Number systems and codes

Number Systems
Positional Number Systems

A number is represented by a string of digits, where each digit position has an associated weight.

The value of a number is a weighted sum of digits, for example:

$$1734=1\cdot1000+7\cdot100+3\cdot10+4\cdot1.$$

Each weight is a power of 10 corresponding to the digit position.

$$23.5=2\cdot10^{1}+3\cdot10^{0}+5\cdot10^{-1}$$

The general form of a number in a positional number system:

$$N(r)=d_{n-1}d_{n-2}d_{n-3}\ldots d_1d_0\cdot d_{-1}d_{-2}d_{-3}\ldots d_{-m}$$

The value of the number is the sum of each digit multiplied by the corresponding power of the radix:

$$D=\sum_{i=-m}^{n-1} d_i \cdot r^i$$

The leftmost digit in such a number is called the most significant digit, the rightmost is the least significant digit.

Binary number system

The general form of a binary number is:

$b_{n-1}b_{n-2}b_{n-3}\ldots b_1b_0\cdot b_{-1}b_{-2}b_{-3}\ldots b_{-m}$

....

$$B = \sum_{i=-m}^{n-1} b_i \cdot 2^i$$

$10011_2=1\cdot16+0\cdot8+0\cdot4+1\cdot2+1\cdot1=19_{10}$

$100010_2=1\cdot32+1\cdot2=34_{10}$

The leftmost bit is called the most significant bit (MSB)

The rightmost is the least significant bit (LSB)

Decimal to binary conversion

- Integers

179:2=89 remainder 1 (LSB)
:2=44 remainder 1
:2=22 remainder 0
:2=11 remainder 0
:2=5 remainder 1
:2=2 remainder 1
:2=1 remainder 0
:2=0 remainder 1 (MSB)

$179_{10}=10110011_2$

- Fractionals

$$\begin{array}{r} 0.58 \\ \times 2 \\ \hline 1.16 \\ \times 2 \\ \hline 0.32 \\ \times 2 \\ \hline 0.64 \\ \times 2 \\ \hline 1.28 \\ \times 2 \\ \hline 0.56 \\ \times 2 \\ \hline 1.12 \end{array}$$

$$0.58_{10} = 0.100101_2$$

Octal and hexadecimal number systems

The octal and hexadecimal number systems are useful for representing multi-bit numbers because their radices are powers of 2.

Since a string of three bits can take on eight different combinations it follows that each 3-bit string can be uniquely represented by one octal digit.

Likewise a 4-bit string can be represented by one hexadecimal digit.

Binary	Decimal	Octal	3-bit string	Hexadecimal	4-bit string
0	0	0	000	0	0000
1	1	1	001	1	0001
10	2	2	010	2	0010
11	3	3	011	3	0011
100	4	4	100	4	0100
101	5	5	101	5	0101
110	6	6	110	6	0110
111	7	7	111	7	0111
1000	8	10	-	8	1000
1001	9	11	-	9	1001
1010	10	12	-	A	1010
1011	11	13	-	B	1011
1100	12	14	-	C	1100
1101	13	15	-	D	1101
1110	14	16	-	E	1110
1111	15	17	-	F	1111

Binary to octal/hexadecimal convertion

Starting at the binary point and working left we simply separate the bits into two groups of three (four) and replace each group with the corresponding octal digit.

$10.1011001011_2 = 010.\ 101\ 100\ 101\ 100_2$
$= 2.5454_8 = 0010.1011\ 0010\ 1100 = 2.BC_{16}$.

Converting in the reverse direction from octal or hexadecimal to binary: replace each octal or hexadecimal digit with the corresponding 3 or 4-bit string.

$2046.17_8 = 010\ 000\ 100\ 110.\ 001\ 111_2$

$9F.46C_{16} = 1001\ 1111.\ 0100\ 0110\ 1100_2$

Conversion Methods for Common Radices

Binary to octal: substitution

$N=10111011001_2=10\ 111\ 011\ 001_2=2731_8$

Binary to hexadecimal: substitution

$N=10111011001_2=101\ 1101\ 1001_2=5D9_{16}$

Binary to decimal: computation

$N=1\cdot 1024+0\cdot 512+1\cdot 256+1\cdot 128+1\cdot 64+0\cdot 92+1\cdot 16+1\cdot 8+0\cdot 4+0\cdot 2+1\cdot 1=1497_{10}$

Conversion Methods for Common Radices

Octal to binary: substitution

$N=1234_8=001\ 010\ 011\ 100_2$

Octal to hexadecimal: substitution

$N=1234_8=001\ 010\ 011\ 100_2=0010\ 1001\ 1100_2=29C_{16}$

Octal to decimal: computation

$N=1234_8=1\cdot512+2\cdot64+3\cdot8+4\cdot1=668_{10}$

Conversion Methods for Common Radices

Hexadecimal to binary: substitution

$N = C0DE_{16} = 1100\ 0000\ 1101\ 1110_2$

Hexadecimal to octal: substitution

$N = C0DE_{16} = 1100\ 0000\ 1101\ 1110_2 = 1\ 100\ 000\ 011\ 011\ 110_2 = 140336_8$

Hexadecimal to decimal: computation

$N = C0DE_{16} = 12 \cdot 4096 + 0 \cdot 256 + 13 \cdot 16 + 14 \cdot 1 = 49374_{10}$

Conversion Methods for Common Radices

Decimal to binary:

division

108_{10}:2=54 remainder 0 (LSB)
:2=27 remainder 0
:2=13 remainder 1
:2=6 remainder 1
:2=3 remainder 0
:2=1 remainder 1
:2=0 remainder 1 (MSB)

$108_{10}=1101100_2$

Decimal to octal:

division

108_{10}:8=13 remainder 4 (Least significant digit)
:8=1 remainder 5
:8=0 remainder 1

$108_{10}=154_8$

Decimal to hexadecimal:

division

108_{10}:16=6 remainder 12 (Least significant digit)
:16=0 remainder 6

$108_{10}=6C_{16}$

Binary Codes for Decimal Numbers

Even though binary numbers are the most appropriate for the internal computations of a digital system, most people still prefer to deal with decimal numbers.

As a result the external interfaces of a digital system may read or display decimal numbers and some digital devices actually process decimal numbers directly.

A decimal number is represented in a digital system by a string of bits. At least 4 bits are needed to represent ten decimal digits. There are millions of ways to choose ten 4-bit code words

$$A_{16}^{10} = \frac{n!}{(n-m)!} = \frac{16!}{6!} = 7*8*9*10*11*12*13*14*15*16 = 29\,059\,430\,400$$

Binary Codes for Decimal Numbers

The most common decimal codes:

BCD (8421)	Exces-3	2421	2 din 5	8421 cu paritate	86(-1)(-4)
0000	0011	0000	00011	00000	0000
0001	0100	0001	00101	10001	0111
0010	0101	0010	00110	10010	0101
0011	0110	0011	01001	00011	1011
0100	0111	0100	01010	10100	1001
0101	1000	1011	01100	00101	0110
0110	1001	1100	10001	00110	0100
0111	1010	1101	10010	10111	1010
1000	1011	1110	10100	11000	1000
1001	1100	1111	11000	01001	1111

Conversion between BCD and decimal representations are trivial, a direct substitution of four bits for each decimal digit.

$358, 196_{10} = 0011\ 0101\ 1000,\ 0001\ 1001\ 0110_{BCD}$

Another codes

- **ASCII (The American Standard Code for Information Interchange).** The ASCII represents each character with a 7-bit string (8 in extended mode), yielding a total of 128 different characters. The code contains the uppercase and lowercase ABC letters, numerals, punctuation and various nonprintable control characters.
- **EBCDIC Extended BCD Interchange code** 8 bits
- **UNICODE** 16 bits