Red Hat's policy has been to share all the **software** innovations that it has developed freely with the open-source community under the GNU General Public License (GPL). Using this approach, it has attained a leading position as a distributor of the Unix-based Linux operating system in the late 1990s and beyond.

FURTHER READING

Young, Robert, and Wendy G. Rohm. *Under the Radar: How Red Hat Changed the Software Business and Took Microsoft by Surprise.* Scottsdale, Ariz.: Coriolis, 1999.

—*Jonathan Bowen*

Reduced Instruction Set Computer

Reduced instruction set computers (RISC) use an approach to **computer architecture** that began in the mid-1980s and soon revolutionized the computer industry. The philosophy behind RISC design can be summarized as: Make it simple, therefore faster.

The motivation for RISC processors arose from technological developments that altered the architectural parameters traditionally used in the computer industry. The general trend until the mid-1970s was the design of ever-richer instruction sets, which could take some of the burden of interpreting high-level computer languages from the **compiler** to the **hardware**. The idea was to reduce the gap between the programming languages and the underlying hardware, by introducing many special instructions. Theoretically, this would make compilers simpler.

The RISC pioneers recognized that the performance of computer systems depended on three factors: the number of instructions in the program, the cycle time of the machine, and the number of instructions executed in each cycle. Bringing the hardware closer to the **software** meant that the number of instructions needed for a program was reduced, but at the cost of increasing the number of instructions per cycle. Dave Patterson (1947–) of the University of California at Berkeley and John Hennessy (1952–) of Stanford University took another route: By simplifying the instruction set and by getting rid of special instructions used only a small fraction of the time, the streamlined machine could use a faster clock and could execute more instructions per cycle.

The RISC movement got started when quantitative measurements of program performance showed that complex instruction set computing (CISC) used fewer than 10 to 20 instructions more than 90 percent of the time. Concentrating the design efforts on these instructions means that the frequently used instructions are made faster. RISC computers eliminated, for example, many addressing modes typical of computers such as the **VAX** but made much faster the few addressing modes that *were* kept. RISC processors also try to optimize the use of a large register file, avoiding frequent accesses to the slower memory. And last but not least, the streamlined instruction set allows the computer architect to pipeline instructions—that is, the processor works on several instructions at the same time, using its several pipeline stages. RISC processors are inherently parallel.

The first processor that would eventually lead to the RISC concept was the **IBM** 801, a machine developed in the 1970s but which remained a laboratory curiosity. The RISC I, built at Berkeley in 1981, and the MIPS processor, built at Stanford a few years later, showed that the approach was feasible and that the necessary technology was available. In the mid-1980s almost all major computer companies started developing RISC processors, and even CISC machines, like the **Intel** 86xx series, were built later using a "RISC kernel."

FURTHER READING

Patterson, Dave, and John Hennessy. *Computer Architecture: A Quantitative Approach.* San Francisco: Morgan Kaufmann, 1992.

—*Raúl Rojas*

Register

A register is a "container" used by the processor of a **digital computer** to store a number that will soon be needed again. Therefore, registers are memory cells, but usually faster that those provided by the main memory. Registers are often implemented with dynamic **RAM** (random access memory) chips, which offer a smaller cycle time than the static chips used