For permanent storages purposes, magnetized surfaces can be used and the two states of magnetization can directly or indirectly ~~as possible~~ represent the binary digits. In the past, punched holes in paper tape or cards have been used for storage purposes. The presence of a hole in the tape or card represents a 1 and the absence of a hole represents a 0.

The number system using the base 16 is known as the hexadecimal number system. This number system is sometimes used to represent binary numbers.
In the hexadecimal number system, there are 16 different digit symbols, the numerals '0 to 9' and the first six letters of the alphabet, A, B, C, D, E and F. The letters are used to represent 10 to 15.

For example: $(1AE)_{16} = 1\times16^2 + A\times16^1 + E\times16^0$

$$= 1\times16^2 + 10\times16^1 + 14\times16^0$$
$$= 1\times256 + 10\times16 + 14\times1$$
$$= (430)_{10}$$

Conversion from one number system to another number system

-conversion between binary and octal or hexadecimal numbers

Conversion between binary and octal or hexadecimal numbers is particularly easy because the octal and hexadecimal bases, 8 and 16, are powers of the binary base, 2 (third and fourth powers respectively). This leads to a direct relationship between groups of digits in octal or hexadecimal numbers and binary numbers.

For octal, groups of three digits of the binary number are equivalent to one digit of the octal number. For example:

i) Conversion of $(1100101)_2$ into octal.
Dividing the binary number into groups of three: 1 100 101
Encoding each group into one octal digit: 1 4 5
Therefore $(1100101)_2 = (145)_8$

ii) Conversion of $(011110111)_2$ into octal.
Dividing the binary number into groups of three: 011 110 111
Encoding each group into one octal digit: 3 6 7
Therefore $(011110111)_2 = (367)_8$

e.g. decimal number $(0.485)_{10}$ into binary

Fractional parts		Integer parts
0.485	$\times 2 = 0.97$	0
0.97	$\times 2 = 1.94$	1
0.94	$\times 2 = 1.88$	1
0.88	$\times 2 = 1.76$	1
0.76	$\times 2 = 1.52$	1
0.52	$\times 2 = 1.04$	1
0.04	$\times 2 = 0.08$	0

Therefore $(0.485)_{10} = (0.0111110\cdots)_2$

A finite fraction in one number system will not necessarilly convert into a finite fraction in another system. Normally the process is terminated when a given number of digits is obtained

e.g. decimal number $(0.347)_{10}$ into hexadecimal:

Fractional parts		Integer parts: Decimal	Integer parts: Hexadecimal
0.347	$\times 16 = 5.552$	5	5
0.552	$\times 16 = 8.832$	8	8
0.832	$\times 16 = 13.312$	13	D
0.312	$\times 16 = 4.992$	4	4
0.992	$\times 16 = 15.872$	15	F

Therefore $(0.347)_{10} = (0.58D4F\ldots)_{16}$

The conversion of a large decimal number into binary can be shortened by first converting into octal and then to binary by inspection, because the number of divisions (or multiplications) of eight to convert into octal is less than the number of divisions (or multiplications) of two to convert into binary.

Conversion from binary/octal/hexadecimal/other to decimal

We could use the same methods as before using arithmetic of the original base for conversion to decimal (or between any base). However it is rather inconvienient to use any base other than decimal for arithmetic.

Hence we can represent normalized positive numbers from $+0.10\cdots00\times2^{-63}$ to $+0.11\cdots11\times2^{+64}$ (or 2^{-64} to approximately 2^{+64}) and normalized negative numbers from $-0.10\cdots00\times2^{-63}$ to $-0.11\cdots11\times2^{+64}$ (or -2^{-64} to approximately -2^{+64}) where in all cases there are 24 bits following the binary point. The approximate ranges in decimal are $+5.4\times10^{-20}$ to $+1.8\times10^{+19}$ and from -5.4×10^{-20} to $+1.8\times10^{+19}$.

Normalized numbers cannot be represented outside these ranges. It is left to the reader to compute the ranges assuming that a stored exponent of zero is reserved for the floating point number zero. A 24-bit mantissa gives the equivalent of eight significant decimal digits.

Binary coded decimal numbers

In some applications (e.g calculators and check-out bills), decimal numbers are continually displayed or entered. Clearly, the decimal numbers can be converted to binary and vice versa as we have seen, and this can be done by computer. However, the decimal nature of the numbers can be retained within the computer using the binary coded decimal (BCD) number system. In BCD, each decimal digit is converted into a four-bit binary equivalent and each four-bit word is joined together to form the coding of the number.

e.g. The number $(259)_{10}$ in BCD is the 12-bit number:

0010 0101 1001

In the BCD coding system, the digits are not 'weighted' in the normal ascending power of the base as in binary or decimal numbers or in any other normal positional number system. The weighting in BCD is:

$\cdots, 100\times2^4\ 100\times2^2\ 100\times2^1\ 100\times2^0, 10\times2^4\ 10\times2^2\ 10\times2^1\ 10\times2^0, 1\times2^4\ 1\times2^2\ 1\times2^1\ 1\times2^0$

Within each group, the binary combinations from decimal 10 to 15, i.e. 1010, 1011, 1100, 1101, 1110 and 1111 are not used and would be invalid if they occurred. Hence more digits are used to represent the number than strictly necessary.

Two BCD digits can be held in one 8-bit word, four BCD digits can be held in one 16-bit word, and eight BCD digits can be held in one 32-bit word. BCD numbers with more than one BCD digit contained in a binary word are called packed BCD numbers.

The Gray code

The Gray code belongs to a class of codes called minimum-change codes, in which only one bit in the code group changes when going from one step to the next. The Gray code is an unweighted code, meaning that the positions in the code groups do not have any specific weight assigned to them. Because of this, the Gray code is not ~~sever~~ suited for arithmetic operations but finds applications in input/output devices and some types of analog-to-digital converters.

Decimal	Binary Code	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

The tables shows the Gray-code representation for the decimal numbers 0 through 15, ~~through~~ together with the straight binary code. If we examine the Gray-code groups for each decimal number, it can be seen that in going from any one decimal number to the next, only one bit of the Gray code changes. For example, in going from 3 to 4, the Gray code changes from 0010 to 0110, which with only the second bit from the left changing. Going from 14 to 15 the Gray-code bits change from 1001 to 1000, with only the last bit changing. This is the principal characteristic of the Gray code. Compare this with the binary code, where anywhere from one to all of the bits change in going from one step to the next.

The Gray code is often used in situations where other codes, such as binary, might produce erroneous or ambiguous result during those transitions in which more than one bit of the code is changing. For instance, using binary code and going from 0111 to 1000 requires that all four bits change simultaneously. Depending on the device or circuit that is generating the bits, there may be a significant differences in the transition times of the different bits. If so, the transition from 0111 to 1000 could produce one or more intermediate states.