



## BACKGROUND

If you are using a word processing package to create a document, and are typing continuously, then you may be surprised to learn that you are probably using the computer at less than 10 per cent of its efficiency. You probably can't manage more than ten keystrokes per second, yet the computer could process more than ten times that number — scanning the keyboard, interpreting the key pressed, storing it in RAM and displaying it on the screen. The result is that the computer spends approximately 90 per cent of its time simply executing a loop that waits for you to press a key.

There are obviously gains to be made if, while it was waiting, the computer could be put to good use elsewhere, and this is where the concept of *background* processing comes in. The computer executes one task in the foreground, which the user sees on his screen and interacts with, while another task is carried out in the background. The user doesn't interface with the background task, except at the start and finish of the job.

It follows, of course, that only certain types of job are suitable for processing in the background. One of these is sorting — a job that requires a lot of computing power, but where the user isn't directly involved. Another is printing: a background system would allow you to create or edit one document, while printing an already created one in the background. Another computer-intensive task that could be done in the background is the posting of invoices into a sales ledger.

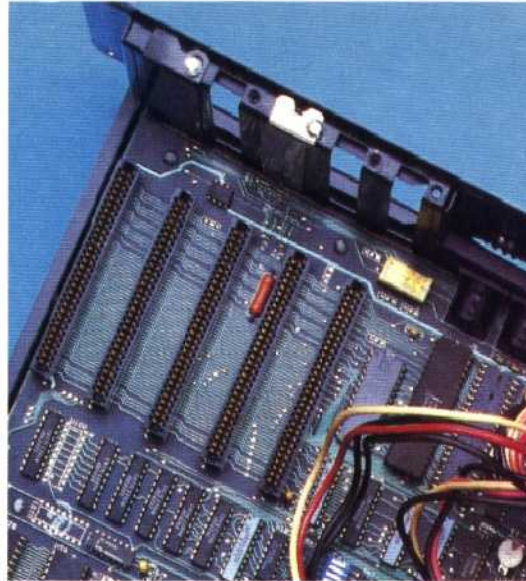
A computer-intensive job like sorting won't execute as fast in the background as it would if it had the whole computer to itself, but this isn't always a disadvantage. Most foreground-background systems are based on timesharing: the processor spends, say, 1/100th of a second seeing to the foreground, then 1/100th of a second in the background, and so on. Large mainframe computers can share their time between hundreds of programs and hundreds of users at once.

A few microcomputer applications programs have such a system built-in, but generally it is necessary to buy a special operating system (referred to as a 'multi-tasking' or 'concurrent' operating system). There is, for example, a concurrent version of the popular CP/M system.

## BACKPLANE

Most home computers are now sold as self-contained units and come with strong warnings about the consequences of opening up the casing. But the first generation of microcomputers were based on a quite different concept. The computer came with a casing, power supply, and something called a *backplane*, which was a kind of board or frame featuring several sockets that connected together any devices or printed circuit boards that were plugged in.

The advantages of the backplane system were



CHRIS STEVENS

not just that it was easy to expand to include additional memory or interfaces, but that you could more or less *configure* (that is, design and put together) the computer as you wanted it.

Some of the backplane concept has been incorporated into contemporary educational and business machines, such as the Research Machines 380Z, the Apple IIe and the IBM PC, for which a large range of add-on boards are available. One business system, the Tycom Microframe, takes this one stage further and permits the user to choose the processor used, as well as the RAM and interfaces.

All the boards and devices that plug into the backplane must use the same set of connections for the bus. One of the common standards among early machines was called the S-100 (so-called because it had one hundred available lines), and there was a cut-down version called the S-50.

## BACKUP

The importance of the concept of backup to microcomputing cannot be overstated. A backup is a duplicate or reserve copy of information, which is made in case something goes wrong with the main one (i.e. the working copy). It is necessary when dealing with magnetic media such as cassette tape or disk because, although the reliability of these has improved over the years, all magnetic media are subject to mishap. They might get lost, have coffee spilt over them, be left near a powerful magnetic field, or simply suffer from *dropouts*, when small pieces of the magnetic surface flake off.

The nature of computing is such that a tiny error can be responsible for rendering an entire file useless. The regularity with which you make backup copies (using the utility commands or programs provided by the operating system) should be proportional to the loss you would suffer if the data became corrupted. It is a lesson that most computer users learn the hard way — there is nothing to describe the way you feel when the computer refuses to retrieve the file you have spent hours creating.

# B