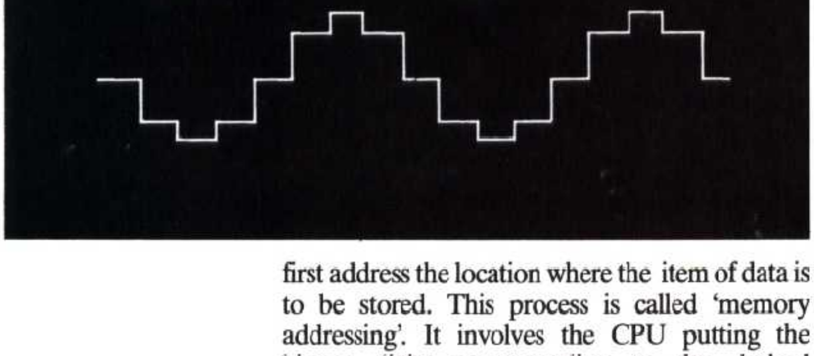
# Digital Dialogue

## Input and Output are essential to the operation of any computer system

**Analogue To Digital**
In the real world, few pieces of information come in discrete, digital steps. Rather, they are infinitely variable like noise levels or the tides.

In order to make these data comprehensible to the computer, the signal must first be digitised. The Analogue-to-Digital (A/D) converter takes samples from the signal source at a known, constant rate — perhaps one hundred every second. Each of these samples is stored in a separate memory location as a digital value, thus allowing calculations of variance to be made, and out-of-limits conditions to be recognised.

Digital-to-Analogue (D/A) converters perform a similar function in reverse, statistical techniques being used to smooth out the peaks into a regular curve


MARK WATKINSON

Input/Output, or I/O as it is commonly abbreviated, is the term used to describe the transfer of information between the CPU (the Central Processing Unit forming the heart of the computer) and the 'outside world'. The 'outside world' in this context means any devices that may be connected to the computer. It does not include RAM and ROM memory, which are considered to be integral to the computer. The distinction between what is held to be 'inside' the computer and 'outside' it is somewhat arbitrary. But all of the logic circuits (see page 92), designed to work in close conjunction with the CPU and main memory, are considered to be part of the 'inside' of the computer.

External devices, which use I/O for communication with the computer, include a wide variety of peripherals ranging from the keyboard to floppy disk drives, joysticks, printers and video display units.

When the CPU wants to retrieve data from memory, it has first to 'address' the location where the byte of data is stored. Similarly, if the CPU wants to store a byte of data for later use, it must first address the location where the item of data is to be stored. This process is called 'memory addressing'. It involves the CPU putting the binary digits corresponding to the desired memory location on a set of 16 wires connected to the CPU's 'address pins'. These wires are called the 'address bus'. Special circuitry in the memory section is able to decode these 16 binary digits to select the correct memory location. (Sixteen binary digits can give 65,536 unique combinations of ones and zeros and can therefore address that many different memory locations.)

If the computer wants to communicate with an external device, it also has to 'locate' that device in a similar way. Only eight address lines are available. This limits the total number of separate I/O locations that can be selected to 256. This is a small number compared with the addressing power of 16 address lines, but in practice 256 is more than adequate. There's usually no need for huge numbers of external devices to be connected to a computer.

## Selecting Devices

To find out how the computer actually selects an external device and sends data to it, let's consider one of the simplest output devices possible — an LED (Light Emitting Diode) mounted on the keyboard of the computer to show when the 'caps lock' key has been pressed (there's a key and an LED like this on the BBC Microcomputer). To the computer, the LED is simply another external device to which it can send data. In the case of a single LED, the data will be either a single 1 (to turn the LED on) or a single 0 (to turn the LED off). Even though it is just a humble LED requiring a single bit of data, it still needs to have an address or location. The CPU can't spend its whole time addressing one LED. It needs to be able to select the LED once only to tell it when it needs to switch on, and again to tell it when to switch off. Suppose, for the sake of argument, the LED has an I/O address of 32. To select it, the address lines will have to be set by the CPU to the binary equivalent of 32. This is 00100000 in binary. The LED will have a special 'decoder' circuit that will ignore all other combinations of bits on the address lines. When the address line becomes 00100000, this decoder circuit recognises it and produces a high voltage and therefore a 'true' output. The next part of the circuit needed to switch on the LED is a small chip called a 'data latch'. This latches or holds the data sent to it so that the LED stays on or off until the next time it is addressed and new data is sent to it. This process is known as 'toggling'.

Most of the external devices with which the computer communicates are considerably more complex than a single LED. A printer is a typical peripheral and each time the computer communicates with it, the data transmitted will represent the code for a whole character to be printed. Usually, when large amounts of