conforms to administrator-defined network security policies. A firewall can prevent unauthorized users from accessing network resources on a company's internal network while preventing internal users from accessing specific resources and **World Wide Web** sites on the Internet. More sophisticated firewalls monitor not only the source and destination of network service requests, but also the content of the data. As an example, some companies monitor e-mail to ensure that sensitive information does not leave the company's internal network.

Firewalls represent one method of solving the myriad security problems related to connecting disparate networks. It is part of a larger security policy that defines the types of services and access levels permitted between two networks. Administrators implement these policies using routers, password policies, and overall network configuration decisions; the firewall plays an integral role in enforcing those policies by forcing all connections to pass through it.

Without a firewall, a company is exposing its internal systems to probes and attacks from hosts elsewhere on the Internet. Network security is completely reliant upon host security, and administrators must ensure a high level of security on all hosts. The larger the network, the more difficult it is for an administrator to secure all the hosts individually. Mistakes and lapses in security become more common, creating holes that a hacker can use to gain unauthorized access.

The primary benefits of a firewall are protection from attacks on vulnerable services, control over access to internal and external systems, concentration of security systems, collection of statistics to monitor network use, abuse, and enforcement of security policies. A firewall is generally located at a high level in the network design, such as a gateway to the company's connection to the Internet. To control access to the network, firewalls do what is called *packet filtering*.

Like a **router**, the firewall examines every packet headed into and out of the network. Depending on the specific policies in place, the firewall may only look at the source or destination IP addresses to determine if the packet should be forwarded. Packet filters may delve deeper into the packet, as necessary. Some firewalls will look at the specific application generating the packet (using the well-known port numbers of TCP and UDP protocols) to determine if the packet should be forwarded. This lets an administrator configure the firewall to support Web traffic while blocking **Telnet** connections.

FURTHER READING

Goncalves, Marcus. *Firewalls Complete*. New York: McGraw-Hill, 1998.

Walker, Kathryn, and Linda Croswhite Cavanaugh. *Computer Security Policies and SunScreen Firewalls*. Palo Alto, Calif.: Sun Microsystems Press, 1998.

Zwicky, Elizabeth, Simon Cooper, and D. Brent Chapman. *Building Internet Firewalls*. Sebastopol, Calif.: O'Reilly, 1995; 2nd ed., Cambridge, Mass., 2000.

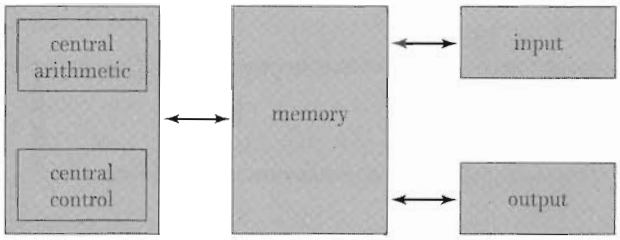
—*Paul Shields*

## *First Draft of a Report on the EDVAC*

by John von Neumann

The *First Draft of a Report on the EDVAC*, written by **John von Neumann** (1903–57) in 1945, outlined the architecture of a stored program computer. It is probably the most famous, the most cited, but also the least read document in the history of computing. A summary of von Neumann's discussions with the inventors of the **ENIAC**, *First Draft* is also one of the most controversial documents in computing history, since the proposals contained in the draft are not exclusively von Neumann's ideas. He was criticized for having omitted the names of his coauthors and was even accused of plagiarism.

According to ENIAC coinventor **John W. Mauchly** (1907–80), writing in 1979, the main architectural concepts for the **EDVAC** (electronic discrete variable automatic computer), the machine that would follow the ENIAC, were developed by his team long before von Neumann visited the laboratory for the first time in September 1944. **J. Presper Eckert** (1919–95), the second coinventor of ENIAC, had written some notes in early 1944 describing the stored program concept; in essence, a mercury delay line or etched disk could be used to store a program. However, the ENIAC team was still hesitant about including this feature in their sec-




*John von Neumann's proposed architecture for the EDVAC.*

ond machine, because of the complexity that it would add to the design. It was von Neumann who convinced them of all the advantages of a stored program computer. He is credited with having refined the concept and its logical implications during his visits to the **Moore School of Electrical Engineering**.

*First Draft* was a report of these discussions, written by von Neumann himself; although it was not published, it was widely distributed in the computer science community and was the first document describing electronic computers to reach a large technical audience. The draft had important implications for Eckert and Mauchly; they were later denied patents for the ENIAC concepts and *First Draft* was ruled to be a disclosure that had put the main ideas into the public domain.

The draft proposed to build a "computer" composed of the following five main units: a central arithmetic part (CA), a central control (CC), a memory (M), input (I) from a recording medium (R), and output (O) to a recording medium. The accompanying figure shows a diagram of the relations between units. The central arithmetic part and the central control are wired together in the C unit. Also, input and output are connected directly to memory, since von Neumann considered this a more convenient layout for the machine.

Von Neumann was very interested in the structure of the human nervous system, and the draft makes many interesting analogies between computers and brains—comparing, for example, the binary system to the "on" and "off" nature of neurons. Finally, it argues that processing in the brain is done in the associative neurons, which correspond to the CA, CC, and M parts of the machine. The wide popularity of *First Draft* led eventually to the concept of **von Neumann architecture**, which refers to machines with a stored program and separate processing and storage units.

FURTHER READING

Mauchly, John. "Amending the ENIAC Story." *Datamation*, Vol. 25, No. 11, 1979, pp. 217–219.

Stern, Nancy. "Minutes of 1947 Patent Conference." Moore School of Electrical Engineering, *Annals of the History of Computing*, Vol. 7, No. 2, 1985, pp. 100–116.

Von Neumann, John. "First Draft of a Report on the EDVAC." Contract W-670-ORD-4926, Moore School of Electrical Engineering, University of Pennsylvania, 30 June 1945. Reprinted in Nancy Stern, *From ENIAC to UNIVAC*. Woburn, Mass.: Digital Press, 1981.

— *Raúl Rojas*

## Flash Memory

Flash memory chips provide nonvolatile storage for computers, cellular telephones, digital cameras, and other electronic devices. In other words, flash memory chips retain the information they store even after the power is cut. These chips can be reprogrammed (rewritten) using the normal voltage levels inside a computer. **EPROM** (erasable programmable read-only memory) chips require much higher voltages and are usually reprogrammed outside the system. Flash memories with much higher density levels than EPROMs are also available.

Earlier generations of flash memories were erased "in bulk"; that is, the entire chip was erased and loaded with new information in a single step. Flash memories cannot be used like DRAM chips, which can be changed one bit at a time. Modern flash memories are erased and rewritten in complete sectors (each device has several sectors). Even if only one byte is different, the entire sector has to be erased and rewritten. Therefore, flash memories are no substitute for dynamic **RAM** (random-access memory), but are perfectly suited for devices in which the information is written once and accessed many times. Modem manufacturers, for example, can change the protocol in the modem by rewriting a flash chip. The **BIOS** (Basic/Input Output Routines) chip in **personal computers** can be implemented with a flash memory and can be reprogrammed to alter the stored routines.