INTERRUPTS

An *interrupt* is a signal to the microprocessor that causes control to pass from the next instruction in the current program to the start of an interrupt-handling routine elsewhere in memory.

There are two kinds of interrupts. *Software interrupts* are generated within a program, usually to transfer control to some part of the operation system. *Hardware interrupts* generate a signal on the reset pin of the processor. This causes a break in execution of the program and is followed by a hardware reset, which restores the system to its power-up state.

Interrupts are used for a variety of different purposes. A program may want to use the operating system for a specific operation and will generate an interrupt — known as a 'supervisor call' — which tells the operating system management process to expect a call to the operating system. This is regarded as a 'voluntary' interrupt.

A similar process will occur with 'involuntary' interrupts, but in this case the interrupt is generated by the BASIC ROM when an error is detected within the program. This interrupt generally causes the program to crash, and displays the appropriate error message. 'Timer' interrupts occur at fixed periods of time when the system clock is updated, and may be used to check if a particular event has occurred — for example, in the case of sprite detection.

Finally, there are interrupts generated by peripheral devices. Information is sent to a peripheral via a buffer queue, which allows the processor to carry out some other operation while the data is being transferred. When the buffer is empty, the peripheral generates an interrupt to stop the processor's current operation and to signal to it to transfer another buffer full of information.

JOB CONTROL LANGUAGE

Job Control Language (JCL) is a language, usually consisting of operating system commands, which controls the way that the system runs a program or a 'job'. JCL is not widely used on microcomputers at present (although CP/M's batching facility is a primitive JCL). However, it is extremely important in mini- and mainframe computers where a suite of programs, each with a different application, is held on tape or disk. For many tasks on a mainframe, several programs may be required to complete one particular job — number crunching, printouts, sorting, etc. Rather than have an operator manually loading and running each program separately, a job control language program is used. A JCL program will normally load and run the programs required for a particular job, initialise any input and output devices, open files and deal with any errors that may be generated. JCLs can control the sequence of the programs being executed and support conditional statement structures, to allow for branching at the end of programs.

Between The Words
Justification — right, left or centred — is available on most word processors. If you look closely, you will see that this is achieved by varying the spaces between words

JUMP

A jump instruction, also known as a 'branch', takes the program out of its normal execution sequence of instructions and passes control to another part of the program. In BASIC, the jump instruction is the GOTO statement. In contrast to the operation of a GOSUB command, GOTO does not cause the current value of the program counter to be saved on the stack for recall later. Should the programmer wish to return to that address, another GOTO has to be contrived.

Jumps can be either *conditional* or *unconditional*. A conditional jump can be represented schematically thus:

```
IF condition is True THEN Jump
```

On the other hand, an unconditional transfer *always* causes a jump whenever it is encountered within the program.

JUNCTION

The boundary between two semiconductors with different electrical properties, or between a metal and a semiconductor, is known as a *junction*. This is a vital component of the transistor. The most common type is the *p-n junction*, where p and n indicate positive and negative semiconductors respectively. Because the n-type semiconductor is negatively charged with respect to the p-type, voltage builds up across the junction when a current is applied to the transistor, and this causes the electrons to cross the junction at a higher voltage. This potential difference between the sides of the junction is applied in the construction of amplifiers and rectifiers.

JUSTIFY

Justification is a common feature of word processing programs. Characters are positioned on the print line so as to produce a uniform right and/or left margin on the screen.

Justification is also used in machine code. A bit pattern can be moved within a register so that the first or last non-zero bit is positioned in the least, or most, significant bit register. This can be important when a particular bit is to be tested, and is required in floating-point formats, where it is called *normalisation*.

IAN McKINNELL