CRASH

When a microcomputer *crashes*, it goes into a state equivalent to a coma — the machine is still 'alive', the screen may even continue to display text or graphics, but normal functions cease and keyboard input is not recognised. In this situation, the user must either press the Reset button (if one is fitted) or disconnect the power supply — which usually results in the loss of all programs or data held in RAM. A crash is therefore often referred to as a *fatal error.*

Microcomputer crashes have three main causes. The power supply is often the culprit; the earliest micros in particular were extremely susceptible to small fluctuations in mains voltage. Such fluctuations could be caused by inconsistent supplies from the local generating station, or by simply plugging in another appliance close to the computer's mains socket. Recent microcomputer designs incorporate more efficient 'smoothing' circuits into the power transformer, but their effectiveness depends on the local mains supply — if your area is prone to mains 'spikes' (bursts of higher voltage) or 'brown-outs' (the lights flicker and become dimmer for a few seconds), then it might be a good idea to invest in an external smoothing unit.

Another cause of a crash is an infinite loop within a program. In BASIC, this is hardly a problem — `100 GOTO 100` is an infinite loop, but it is quite simple to break out of by pressing the Break or Stop key. Few machines are able to interrupt a machine code program from the keyboard, so a JMP loop with no exit condition will cause the system to crash.

Most microprocessors will crash if they are confronted with an op-code that does not correspond to anything in their instruction set. This may result from a program that has been wrongly assembled — perhaps a JMP instruction has been misused so that the processor mistakes a piece of data for program code.

CROSS-ASSEMBLER

A *cross-assembler* is a utility program that is used to assemble source code into object code that will run on a different type of machine. For example, a cross-assembler may be used on a BBC Micro to develop machine code programs for a Commodore 64. This should not be confused with converting a BBC program to run on the 64 — a far more difficult exercise.

Cross-assembly allows a programmer to develop code for many different machines by using one system. This development system will probably have a better keyboard and screen and more RAM than a home computer. It will almost certainly execute code faster, and will have a library of editing and utility programs that are not available on the target machines.

The development system may even have a different processor from the one on the target machine — the cross-assembler can deal with the different instruction sets and will ensure that the assembled code is located in the correct memory area. Many home computer programs are in fact developed on mainframe computers. A *cross-compiler* works in a similar way, but is used to compile high-level languages rather than Assembly code.

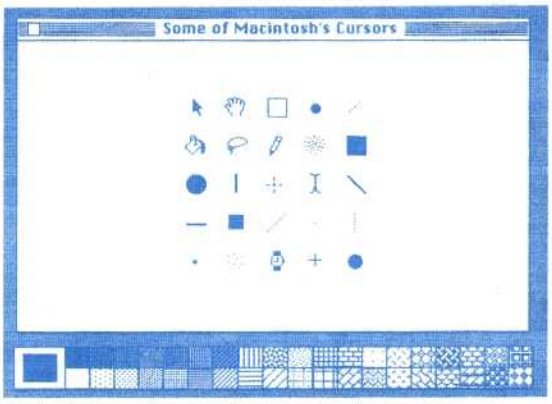
CURRENT LOOP

The most widely established standard for serial data communication is RS232, which specifies the format for the data bits (including start and stop bits), plus the 'handshaking' lines to control the transmission. ('Handshaking' is the name given to the procedure in which, when data is transferred across an interface, one side introduces itself to — or 'shakes hands' with — the other in a pre-defined sequence of actions.)

Within the RS232 standard, there are two variations: V24 and 20 mA (milli-amp) *current loop.* The first of these is the most common; here the data bits are represented by voltage levels, with the difference between a zero and a one bit being 24 volts. Current loop was used in the original teletype terminals. When this system is used, the data line and the ground (or earth) line form a loop between two peripherals, and a current of 20 mA signifies a one while an absence of current denotes a zero.

CURSOR

A computer cursor is simply an on-screen symbol that indicates where the next character will appear. Its name was originally derived from the Latin (it means 'to run before' or 'to herald'), but it was more recently used to describe the movable marker on a slide rule or the pointer on a mechanical adding machine.



Microcomputer cursor symbols have traditionally been either a flashing square or an underline character. Some have simply been a flashing letter of the alphabet (the Sinclair Spectrum displays 'K', 'C', 'L', or 'G' depending on the current mode), but on machines like the Apple Lisa and Macintosh the nature of the cursor changes according to the task being performed. For example, a wristwatch symbol indicates that the user must wait, and a small picture of a hand is used to point to an object that can be moved.