Positional Numbering System

A **positional numbering system** uses a set of symbols. The value that each symbol represents, however, depends on its **face value** and its **place value,** the value associated with the position it occupies in the number. In other words, we have the following.

$$\text{Symbol value} = \text{Face value} \times \text{Place value}$$
$$\text{Number value} = \text{Sum of Symbol values}$$

In this appendix, we discuss only integers, numbers with no fractional part; the discussion of reals, numbers with a fractional part, is similar.

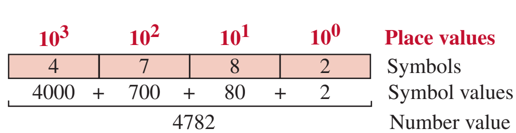
B.1 DIFFERENT SYSTEMS

We first show how integers can be represented in four different systems: base 10, base 2, base 16, and base 256.

B.1.1 Base 10: Decimal

The first positional system we discuss is called the **decimal system.** The term *decimal* is derived from the Latin root *decem* (meaning *ten*). The decimal system uses 10 symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9) with the same face values as the symbols. The place values in the decimal number system are powers of 10. Figure B.1 shows the place values and the symbol values in the integer 4782.

Figure B.1 *An example of a decimal number*



The decimal system uses 10 symbols in which the place values are powers of 10.

B.1.2 Base 2: Binary

The second positional system we discuss is called the **binary system.** The term *binary* is derived from the Latin root *bi* (meaning *two by two*). The binary system uses 2 symbols (0 and 1) with the same face values as the symbols. The place values in the binary number system are powers of 2. Figure B.2 shows the place values and the symbol values in the binary $(1101)_2$. Note that we use subscript 2 to show that the number is in binary.

Figure B.2 *An example of a binary number*

2^3		2^2		2^1		2^0	**Place values**
1		1		0		1	Symbols
8	+	4	+	0	+	1	Symbol values
13							Number value (in decimal)

The binary system uses 2 symbols in which the place values are powers of 2.

B.1.3 Base 16: Hexadecimal

The third positional system we discuss is called the **hexadecimal system.** The term *hexadecimal* is derived from the Greek root hex (meaning 6) and the Latin root *decem* (meaning *ten*). The hexadecimal system uses 16 symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F). The face value of the first ten symbols are the same as the symbols, but the face values of the symbols A to F are 10 to 15 respectively. The place values in the hexadecimal number system are powers of 16. Figure B.3 shows the place values and the symbol values in the hexadecimal $(A20E)_{16}$. Note that we use subscript 16 to show that the number is in hexadecimal.

Figure B.3 *An example of a hexadecimal number*

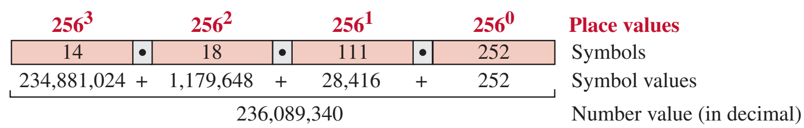
16^3		16^2		16^1		16^0	**Place values**
A		2		0		E	Symbols
40,960	+	512	+	0	+	14	Symbol value (in decimal)
41,486							Number value (in decimal)

The hexadecimal system uses 16 symbols in which the place values are powers of 16.

B.1.4 Base 256: Dotted-Decimal Notation

The fourth positional system we discuss is base 256, which is called ***dotted-decimal notation*.** This system is used to represent IPv4 addressing. The place values in this system are powers of 256. However, since using 256 symbols is almost impossible, the symbols in this system are decimal numbers between 0 and 255, with the same face values as the symbols. To separate these numbers from each other, the system uses a dot, as discussed in Chapter 18. Figure B.4 shows the place values and the symbol values of the address (14.18.111.252). Note that we never use more than four symbols in an IPv4 address.

Figure B.4 *An example of a dotted-decimal notation*



Dotted-decimal notation uses decimal numbers (0 to 255) as symbols, but inserts a dot between each symbol.

B.1.5 Comparison

Table B.1 shows how three different systems represent the decimal numbers 0 through 15. For example, decimal 13 is equivalent to binary $(1101)_2$, which is equivalent to hexadecimal D.

Table B.1 *Comparison of three systems*

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

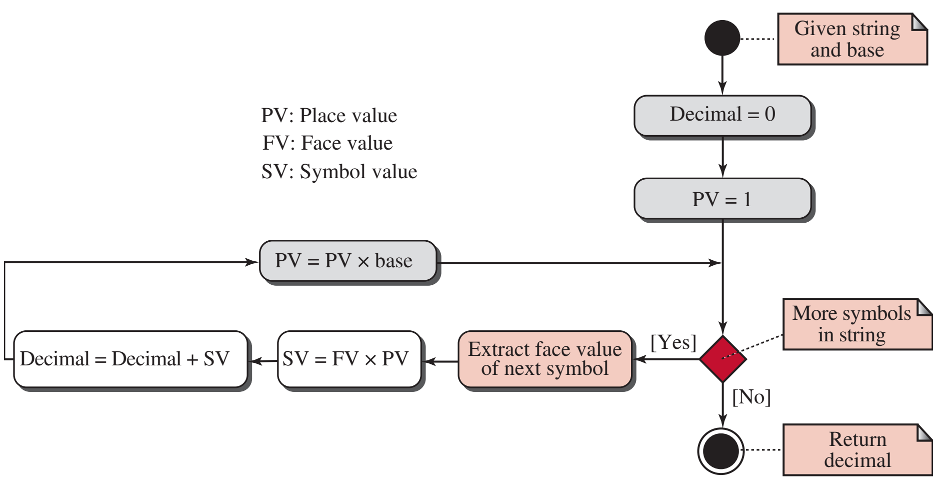
B.2 CONVERSION

We need to know how to convert a number in one system to the equivalent number in another system.

B.2.1 Conversion from Any Base to Decimal

Figures B.2 to B.4 actually show how we can manually convert a number in any base to decimal. However, it is easier to use the algorithm in Figure B.5. The algorithm uses the fact that the next place value is the previous value multiplied by the base (2, 16, or 256). The algorithm is a general one that can be used to convert a string of symbols in a given base to a decimal number. The only section in the algorithm that is different for each base is how to extract the next symbol in the string and find its face value. In the case of base 2, it is simple: the face value can be found by changing the symbol to a numeric value. In the case of base 16, we need to consider the case that the face value of symbol A is 10, the face value of symbol B is 11, and so on. In the case of base 256, we need to extract each string delimited by dots and change the string to its numerical value.

Figure B.5 *Algorithm to convert from any base to decimal*



The manual implementation of the above algorithm for small numbers can be shown in a few examples.

Example B.1

Show the equivalent of the binary number $(11100111)_2$ in decimal.

Solution

We follow the algorithm as shown below.

128		**64**		**32**		**16**		**8**		**4**		**2**		**1**	Place values
1		1		1		0		0		1		1		1	Face values
128		64		32		0		0		4		2		1	Symbol values
231		103		39		7		7		7		3		1	Decimal = 0

The value of decimal is initially set to 0. When the loop is terminated, the value of decimal is 231.

Example B.2

Show the equivalent of the IPv4 address 12.14.67.24 in decimal.

Solution

We follow the algorithm as shown below.

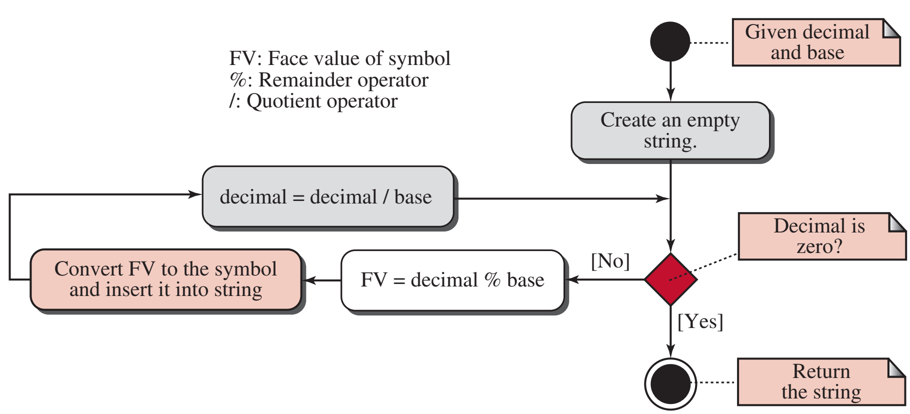
16,777,216		**65,536**		**256**		**1**	Place values
12	•	14	•	67	•	24	Face values
201,326,592		917,504		17,152		24	Symbol values
202,261,272		934,680		17,176		24	Decimal = 0

The value of decimal is initially set to 0. When the loop is terminated, the value of decimal is 202,261,272.

B.2.2 Conversion from Decimal to Any Base

Conversion from a decimal value to any base can be done if we continuously divide the decimal number by the base to find the remainder and the quotient (see Figure B.6). The remainder is the face value of the next symbol; the quotient is the decimal value to be used in the next iteration. As in the case of inverse conversion, we need to have a separate algorithm to change the face value of a symbol, in the corresponding base, to the actual symbol and insert it in the string representing the converted number.

Figure B.6 *Conversion from decimal to any base*



We can show how we can manually follow the algorithm in a few examples.

Example B.3

Convert the decimal number 25 to its binary equivalent.

Solution

We continuously divide the decimal value by 2 (the base of the binary system) until the quotient becomes 0. In each division we interpret the value of the remainder as the next symbol to be inserted in the hexadecimal string. The down arrow shows the remainder; the left arrow shows the quotient. When the decimal value becomes 0, we stop. The result is the binary string $(11001)_2$.

0	←	1	←	3	←	6	←	12	←	**25**	Decimal
		↓		↓		↓		↓		↓	
		1		1		0		0		1	Binary

Example B.4

Convert the decimal number 21,432 to its hexadecimal equivalent.

Solution

We continuously divide the decimal value by 16 (the base of the hexadecimal system) until the quotient becomes 0. In each division we interpret the value of the remainder as the next symbol to be inserted in the hexadecimal string. The result is the hexadecimal string $(53B8)_{16}$.

0	←	5	←	83	←	1339	←	**21432**	Decimal
		↓		↓		↓		↓	
		5		3		B		8	Hexadecimal

Example B.5

Convert the decimal number 73,234,122 to base 256 (IPv4 address).

Solution

We continuously divide the decimal value by 256 (the base) until the quotient becomes 0. In each division we interpret the value of the remainder as the next symbol to be inserted in the IPv6 address. We also insert dots as required in the dotted-decimal notation. The result is the IPv4 address 4.93.118.202.

0	←	4	←	1117	←	286,070	←	**73,234,122**	Decimal
		↓		↓		↓		↓	
		4	•	93	•	118	•	202	IPv4 address

B.2.3 Other Conversions

Conversion from a nondecimal system to another nondecimal system is often easier. We can easily convert a number in binary to hexadecimal by converting a group of 4 bits into 1 hexadecimal digit. We can also convert a hexadecimal digit into a group of 4 bits. We give a few examples to show the process.

Example B.6

Convert the binary number $(1001111101)_2$ to its equivalent in hexadecimal.

Solution

We create groups of 4 bits from the right. We then replace each group with its equivalent hexadecimal digit. Note that we need to add two extra 0s to the last group.

0010	**0111**	**1101**	Binary
↓	↓	↓	
2	7	D	Hexadecimal

The result is $(27\text{D})_{16}$.

Example B.7

Convert the hexadecimal number $(3\text{A}2\text{B})_{16}$ to its equivalent in binary.

Solution

We change each hexadecimal digit to its 4-bit binary equivalent.

3	**A**	**2**	**B**	Hexadecimal
↓	↓	↓	↓	
0011	1010	0010	1011	Binary

The result is $(0011\ 1010\ 0010\ 1011)_2$.

Example B.8

Convert the IPv4 address 112.23.78.201 to its binary format.

Solution

We replace each symbol with its equivalent 8-bit binary.

112	•	**23**	•	**78**	•	**201**	IPv4 address
↓		↓		↓		↓	
01110000		00010111		01001110		11001001	Binary

The result is $(01110000\ 00010111\ 01001110\ 11001001)_2$.