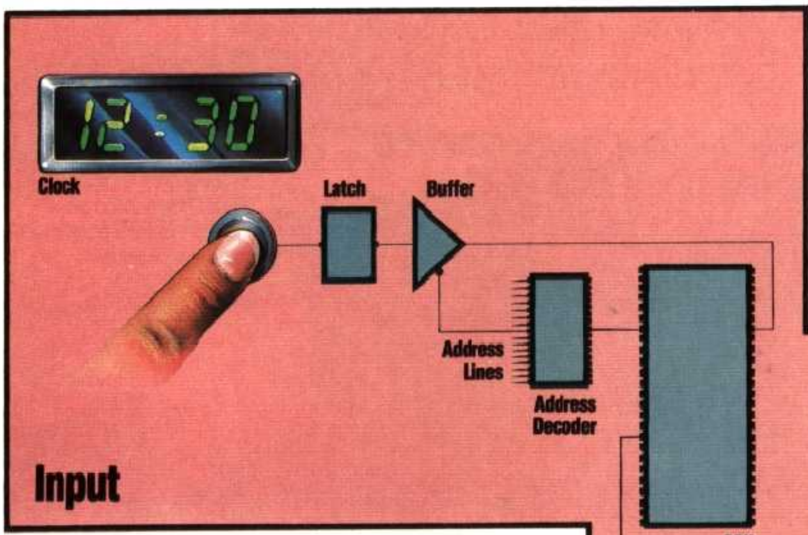
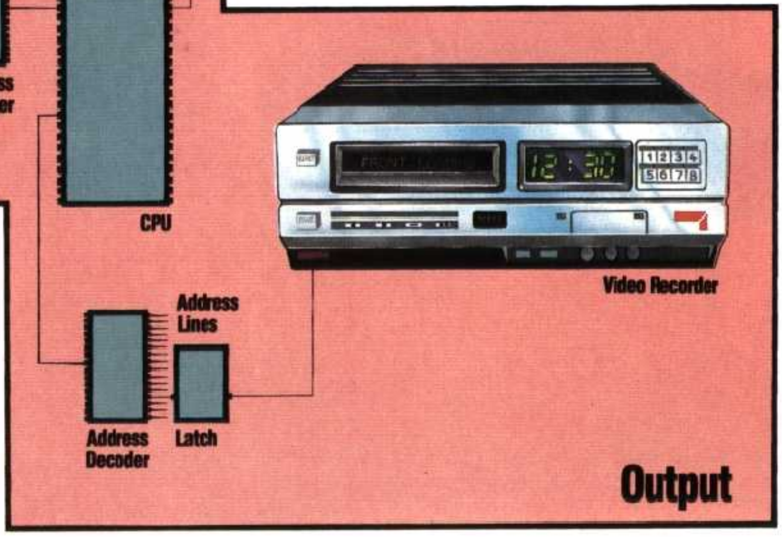




Input/Output
In the simplest control applications, such as that illustrated, the CPU handles only a single piece of information, whether or not a switch has been pressed. The buffer — itself a short-term memory — simply holds the data until the CPU next 'polls' the device in question. The address decoder indicates the source of each signal, and when a change of state is recognised, i.e., that the switch has been pressed, the CPU delivers an appropriate response, in this case changing the clock display from actual time to the time at which the VCR's auto timer will turn on the recorder. Within the output stage, the same procedure works in reverse

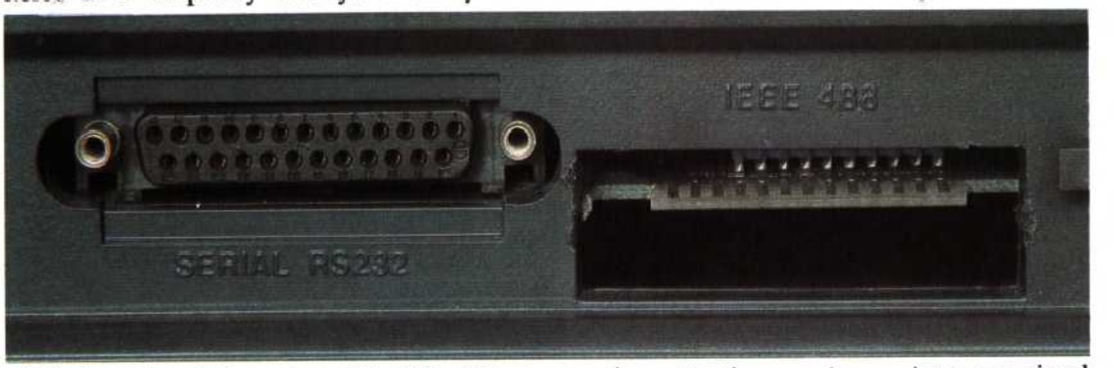
information are to be transferred, as for a printer, a special I/O interface chip is used. Such chips simplify the task of the computer engineer, because the interface circuit is designed to incorporate most of the circuitry needed into one chip. One of the most popular of these is the 8255 PPI (Programmable Peripheral Interface). This 40-pin chip contains three eight-bit I/O ports. That means there are 24 I/O pins on the chip, eight pins each for I/O ports A, B and C. Each of these ports can send eight bits (one byte's worth) of data at a time to a peripheral device such as a printer, or receive eight bits of data at a time from an input device such as a keyboard.

To send eight bits of data to a printer, the CPU will first address the PPI and then send it the eight bits of data on the data bus. This data will be stored in a temporary one-byte memory cell within the chip, called a register. The PPI will then make this data available on the appropriate set of I/O pins. A similar principle, but working in reverse, allows data from external input devices to be stored in a register in the chip, and then put onto the data bus when the CPU sends it the appropriate signal. As noted above, external devices cannot be allowed to put their data onto the computer's data bus continuously — it is needed to transfer data to and from memory and by other I/O devices. The I/O chip stores the data temporarily and only puts this data on the data bus (to be picked up by the CPU) when the CPU tells it to do so.

How does the CPU know if an external device is trying to send data to the computer? Briefly, there are two main techniques that can be used. The CPU stops running the program it is executing periodically and takes a quick look at all the input ports. If it finds data there waiting to be input, it instructs the port to put the data on the data bus. The process of enquiry of the input devices is known as 'polling'.

The other method uses 'interrupts'. The device wanting attention sends an interrupt signal directly to the CPU and this forces the program being executed to stop while the input port is attended to. The advantages and disadvantages of these two methods will be described in more detail later in the course.

The I/O we have described so far is called 'parallel I/O' because data is input or output one byte at a time using eight I/O wires or lines (eight bits in parallel). Another technique is called 'serial I/O'. Here the information in each byte is fed in or out a bit at a time, one bit after the other. Some printers use serial interfaces, and the output from modems (see page 108) is also in serial form. The main advantage is that, essentially, serial communication allows a single pair of wires to be used instead of eight or more.



Serial And Parallel Ports
Most modern microcomputers provide both serial and parallel ports, the former passing data a bit at a time, the latter in whole bytes. The most common type of serial convention, known as RS232C, uses either a 'D-type' sub-miniature connector, a 25-pin example of which is illustrated here (left), or more rarely a DIN plug like those used in hi-fi systems.

The parallel port (right) follows the IEEE488 convention, developed by Hewlett Packard and adopted as an industry standard by the Institute of Electrical and Electronics Engineers in the USA