as a solid-state disk or wireless communication with a cellular phone network. Lastly, we see that we can connect analog audio inputs and outputs, such as electronic musical instruments and headphones, plus an external microphone.

Physical size and weight are important parameters for a laptop that will be carried regularly. This is a mid-size, mid-weight model. At 5.6 pounds, it weighs over twice as much as this book. A lightweight laptop has roughly the same weight as this book, and heavier models, sometimes called desktop replacements, can weigh in at around 8 pounds. Generally, to reduce weight, the size shrinks and we give up features and battery life. However, it is also possible to reduce weight by replacing plastic in the case with aluminum, but for greater cost.

Lastly, the ad lists software that is preinstalled on the laptop. These include the operating system (Windows 7®), the Microsoft® Office suite of programs that includes a word processor, spreadsheet, and so on for performing common tasks, and a 3-year subscription to updates for a malware detection package. Malware is software that intends to do harm, and comes in many forms, such as viruses that can take over your computer when you open a downloaded file. Malware detection software constantly watches for such programs in files and web content to prevent them from running. But hackers are constantly creating new forms of malware, so it is necessary to regularly update the detection software to keep up with the latest threats.

Putting sizes in perspective
Admiral Grace Murray Hopper demonstrated the relative sizes of computer jargon by displaying a coil of wire nearly 1000 feet long, a short piece of wire about as long as your forearm, and a bag containing grains of pepper. She would point out that the wire coil was the distance traveled by an electron along the wire in the space of a microsecond. The short piece of wire was the distance traveled by an electron along the wire in the space of a nanosecond. The grains of pepper represented the distance traveled by an electron in a picosecond. She would admonish the members of her audience to remember their nanoseconds.

Within this ad, multiple size measures have been used. Let's summarize the prefixes that are used frequently in computing.

Power of 10	Power of 2	Value of Power of 2	Prefix	Abbreviation	Derivation
10^{-12}			pico	p	Italian for little
10^{-9}			nano	n	Greek for dwarf
10^{-6}			micro	μ	Greek for small
10^{-3}			milli	m	Latin for thousandth
10^{3}	2^{10}	1024	kilo	K	Greek for thousand
10^{6}	2^{20}	1,048,576	mega	M	Greek for large
10^{9}	2^{30}	1,073,741,824	giga	G	Greek for giant
10^{12}	2^{40}	not enough room	tera	T	Greek for monster
10^{15}	2^{50}	not enough room	peta	P	Greek prefix for five

Did you notice that we used powers of 10 when referring to time and powers of 2 when referring to storage? Time is expressed in multiples of seconds in decimal notation. Storage capacity is expressed in multiples of bytes in binary notation. If you keep this distinction in mind, it is clear that K is 1000 when referring to speed and 1024 when referring to storage.

We now move from the specific to the general. In the next several sections we look at each of the pieces of hardware that make up a computer from the logical level, rather than from a specific computer configuration.

5.2 Stored-Program Concept

A major defining point in the history of computing was the realization in 1944–1945 that data and instructions to manipulate the data were logically the same and could be stored in the same place. The computer design built upon this principle, which became known as the *von Neumann architecture*, is still the basis for computers today. Although the name honors John von Neumann, a brilliant mathematician who worked on the construction of the atomic bomb, the idea probably originated with J. Presper Eckert and John Mauchly, two other early pioneers who worked on the ENIAC at the Moore School at the University of Pennsylvania during the same time period.

Does it matter who was the father of the modern computer?
All of the people involved in the research and development of electronic computing devices in the late 1930s and 1940s undoubtedly contributed to the computer as we know it. This list includes John Atanasoff, Clifford Berry, and Konrad Zuse, in addition to von Neumann, Eckert, and Mauchly.
In 1951, Sperry Rand bought the patent for the ENIAC and its underlying concepts and began charging royalties to other computer manufacturers. Not wanting to pay royalties, Honeywell researched the history of modern computers and presented evidence that the work of John Atanasoff at Iowa State College had directly influenced Mauchly and Eckert. Because of this evidence, the patent for the ENIAC was invalidated in 1973.

von Neumann Architecture

Another major characteristic of the von Neumann architecture is that the units that process information are separate from the units that store information. This characteristic leads to the following five components of the von Neumann architecture, shown in Figure 5.1:

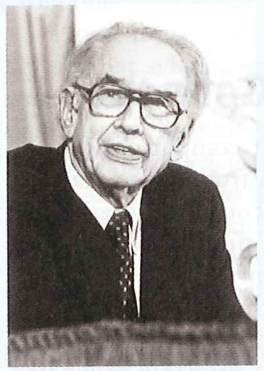
- The memory unit that holds both data and instructions
- The arithmetic/logic unit that is capable of performing arithmetic and logic operations on data
- The input unit that moves data from the outside world into the computer
- The output unit that moves results from inside the computer to the outside world
- The control unit that acts as the stage manager to ensure that all the other components act in concert

Memory

Recall from the discussion of number systems that each storage unit, called a bit, is capable of holding a 1 or a 0; these bits are grouped together into bytes (8 bits), and these bytes are in turn grouped together into words. Memory is a collection of cells, each with a unique physical address. We use the generic word *cell* here rather than byte or word, because the number of bits in each addressable location, called the memory's **addressability**, varies from one machine to another. Today, most computers are byte addressable.

Addressability The number of bits stored in each addressable location in memory

John Vincent Atanasoff



Courtesy of ISU Photo Service

John Vincent Atanasoff was born in Hamilton, New York, on October 4, 1903, one of nine children. When he was about ten, his father bought a new slide rule. After reading the instructions, John Vincent became more interested in the mathematics involved than in the slide rule itself. His mother picked up on his interest and helped him study his father's old college algebra book. He continued his interest in mathematics and science and graduated from high school in two years. His family moved to Old Chicara, Florida, where John Vincent graduated from the University of Florida in 1925 with a degree in electrical engineering because the university didn't offer a degree in theoretical physics. A year later, he received a Master's degree in mathematics from Iowa State College. In 1930, after receiving his PhD in theoretical physics, he returned to Iowa State College as an assistant professor in mathematics and physics.

Dr. Atanasoff became interested in finding a machine that could do the complex mathematical work he and his graduate students were doing. He examined computational devices in existence at that time, including the Monroe calculator and the IBM tabulator. Upon concluding that these machines were too slow and inaccurate, he became obsessed with finding a solution. He said that at night in a tavern after a drink of bourbon he began generating ideas of how to build this computing device. It would be electronically operated and would compute by direct logical action rather than enumeration, as in analog devices. It would use binary numbers rather than decimal numbers, condensers for memory, and a regenerative process to avoid lapses due to leakage of power.

In 1939, with a $650 grant from the school and a new graduate assistant named Clifford Berry, Dr. Atanasoff began work on the first prototype of the Atanasoff Berry Computer (ABC) in the basement of the physics building. The first working prototype was demonstrated that year.

In 1941, John Mauchly, a physicist at Ursinus College whom Dr. Atanasoff had met at a conference, came to Iowa State to visit the Atanasoffs and see a demonstration of the ABC machine. After extensive discussions, Mauchly left with papers describing its design. Mauchly and J. Presper Eckert continued their work on a computation device at the Moore School of Electrical Engineering at the University of Pennsylvania. Their machine, the ENIAC, completed in 1945, became known as the first computer.

Dr. Atanasoff went to Washington in 1942 to become director of the Underwater Acoustics Program at the Naval Ordnance Laboratory, leaving the patent application for the ABC computer in the hands of the Iowa State attorneys. The patent application was never filed and the ABC was eventually dismantled without either Atanasoff or Berry being notified. After the war, Dr. Atanasoff was chief scientist for the Army Field Forces and director of the Navy Fuse program at the Naval Ordnance Laboratory.

In 1952, Dr. Atanasoff established The Ordnance Engineering Corporation, a research and engineering firm, which was later sold to

» continued

John Vincent Atanasoff, continued

Aerojet General Corporation. He continued to work for Aerojet until he retired in 1961.

Meanwhile, in 1947 Mauchly and Eckert applied for the patent on their ENIAC computer. Sperry Rand brought suit. The subsequent trial lasted 135 working days and filled more than 20,000 pages of transcript from the testimony of 77 witnesses, including Dr. Atanasoff. Judge Larson found that Mauchly and Eckert "did not themselves first invent the automatic electronic digital computer, but instead derived that subject matter from one Dr. John Vincent Atanasoff."

In 1990, President George Bush acknowledged Dr. Atanasoff's pioneering work by awarding him the National Medal of Technology. Dr. Atanasoff died on June 15, 1995.

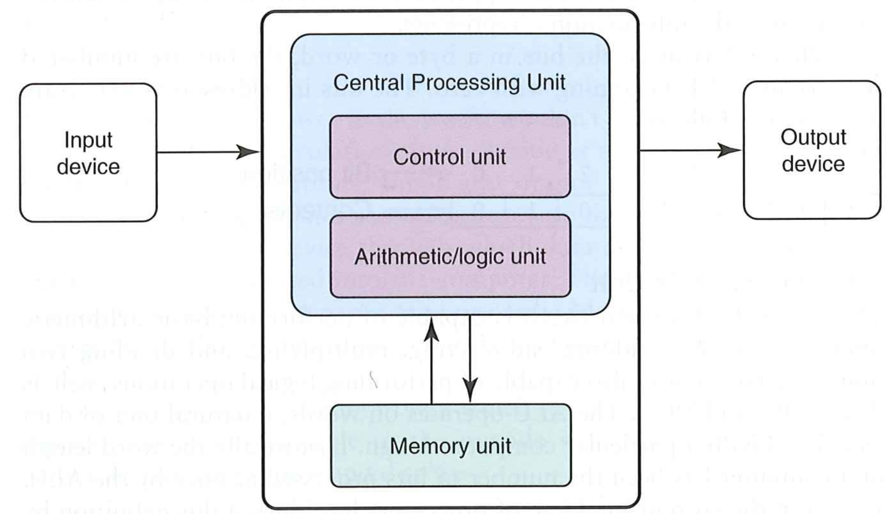


FIGURE 5.1 The von Neumann architecture

The ad in the previous section describes a memory of 4×2^{30} bytes. This means that each of the 4GB is uniquely addressable. Therefore, the addressability of the machine is 8 bits. The cells in memory are numbered consecutively beginning with 0. For example, if the addressability is 8, and there are 256 cells of memory, the cells would be addressed as follows:

Address	Contents
00000000	11100011
00000001	10101001
:	:
11111100	00000000
11111101	11111111
11111110	10101010
11111111	00110011

What are the contents of address 11111110? The bit pattern stored at that location is 10101010. What does it mean? We can't answer that question in the abstract. Does location 11111110 contain an instruction? An integer with a sign? A two's complement value? Part of an image? Without knowing what the contents represent, we cannot determine what it means: It is just a bit pattern. We must apply an interpretation on any bit pattern to determine the information it represents.

When referring to the bits in a byte or word, the bits are numbered from right to left beginning with zero. The bits in address 11111110 are numbered as follows:

7	6	5	4	3	2	1	0	← Bit position
1	0	1	0	1	0	1	0	← Contents

Arithmetic/Logic Unit

The **arithmetic/logic unit** (**ALU**) is capable of performing basic arithmetic operations such as adding, subtracting, multiplying, and dividing two numbers. This unit is also capable of performing logical operations such as AND, OR, and NOT. The ALU operates on words, a natural unit of data associated with a particular computer design. Historically the word length of a computer has been the number of bits processed at once by the ALU. However, the current Intel line of processors has blurred this definition by defining the word length to be 16 bits. The processor can work on words (16 bits), double words (32 bits), and quadwords (64 bits). In the rest of this discussion we continue to use "word" in its historical sense.

Most modern ALUs have a small number of special storage units called **registers**. These registers contain one word and are used to store information that is needed again immediately. For example, in the calculation of

One * (Two + Three)

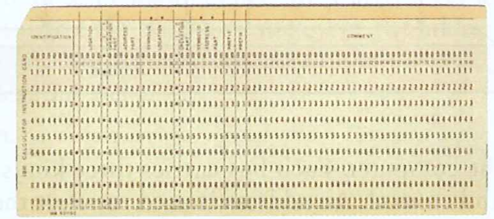
Two is first added to Three and the result is then multiplied by One. Rather than storing the result of adding Two and Three in memory and then retrieving it to multiply it by One, the result is left in a register and

Arithmetic/logic unit (ALU) The computer component that performs arithmetic operations (addition, subtraction, multiplication, division) and logical operations (comparison of two values)

Register A small storage area in the CPU used to store intermediate values or special data

Who Was Herman Hollerith?

In 1889 the United States Census Bureau realized that unless it found a better way to count the 1890 census, the results might not be tabulated before the next required census in 1900. Herman Hollerith had designed a method of counting based on cards with holes punched in them. This method was used for tabulating the census and the cards became known as Hollerith cards. Hollerith's electrical tabulating system led to the founding of the company known today as IBM.



Courtesy of Douglas W. Jones

the contents of the register are multiplied by One. Access to registers is much faster than access to memory locations.

Input/Output Units

All of the computing power in the world wouldn't be useful if we couldn't input values into the calculations from outside or report to the outside the results of those calculations. Input and output units are the channels through which the computer communicates with the outside world.

An **input unit** is a device through which data and programs from the outside world are entered into the computer. The first input units interpreted holes punched on paper tape or cards. Modern-day input devices include the keyboard, the mouse, and the scanning devices used at supermarkets.

An **output unit** is a device through which results stored in the computer memory are made available to the outside world. The most common output devices are printers and displays.

Control Unit

The **control unit** is the organizing force in the computer, for it is in charge of the fetch–execute cycle, discussed in the next section. There are two special registers in the control unit. The **instruction register** (**IR**) contains the instruction that is being executed, and the **program counter** (**PC**) contains the address of the next instruction to be executed. Because the ALU and the control unit work so closely together, they are often thought of as one unit called the central processing unit, or **CPU**.

Figure 5.2 shows a simplified view of the flow of information through the parts of a von Neumann machine. The parts are connected to one another by a collection of wires called a bus, through which data travels in

Input unit A device that accepts data to be stored in memory

Output unit A device that prints or otherwise displays data stored in memory or makes a permanent copy of information stored in memory or another device

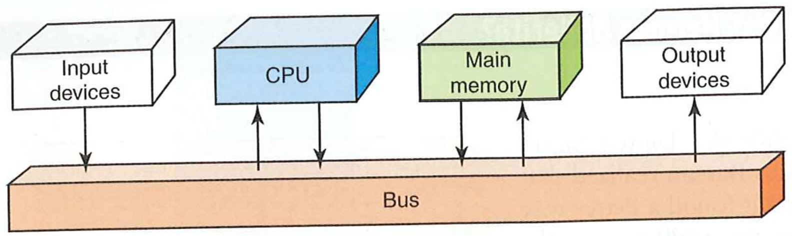
Control unit The computer component that controls the actions of the other components so as to execute instructions in sequence

Instruction register (IR) The register that contains the instruction currently being executed

Program counter (PC) The register that contains the address of the next instruction to be executed

CPU The central processing unit, a combination of the arithmetic/logic unit and the control unit; the "brain" of a computer that interprets and executes instructions

FIGURE 5.2 Data flow through a von Neumann machine



Bus width The number of bits that can be transferred in parallel over the bus

Cache memory A type of small, high-speed memory used to hold frequently used data

Pipelining A technique that breaks an instruction into smaller steps that can be overlapped

Motherboard The main circuit board of a personal computer

the computer. Each bus carries three kinds of information: address, data, and control. An address is used to select the memory location or device to which data will go, or from which it will be taken. Data then flows over the bus between the CPU, memory, and I/O devices. The control information is used to manage the flow of addresses and data. For example, a control signal will typically be used to determine the direction in which the data is flowing, either to or from the CPU. The **bus width** is the number of bits that it can transfer simultaneously. The wider the bus, the more address or data bits it can move at once.

Because memory accesses are very time consuming relative to the speed of the processor, many architectures provide **cache memory**. Cache memory is a small amount of fast-access memory into which copies of frequently used data are stored. Before a main memory access is made, the CPU checks whether the data is stored in the cache memory. **Pipelining** is another technique used to speed up the fetch–execute cycle. This technique splits an instruction into smaller steps that can be overlapped.

In a personal computer, the components in a von Neumann machine reside physically in a printed circuit board called the **motherboard**. The motherboard also has connections for attaching other devices to the bus, such as a mouse, a keyboard, or additional storage devices. (See the section on secondary storage devices later in this chapter.)

So just what does it mean to say that a machine is an *n*-bit processor? The variable *n* usually refers to the number of bits in the CPU general registers: Two *n*-bit numbers can be added with a single instruction. It also can refer to the width of the address bus, which is the size of the addressable memory—but not always. In addition, *n* can refer to the width of the data bus—but not always.

The Fetch-Execute Cycle

Before looking at *how* a computer does what it does, let's look at *what* it can do. The definition of a computer outlines its capabilities: A computer is a device that can store, retrieve, and process data. Therefore, all of the instructions that we give to the computer relate to storing, retrieving, and processing data. In Chapters 6 and 9, we look at various languages that we can use to give instructions to the computer. For our examples here, we use simple English-like instructions.

Recall the underlying principle of the von Neumann machine: Data and instructions are stored in memory and treated alike. This means that instructions and data are both addressable. Instructions are stored in contiguous memory locations; data to be manipulated are stored together in a different part of memory. To start the fetch–execute cycle, the address of the first instruction is loaded into the program counter.

The processing cycle includes four steps:

- Fetch the next instruction.
- Decode the instruction.
- Get data if needed.
- Execute the instruction.

Let's look at each of these steps in more detail. The process starts with the address in memory of the first instruction being stored in the program counter.

Fetch the Next Instruction

The program counter contains the address of the next instruction to be executed, so the control unit goes to the address in memory specified in the PC, makes a copy of the contents, and places the copy in the instruction register. At this point the IR contains the instruction to be executed. Before going on to the next step in the cycle, the PC must be updated to hold the address of the next instruction to be executed when the current instruction has been completed. Because the instructions are stored contiguously in memory, adding the number of bytes in the current instruction to the program counter should put the address of the next instruction into the PC. Thus the control unit increments the PC. It is possible that the PC may be changed later by the instruction being executed.

In the case of an instruction that must get additional data from memory, the ALU sends an address to the memory bus, and the memory responds by returning the value at that location. In some computers, data retrieved from memory may immediately participate in an arithmetic or logical operation. Other computers simply save the data returned by the memory into a register for processing by a subsequent instruction. At the end of execution, any result from the instruction may be saved either in registers or in memory.

Decode the Instruction

To execute the instruction in the instruction register, the control unit has to determine what instruction it is. It might be an instruction to access data from an input device, to send data to an output device, or to perform some operation on a data value. At this phase, the instruction is decoded into control signals. That is, the logic of the circuitry in the CPU determines which operation is to be executed. This step shows why a computer can execute only instructions that are expressed in its own machine language. The instructions themselves are literally built into the circuits.

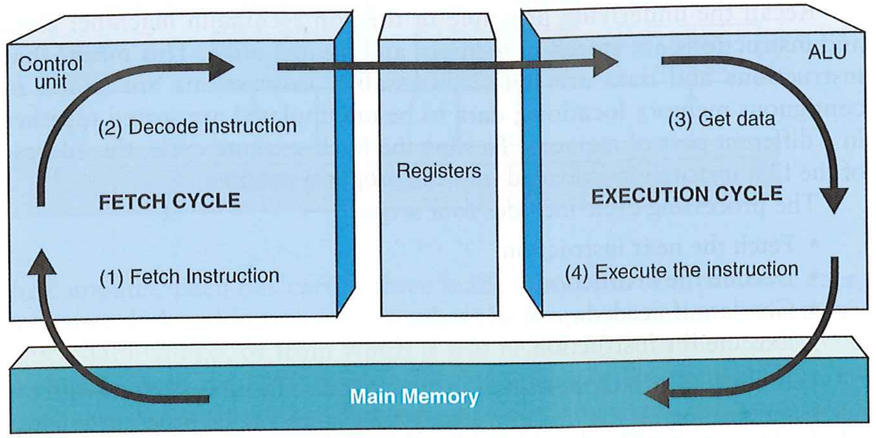


FIGURE 5.3 The fetch-execute cycle

Get Data If Needed

The instruction to be executed may potentially require additional memory accesses to complete its task. For example, if the instruction says to add the contents of a memory location to a register, the control unit must get the contents of the memory location.

Execute the Instruction

Once an instruction has been decoded and any operands (data) fetched, the control unit is ready to execute the instruction. Execution involves sending signals to the arithmetic/logic unit to carry out the processing. In the case of adding a number to a register, the operand is sent to the ALU and added to the contents of the register.

When the execution is complete, the cycle begins again. If the last instruction was to add a value to the contents of a register, the next instruction probably says to store the results into a place in memory. However, the next instruction might be a control instruction—that is, an instruction that asks a question about the result of the last instruction and perhaps changes the contents of the program counter.

Figure 5.3 summarizes the fetch–execute cycle.

Hardware has changed dramatically in the last half-century, yet the von Neumann machine remains the basis of most computers today. As Alan Perlis, a well-known computer scientist, said in 1981, "Sometimes I think the only universal in the computing field is the fetch–execute cycle." [1] This statement is still true today, nearly three decades later.

RAM and ROM

As mentioned, RAM stands for random-access memory. RAM is memory in which each cell (usually a byte) can be directly accessed. Inherent in the idea of being able to access each location is the ability to *change* the contents of each location. That is, storing something else into that place can change the bit pattern in each cell.

In addition to RAM, most computers contain a second kind of memory, called ROM. ROM stands for read-only memory. The contents in locations in ROM cannot be changed. Their contents are permanent and cannot be altered by a stored operation. Placing the bit pattern in ROM is called *burning*. The bit pattern is burned either at the time the ROM is manufactured or at the time the computer parts are assembled.

RAM and ROM are differentiated by a very basic property: RAM is volatile; ROM is not. This means that RAM does not retain its bit configuration when the power is turned off, but ROM does. The bit patterns within ROM are permanent. Because ROM is stable and cannot be changed, it is used to store the instructions that the computer needs to start itself. Frequently used software is also stored in ROM so that the system does not have to read the software in each time the machine is turned on. Main memory usually contains some ROM along with the general-purpose RAM.

Secondary Storage Devices

As mentioned earlier, an input device is the means by which data and programs are entered into the computer and stored into memory. An output device is the means by which results are sent back to the user. Because most of main memory is volatile and limited, it is essential that there be other types of storage devices where programs and data can be stored when they are no longer being processed or when the machine is not turned on. These other types of storage devices (other than main memory) are called *secondary* or *auxiliary* storage devices. Because data must be read from them and written to them, each secondary storage device is also an input and an output device.

Secondary storage devices can be installed within the computer box at the factory or added later as needed. Because these devices can store large quantities of data, they are also known as mass storage devices. For example, the hard disk drive that comes with the laptop specified in the ad can store 500×2^{30} bytes as opposed to 4×2^{30} bytes in main memory.

The next sections describe some secondary storage devices.

Magnetic Tape

Card readers and card punches were among the first input/output devices. Paper tape readers and punches were the next input/output devices. Although paper tapes, like cards, are permanent, they cannot hold much data. The first truly mass auxiliary storage device was the *magnetic tape drive*. A magnetic tape drive is like a tape recorder and is most often used to back up (make a copy of) the data on a disk in case the disk is later damaged. Tapes come in several varieties, from small streaming-tape cartridges to large reel-to-reel models.