G

GATES

The electronic circuitry that comprises most of a computer's working parts consists of thousands of different kinds of switches: information is represented in the machine by patterns of electrical currents, and its processing is effected by switching it through the various routes that these switches, or *gates,* provide. Gates are so called because their function is to allow or deny the passage of information according to the properties of the information and the nature of the gate.

The elementary gates are the AND and OR gates, corresponding to the logical operations of Boolean algebra (see page 32). All logical expressions can be reduced to expressions of these operators, so all the processing functions of the computer can be built using only these gates. The practicalities of integrated circuit construction, however, make NAND and NOR (AND and OR gates with an inverter on the output) cheaper and more convenient to use in large quantities.

GLOBALITY

In a comparatively simple programming language such as BASIC, the variables used in one part of a program are usually available in all other parts of the program, so if X is initialised as 134, say, at the start of the program, then any subsequent reference to X will be to that variable containing the value 134. Such variables are said to be *global* in scope. This is often a convenient feature, and is so commonplace in BASIC that alternatives are rarely considered. However, it can present problems, particularly in large programs, and especially those written by more than one programmer, or when library routines are merged with existing programs. Using a variable in one part of a program, then inadvertently re-using it elsewhere for a different purpose is a very common error, especially when the dialect insists on single-character loop-counter variables. The real shortcoming of global variables is the threat they pose to program structure. A logically independent block of code such as a subroutine or procedure should be accessible only through the calling mechanism (`GOSUB` or `PROC`), and should affect only those variables that are passed to and from the routine as parameters; if variables are global, however, then the subroutine can affect their value whether or not they are passed as parameters.

Some BASIC dialects, and many other languages (such as PASCAL), support *local* variables, whose scope is limited to the logical block in which they are defined. If X is set at 134 in the main program, for example, and control passes to a subroutine containing the statement `LOCAL X`, then inside that subroutine X can be used and re-used without affecting the value of X in the main program.

GRANDFATHERING

Grandfathering is the name given to a system of updating files that retains a copy of the original file, as well as creating new, amended files. This system ensures the safe storage of information, as if a file is accidentally destroyed, there is always a copy in existence. Once a file is amended it is referred to as the *father* file, and the file from which it was created the *grandfather* file. Any subsequent update of the father file is known as the *son* file. As the file is updated, the latest version is always the son file and the most out-of-date the grandfather file. Only three *generations* of files are stored at once, so as new files are created, the current grandfather file is deleted.

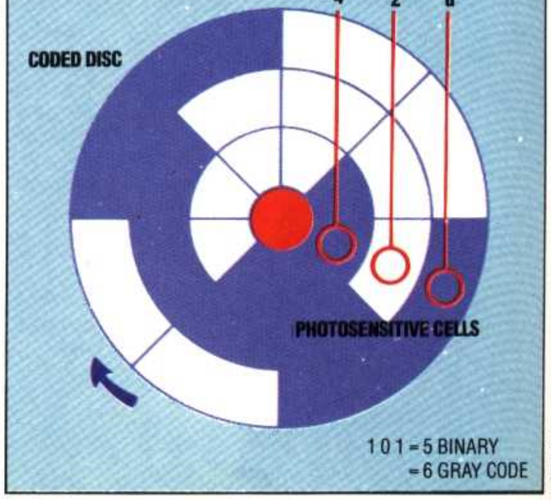
GRAY CODE

In computerised control applications, positional data written on a moving object must often be read by a mechanical reader. The accuracy of the reading is subject to errors in the timing of the read cycle, and it often happens that a small error causes the sensors to point to the gaps between the data rather than to the data itself. It is then a matter of chance which of the two numbers is read by the sensor. The *Gray code* is a way of encoding binary data to minimise the effect of such errors. The principle of the code is that only one bit of a number changes with every successive increment of the number, and the bit that does change should be as far to the right in the number as possible. Compare these numbers in binary and Gray code:

Decimal	Binary	Gray Code
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111

If the mechanical reader is positioned between two consecutive Gray code numbers and reads the wrong one, then the error is confined to one bit, since there is only one bit that changes.

Gray Disc
The disc is attached to a rotating shaft to enable the photocells to read the shaft's position. The three parts of each segment are coloured to represent the numbers zero to seven in Gray code and are read by the photocells positioned above the disc. Each segment of the disc differs from its neighbour in only one respect — the basis of the Gray code



LIZ DIXON