SECTION 2 - STORAGE OF INFORMATION WITHIN A WORD

2.1. Binary Form

It has been said that each of the 1024 registers of the immediate access store has a capacity of 36 binary bits, and that this group of 36 bits is known as a computer word.

A register may be envisaged thus:

It will be noted that the 'bits' are numbered from 0 at the right-hand end to 35 at the left-hand end. This is the convention adopted since binary numbers are considered, in the same way as normal numbers, with the least significant digit on the right. The digit positions from 0 to 35 are described as D_0 D_1 D_2 and so on up to D_{35}.

A binary number consists of a series of ones and zeros. Now each of the bits in a register may be electrically energised into one of two states, one representing "one" and the other "zeros".

Thus for example, the binary equivalent of 10,000 is 10011100010000 and this will appear:-

where the shaded bits are energised to the state representing "one" and the plain bits to the state representing "zero". It will be noted also that the "one" in D4 represents 2^4, the "one" in D8 represents 2^8 and so on. Negative as well as positive numbers may be held in the machine, and negative numbers are represented as complements of 2^{36}. To represent, for example -10000 this number is subtracted from zero, thus:-

Decimal	Binary
00000	000000000000000000000000000000000000
10000	10011100010000
-10000	111111111111111111111101100011110000

The register containing this binary number will then appear as:-

It will be seen that, for any negative number less than 2^{35}, the D_{35} position will be in the state of "one", while for the corresponding positive number this position will be in the "zero" state. This fact is used by the computer control to distinguish negative numbers, and while therefore it is possible to hold positive numbers up to 2^{36} -1 it is not usual to hold anything higher than 2^{35} -1, since to do so would invalidate the positive-negative testing. The capacity of a register or word is therefore 34,359,738,367 or £143,165,576-10-7. (For double-length register operations, however, up to 2^{71} -1 may be used).

All numbers on which arithmetic operations are to be performed must be held in the pure binary form demonstrated above. Since, however, EMIDEC is a stored program machine with full capacity for numeric and alphabetic storage its store will also hold two other types of information, these being

1) Alphanumeric characters required for storage only, and

2) Program Instructions.

The same registers containing 36 binary "ones" or "zeros" are used for each of the three types of information. Thus, while pure binary numbers are interpreted by regarding the whole word as a single entity with each digit in its appropriate significant place, alphanumeric characters (as also program instructions) are interpreted in specific groups of digits according to a pre-determined code. The machine itself is unable to distinguish between the three types of information; each being represented by a row of binary digits.

2.2. Alphanumeric Form

The alphanumeric code used within EMIDEC and in the buffers of input and output units, employs six binary digits to represent each character. The usual allocation of these groups of six is shown in Appendix I.

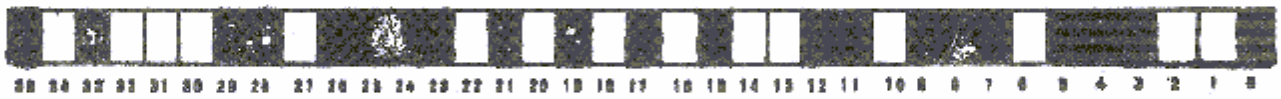
It will be seen that, when information is held in registers in alphanumeric form, each register will hold six groups of six binary digits each. To hold, for instance, the number 123, the first group (from the right or least significant end) will be the coded equivalent of three, i.e. 000011, the second will be the coded equivalent of 2 i.e. 000010, and the third, the coded equivalent of 1 i.e. 000001. The remaining three groups will all be zeros, and the whole word will then be

000000,000000,000000,000001,000010,000011,

and the register will appear as

It should be apparent that this is identical with the pure binary representation of 4227 i.e. 1000010000011. The computer is however able to treat the contents of a register in the way appropriate to the instruction being performed - though it should be borne in mind that a programming error could cause a misinterpretation.

As a further illustration of the alphanumeric code the word CREDIT would be coded in binary as 101000/110111/101010/101001/101110/111001 and would appear in the register as;



which it will be observed, is also equivalent, in pure binary to the negative number - 24,835,875,911.