F

FORTH

FORTH was invented by astronomer Charles Moore in 1972 when he became dissatisfied with FORTRAN as a language in which to write telescope control programs. The specialised functions that he needed were difficult to write in FORTRAN because its structure and processes were too strongly oriented towards that language's scientific/mathematical purposes. Accordingly, he designed FORTH as a dictionary of *primitives*— the elementary functions of the language — and an editor/compiler/interpreter.

The editor is used to define new functions as expressions (or 'subroutines') created from the existing dictionary; the new functions are named and compiled, and thus added to the language. A function can be executed at any time through the interpreter by simply issuing its name as a command. FORTH treats all of user memory as one big Last In First Out (LIFO) stack, while program memory is a series of independent stacks (one per function) of machine code subroutine addresses. A function is executed by jumping to the first address on its program stack, popping any needed values off the main stack, pushing any results back onto the stack, and exiting to the next stacked address, until execution is complete. Arithmetic expressions are, therefore, written in *reverse Polish* — or *postfix* — *notation* (operands are grouped together and followed by their operators; thus `A+B*C` is written `B C A * +`) because this is a stacked-oriented notation.

Programming in FORTH, then, really consists of developing a customised version of the language to suit each application. The language's chief virtues are its 'extensibility' and speed of operation. Because of its extensibility and because it brings the user closer to the computer's operations than is normal with high-level languages, FORTH has been acclaimed as a replacement for BASIC but, although it is available in various versions for most micros, only one — the now-defunct Jupiter Ace — has been manufactured with FORTH rather than BASIC as its resident language.

FORTRAN

Developed by IBM in 1956, FORTRAN (derived from FORmula TRANslation) was the first commercially available high-level language. It had two main purposes: to demonstrate that high-level, quasi-English programming languages could be compiled quickly and efficiently, and to make computers more generally accessible to scientists and engineers who might be prepared to learn a language rather like the algebraic expressions in which they formulated their ideas, but who had neither the time nor the patience to learn machine code. In both of these aims FORTRAN has been enormously successful; it is still the most widely used of the high-level languages, and a new version is due in the late 1980s. Several versions are also available for microcomputers.

An important early development was the ability to create system libraries of independently-compiled FORTRAN subroutines: all mainframe systems have such libraries, and so important a resource are they that other languages — such as PASCAL — are configured so that they can call FORTRAN routines from the libraries.

FORTRAN'S legacy is the group of languages descending from it—chiefly ALGOL, PASCAL and BASIC—but its true historical significance is probably that it enabled computers to move out of university computing laboratories and into the classrooms and workshops, where they could become taken for granted as everyday scientific equipment. From there it was a short step into offices and homes. FORTRAN brought computing within the reach of the non-specialist, and was perhaps, therefore, the first step on the road to user-friendliness.

FOURTH GENERATION

A generation in the development of computers seems to span about ten years, and begins with the development of an expensive new technology which is commonplace by the end of that period. The first generation began in the late 1940s with the first stored-program thermionic valve machines; the second generation machines appeared in the late 1950s and used discrete transistor logic; the third generation — typified by the IBM 360 — began in the early 1960s with integrated families of machines and comprehensive operating systems; and the *fourth generation* appeared in the early 1970s with the introduction of Large Scale and Very Large Scale Integration (LSI and VLSI) chip circuitry. As such, it includes mainframe, mini and microcomputers. Micros have themselves gone through several stages of development to arrive in the middle 1980s as credible small-scale computers, supporting fourth generation features such as large memories (one megabyte or more), networking, multi-tasking and integrated software.

The fifth generation is expected to appear in the late 1980s. Its characteristic features are likely to be natural-language programming, speech recognition and generation, and a degree of artificial intelligence in its operating systems and applications software; it will probably be developed in Japan.