How Computers Work: The CPU and Memory

Figure 0 shows the parts of a computer:

- The Central Processing Unit:
    - (CPU),
    - Buses,
    - Ports and controllers,
    - ROM;
- Main Memory (RAM);
- Input Devices;
- Output Devices;
- Secondary Storage;
    - floppy disks,
    - hard disk,
    - CD-ROM

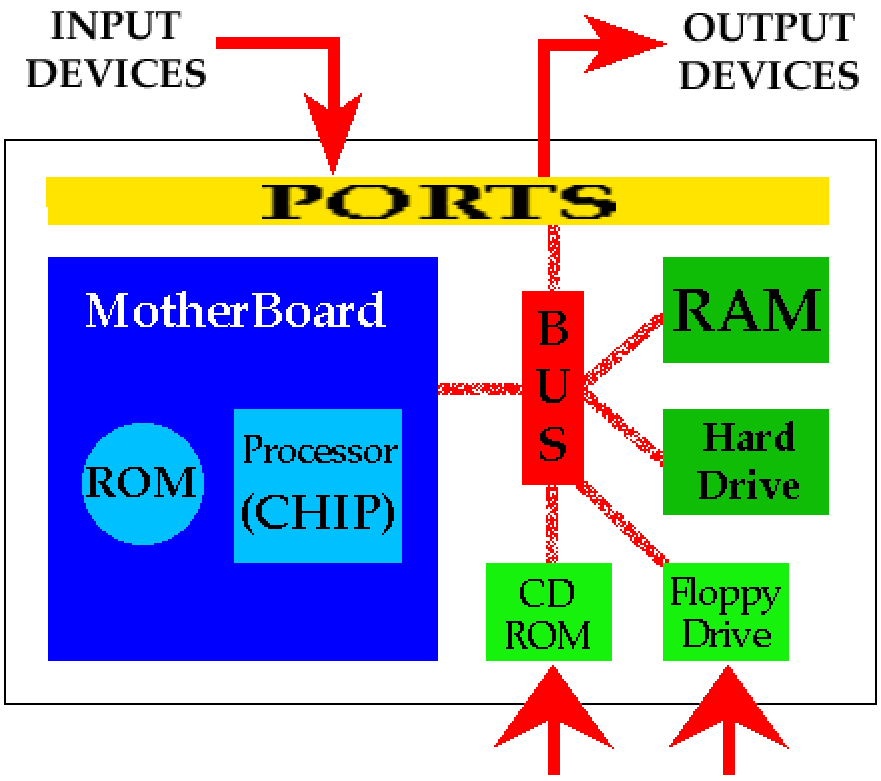


Figure 0: Inside The Computer

This part of the reading will examine the CPU, Buses, Controllers, and Main Memory. Other sections will examine input devices, output devices, and secondary memory.

The Central Processing Unit (CPU)

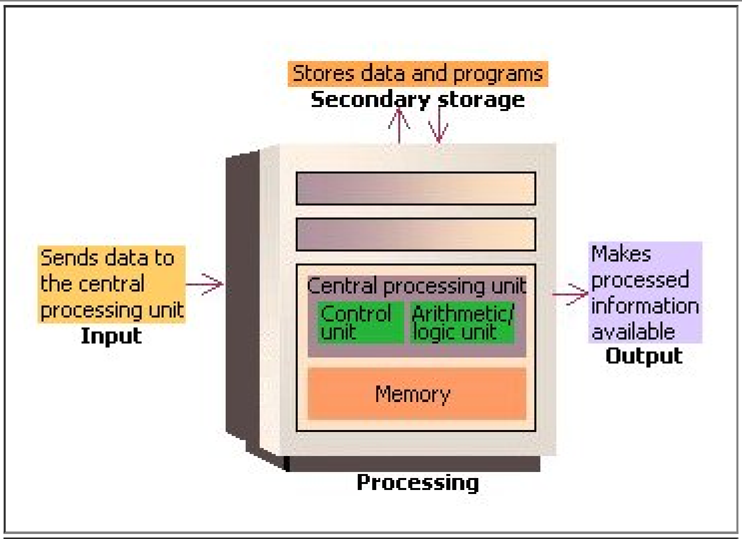


Figure 1: The Central Processing Unit

The computer does its primary work in a part of the machine we cannot see, a control center that converts data input to information output. This control center, called the central processing unit (CPU), is a highly complex, extensive set of electronic circuitry that executes stored program instructions. All computers, large and small, must have a central processing unit. As Figure 1 shows, the central processing unit consists of two parts: The control unit and the arithmetic/logic unit. Each part has a specific function.

Before we discuss the control unit and the arithmetic/logic unit in detail, we need to consider data storage and its relationship to the central processing unit. Computers use two types of storage: Primary storage and secondary storage. The CPU interacts closely with primary storage, or main memory, referring to it for both instructions and data. For this reason this part of the reading will discuss memory in the context of the central processing unit. Technically, however, memory is not part of the CPU.

Recall that a computer's memory holds data only temporarily, at the time the computer is executing a program. Secondary storage holds permanent or semi-permanent data on some external magnetic or optical medium. The diskettes and CD-ROM disks that you have seen with personal computers are secondary storage devices, as are hard disks. Since the physical attributes of secondary storage devices determine the way data is organized on them, we will discuss secondary storage and data organization together in another part of our on-line readings.

Now let us consider the components of the central processing unit.

- **The Control Unit**

The control unit of the CPU contains circuitry that uses electrical signals to direct the entire computer system to carry out, or execute, stored program instructions. Like an orchestra leader, the control unit does not execute program instructions; rather, it directs other parts of the system to do so. The control unit must communicate with both the arithmetic/logic unit and memory.

- **The Arithmetic/Logic Unit**

The arithmetic/logic unit (ALU) contains the electronic circuitry that executes all arithmetic and logical operations.

The arithmetic/logic unit can perform four kinds of arithmetic operations, or mathematical calculations: addition, subtraction, multiplication, and division. As its name implies, the arithmetic/logic unit also performs logical operations. A logical operation is usually a comparison. The unit can compare numbers, letters, or special characters. The computer can then take action based on the result of the comparison. This is a very important capability. It is by comparing that a computer is able to tell, for instance, whether there are unfilled seats on airplanes, whether charge- card customers have exceeded their credit limits, and whether one candidate for Congress has more votes than another.

Logical operations can test for three conditions:

- **Equal-to condition.** In a test for this condition, the arithmetic/logic unit compares two values to determine if they are equal. For example: If the number of tickets sold equals the number of seats in the auditorium, then the concert is declared sold out.
- **Less-than condition.** To test for this condition, the computer compares values to determine if one is less than another. For example: If the number of speeding tickets on a driver's record is less than three, then insurance rates are $425; otherwise, the rates are $500.
- **Greater-than condition.** In this type of comparison, the computer determines if one value is greater than another. For example: If the hours a person worked this week are greater than 40, then multiply every extra hour by 1.5 times the usual hourly wage to compute overtime pay.

A computer can simultaneously test for more than one condition. In fact, a logic unit can usually discern six logical relationships: *equal to, less than, greater than, less than or equal to, greater than or equal to*, and *not equal*.

The symbols that let you define the type of comparison you want the computer to perform are called relational operators. The most common relational operators are the equal sign(=), the less-than symbol(<), and the greater-than symbol(>).

- **Registers: Temporary Storage Areas**
  Registers are temporary storage areas for instructions or data. They are not a part of memory; rather they are special additional storage locations that offer the advantage of speed. Registers work under the direction of the control unit to accept, hold, and transfer instructions or data and perform arithmetic or logical comparisons at high speed. The control unit uses a data storage register the way a store owner uses a cash register-as a temporary, convenient place to store what is used in transactions.

Computers usually assign special roles to certain registers, including these registers:

- **An accumulator**, which collects the result of computations.
- **An address register**, which keeps track of where a given instruction or piece of data is stored in memory. Each storage location in memory is identified by an address, just as each house on a street has an address.
- **A storage register**, which temporarily holds data taken from or about to be sent to memory.
- A general-purpose **register**, which is used for several functions.

- **Memory and Storage**
  Memory is also known as primary storage, primary memory, main storage, internal storage, main memory, and RAM (Random Access Memory); all these terms are used interchangeably by people in computer circles. Memory is the part of the computer that holds data and instructions for processing. Although closely associated with the central processing unit, memory is separate from it. Memory stores program instructions or data for only as long as the program they pertain to is in operation. Keeping these items in memory when the program is not running is not feasible for three reasons:
  - Most types of memory only store items while the computer is turned on; data is destroyed when the machine is turned off.
  - If more than one program is running at once (often the case on large computers and sometimes on small computers), a single program can not lay exclusive claim to memory.
  - There may not be room in memory to hold the processed data.

  How do data and instructions get from an input device into memory? The control unit sends them. Likewise, when the time is right, the control unit sends these items from memory to the arithmetic/logic unit, where an arithmetic operation or logical operation is performed. After being processed, the information is sent to memory, where it is hold until it is ready to he released to an output unit.

  The chief characteristic of memory is that it allows very fast access to instructions and data, no matter where the items are within it. We will discuss the physical components of memory-memory chips-later in this chapter.

  To see how registers, memory, and second storage all work together, let us use the analogy of making a salad. In our kitchen we have:

  - a refrigerator where we store our vegetables for the salad;
  - a counter where we place all of our veggies before putting them on the cutting board for chopping;
  - a cutting board on the counter where we chop the vegetables;
  - a recipe that details what veggies to chop;
  - the corners of the cutting board are kept free for partially chopped piles of veggies that we intend to chop more or to mix with other partially chopped veggies.
  - a bowl on the counter where we mix and store the salad;
  - space in the refrigerator to put the mixed salad after it is made.

  The process of making the salad is then: bring the veggies from the fridge to the counter top; place some veggies on the chopping board according to the recipe; chop the veggies, possibly storing some partially chopped veggies temporarily on the corners of the cutting board; place all the veggies in the bowl to either put back in the fridge or put directly on the dinner table.

  The refrigerator is the equivalent of secondary (disk) storage. It can store high volumes of veggies for long periods of time. The counter top is the equivalent of the computer's motherboard - everything is done on the counter (inside the computer). The cutting board is the ALU - the work gets done there. The recipe is the control unit - it tells you what to do on the cutting board (ALU). Space on the counter top is the equivalent of RAM memory - all veggies must be brought from the fridge and placed on the counter top for fast access. Note that the counter top (RAM) is faster to access than the fridge (disk), but can not hold as much, and can not hold it for long periods of time. The corners of the cutting board where we temporarily store partially chopped veggies are equivalent to the registers. The corners of the cutting board are very fast to access for chopping, but can not hold much. The salad bowl is like a temporary register, it is for storing the salad waiting to take back to the fridge (putting data back on a disk) or for taking to the dinner table (outputting the data to an output device).

Now for a more technical example. let us look at how a payroll program uses all three types of storage. Suppose the program calculates the salary of an employee. The data representing the hours worked and the data for the rate of pay are ready in their respective registers. Other data related to the salary calculation-overtime hours, bonuses, deductions, and so forth-is waiting nearby in memory. The data for other employees is available in secondary storage. As the CPU finishes calculations about one employee, the data about the next employee is brought from secondary storage into memory and eventually into the registers.

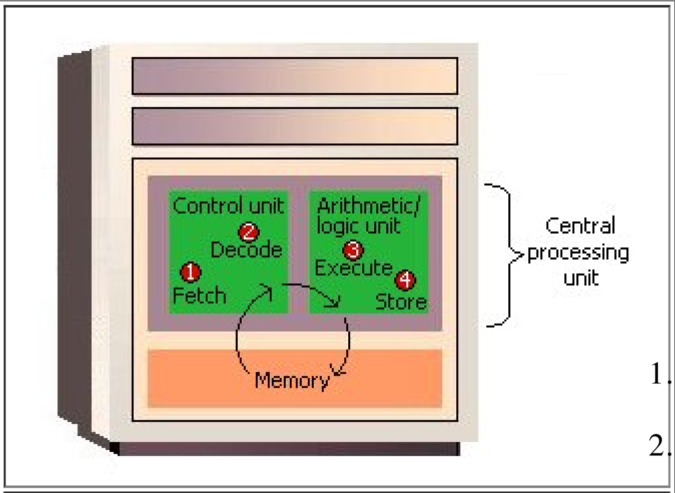
The following table summarizes the characteristics of the various kinds of data storage in the storage hierarchy.

Storage	Speed	Capacity	Relative Cost ($)	Permanent?
Registers	Fastest	Lowest	Highest	No
RAM	Very Fast	Low/Moderate	High	No
Floppy Disk	Very Slow	Low	Low	Yes
Hard Disk	Moderate	Very High	Very Low	Yes

Modern computers are designed with this hierarchy due to the characteristics listed in the table. It has been the cheapest way to get the functionality. However, as RAM becomes cheaper, faster, and even permanent, we may see disks disappear as an internal storage device. Removable disks, like Zip disks or CDs (we describe these in detail in the online reading on storage devices) will probably remain in use longer as a means to physically transfer large volumes of data into the computer. However, even this use of disks will probably be supplanted by the Internet as the major (and eventually only) way of transferring data. Floppy disks drives are already disappearing: the new IMac Macintosh from Apple does not come with one. Within the next five years most new computer designs will only include floppy drives as an extra for people with old floppy disks that they must use.

For more detail on the computer's memory hierarchy, see the How Stuff Works pages on computer memory.. *This is optional reading*.

- **How the CPU Executes Program Instructions**
  Let us examine the way the central processing unit, in association with memory, executes a computer program. We will be looking at how just one instruction in the program is executed. In fact, most computers today can execute only one instruction at a time, though they execute it very quickly. Many personal computers can execute instructions in less than one-millionth of a second, whereas those speed demons known as supercomputers can execute instructions in less than one-billionth of a second.



Before an instruction can be executed, program instructions and data must be placed into memory from an input device or a secondary storage device (the process is further complicated by the fact that, as we noted earlier, the data will probably make a temporary stop in a register). As Figure 2 shows, once the necessary data and instruction are in memory, the central processing unit performs the following four steps for each instruction:

1. The control unit fetches (gets) the instruction from memory.
2. The control unit decodes the instruction (decides what it means) and directs that the necessary data

Figure 2: The Machine Cycle

be moved from memory to the arithmetic/logic unit. These first two steps together are called instruction time, or I-time.

3. The arithmetic/logic unit executes the arithmetic or logical instruction. That is, the ALU is given control and performs the actual operation on the data.
4. Thc arithmetic/logic unit stores the result of this operation in memory or in a register. Steps 3 and 4 together are called execution time, or E-time.

The control unit eventually directs memory to release the result to an output device or a secondary storage device. The combination of I-time and E-time is called the machine cycle. Figure 3 shows an instruction going through the machine cycle.

Each central processing unit has an internal clock that produces pulses at a fixed rate to synchronize all computer operations. A single machine-cycle instruction may be made up of a substantial number of sub-instructions, each of which must take at least one clock cycle. Each type of central processing unit is designed to understand a specific group of instructions called the instruction set. Just as there are many different languages that people understand, so each different type of CPU has an instruction set it understands. Therefore, one CPU-such as the one for a Compaq personal computer-cannot understand the instruction set from another CPU-say, for a Macintosh.

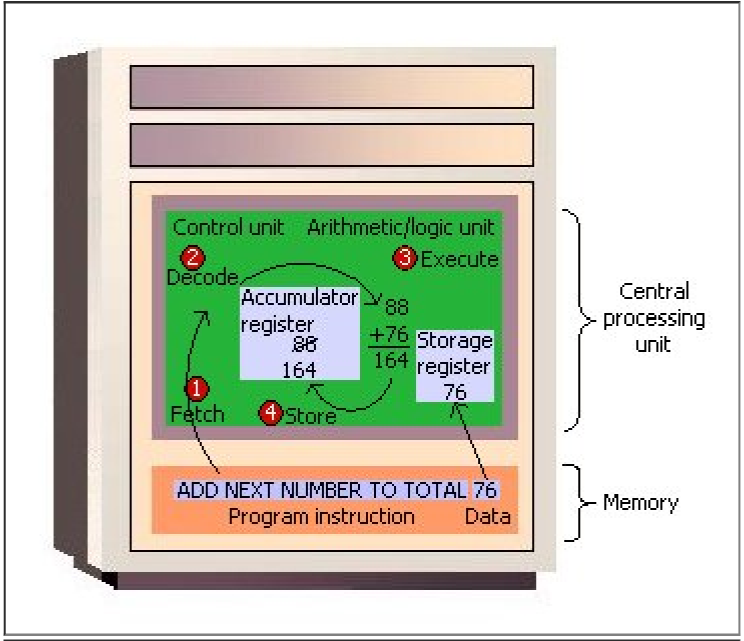


Figure 3: The Machine Cycle in Action

It is one thing to have instructions and data somewhere in memory and quite another for the control unit to be able to find them. How does it do this?

The location in memory for each instruction and each piece of data is identified by an address. That is, each location has an address number, like the mailboxes in front of an apartment house. And, like the mailboxes, the address numbers of the locations remain the same, but the contents (instructions and data) of the locations may change. That is, new instructions or new data may be placed in the locations when the old contents no longer need to be stored in memory. Unlike a mailbox, however, a memory location can hold only a fixed amount of data; an address can hold only a fixed number of bytes - often two bytes in a modern computer.

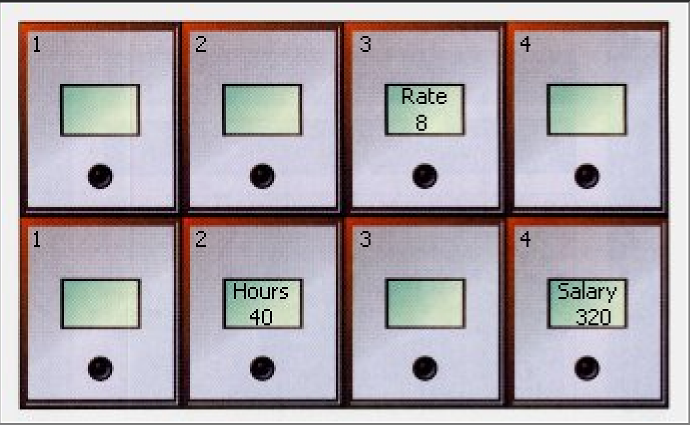


Figure 4: Memory Addresses Like Mailboxes

Figure 4 shows how a program manipulates data in memory. A payroll program, for example, may give instructions to put the rate of pay in location 3 and the number of hours worked in location 6. To compute the employee's salary, then, instructions tell the computer to multiply the data in location 3 by the data in location 6 and move the result to location 8. The choice of locations is arbitrary - any locations that are not already spoken for can be used. Programmers using programming languages, however, do not have to worry about the actual address numbers, because each data address is referred to by a name. The name is called a symbolic address. In this example, the symbolic address names are Rate, Hours, and Salary.

Now that we see conceptually how a computer works, we will look at the hardware components that make up the internals os a modern computer. Click here to continue the required reading.