Central Processing Unit

The central processing unit (CPU) is the part of a computer that contains the arithmetic and logical circuits of the machine. In sequential computers using **von Neumann architecture**, a computer consists of a CPU, a memory, and input/output devices, such as **keyboard**, screen, and **hard disk**. Figure 1 shows a simplified diagram of such a computer. In this machine, which could be a microcomputer of the early 1970s, all devices exchange data through the bus. Programs and data are stored in the main memory and the hard disk provides a nonvolatile storage medium; that is, the information remains stored even if the machine is turned off.

The distinction between CPU and memory was not as clear-cut in early computers. The **ENIAC**, for example, had storage units, called *accumulators*, which included their own arithmetic facilities. One could add a number A to another number B by sending A to the accumulator containing B, ordering it to perform an addition. The **Harvard Mark I** used a similar approach with its 72 accumulators. In both machines CPU and memory were intermixed.

There is a good reason for separating the processor from the memory. Most of the data in a computer's memory are not accessed in every cycle; the bulk of it just sits there waiting to be used. By having a separate processor, only the data that are needed for each operation are brought into the CPU, transformed, and sent back to memory. The total number of components needed is lower than when each memory cell has its own small attached processor.

Further, memory chips are usually slower than the processor logic. The processor tries to hold the information internally, in a few fast memory cells called *registers*, as long as it is feasible, exchanging information with the memory only when needed. A fast processor can work with several slower memory units using a technique called *interleaving*. The resulting system is fast and simple.

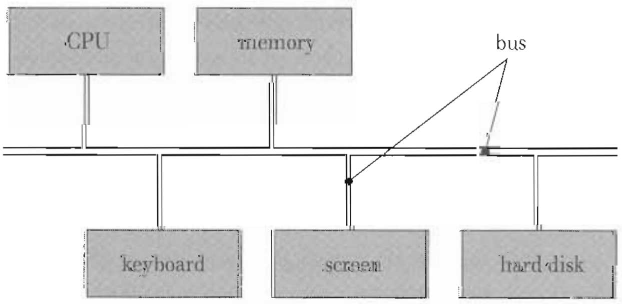

Figure 1. *Von Neumann architecture.*

A small CPU consists of registers, an arithmetic-logic unit (ALU) and several other registers that are useful for diverse purposes. Figure 2 shows a simplified view of a small CPU, depicting a machine with eight registers numbered 0 to 7. If two registers are to be added, they are read from the register file and go to both inputs of the ALU. The ALU performs the addition and the result is stored back in one of the registers. The ALU can be limited to perform addition and subtraction of numbers. Other arithmetical operations can be implemented using these two basic operations repeatedly.

The instruction register (IR) is used to hold the instruction being executed. Decoding logic sends the appropriate signals to the rest of the machine in order to activate the necessary transfers. The address register (AR) is used to hold the address of the data that will be read from memory or stored to it. The contents of the AR are sent to the data bus and the CPU then asks for permission to start a memory access. The data register (DR) holds the data to be sent to memory during a memory write, or the data read from memory after a load. DR has a connection path to the registers. The address of the next instruction to be executed is kept in the program counter (PC), which is increased by one during normal execution of a sequence of operations. If the program needs to perform a conditional branching, the contents of the PC are altered.

The execution of an instruction in our simple machine could be divided in four steps: fetch, decode, execute, and write back. In the fetch phase, the next instruction is loaded from memory and is stored in IR. In the next phase, the instruction is decoded and the appropriate arguments for the ALU are selected. In the execution phase, the ALU performs the desired operation, and in the write-back phase, the result is written to the result register.

The first generation of **microprocessors** consisted of CPUs with a word length of eight bits (i.e., the number of bits used in the internal arithmetical registers

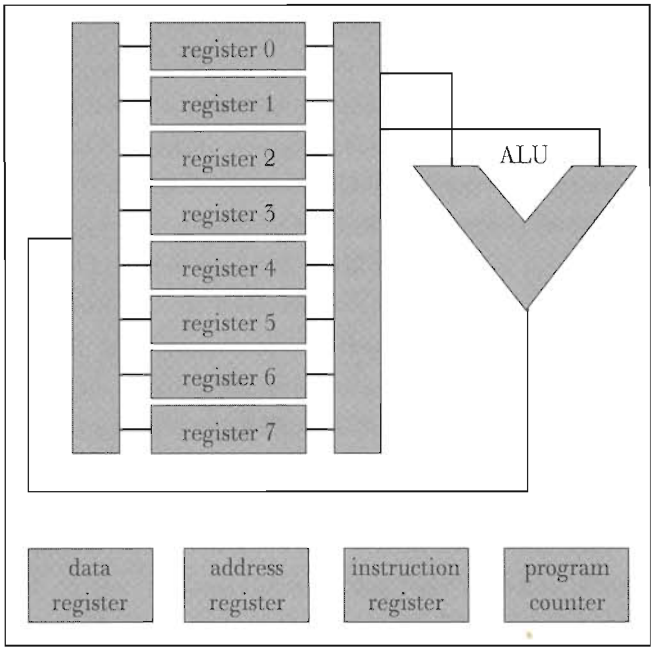

Figure 2. *Small CPU.*

was eight). The second generation used 16 bits, and the third generation used 32 bits. The transition to 64-bit architectures started in the mid-1990s.

Modern CPUs contain many more registers than the CPU discussed in this entry. The IA-64 architecture developed by **Intel** and **Hewlett-Packard** as their new flagship for the 21st century contains no less than 128 general-purpose and 128 floating-point registers, together with registers for many other special purposes.

FURTHER READING

Hennessy, John and David Patterson. *Computer Architecture: A Quantitative Approach*, 2nd ed. San Francisco: Morgan Kaufmann, 1996.

Patterson, David, and John Hennessy. *Computer Organization and Design: The Hardware/Software Interface*, 2nd ed. San Francisco: Morgan Kaufmann, 1998.

—*Raúl Rojas*

Cerf, Vinton

1943–

U.S. Internet Pioneer

Vinton G. Cerf is known as the "father of the **Internet**" for his early work in bringing about the underlying protocols of the network. With **Robert Kahn** (1938–), he was the codesigner of **TCP/IP**, the digital telecommunications protocol that gave birth to the Internet and is commonly used today. Later, especially during his time with the U.S. Department of Defense's Advanced Research Projects Agency (**ARPA**) from 1976 to 1982, Cerf performed a key role leading the development of Internet and its related data packet and security technologies.

Cerf's introduction to the concepts underlying the Internet came when he was a graduate student at the University of California–Los Angeles (UCLA), 1967–72. He worked in Len Kleinrock's (1934–) laboratory and helped to develop the host-level protocols of the **ARPANET**, the forerunner of the Internet. He contributed to the host-to-host, file transfer (**FTP**), and telnet protocols. He also worked on software for the Network Measurement Center, which made performance measurements of the ARPANET, comparing them with predictions from queuing models developed under Kleinrock's supervision. At UCLA, Cerf met Kahn in 1969 when he visited from the firm **Bolt, Beranek and Newman**, with David Walden, to perform tests on the ARPANET.

In late 1972, Cerf moved to Stanford University, joining the electrical engineering and computer science faculties. At Stanford, he met Bob Kahn again to discuss his work at ARPA. Kahn was investigating packet radio and packet satellite ideas that had worked well using telephone circuits for the ARPANET. The problem under consideration was how to interconnect these networks. In particular, he was aiming for an open architecture that would allow many networks to work together so that the computers involved need not know about the topology and specific characteristics of the connected hardware.

During 1973, Cerf and Kahn developed the concepts underlying the Internet. They prepared a draft paper in September for presentation to the International Network Working Group (INWG), founded the preceding year. The INWG was an informal group of researchers looking at networks, mainly from the United States and Europe. All those involved had an interest in packet-switching technology. Cerf and Kahn revised their paper after the INWG meeting and