Digital Computer

A digital computer is a device for storing and manipulating numbers using a small set of discrete states of a given physical system. It is the opposite of an analog computer, in which quantities are represented using a continuous variable. A watch, with blades rotating smoothly and continuously around the clock face, is a good example of an analog system: any position of the blades is allowed. In contrast, counting using our fingers is an example of a digital process. Only full values are meaningful and we don't count using "half a finger". In fact, the word "digital" comes from "digits". Digital computers count using "digits", i.e. a discrete representation of numbers.

The earliest example of a digital computer is the Analytical Engine, designed by Charles Babbage in England in the nineteenth century, but never completed. The Analytical Engine, stored numbers using rotating wheels. Each of them could represent ten digits, from zero to nine, depending on its angle relative to the rest position. This was the method used to store numbers in mechanical calculators until they were replaced by electronic computers and electronic calculators in the twentieth century.

Modern digital computers operate using the binary system. The only digits needed are "0" and "1", i.e. the binary digits or "bits". Binary components are usually cheaper and easier to manufacture than components for other numerical bases. However, when the first digital computers were built in the USA and in Europe, it was not yet totally clear which technology should be used for the internal design. The American ENIAC and Mark I computers used binary signals but also a decimal internal representation for numbers. They were digital computers with a hybrid numerical base.

It is easy to illustrate the importance of the numerical base for digital computers. Assume that numbers are represented using small pieces of cardboard. On each of them we can only write a single decimal digit, from zero to nine. If we want to represent decimal numbers with 5 digits, we need 50 cardboard pieces, that is, five nines, five eights, etc. We need each digit five times because we do not know in advance which numbers we will be asked to display with our cardboard pieces. If each cardboard costs $1 to be produced, then our "cardboard display" has a total cost of $50 and can display all integers from 00000 to 99999.

However, if we invest the same $50 to print 25 cardboards with a "0" and 25 with a "1", we can represent numbers using 25 bits. With so many bits we can display any number between 0 and 2^{25}, that is, between 0 and 33,554,432. These are far more numbers than in the previous case (almost a factor 34 more!) and from this simple example we conclude that the binary system is more efficient for representing numbers, if the cost per digit is the same among the different bases. Curiously, central bank authorities face the same problem when setting the nominal value of paper money. Usually a compromise is made between base 2 (which is efficient) and base 10 (which is easier to read) as shown by the fact that in many countries the bills have a face value of 1,2,5,10,20,50,100, etc.

Actually, it can be proved that base 3 is still more efficient than base 2, which is only the second most efficient base. However, logic components for base 3 are more expensive than for base 2 and therefore digital computers have stayed with base 2. Some computer prototypes with base 3 logic elements were built in the 1960s in the US and in the former USSR (the SETUN machine).

References

Malvino, Albert, *Digital Computer Electronics*, McGraw Hill, New York, 1993.

Min, J.L, Schrage, J.J., *Design of Analog and Digital Control Systems*, Horwood, Chichester, 1988.

Raul Rojas