Number Systems

Dr. Chandan Karfa

CSE IIT Guwahati

4/9/2020

Text Book

- Chapter 1: Z. Kohavi and N. Jha, Switching and Finite Automata Theory, 3rd Ed., Cambridge University Press, 2010.
- Chapter 1, Sec 1.6 M. M. Mano and M. D. Ciletti, Digital Design, 5th Ed., Pearson Education (Signed numbers)

Number System

- Convenient as the decimal number system generally is, its usefulness in machine computation is limited because of the nature of practical electronic devices.
- In most present digital machines, the numbers are represented, and the arithmetic operations performed, in a different number system called the **binary number system.**
- The representation of numbers in various systems and with methods of conversion from one system to another.

Number representation

- An ordinary decimal number actually represents a polynomial in powers of 10.
  For example, the number 123.45 represents the polynomial

  $$123.45 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}.$$

  This method of representing decimal numbers is known as the **decimal number system**, and the number 10 is referred to as **the base (or radix)** of the system.

- In a system whose base is b, a positive number N represents the polynomial. where the base or radix *b* is an integer greater than 1 and the *a*'s are integers in the range $0 \le a_i \le b - 1$.

$$\begin{aligned} N &= a_{q-1}b^{q-1} + \cdots + a_0 b^0 + \cdots + a_{-p}b^{-p} \\ &= \sum_{i=-p}^{q-1} a_i b^i, \end{aligned}$$

Representation of integers

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Conversion of bases

Conversion a number N from base b1 to base b2. $(N)_{b1}$ => $(X)_{b2}$

Case 1: b1 < b2:

- base-b2 arithmetic can be used in the conversion process.
- The conversion technique involves expressing number $(N)_{b1}$ as a polynomial in powers of b1 and evaluating the polynomial using base-b2 arithmetic.

Example We wish to express the numbers $(432.2)_8$ and $(1101.01)_2$ in base 10. Thus

$$(432.2)_8 = 4 \times 8^2 + 3 \times 8^1 + 2 \times 8^0 + 2 \times 8^{-1} = (282.25)_{10},$$
$$(1101.01)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (13.25)_{10}.$$

In both cases, the arithmetic operations are done in base 10.

$(145)_{10}$ == $(X)_{16}$

Conversion Procedure

Case 2: b1 > b2

- It is more convenient to use base-b1 arithmetic. The conversion procedure will be obtained by considering separately the integer and fractional parts of N .
- Let $(N)_{b1}$ be an integer whose value in base b2 is given by

$$(N)_{b_1} = a_{q-1}b_2^{q-1} + a_{q-2}b_2^{q-2} + \cdots + a_1 b_2^1 + a_0 b_2^0.$$

- To find the values of the a's, let us divide the above polynomial by b2 .

$$\frac{(N)_{b_1}}{b_2} = \underbrace{a_{q-1}b_2^{q-2} + a_{q-2}b_2^{q-3} + \cdots + a_1}_{Q_0} + \frac{a_0}{b_2}.$$

Cont'd...

- Thus, the least significant digit of $(N)_{b2}$, i.e., a_0 , is equal to the first remainder. The next most significant digit, a_1 , is obtained by dividing the quotient Q_0 by b2 , i.e.,

$$\left(\frac{Q_0}{b_2}\right)_{b_1} = \underbrace{a_{q-1}b_2^{q-3} + a_{q-2}b_2^{q-4} + \cdots}_{Q_1} + \frac{a_1}{b_2}.$$

- The remaining a's are evaluated by repeated divisions of the quotients until Q_{q-1} is equal to zero.
- If N is finite, the process must terminate.

Integral Part conversion

Q_i	r_i
68	$4 = a_0$
8	$4 = a_1$
1	$0 = a_2$
	$1 = a_3$

$(548)_{10} = (1044)_8$

Q_i	r_i
57	$3 = a_0$
9	$3 = a_1$
1	$3 = a_2$
	$1 = a_3$

$(345)_{10} = (1366)_6$

Fractional Part Conversion

- If $(N)_{b1}$ is a fraction, a dual procedure is employed. It can be expressed in base $b2$ as follows:

$$(N)_{b_1} = a_{-1}b_2^{-1} + a_{-2}b_2^{-2} + \cdots + a_{-p}b_2^{-p}.$$

- The most significant digit, a_{-1}, can be obtained by multiplying the polynomial by $b2$:

$$b_2 \cdot (N)_{b_1} = a_{-1} + a_{-2}b_2^{-1} + \cdots + a_{-p}b_2^{-p+1}.$$

- If the above product is less than 1 then a_{-1} equals 0; if the product is greater than or equal to 1 then a_{-1} is equal to the integer part of the product.

Fractional Part Conversion

- The next most significant digit, a_{-2}, is found by multiplying the fractional part of the above product part by $b2$ and determining its integer part; and so on.
- This process does not necessarily terminate since it may not be possible to represent the fraction in base $b2$ with a finite number of digits.

Example To convert $(0.3125)_{10}$ to base 8, find the digits as follows:

$$0.3125 \times 8 = 2.5000, \quad \text{hence} \quad a_{-1} = 2;$$
$$0.5000 \times 8 = 4.0000, \quad \text{hence} \quad a_{-2} = 4.$$

Thus $(0.3125)_{10} = (0.24)_8$.
Similarly, the computation below proves that $(0.375)_{10} = (0.011)_2$:

$$0.375 \times 2 = 0.750, \quad \text{hence} \quad a_{-1} = 0;$$
$$0.750 \times 2 = 1.500, \quad \text{hence} \quad a_{-2} = 1;$$
$$0.500 \times 2 = 1.000, \quad \text{hence} \quad a_{-3} = 1.$$

Example

- Convert $(432.354)_{10}$ to Binary.

Q_i	r_i
216	$0 = a_0$
108	$0 = a_1$
54	$0 = a_2$
27	$0 = a_3$
13	$1 = a_4$
6	$1 = a_5$
3	$0 = a_6$
1	$1 = a_7$
	$1 = a_8$

Hence $(432)_{10} = (110110000)_2$.

$$
\begin{array}{lll}
0.354 \times 2 = 0.708, & \text{hence} & a_{-1} = 0, \\
0.708 \times 2 = 1.416, & \text{hence} & a_{-2} = 1, \\
0.416 \times 2 = 0.832, & \text{hence} & a_{-3} = 0, \\
0.832 \times 2 = 1.664, & \text{hence} & a_{-4} = 1, \\
0.664 \times 2 = 1.328, & \text{hence} & a_{-5} = 1, \\
0.328 \times 2 = 0.656, & \text{hence} & a_{-6} = 0, \\
 & & a_{-7} = 1, \\
 & & \text{etc.}
\end{array}
$$

$$(0.354)_{10} = (0.0101101\cdots)_2.$$

The conversion is usually carried up to the desired accuracy. In our example, reconversion to base 10 shows that

$$(110110000.0101101)_2 = (432.3515)_{10}$$

Binary to Octal/Hexadecimal Conversion

- The conversion from and to binary, octal, and hexadecimal plays an important role in digital computers, because shorter patterns of hex characters are easier to recognize than long patterns of 1’s and 0’s.
- Conversion of binary to octal is accomplished by partitioning the binary number into groups of three digits each, starting from the binary point and proceeding to the left and to the right. The corresponding octal digit is then assigned to each group

$$\begin{array}{ccccccccccl} (10 & 110 & 001 & 101 & 011 & \cdot & 111 & 100 & 000 & 110)_2 & = (26153.7406)_8 \\ 2 & 6 & 1 & 5 & 3 & & 7 & 4 & 0 & 6 & \end{array}$$

Octal/Hexadecimal to Binary Conversion

- Conversion from binary to hexadecimal is similar, except that the binary number is divided into groups of *four* digits:

$$\begin{array}{cccccccl} (10 & 1100 & 0110 & 1011 & \cdot & 1111 & 0010)_2 & = (2C6B.F2)_{16} \\ 2 & C & 6 & B & & F & 2 & \end{array}$$

- Octal or hexadecimal to binary is done by reversing the preceding procedure. Each octal digit is converted to its three-digit binary equivalent. Similarly, each hexadecimal digit is converted to its four-digit binary equivalent.

$$\begin{array}{lccccccc} (673.124)_8 = & (110 & 111 & 011 & \cdot & 001 & 010 & 100)_2 \\ & 6 & 7 & 3 & & 1 & 2 & 4 \end{array}$$

$$\begin{array}{lccccc} (306.D)_{16} = & (0011 & 0000 & 0110 & \cdot & 1101)_2 \\ & 3 & 0 & 6 & & D \end{array}$$

Complements of a Number

- To **simplify the subtraction operation**
- N in base r having n digits, the **(*r* - 1)'s complement of *N***, i.e., its diminished radix complement, is defined as $(r^n - 1) – N$
- **The 1's complement of a binary number is formed by changing 1's to 0's and 0's to 1's.**
  - The 1's complement of 1011000 is 0100111.
- The ***r*'s complement** of an n-digit number N in base r is as $r^n - N$ for N != 0 and as 0 for N = 0.
- r's complement is obtained by adding 1 to the $(r - 1)$'s complement, i.e., $r^n - N = [(r^n - 1) – N$
  - 2's complement of 0110111 is 1001001
- The 2's complement can be formed by leaving all least significant 0's and the first 1 unchanged and replacing 1's with 0's and 0's with 1's in all other higher significant digits.

- **The complement of the complement restores the number to its original value .**
  - the *r*'s complement of *N* is r^n - *N*,
  - the complement of the complement is $r^n - (r^n - N) = N$ and is equal to the original number

Subtraction of Unsigned Numbers

The subtraction of two n-digit unsigned numbers $M - N$ in base r can be done as follows:

1. Add the minuend M to the r's complement of the subtrahend N. Mathematically, $M + (r^n - N) = M - N + r^n$.
2. If $M \geq N$, the sum will produce an end carry r^n, which can be discarded; what is left is the result $M - N$.
3. If $M < N$, the sum does not produce an end carry and is equal to $r^n - (N - M)$, which is the r's complement of $(N - M)$. To obtain the answer in a familiar form, take the r's complement of the sum and place a negative sign in front.

Example

Given the two binary numbers $X = 1010100$ and $Y = 1000011$, perform the subtraction **(a)** $X - Y$ and **(b)** $Y - X$ by using 2's complements.

(a)

$$\begin{array}{rr}
X = & 1010100 \\
\text{2's complement of } Y = & +\ \underline{0111101} \\
\text{Sum} = & 10010001 \\
\text{Discard end carry } 2^7 = & -\ \underline{10000000} \\
\textit{Answer}: X - Y = & 0010001
\end{array}$$

(b)

$$\begin{array}{rr}
Y = & 1000011 \\
\text{2's complement of } X = & +\ \underline{0101100} \\
\text{Sum} = & 1101111
\end{array}$$

There is no end carry. Therefore, the answer is $Y - X = -(\text{2's complement of } 1101111) = -0010001$.

Signed Binary Number

- If binary number is signed, then the leftmost bit represents the sign and the rest of the bits represent the number.
    - Signed Magnitude: 10001001 (-9)
    - 1’s complement: 11110110 (-9)
    - 2’s complement: 11110111 (-9)
- The signed-2’s-complement representation of -N is obtained by taking the 2’s complement of the positive number N, including the sign bit.
    - Signed-2’s-complement representation of -9: 11110111
- 2’s complement form is used for number representation in Digital systems

Signed Binary Numbers

Decimal	Signed-2's Complement	Signed-1's Complement	Signed Magnitude
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
−0	—	1111	1000
−1	1111	1110	1001
−2	1110	1101	1010
−3	1101	1100	1011
−4	1100	1011	1100
−5	1011	1010	1101
−6	1010	1001	1110
−7	1001	1000	1111
−8	1000	—	—

Arithmetic Addition

- The addition of two signed binary numbers with negative numbers represented in signed- 2’s-complement form is obtained from the addition of the two numbers, including their sign bits. A carry out of the sign-bit position is discarded.

$$\begin{array}{rr} +\ 6 & 00000110 \\ +13 & 00001101 \\ \hline +19 & 00010011 \end{array} \qquad \begin{array}{rr} -\ 6 & 11111010 \\ +13 & 00001101 \\ \hline +\ 7 & 00000111 \end{array}$$

$$\begin{array}{rr} +\ 6 & 00000110 \\ -13 & 11110011 \\ \hline -\ 7 & 11111001 \end{array} \qquad \begin{array}{rr} -\ 6 & 11111010 \\ -13 & 11110011 \\ \hline -19 & 11101101 \end{array}$$

Arithmetic Subtraction

- Subtraction of two signed binary numbers when negative numbers are in 2’s-complement form is simple and can be stated as follows:
  - Take the 2’s complement of the subtrahend (including the sign bit) and add it to the minuend (including the sign bit). A carry out of the sign-bit position is discarded.

$$(\pm A) - (+B) = (\pm A) + (-B);$$
$$(\pm A) - (-B) = (\pm A) + (+B).$$

2’s Complement

- It is worth noting that binary numbers in the **signed-complement** system are added and subtracted by the same basic addition and subtraction rules as **unsigned numbers**.
- Therefore, **computers need only one common hardware circuit to handle both types of arithmetic.**
- This consideration has resulted in the **signed-complement** system being used in virtually all arithmetic units of computer systems.