INPUT DEVICE

An *input device* feeds information, in the form of digital electrical signals, into a computer or some other processor-based system. The most commonly-used input device is the keyboard, although other methods, such as speech input and the hand-held mouse, are becoming increasingly popular. Other input systems include light pens, graphics tablets and, of course, the joystick. Many mainframe computers use paper tape and punch card readers to input data.

Input Device
The mouse is an attempt to create an input device that is easier to use, particularly for novices to the standard typewriter keyboard

CHRIS STEVENS

INPUT/OUTPUT

Input/Output (I/O) is the part of a computer system that handles the transfer of information to and from the central processing unit. Without a system of input/output, the CPU would be useless, since it would lack any communication with the outside world. The major purpose of input/output is to allow users to communicate with the processor.

The primary function of an I/O system, then, is to translate signals from the outside world into a form of data that can be understood by the CPU. However, this is only a part of the job of input/output. The system must also decide which of the several possible external devices (keyboard, joystick, printer, etc.) is being used at the time, and whether data is being written to or read from that device. Clearly, input/output tasks require a lot of attention — particularly if you consider external devices such as televisions, which have displays that need to be constantly updated. Therefore, most computers have a separate I/O chip set aside specifically to perform these tasks.

Input/output is most commonly associated with mass storage devices that can be used for either input or output, such as cassette tape decks and floppy disk drives.

INSTRUCTION

An *instruction* is a group of characters or digits that specifies an operation to be carried out by the computer. An instruction consists of an operation code (*op-code*), which is the action to be performed, and an *operand*, the data on which the operation is to be performed or its address. For example, in the instruction LDY 8, the op-code is LDY and the operand is 8.

Instructions can be grouped into three different types; *arithmetic* instructions, which perform arithmetical operations such as add, subtract, divide and multiply; *logic* instructions, which perform logical operations such as AND or OR; and *input/output* instructions, which perform I/O operations.

INSTRUCTION COUNTER

The *instruction counter* (also known as the 'program counter' or the 'current address register') is a register that is normally found in the CPU, which holds the memory address of the current instruction being executed. As the execution of a program proceeds, the instruction counter is added to in single steps, continually updating the address held as it moves through the sequence of instructions.

When the program encounters a branch or jump instruction, the instruction counter will have its contents altered by the operation. If this is a branch command, the current address held in the instruction counter will be PUSHed onto the stack, and the new address placed in the instruction counter. At the end of the subroutine, the address will be PULLed off the stack and placed back in the instruction counter. This will usually be done automatically by the processor and there is no need for the programmer to update the instruction counter manually.

INSTRUCTION SET

The *instruction set* is the list of instructions that a given microprocessor is able to perform. This is the irreducible set of instructions from which all other higher-level operations, such as BASIC commands, can be constructed. The instruction set is burned into the ROM on construction, and the extent and power of the instruction set can, therefore, vary according to the architecture of the chip itself. An instruction set should include operations to access and manipulate the contents of the registers and the accumulator. There should also be provision for branching — both conditional and non-conditional. Finally, there should be a number of instructions set aside to test and clear the status flags.

INTEGER

An *integer*, also known as a 'whole number', is a number that contains no fractional element. An integer can be positive, negative or zero. For example, 56, −43 and 0 are all integers, but 3.8, −2.001 and $3\frac{1}{2}$ are not.

An integer is not recognised as such by a computer, it is simply that we interpret it that way. For example, taking the eight-bit binary pattern 10000001, we can look on it in several different ways:

Binary Number	Interpretation	Decimal Equivalent
10000001	Unsigned integer	129
10000001	Sign and magnitude	−1
10000001	Two's complement	−127