indicates whether a read or write is to be performed), then the selected byte will place its contents on the data bus, in the form of a pattern of eight bits. Similarly, the CPU can write a pattern of eight bits into any chosen location. The CPU has no knowledge of which parts of memory are ROM and RAM, so getting the addresses right is another crucial responsibility of the programmer.

Inside the microprocessor, there are perhaps half a dozen 'registers', which are like individual memory locations and are used for storing temporary results and performing the logic and binary arithmetic functions. Most of these registers are equivalent to one byte of memory, though some are 16 bits wide. One of the latter type is called the Program Counter (PC) register, and this contains the address in memory of the machine code instruction that is currently being performed. You can think of this as being similar to the line number in a BASIC program.

Another of the most important registers (but this time just eight bits wide) is the 'accumulator'. As the name suggests, this register can accumulate totals (that is to say, bytes can be added to it or subtracted from it), and indeed this is usually the only register that can perform any kind of arithmetic. So, a very simple machine code program might be specified as follows:

1) Load the accumulator with the contents of memory location $3F80. Addresses in machine code are usually written in hexadecimal (see page 179). Hexadecimal numbers are indicated in writing by prefixing a special sign, usually a $.

2) Add to the accumulator the contents of memory location $3F81, allowing for the fact that the result may be larger than can be stored in a single byte — in which case there will be a 'carry bit' as well.

3) Store the new contents of the accumulator (i.e. the result) in memory location $0493.

Each of these constitutes a machine code instruction, and the program would normally be written thus:

```
LDA $3F80   (LoaD Accumulator)
ADC $3F81   (ADd with Carry)
STA $0493   (STore Accumulator)
```

The comments in brackets, like BASIC REMark statements, have no effect. The first entry on each line is called the 'opcode', and this indicates the nature of the operation. The second column contains the 'operand' — the details of, or whereabouts of, the data that is to be operated on. A microprocessor will usually feature several dozen possible opcodes (that is to say, it can perform several dozen types of simple operation), and each opcode will occupy just one byte of memory when it has been entered into the machine.

An opcode can therefore by specified as a number in the range 0-255 (or, more properly, in the hex range $00 to $FF). However, while a program is being developed, it is more usual to make the listing more readable by using three letter mnemonics, such as LDA, ADC and STA.

Flashing Lights
The idea for the huge panels of lights often seen on computers in films came from the 'front panel' found on many mini-computers. This front panel was a line of lights and switches representing the CPU's address and data buses. Before keyboards were interfaced, all machine code programs had to be entered in binary in this form

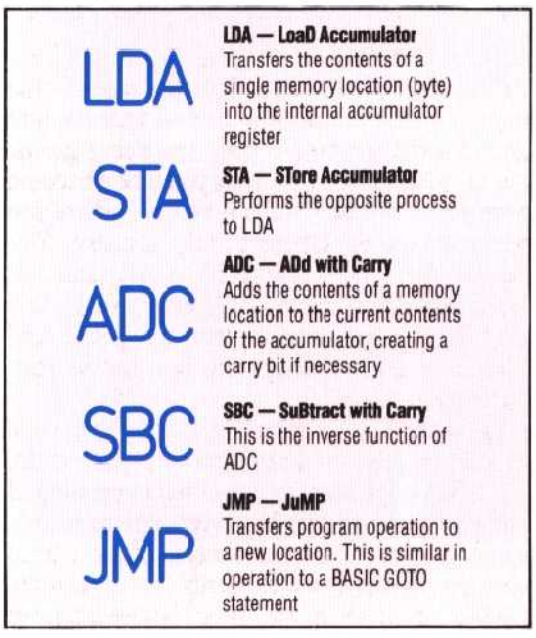
KEVIN JONES

Each of the three operands shown consists of a hex number in the range $0000 to $FFFF, and uses up two bytes of program memory space. However, some operands are just one byte long, and some opcodes don't have operands at all. The short program that we have given would therefore occupy a total of only nine bytes — not including the three memory locations ($3F80, $3F81, and $0493) that the program will operate on. For this trivial exercise, the following BASIC program would achieve exactly the same effect, but would occupy nearly 50 bytes and perform the operation at least a hundred times slower, because of all the time taken by the interpreter to translate it:

```
10 A = PEEK (16256)
20 A = A + PEEK (16257)
30 POKE 1171,A
```

N.B. The locations used by this particular program may not be suitable for your machine.

In the next instalment of THE HOME COMPUTER COURSE, we'll look at how machine code is entered into a home computer and run, and the different ways in which machine code is expressed.



Opcodes
These are just a few of the opcodes (types of operation or instruction) that a typical microprocessor can execute