1

Number Systems

1.1. Introduction

Digital systems are used to process data and to perform calculations in most instrumentation, monitoring and communication devices. As physical quantities and signals can only take discrete values in a digital system, the interpretation of real-world information requires the use of interface circuits such as data converters.

In general, numbers may be represented in different numeration systems. The decimal system is commonly used in routine transactions while the binary system is the basis for digital electronics. Every number (or numeration) system is defined by a base (or *radix*), which is a collection of distinct symbols. The representation of a number in a numeration system may be considered as a change in base. In a positional number system, a value of a number depends on the place occupied by each of its digits in the representation.

1.2. Decimal numbers

The decimal number system uses the following 10 numbers or symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. The radix is thus 10.

EXAMPLE 1.1.– Decompose the numbers 734 and 12345 into powers of 10.

The decomposition of the number 734 takes the form:

$$\begin{aligned} 734 &= (7 \times 10^2) + (3 \times 10^1) + (4 \times 10^0) \\ &= 734_{10} \end{aligned}$$

For the number 12345, we have:

$$\begin{aligned} 12\ 345 &= (1 \times 10^4) + (2 \times 10^3) + (3 \times 10^2) + (4 \times 10^1) + (5 \times 10^0) \\ &= 12\ 345_{10} \end{aligned}$$

Depending on its position, each number is multiplied by the appropriate power of 10. The right-most digit represents the unit digit.

1.3. Binary numbers

Binary number system is based on two-level logic, conventionally noted as 0 (low level) and 1 (high level). It is a system with a radix of two.

EXAMPLE 1.2.– Convert the decimal numbers 13 and 125 into binary numbers.

The decomposition of the number 13 in powers of 2 is written as:

$$\begin{aligned} 13_{10} &= (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\ &= 1101_2 \end{aligned}$$

For the number 125, we have:

$$\begin{aligned} 125_{10} &= (1 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) \\ &+ (1 \times 2^0) = 1111101_2 \end{aligned}$$

The binary code that is then obtained for a positive number is called a natural binary code.

The coefficients or numbers (0 or 1) used in the binary representation of a number are called bits.

The right-most bit is called the *least significant bit* (LSB), while the left-most bit is called the *most significant bit* (MSB).

In practice, the conversion of a decimal number to a binary number can be carried out by reading, from last to first, the remainders of a series of integer divisions as illustrated by Figure 1.1.

The arithmetic and logic unit of a microprocessor manipulates binary numbers or *words* with a fixed number of bits.

$$\begin{aligned} 13 \div 2 &= 6 \quad \text{Remainder } 1 \text{ (LSB)} \\ 6 \div 2 &= 3 \quad \text{Remainder } 0 \\ 3 \div 2 &= 1 \quad \text{Remainder } 1 \\ 1 \div 2 &= 0 \quad \text{Remainder } 1 \text{ (MSB)} \end{aligned}$$

$$13_{10} = 1101_2$$

$$\begin{aligned} 125 \div 2 &= 62 \quad \text{Remainder } 1 \text{ (LSB)} \\ 62 \div 2 &= 31 \quad \text{Remainder } 0 \\ 31 \div 2 &= 15 \quad \text{Remainder } 1 \\ 15 \div 2 &= 7 \quad \text{Remainder } 1 \\ 7 \div 2 &= 3 \quad \text{Remainder } 1 \\ 3 \div 2 &= 1 \quad \text{Remainder } 1 \\ 1 \div 2 &= 0 \quad \text{Remainder } 1 \text{ (MSB)} \end{aligned}$$

$$125_{10} = 1111101_2$$

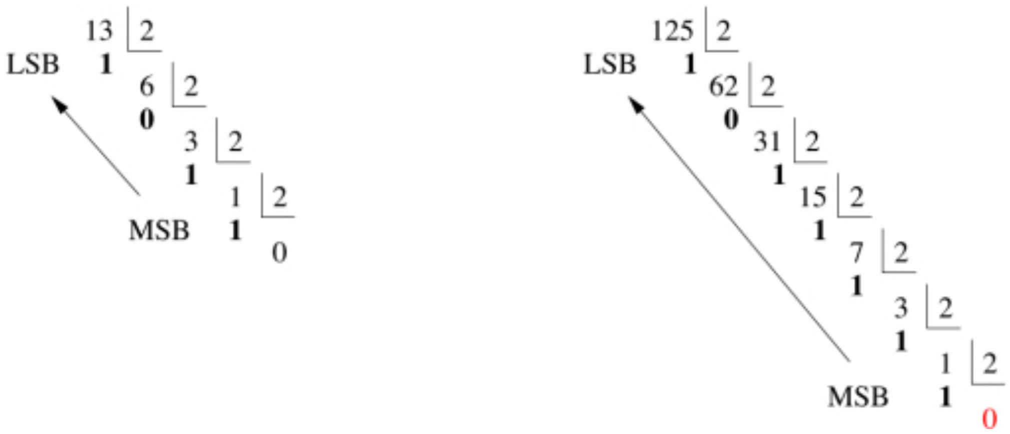

Figure 1.1. *Decimal-binary conversion using successive division methods*

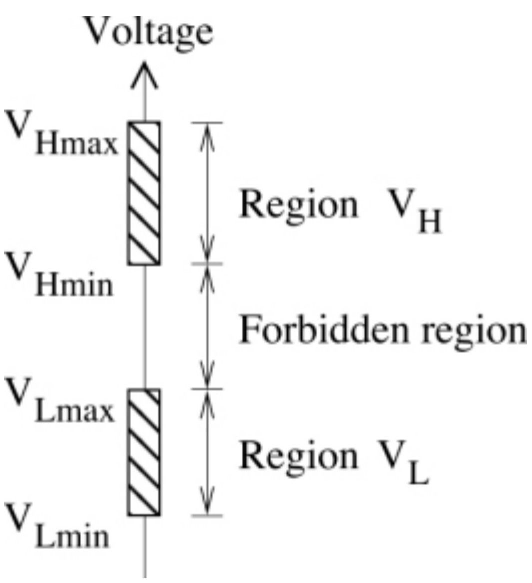

Figure 1.2. *Representation of logic voltage levels*

A *byte* is an 8-bit word.

In practice, the bits 0 and 1 are represented by voltage or current levels. Figure 1.2 shows the representation of logic voltage levels. The two regions V_H and V_B are separated by a forbidden region where the logical level is undefined.

Logical states may be assigned to regions based on positive logic or negative logic. In the case of positive logic, the region V_H corresponds to 1 (or the high level), and the region V_B corresponds to 0 (or the low level); and in the case of negative logic, the region V_H corresponds to 0 (or low level), and the region V_B corresponds to 1 (or high level).

1.4. Octal numbers

The octal number system or a representation with radix eight consists of the following symbols: 0, 1, 2, 3, 4, 5, 6, 7.

EXAMPLE 1.3.– Convert the decimal numbers 250 and 777 to octal numbers.

In radix 8 representation, the number 250 takes the form:

$$\begin{aligned} 250_{10} &= (3 \times 8^2) + (7 \times 8^1) + (2 \times 8^0) \\ &= 372_8 \end{aligned}$$

In the case of the number 777, we have:

$$\begin{aligned} 777_{10} &= (1 \times 8^3) + (4 \times 8^2) + (1 \times 8^1) + (1 \times 8^0) \\ &= 1\,411_8 \end{aligned}$$

The right-most digit is called the *least significant digit* (LSD), while the left-most digit is called the *most significant digit* (MSD).

A practical approach to converting a decimal number to an octal number consists of carrying out a series of integer divisions as illustrated in Figure 1.3.

$$\begin{array}{ll} 250 \div 8 = 31 & \text{Remainder 2 (LSD)} \\ 31 \div 8 = 3 & \text{Remainder 7} \\ 3 \div 8 = 0 & \text{Remainder 3 (MSD)} \end{array}$$

$250_{10} = 372_8$

$$\begin{array}{ll} 777 \div 8 = 97 & \text{Remainder 1 (LSD)} \\ 97 \div 8 = 12 & \text{Remainder 1} \\ 12 \div 8 = 1 & \text{Remainder 4} \\ 1 \div 8 = 0 & \text{Remainder 1 (MSD)} \end{array}$$

$777_{10} = 1411_8$

Figure 1.3. *Decimal-octal conversion using the successive division method*

Octal numeration may be deduced from binary numeration by grouping, beginning from the right, consecutive bits in triplets or, conversely, by replacing each octal number by its three corresponding bits.

EXAMPLE 1.4.– Determine the radix 8 representation for the decimal numbers 85 and 129.

Radix 8 representations are obtained by replacing each group of three bits by the equivalent octal number. We can therefore write:

$$85_{10} = 1010101_2 = \underbrace{001}_{1}\ \underbrace{010}_{2}\ \underbrace{101}_{5} = 125_8$$

Similarly,

$$129_{10} = 10000001_2 = \underbrace{010}_{2}\ \underbrace{000}_{0}\ \underbrace{001}_{1} = 201_8$$

1.5. Hexadecimal numeration

The hexadecimal number system or a representation with a radix 16 consists of the following symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

EXAMPLE 1.5.– Convert the decimal numbers 291 and 1000 to hexadecimal.

The number 291 is represented in radix 16 by:

$$\begin{aligned} 291_{10} &= (1 \times 16^2) + (2 \times 16^1) + (3 \times 16^0) \\ &= 123_{16} \end{aligned}$$

For the number 1000, we obtain:

$$\begin{aligned} 1\ 000_{10} &= (3 \times 16^2) + (14 \times 16^1) + (8 \times 16^0) \\ &= 3E8_{16} \end{aligned}$$

In practice, a series of integer divisions makes it possible to convert a decimal number to a hexadecimal number. The different remainders constitute the results of the conversion, beginning with the last, which is the MSD, to the first, which represents the LSD. We thus have:

291 ÷ 16 = 18 Remainder 3 (LSD)	1000 ÷ 16 = 62 Remainder 8 (LSD)
18 ÷ 16 = 1 Remainder 2	62 ÷ 16 = 3 Remainder 14
1 ÷ 16 = 0 Remainder 1 (MSD)	3 ÷ 16 = 0 Remainder 3 (MSD)
$291_{10} = 123_{16}$	$1000_{10} = 3E8_{16}$

Figure 1.4. *Decimal-hexadecimal conversion using the successive division method*

We can also proceed as demonstrated in Figure 1.4, the result of each conversion being made up of the successive remainders of the divisions.

Binary to hexadecimal conversion is done by grouping the bits representing the binary four by four and beginning from the right, conversely, replacing each hexadecimal digit by its four corresponding bits.

EXAMPLE 1.6.– Convert the decimal numbers 31 and 2, 988 into hexadecimal.

To obtain the equivalent hexadecimal from the binary representation, each group of four bits is replaced by the corresponding hexadecimal digit. We therefore have:

$$31_{10} = 11111_2 = \underbrace{0001}_{1}\ \underbrace{1111}_{15=F} = 1F_{16}$$

Similarly,

$$2\ 988_{10} = 101110101100_2 = \underbrace{1011}_{11=B}\ \underbrace{1010}_{10=A}\ \underbrace{1100}_{12=C} = BAC_{16}$$

It is generally more convenient to represent the value of an octet using two hexadecimal digits as it is more compact.

1.6. Representation in a radix B

In general, in radix *B* representation, a decimal number *N* may be decomposed as follows:

$$N_{10} = b_{n-1}B^{n-1} + \cdots + b_2B^2 + b_1B^1 + b_0B^0 \qquad [1.1]$$

$$= \sum_{i=0}^{n-1} b_i B^i \qquad [1.2]$$

where $B \geq 2$. Thus, the decimal number N is represented in radix B with n digits, $b_{n-1}\cdots b_2b_1b_0$.

Using n digits in a radix B numeration, we can code the decimal numbers from 0 to $B^n - 1$.

For an integer represented by n digits with a radix B, the formulas for conversion are as follows:

$$\begin{aligned}(b_{n-1}b_{n-2}\cdots b_2b_1b_0)_B &= \sum_{i=0}^{n-1} b_i B^i \\ &= b_{n-1}B^{n-1} + b_{n-2}B^{n-2} + \cdots + b_2B^2 + b_1B^1 + b_0B^0 \\ &= b_0 + B(b_1 + B(b_2 + B(\cdots + B(b_{n-2} + Bb_{n-1})\cdots))) \\ &= N_{10}\end{aligned} \qquad [1.3]$$

EXAMPLE 1.7.– Convert the binary number 110101_2, the octal number 5671_8 and the hexadecimal number $5CAD_{16}$ to decimal.

In decimal form, the number 110101_2 is written as:

$$\begin{aligned}110111_2 &= \mathbf{1} \times 2^5 + \mathbf{1} \times 2^4 + \mathbf{0} \times 2^3 + \mathbf{1} \times 2^2 + \mathbf{1} \times 2^1 + \mathbf{1} \times 2^0 \\ &= \mathbf{1} + 2(\mathbf{1} + 2(\mathbf{1} + 2(\mathbf{0} + 2(\mathbf{1} + 2 \times \mathbf{1})))) \\ &= 55_{10}\end{aligned}$$

For the number 5671_8, we get:

$$\begin{aligned}5671_8 &= \mathbf{5} \times 8^3 + \mathbf{6} \times 8^2 + \mathbf{7} \times 8^1 + \mathbf{1} \times 8^0 \\ &= \mathbf{1} + 8(\mathbf{7} + 8(\mathbf{6} + 8 \times \mathbf{5})) \\ &= 3001_{10}\end{aligned}$$

The conversion of the number $5CAD_{16}$ to decimal is effected by:

$$\begin{aligned}5CAD_{16} &= \mathbf{5} \times 16^3 + \mathbf{12} \times 16^2 + \mathbf{10} \times 16^1 + \mathbf{13} \times 16^0 \\ &= \mathbf{13} + 16(\mathbf{10} + 16(\mathbf{12} + 16 \times \mathbf{5})) \\ &= 23725_{10}\end{aligned}$$

1.7. Binary-coded decimal numbers

To represent a 8421-type *binary-coded decimal* (BCD) number, each digit must be replaced by its equivalent 4-bit binary.

EXAMPLE 1.8.– Give the BCD representation for the decimal numbers 90 and 873.

The BCD representation of the number 90 is written as follows:

$$90_{10} = 1001\ 0000_{BCD}$$

For the number 873, we have:

$$873_{10} = 1000\ 0111\ 0011_{BCD}$$

Table 1.1 gives the hexadecimal, octal, binary and BCD representations of numbers from 0 to 15.

Table 1.1. *Conversion tables for 0 numbers to 15*

Decimal number	Representation			
	Hexadecimal	Octal	Binary	BCD
0	0	0	0000	0000
1	1	1	0001	0001
2	2	2	0010	0010
3	3	3	0011	0011
4	4	4	0100	0100
5	5	5	0101	0101
6	6	6	0110	0110
7	7	7	0111	0111
8	8	10	1000	1000
9	9	11	1001	1001
10	A	12	1010	0001 0000
11	B	13	1011	0001 0001
12	C	14	1100	0001 0010
13	D	15	1101	0001 0011
14	E	16	1110	0001 0100
15	F	17	1111	0001 0101

It must be noted that with n bits, we can represent the decimal numbers between 0 and $10^{n/4} - 1$. In addition to the 8421 BCD code, there are other types of BCD codes.

1.8. Representations of signed integers

Several approaches may be adopted to represent signed integers in digital systems: the *sign-magnitude (SM) representation, two's complement (2C) representation, and excess-E (XSE) representation*. Each of these approaches assumes the use of a format (or number of bits) fixed beforehand.

1.8.1. *Sign-magnitude representation*

The simplest approach allowing for the representation of a signed integer consists of reserving the MSB for the number sign and the remaining bits for the number magnitude. If the sign bit is set to 0, the number is positive, and if the sign bit is set to 1, the number is negative.