Detective Work

When data is passed from one computer to another it runs the risk of becoming corrupted. Hamming codes can detect and correct these errors

We must all have heard stories about computers making dreadful mistakes — like mailing 500 copies of the same company leaflet to one person. The truth is, of course, that the machine is not to blame: the mistake will have originated from a human failing, perhaps as simple as a typing error. The computer merely serves to amplify the problem. Occasionally, errors arise because the applications program hasn't been written to cope with all eventualities — as in the case of computer-generated final demands for gas bills of £0.00.

Sometimes, though, computers make mistakes that can't be attributed to human intervention, and these are usually manifested in the form of 'bit errors'. A bit error occurs when a single bit in a section of data is transposed from a 1 to a 0 or vice-versa. Bit errors can be caused when a hardware component, such as a RAM chip, fails. That's why many home computers go through a 'diagnostic' error checking software routine whenever the power is turned on.

Most bit errors, however, are 'soft errors' — bits get 'flipped' even though all the RAM has passed the diagnostic test. Home computers are designed to operate in domestic environments, but during a summer heatwave it is quite possible for the temperature to exceed the operating temperature range of the components. Damage is unlikely to be permanent, but bit errors may result in a character on the screen suddenly changing from an 'A' to a 'B', for example, or if the bit happens to form part of an important pointer, it may 'crash' the program, requiring the computer to be reset.

Bit errors can also arise during periods of high sunspot activity, when sub-atomic particles can penetrate the atmosphere and interfere with the flow of electrons in a miniature circuit. In applications such as military systems, industrial control, scientific experimentation or international banking, errors could bring disastrous consequences, so a variety of methods have been adopted to detect them.

The simplest is parity checking (see page 253). An alternative method is the 'checksum', which is widely used when writing data onto magnetic tape or disk. Data is typically handled in blocks of 128 bytes, the last of which to be read or written will be a checksum byte. This byte represents the sum of all the other bytes (each having a value in the range 0 to 255) modulo 256 — meaning the remainder of the sum when divided by 256. Here's an example:

Data: 114,67,83...(121 other values)...
36,154,198
Total of these 127 bytes = 16,673
Total divided by 256 = 65, remainder 33
Therefore checksum = 33

The total of the bytes (16,673) is equal to 65 lots of 256 plus a remainder of 33 — the value that is written into the 128th byte as a checksum. When the computer reads the block back again, it performs its own checksum calculation on the data and if this value differs from 33 then it knows that a bit error has occurred in the recording process.

With both parity and checksum, the computer has no way of knowing which bit of the data has been corrupted. If the error occurred in transmission, then the receiving computer can request a particular byte or block of bytes to be transmitted again; in the case of a recording error, there may well be no way of retrieving the uncorrupted data.

Where errors would be unacceptable, a system must be used that will both detect and correct them. Hamming codes, named after their inventor R W Hamming of Bell Telephone Laboratories, perform this function.

All error correction systems work on the principle of redundancy. Human languages contain a high degree of redundancy — if a typing

Exclusive-Or
A simple Exclusive-Or gate has two inputs and one output. If both inputs are at logical 0 then the output is 0. If either input is 1 then the output is 1. However, if both inputs are 1 then the output is 0. This last condition differentiates the Or gate from the Ex-Or (for short). The operation can be represented by a truth table. Where an Ex-Or has more than two inputs, the output will be 1 if there is an odd number of 1s at the input. Such devices are the means by which parity and error-checking bits are created