VLSI See Very Large Scale Integration.

VMS

VMS is a proprietary **operating system** (OS) that originated at the **Digital Equipment Corporation** (DEC) in the late 1970s. When in the mid-1980s DEC was the second-largest computer company in the world, VMS was the leading OS for its most popular products, then called **minicomputers. Compaq Computer Corporation** acquired DEC in 1998, and the OS is now available from Compaq under the OpenVMS trademark. OpenVMS remains a supported OS in the year 2000, with a reputation for reliability, comprehensive documentation, and technical excellence.

DEC released version 1.0 of VMS in 1978 for the first of its extremely successful VAX line of computers, the first 32-bit commercial machines. By 1984, VMS included sophisticated security, networking, and clustering abilities, many of which remain unavailable in other operating systems 15 years later. To compete more effectively with **Unix** rivals, Version 5.5 implemented the entire Portable Operating System Interface (POSIX) standard programming interface in 1991; this made OpenVMS a candidate for a range of U.S. government contracts that previously could be filled only by Unix machines.

The next year, 1992, DEC released the new AXP **central processing unit** (CPU) architecture with VMS. As 2000 begins, OpenVMS is at release 7.2 as the property of Compaq, which acquired DEC in 1998. Note that OpenVMS is merely a new marketing label for VMS; there is no technical difference between VMS and OpenVMS.

Structurally, VMS most resembles **IBM**'s OS/400, for the latter is also a proprietary midrange OS designed in parallel with its underlying **hardware** and noted for its high level of reliability. In the marketplace, however, the Unix OS has been VMS's principal competitor in its traditional niches of scientific computing, industrial control, and database hosting. Like traditional Unix, a command-line interface (CLI) is most common for VMS users. The most common application languages for VMS development are DCL (the VMS "shell"), **Fortran**, **BASIC**, and C. Among VMS's many attractions is that different languages are compatible in object format. This means that it is straightforward to code different parts of an application using Fortran and C, for example, and have them work together correctly.

Although there have been several projects to host VMS on less expensive hardware than the VAX and AXP (now called Alpha) CPUs, none of these moved beyond the point of laboratory experimentation. One footnote to VMS's legacy is that the Microsoft **Windows NT** (and later Windows 2000) OS recruited several designers and took many ideas from VMS.

FURTHER READING

Holmay, Patrick, and James F. Peters III. *The VMS User's Guide.* Bedford, Mass.: Digital Press, 1990; 2nd ed., published under the title *The OpenVMS User's Guide*, Boston: Digital Press, 1998.

Miller, David Donald. *Open VMS Operating System Concepts.* Boston: Digital Press, 1997.

—*Cameron Laird*

Von Neumann Architecture

Von Neumann architecture is the basic arrangement of the **hardware** associated with **digital computers** since the advent of the first stored program machines in the late 1940s. The name has been controversial, since **John von Neumann** (1903–57), the brilliant Hungarian mathematician, was just a consultant to the team of engineers that built the **ENIAC** in 1944–45.

The ENIAC team realized that their machine's design made it difficult to program and that this would be an obstacle to widespread adoption of computer technology. They came to the conclusion that storing the program, together with the data, in memory would make the machine more flexible and easy to use. Von Neumann volunteered to write a report of the discussions and his "**First Draft of a Report on the EDVAC**" was soon circulating widely. Although von Neumann was not the sole intellectual source of the ideas contained in the report, the term *von Neumann architecture* stuck and is still used today.

The von Neumann architecture consists of five main parts: an arithmetic and control unit, a memory, and input/output devices. Since the processor is usually faster that memory, the connection between both is typically called the *von Neumann bottleneck.* Stored programs are considered a main feature of the von Neumann architecture, and usually von Neumann architectures are sequential—that is, every instruction in executed in turn.

John Backus (1924–) coined another term, the *von Neumann style of programming.* This refers to the fact that data have to be moved between processor and memory explicitly by the programmer. This can lead to errors and poor productivity.

As can be seen, the term *von Neumann architecture* is used in many different senses, but usually refers to (1) the **stored program** concept, (2) sequential machines and (3) computers with separate processor, memory and input/output. The term *non-von Neumann architecture* is also used; it can refer to parallel machines, functional machines (e.g. **LISP** machines), or machines in which processing and memory are not clearly separated.

FURTHER READING

Backus, John W. "Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs." *Communications of the ACM,* Vol. 21, 1978, pp. 613–641.

Hurd, Cuthbert. "Early IBM Computers: Edited Testimony." *Annals of the History of Computing,* Vol. 3, No. 1, 1981, pp. 163–182.

Von Neumann, John. "First Draft of a Report on the EDVAC." In *Papers of John von Neumann on Computers and Computer Theory.* Cambridge, Mass.: MIT Press; Los Angeles/San Francisco: Tomash Publishers, 1987, pp. 17–82.

—*James Tomayko and Raúl Rojas*

Von Neumann, John

1903–57

Hungarian–U.S. Mathematician and Scientist

John von Neumann (Neumann Janos in Hungarian) was one of the most outstanding mathematicians and logicians of the twentieth century. He made significant contributions to pure mathematics, quantum physics, meteorology, game theory for economics, and to the theory and practice of high-speed computing machines and their programming. He was associated with the earliest all-electronic computers in the United States: the **ENIAC**, the **EDVAC**, and the **IAS machine**. Von Neumann's enthusiasm and logical design efforts helped set a strong mathematical and scientific foundation for the development of the modern computer.

John von Neumann grew up in an environment that encouraged intellectual activity and curiosity. At extended lunches and dinners, family members were encouraged to make scholarly or literary presentations. Intellectual achievement was held in high esteem not just at home but in much of the middle-class community in Budapest. At school, John stood out as particularly intelligent, and his schoolmaster arranged for him to be tutored by mathematicians from Budapest University, who were amazed at the brilliant solutions John proposed for the problems they posed him.

In 1926, when von Neumann was at the University of Göttingen in Germany, again the atmosphere was intellectually vibrant. Werner Heisenberg (1901–76) had just developed the matrix approach to quantum mechanics, which was soon challenged by the wave formulation of Erwin Schrödinger (1887–1961). In the midst of the debate, von Neumann and others drew the logical conclusion that since both theories were arriving at the same results, there must be different ways of saying the same thing. Von Neumann went on to show that both the Heisenberg and Schrödinger formulations of quantum mechanics satisfy the same set of axioms, and therefore the wave and particle descriptions are equivalent at the mathematical level.

World War II was a turning point in the development of important new technologies such as radar and computing. By then von Neumann had been in the United States for almost a decade, was a naturalized citizen, and was much in demand by U.S. government agencies involved with the war effort. He was consulted on the solution of ballistics, detonation, and aerodynamics problems and worked at Los Alamos on the atomic bomb project. He was intimately involved with problems that needed massive calculations for their solution and therefore required computing machines.