



ALU

The Arithmetic Logic Unit forms the heart of any computer that follows the von Neumann architecture — in other words all existing home computers. The ALU is a part of, and therefore found within, the CPU (Central Processing Unit), which on a home computer will be a 6502, Z80, or perhaps a Motorola 68000. While the rest of the CPU is concerned with shuffling data around the computer, at the right time and to the right places, the ALU is the only part of it that can actually change the data.

As the name suggests, the ALU can perform elementary *arithmetical* and *logical* functions on the data. Addition and subtraction are possible, as are functions such as AND, OR and NOT, but only on one byte at a time. The ALU performs its processing in the CPU's internal registers, in particular the accumulator (see page 13).

AND

AND is one of the fundamental logical functions, with which readers of THE HOME COMPUTER ADVANCED COURSE will already be familiar. AND takes on two arguments, each of which may be true or false — usually represented by binary 1 and 0 respectively. If, and only if, both inputs are true then the output or result of the AND function will be true also.

An AND gate is a hardware circuit that implements the AND function. A typical home computer will contain many of these, either in the form of dedicated logic chips, or integrated into the design of larger chips such as the microprocessor. But the AND function can also play an important role in programming, both in BASIC and machine code; in both cases it performs the function on several pairs of bits simultaneously. In machine code there will normally be an opcode for ANDing the contents of the accumulator with another byte in memory. In the resulting byte, a bit will have the value 1 if, and only if, the corresponding bits of the original bytes were both 1.

Most BASICs will allow AND to be used in a similar way, so that:

```
LET C=A AND B
```

will cause the computer to perform an AND on the binary representations of A and B, and store the result in the variable C. So if A is 13 (1101) and B is 11 (1011), C will take on the value 9 (1001). There are many applications for such a construct. For example:

```
IF (TURN AND 1)=1 THEN GOTO 1000
```

will only jump to line 1000 if TURN contains an odd (binary) number.

APL

Computer jargon is full of strange acronyms, but they don't come much stranger than this one. APL is a programming language and it stands for A Programming Language! Despite the simplicity of

its name, it is a very sophisticated, high-level language, and its users boast that they can program in a single line of APL what could take dozens of lines in BASIC. Though favoured in some academic and mathematical circles, its appeal is very limited, and it is consequently not generally available for microcomputers. Another reason for its unpopularity is that APL requires special symbols that are not available on most keyboards and screens. One microcomputer that could run APL was the SuperPET (a modification of the Commodore 8000 PET), which possessed a different processor (the 6809), more RAM and a character generator capable of producing the APL symbols on the screen.

APPLICATIONS GENERATOR

The idea of a program that can accept as its input a user's specification, and produce as its output a freestanding program that will perform the required task is a very appealing one — and not at all beyond the bounds of possibility. However, most of the commercial packages sold as 'program generators' are severely limited in scope, and produce programs that are considerably less efficient than code written by humans.

An *applications generator* is a similar but more popular idea. The applications programs that it produces are not freestanding — the original generator program is needed to run them. However, it does allow you to produce programs that are on the one hand tailor-made to your specifications, but on the other hand use efficient code that was originally written by a human programmer.

They work on the principle that within a particular area of application (say adventure games or book-keeping) all programs have to perform a minimum set of functions, which the generator provides ready-written. In specifying the exact operation of the required program, all the user is doing is creating a file of reference data that tells the applications generator how to apply its standard routines in the main program to the task in hand.



IAN MCKINWELL

A