

Computers input, process, and output information

Richard K. Herz, ReactorLab.net

Input

Examples of input devices include keyboards, touch screens, joy sticks, cameras, microphones, strain gauges, photocells, credit card chip readers.

A sensor responds to information from the environment (light, pressure, chemical) to produce a signal of some type, often an electrical current or voltage. A transducer is a device that can convert one type of signal into another. A digital input/output (I/O) device converts a signal to binary numbers. An analog-to-digital (A/D) converter converts an electrical voltage or current into binary numbers.

Encoding of input information

All information that comes into a digital computer must be converted to numbers in the binary number system. This is because all communication and processing in digital computers and digital communication systems uses binary numbers, for reasons explained below.

Alphanumeric characters are converted to binary numbers via the process of a table of conversions. Conversion table systems include ASCII and several varieties of Unicode. For example, "A" is first converted in ASCII to the decimal number 65 and then to the binary equivalent of 65, which is 01000001. Bits are added to denote that the value is a character value. ("a" is 97)

Sound is converted to a continuous electrical voltage signal in a microphone. Then the continuous signal is periodically "sampled" by converting the signal value to a binary number in an analog-to-digital converter. This results in a series of binary numbers, each representing a signal value separated by a known time interval.

Images can be segmented into a grid of picture elements called pixels. The color values and intensity of each pixel can be stored as numbers in 2D arrays. These values are converted to binary numbers.

Processing of information - hardware

A digital computer or a digital communication system is built out of nanometer sized physical devices, each of which can be in one of two, binary states. Each binary state is called a bit. The states of a collection of devices - a collection of bits called a byte - can represent the binary equivalent of a decimal number.

Examples of physical data storage devices include transistors in integrated circuits (on-off), capacitors (charged-discharged), field orientation of magnetic elements in hard disks (N-S), and optical reflectivity of organic dyes or aluminum films in DVDs (specular-diffuse).

There are several reasons that digital computers process digital information. These reasons include:

- high density of information
 - binary-state devices can be simple and made very small

- math can be used to compress information into a smaller number of bit states.
- insensitivity to noise - binary states are only sensitive to values above or below a threshold value, not to specific values to which noise may have been added
- error detection and correction - since data have been converted to numbers, math can be used to do this

Processing of information - software

There are two general types of information: data and instructions. Instructions are statements that tell the computer what to do. Collections of instructions are software programs.

Both types of information are in the form of binary numbers when they interact with the computer's hardware. Any one set of physical devices can store either data or an instruction. This is a very useful and important characteristic of digital computers.

Types of software programs include the operating system, applications (apps) that we use, and software programs that we are learning to write.

When we write a program, we input information that we can read. In many cases, we use a computer language that looks vaguely similar to English. Our program must be converted to binary numbers or "machine code" in order to run on the computer hardware. This can occur in three ways.

- For "compiled" computer languages such as C, C++ and Fortran, we must "compile" or translate our program using an app into binary machine code that can be processed or "run" directly on the hardware.
- For computer languages such as Java and LiveCode, we compile the program using an app into a "byte code" program that can be run on any of several types of operating systems and hardware. A byte code program then would be run by a compiled program designed for a specific type of operating system and hardware.
- For "interpreted" computer languages such as Matlab, we input single instructions in the Command Window, or a series of instructions in m-files. These instructions are "interpreted" and processed by a compiled program designed for a specific type of operating system and hardware.

In the early days of computers, compiled programs were preferred over interpreted programs for long calculations. This was because compiled programs run directly on the hardware and, thus, run much faster. Now, computer hardware is so fast that we can often use an interpreted program in many cases and not notice the difference in speed (what's a millisecond or two?).

Decoding of input information

Before results are output, they are converted from binary numbers into a form that can be processed by the output device. This decoding can be done by a digital input-output (I/O) device that is located in the computer, in a box on the signal cable, or in the output unit.

Output

Examples of output devices include video display screens, audio speakers, and mechanical actuators.

Examples of mechanical actuators include the device that spits \$20 bills out of an ATM, the vibrator of a cell phone or an Apple Watch (haptic device in fancy words), and fluid flow control valves such as those in automotive anti-lock braking systems.

Vocabulary

sensor
transducer
digital input/output (I/O) device
analog-to-digital converter (A/D or ADC)
digital computer
binary number system
Alphanumeric character
ASCII
Unicode
Sound sample
pixel
nanometer
bit
byte
threshold value
computer instructions, statements, commands
machine code
compile
byte code program
interpreted computer language