

As a computer architect, he rewrote the rulebook, and now, as president of a major U.S. university, he's ready to shake things up once again

# RISC Maker

BY TEKLA S. PERRY  
Senior Editor

**C**omputer scientists are more than just nerds with pocket packs," says John Hennessy. "They can have an appreciation for the arts and humanities and history as well as the sciences."

This pioneer in reduced-instruction-set computing (RISC) now has a chance to prove his point. Installed as president of Stanford University in 2000, Hennessy is responsible for a student population of some 14 000 and a program that boasts graduate schools of business, law, medicine, and engineering and offers undergraduate degrees in over 60 fields, including archaeology, drama, feminist studies, and music. He is the first computer scientist to head such a large and diverse U.S. university.

Hennessy has always been a pioneer. He helped evangelize RISC in the early 1980s when most people didn't believe computers with such an architecture would ever be much use, and he started a company that proved them wrong. He led the development of a new way of organizing cache memory in multiprocessors that in the late 1980s experts said was unworkable, but is today in wide use.

"My career parallels the coming of age of computer science," Hennessy, an IEEE Fellow, told *IEEE Spectrum*.

Hennessy's involvement in computing began some 35 years ago in New York City when he was 16. The dominant small computer at that time was Digital Equipment Corp.'s PDP-8



## Vital Statistics

**Name:** John Hennessy

**Current job:** president, Stanford University (California), since 2000

**Date of birth:** 22 September 1952

**Birthplace:** New York City

**Family:** wife, Andrea; sons, Thomas (19) & Christopher (17)

**Education:** BSEE, Villanova University, 1973; M.S. and Ph.D. in computer science, State University of New York at Stony Brook, 1975 and 1977

**First job:** grocery store stocker and bagger

**Patents:** one, for a method of extending computer word lengths from 32 bits to 64 bits, first used in the MIPS R4000

**Most recent book read:** *Truman* by David McCullough

**Favorite books:** *Tale of Two Cities* by Charles Dickens, *The Foundation Trilogy* by Isaac Asimov, and *Lincoln* by David Herbert Donald

**Favorite periodical:** *New York Review of Books*

**Favorite music:** Michael Kamen's

"Symphony for a New Millennium," any Puccini opera

**Computer:** Macintosh G4 Cube with 21-inch Cinema display

**Favorite Web sites:** Google and MyYahoo

**Favorite expression:** "Charge!"

**Leisure activity:** hiking, golf, bicycling

**Pet peeve:** fuzzy thinking

**Management creed:** "Everybody's opinion has value."

**Memberships:** IEEE, Association for Computing Machinery, American Society for Engineering Education

**Favorite awards:** IEEE John von Neumann Medal (jointly with David A. Patterson), IEEE Seymour Cray Computer Engineering Award, election to National Academy of Engineering, National Academy of Sciences



minicomputer, with a memory of up to 4096 12-bit words and a processing speed of a few hundred thousand instructions per second. The microprocessor had yet to be invented.

Relays, meanwhile, were cheap and abundant, costing less than a dollar each. "And," Hennessy says, "you could do some interesting computing if you had enough relays." So he and a friend decided to build a computer that would play tic-tac-toe. They used about 20 multipole relays.

"We didn't know the basics of logic design or switching theory," Hennessy recalls. "But we were able to figure out the decision tree for tic-tac-toe. And then we made a really important decision—we put red and green lights on the front

and covered the whole thing in black Contac paper, which was absolutely crucial to get the right effect." People were astounded that this device could win.

Flush with the success of his tic-tac-toe computer, which won an award at a local science fair, Hennessy decided to major in electrical engineering in college. He blasted through Villanova University (Villanova, Pa.) in three and a half years, then went on to the State University of New York at Stony Brook. Within six months, in 1974, he had started the research that would be his Ph.D. thesis.

A manager at Brookhaven National Laboratory (Upton, N.Y.) initiated that Ph.D. project when he asked Stony Brook's computer science department for

help in solving a control problem. Brookhaven feared that its workers might, because of long-term radiation exposure, be losing bone density. Monitoring this loss required regular X-ray scans; but a device was needed to control the scanner and keep the X-ray intensity as low as possible by scanning as quickly as possible, yet still capture an adequate image.

Hennessy's faculty advisor suggested he try using a microprocessor to solve the problem. In those early days of microprocessor technology, the state of the art was the Intel 8008, which preceded Intel's 8086 and was programmed largely in assembly language. For his thesis, Hennessy decided to build a programming language that, along with the necessary com-



piler, would do real-time control. He worked on this project for three years. While the resulting system was only an academic prototype and was never used commercially, Hennessy says that some of the ideas that emerged in both specifying and checking real-time constraints were later incorporated in real systems.

When Hennessy started his research, the wider computer world had only limited interest in real-time systems. But just as he was finishing his Ph.D. in 1977, the technology suddenly began getting a lot of attention as its potential became apparent. "So this little field I was working in just exploded," he told *Spectrum*.

For someone about to enter the job market, this was not a bad position to be in. Hennessy's job search focused on academia. "I loved working on research and with students and never thought about doing anything else," he says.

He started as an assistant professor at Stanford, teaching a microprocessor laboratory course to graduate students and a systems programming course to undergraduates. His research focused on compiler optimization.

In 1980, the U.S. Defense Advanced Research Projects Agency (DARPA) began funding researchers as it developed what would become its very large-scale integration (VLSI) research program. Stanford's project for DARPA, led by Forest Baskett and Jim Clark, was to design an IC chip to do three-dimensional graphics transformations. This design evolved into the geometry engine, the chip that was to be the foundation for launching Silicon Graphics Inc. (Mountain View, Calif.), one of the first successful manufacturers of high-performance graphics workstations.

To implement the complex 3-D transformations, which required using floating-point arithmetic, the design resorted to microcode, a series of stored instructions that are not program addressable. The microcode translated the high-level geometric operations into simple instructions that could be implemented with a small number of transistors. As part of the Stanford team, Hennessy developed the high-level programming language and the compiler used to generate that microcode.

Hennessy was perfectly happy in academia. "I never envisioned trying to take

my ideas out to industry," he said. But in 1981, Carver Mead, a professor at the California Institute of Technology (Pasadena) known to Hennessy through Mead's landmark work on VLSI, started a company to commercialize some Caltech research that demonstrated how to design VLSI chips from high-level descriptions. The company was called Silicon Compilers Inc. (now part of Mentor Graphics Inc.). Aware of the work Hennessy had done on the microcode generation language for the geometry engine, the charismatic Mead easily convinced him to spend one day a week working for his nascent company.

Hennessy ended up helping to define the microarchitecture, which included a pipeline and the engine for implementing the instruction set, for what was to become the MicroVAX 1, a computer released by DEC in 1984. This was the first single-chip implementation of the VAX minicomputer architecture. It was also the first large-scale use of high-level synthesis tools, often called silicon compilers.

This stint at Silicon Compilers was the beginning of what was to become a parallel life for Hennessy in industry. When Silicon Graphics was formed, Hennessy left his day-a-week job at Silicon Compilers for a day-a-week job there, where he worked on the system architecture for what was to become the company's first workstation.

### Back at The Farm

Meanwhile, back at Stanford, often called The Farm, Hennessy had begun looking for a new area of computing to explore. To generate ideas, he organized a series of brainstorming sessions as a graduate seminar, asking this question of the group: given that VLSI will soon become the technology of choice, how should we change the way we design computers?

The group read papers, brought in guest speakers, and candidly exchanged ideas. The concepts that came out of the seminar helped to form the basis of what came to be called RISC. At the core was the idea of eliminating microcode. Instead of needing microcode to translate the instruction set, the processor would encounter instructions so simple that it could execute them directly. It was Hennessy's experience with microcode

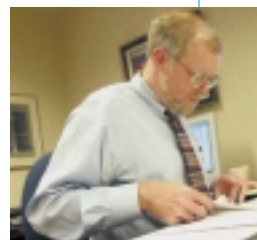
## To-Do List

Themes of John Hennessy's ambitious agenda as president of Stanford University include:

### MULTIDISCIPLINARY RESEARCH.

Hennessy believes researchers walled off into separate, individual worlds can't solve the problems of the future. "Look at the environment. If you want to attack environmental problems, the right way to do so is to bring together geoscientists, plant biologists, ecologists, engineers, and policy people," he says.

His first effort along those lines was helping create a new biomedical engineering and science program, bringing together within one building faculty from engineering, the medical school, and departments such as physics and biology.



Hennessy secured US \$150 million from Jim Clark to fund this effort, the largest single donation Stanford has ever received from a nonalumnus.

### UNDERGRADUATE EDUCATION.

At Hennessy's inauguration, he announced a billion-dollar funding campaign to strengthen Stanford's undergraduate programs by creating an endowment for innovations in that area. One of the most important innovations so far is a series of over 200 small seminars (with 16 students or less) offered to freshmen and sophomores. Undergrads also have more opportunities to engage in research worldwide.

**RESEARCH.** As the major industrial laboratories shut their doors or redirect their efforts away from basic research and toward development, Hennessy says it's time for basic research to play an even greater role in research universities, and that corporations need to commit to funding such research. As a first step in that effort, Hennessy helped establish the Stanford Networking Research Center, a collaboration of 3Com, Cisco, Sony, and other communications companies that make large initial grants and fund ongoing projects. —T.S.P.

in the MicroVAX 1 project that helped him see that it could be eliminated.

Hennessy's group was not alone in coming to this conclusion. Researchers at IBM Corp. had for several years been working on the 801 project, an attempt to implement similar ideas. (The project was never brought to market, although the ideas influenced later IBM products, including the PowerPC architecture.)

Somebody else who also concluded that microcode was unnecessary was David Patterson, a professor of computer science at the University of California at Berkeley. Some six months earlier, he had discovered that computers could run more efficiently without it. Patterson's research team coined the acronym RISC.

But most of the rest of the computer world was skeptical that RISC would work. The fact that the original performance results were obtained from simulations only added to their doubts.

So Hennessy, with a handful of graduate students and the occasional collaboration of other professors, spent about 18 months designing, building, and testing a VLSI chip, along with compilers and a simple test computer, to prove the technology. Looking back, he admits they made a fair number of design errors and the chip wasn't as efficient as it could have been.

But, he said, "the opportunity for improvement was so big that the fact we made mistakes didn't matter.

We built this little 25 000-transistor microprocessor, and it was twice as fast as a big minicomputer that cost US \$250 000."

Hennessy wrote papers and began giving talks about this new computer architecture, thinking that existing computer companies would be quick to embrace such an obvious technical improvement. "To say I was naïve," Hennessy told *Spectrum*, "is an understatement. People just didn't believe it."

Patterson, who was also promoting the new concept, did not see Hennessy as a rival. "We were a persecuted minority," Patterson told *Spectrum*. "It was better for us to be on the same team."

There were two issues to deal with, Patterson recalls, a technical one and a

business one. "Technically, what we were doing was designing computers based on careful measurements at a time when computer designs were being driven by aesthetics. Computer architectures were supposed to be beautiful, not driven by efficiency," he said.

On the business side, people questioned whether RISC made sense economically. Using the analogy of train tracks, Patterson added, "While perhaps changing the width of train tracks might make technical sense, given the investment of infrastructure, it will never happen." Software would have to be recompiled for these new machines.



To get around town, John Hennessy drives a GEM, an electric vehicle made by DaimlerChrysler.

C. Gordon Bell, DEC's former vice president of R&D, thought otherwise, even though DEC's own RISC project had never come to fruition. He had just started Encore Computer Corp. (now Encore Real Time Computing Inc., Fort Lauderdale, Fla.) to create massively scalable computer systems, and he was aware of Hennessy's work on RISC. He told Hennessy that to get RISC ideas out into the world, he would have to start his own company, and that, if he did, Encore Computer would be his first customer and an investor.

"It was clear to me that a chip with RISC architecture needed to be built," says Bell, currently a senior researcher at Microsoft Corp. "I thought it would

give us at least a one-time performance gain of a factor of two or three."

So Hennessy, along with John Mousouris, who had worked on the IBM RISC project, and Edward ("Skip") Stritter, who had worked on the Motorola 68000 microprocessor, decided that starting a company was just what they would do.

### Backing into business

"We went to the venture capitalists with the most ridiculous business plan you'd ever want to see," Hennessy says. "We had a bunch of slides about why this was a great technology and a spreadsheet showing how we'd spend the money. We didn't have a realistic technical schedule; we didn't have any marketing plan."

But they got funded in the summer of 1984 for \$1 million from the Mayfield Fund, a venture capital company based in Menlo Park, Calif. That summer, MIPS Computer Systems Inc. set up its first office in subleased space in Mountain View.

Unwilling to completely give up his academic career, Hennessy took an 80 percent leave from Stanford, working there one day a week. He planned to return full-time once the company was solidly established. Since the initial technical research at MIPS was done, Hennessy, as co-founder and chief scientist, ended up becoming chief evangelist, doing cold calls on computer companies in an effort to convince potential customers that this first RISC

microprocessor was going to change the world of computing.

About six months later, MIPS signed a letter of intent with its first customer, Prime Computer Ltd. (now defunct), committing to deliver chips by the end of 1985. In the final deadline crunch, Hennessy jumped in as needed, from working on the compiler team, to writing test code, to debugging the processor, and the MIPS R2000 came out on schedule. Then Hennessy turned his focus back to Stanford, though he still spends an occasional day at MIPS.

MIPS chips are now used primarily in embedded applications, including products such as the Sony PlayStation, Hewlett-Packard color laser writers, Cisco Systems network switches, and digital set-

top boxes. The company had its initial public offering in 1989, was acquired by Silicon Graphics in 1992, and later spun out as MIPS Technologies. Countless other RISC chips have been designed since, and RISC is now the dominant architecture for embedded applications.

Hennessy's time in the corporate world fed positively into his university career. "In academia," he says, "we have a tendency not to make decisions until we get all the data in, know every single fact, and it's crystal clear what the right decision is. In industry, it doesn't work that way; time is considered a valuable commodity." Hennessy found that learning how to be decisive helped him in academia, particularly as he climbed up the administrative hierarchy.

With MIPS demonstrating that RISC was an important breakthrough in computer architecture, Hennessy decided to try to teach this new design method to students. Patterson, still at Berkeley, was doing the same thing. Of course, there were no textbooks. So the two decided to write one.

Taking sabbatical time, they moved into an office donated to them by DEC's Western Research Laboratory in Palo Alto. The fall of 1987 was spent figuring out the framework for the text: what it needed to teach and what its structure would be. They divided the chapters between them and edited each other's work. They then asked their colleagues in the industry to critique the book and, after reviewing suggestions, revised it. Finally, in the spring of 1990, *Computer Architecture: A Quantitative Approach* was published by Morgan Kaufmann (San Mateo, Calif.). The publisher expected lifetime sales of the book to be about 15 000; it hit that number the first year. Hennessy and Patterson's book is still being widely used and has sold over 100 000 copies.

### Back at The Farm—again

In 1988 Hennessy was running the Computer Systems Laboratory at Stanford, teaching introductory computer architecture to graduate students, and putting together a group of researchers to explore how the RISC approach might be employed in parallel computing.

"The key questions we asked," he says, "were: what was the right balance of hardware and software mechanisms, and how

should the mechanisms change with larger numbers of processors?"

In small-scale multiprocessors, a technique called a snooping cache is used to maintain consistency among the individual cache memories associated with each processor. A snooping cache requires that each processor send notifications to all the other processors when it changes any shared data. So, if more than 20 or 30 processors are used, they end up spending most of their time on notification.

One solution would be to use a single shared memory and not cache data that could be shared. But with large numbers of processors, that approach falls apart: the memory becomes a bottleneck and is much slower than the individual caches. "The conventional wisdom," Hennessy says, "became that because of the cache coherency problem, you could not build a large-scale multiprocessor with a shared memory."

**We didn't know logic-design or switching-theory basics. But we could figure out the decision tree for tic-tac-toe.**

So in the 1980s, several groups around the country, including researchers at Caltech and Intel, were trying to optimize an approach called message passing, which avoids shared memory. Rather, it requires that data be explicitly communicated by messages between processors. Unfortunately, that setup uses a different programming model from that used for multiprocessors with a few tens of processors.

Hennessy and his students at Stanford instead decided to reconsider cache coherency. Rather than physically building a central shared memory, they created one logically. That is, each of the multiple, physically distributed memories has a dedicated cache, and the computer maintains a directory with each memory, keeping track of which processor has which memories in its cache. When information is changed or needed, each processor simply refers to the directory and then just sends out a single message.

The first paper describing this directory concept was published by Hennessy and his colleagues in 1988; the group built a working design in 1992. The

machine was called DASH, for Directory Architecture for SHared memory. Silicon Graphics used the directory approach embodied in DASH in 1996 in its Origin workstation, a scalable high-performance computer, and several other companies have employed the technology since.

### Steps to the presidency

While Hennessy went on being involved in research for some years, his administrative functions at Stanford were increasing, as he stepped up to department chairman, then dean of engineering, then provost, and, now, president.

"John shot through the academic hierarchy faster than anyone I've seen," Patterson says.

Even as president, Hennessy does some work outside the university, although it now takes less than 10 percent of his time. This past year, much of that

has been spent with Patterson preparing the third edition of their textbook. He also supervises one or two Ph.D. students and advises 15 undergraduates.

As president, Patterson observes, Hennessy runs the university with an engineering perspective, assuming he can analyze every problem and find the best solution. He has earned broad loyalty on campus because, Patterson believes, he is honest, doesn't play games, and speaks the truth, even if his positions aren't popular.

Hennessy sees a university presidency as being something one does for about a decade. These years will be busy ones, for Hennessy has made himself a long to-do list [see "To-Do List," p. 35]. After his term is over, he says, he might take a real sabbatical, instead of starting a company or writing a book. Or not.

"The great thing about being a university president is you can keep that most wonderful of titles—professor. So you are able to go back to teaching and working with students," he says. And, undoubtedly, doing a little something on the side. ●