Chapter 1. CONVENTIONAL NUMBER SYSTEMS

§1. Conventional Number Systems

§1.1. Binary number system

A binary number of length n is an ordered sequence

$$X = (x_{n-1}, x_{n-2}, \ldots, x_1, x_0)_2 \tag{1}$$

of binary digits (bits) where each bit $x_i \in \{0, 1\}$ can assume one of the values 0 or 1. The above n-tuple represents integer value

$$X = x_{n-1} \times 2^{n-1} + x_{n-2} \times 2^{n-2} + \cdots + x_1 \times 2^1 + x_0 \times 2^0 = \sum_{i=0}^{n-1} x_i 2^i. \tag{2}$$

The weight of the digit x_i, is 2^i, where 2 is called the *radix* of the number system. The radix can be denoted as the subscript of a representation as shown in (1). If the radix is understood in context, the radix is often omitted in a number representation. The digit set of this number system is $(0, 1)$.

Sometimes it may be useful to differentiate between representation of a number (1) and its value (2).

§1.2. Radix-r Number Systems $(r \geq 2)$

A n-tuple

$$(x_{n-1}, x_{n-2}, \ldots, x_1, x_0)_r, \;\; x_i \in \{0, 1, \ldots, r-1\},$$

represents the integer value

$$X = x_{n-1} \times r^{n-1} + x_{n-2} \times r^{n-2} + \cdots + x_1 \times r^1 + x_0 \times r^0 = \sum_{i=0}^{n-1} x_i r^i. \tag{3}$$

The radix-r number system has the digit set $\{0, 1, ...r-1\}$.

§1.3. Conventional Number System

- A number system is in general defined by the set of values and a set of rules that gives the mapping between the sequence of digits and their numerical values.
- A conventional number system is a nonredundant, weighted and positional.
- A conventional number system is fixed-radix system.
- A number $(x_{n-1}, \ldots, x_0)$ in conventional number system can be evaluated with

$$X = \sum_{i=0}^{n-1} x_i r^i.$$

§2. Machine Representations of Numbers

Since the register for storing a number always has a fixed length, there is a finite number of distinct values that can be represented. Let Xmm and Xmax denote the smallest and largest representable values, respectively. We say that $[X_{\min}, X_{\max}]$ is the range of the representable numbers. Any arithemtic operation that attempts to produce a result larger than $X_{\max}$ (or smaller than $X_{\min}$ will produce an incorrect result (overflow).

§2.1. Fix-point representations

A sequence of n digits does not necessarily have to represent an integer. We may use such a sequence to represent a mixed number that has both integral and fractional parts. Let $n = k + m$. Then a representation

$$(x_{k-1}, x_{k-2} \cdots, x_1, x_0, .x_{-1}, x_{-2}, \cdots, x_{-m})_r$$

has value

$$X = x_{k-1}r^{k-1} + x_{k-2}r^{k-2} + \cdots + x_1 r + x_0 + x_{-1}r^{-1} + \cdots + x_{-m}r^{-m} = \sum_{i=-m}^{k-1} x_i r^i.$$

The radix point is not stored in the register but is understood to be in a fixed position between the k most significant digits and the m least significant digits. For this reason we call such representations *fixed-point* representations.

To avoid the need to tell where the radix point is, we introduce the notion of a *unit in the last position (ulp)*, which is the weight of the least significant digit,

$$ulp = r^{-m}.$$

The other popular method to represent a number in hardware is called floating-point representation, which will be discussed later.

§2.2. Radix Conversions

Given a number X in radix-r_s number system (source number system), we wish to find its resentation in the radix-r_d number system (destination number system). That is, to find x_i in $X = (x_{k-1} \cdots x_{-m})_{r_d}$.

Let $X = X_I + X_F$, where X_I and X_F are respectively the integral and fractional parts of X. Then X_I and X_F can be evaluated respectively by

$$X_I = \sum_{i=0}^{k-1} x_i r_s^i,$$

and

$$X_F = \sum_{i=-m}^{-1} x_i r_s^i.$$

Then we write

$$\begin{aligned} X_I &= x_{k-1}r_d^{k-1} + \cdots + x_1 r_d + x_0 \\ &= r_d(x_{k-1}r_d^{k-2} + \cdots + x_1) + x_0. \end{aligned}$$

Clearly, x_0 can be obtained as the remainder of r_d dividing X_I, and the quotient is $x_{k-1}r_{r_d}^{k-2} + \cdots + x_1$. Then, x_1 can be obtained as the remainder of r_d dividing the above quotient $x_{k-1}r_{r_d}^{k-2} + \cdots + x_1 = r_d(x_{k-1}r_d^{k-3} + \cdots + x_2) + x_1$. Repeat this procedure until a zero quotient is reached. For the fractional part,

$$X_F = x_{-1}r_d^{-1} + x_{-2}r_d^{-2} + \cdots .$$

To obtain x_{-1}, we multiply X_F by r_d and it follows

$$r_d \cdot X_F = x_{-1} + x_{-2}r_d^{-1} + \cdots .$$

Clearly, the integral part of $r_d \cdot X_F$ is x_{-1} and we keep the fractional part $x_{-2}r_d^{-1} + x_{-3}r_d^{-2} + \cdots$. To obtain x_{-2}, we multiply the above fractional part by r_d and it follows

$$x_{-2} + x_{-3} \cdot r_d^{-1} + \cdots ,$$

then we obtain x_2 as the integral part in the above expression. Repeat this procedure until a required precision is met or the fractional part is zero.

Example 1 *Convert* $X = 46.375_{10}$ *to the binary form.*

Solution:

$$X_I = 46, \quad \text{and } X_F = 0.375.$$

For the integral part, we have the following

Integer: Decimal-to-binary		
Dividing-by-r_d	Quotient	Remainder
46/2	23	$0 = x_0$
23/2	11	$1 = x_1$
11/2	5	$1 = x_2$
5/2	2	$1 = x_3$
2/2	1	$0 = x_4$
1/2	0	$1 = x_5$

For the fractional part, it follows

Fraction: Decimal-to-binary		
Multiplying-by-r_d	Fractional part	Integral part
0.375×2	0.75	$0 = x_{-1}$
0.75×2	0.5	$1 = x_{-2}$
0.5×2	0	$1 = x_{-3}$

§3. Representations of Negative Numbers

In this section we will talk about three methods for representing a negative number:

1. Sign-magnitude method;

2. Biased method;

3. Radix complement representation method;

4. Diminished-radix complement representation method.

§3.1. Sign-magnitude method

In this method, The sign and magnitude are represented separately, where the sign is represented with the first digit while the remaining $n-1$ digits represent the magnitude.

	Minimal	Maximal	Range
Positive	$0\,0\,\cdots\,0$	$0\,(r-1)\cdots(r-1)$	$[0, r^{k-1}-ulp]$
Negative	$(r-1)\,(r-1)\cdots(r-1)$	$(r-1)\,0\,\cdots\,0$	$[-(r^{k-1}-ulp), 0]$

Disadvantage of this method: The operation to be performed may depend on the signs of the operands.

§3.2. Biased method

The basic idea of this method is to let $[0, \mathrm{Max}]$ to represent $[-\mathrm{Bias}, \mathrm{Max}-\mathrm{Bias}]$, where the Bias is a fixed positive integer. This method sometimes is also called "excess-*Bias*" method.

For example, to apply a biased representation to the integers falling into $[-8, +7]$, we can choose bias equal to 8 such that the biased representation $x+8$ is always greater than or equal to zero. Then we use $x+8$ to represent the value x. This is also called excess-8 method.

What are the advantages and disadvantages of this method? Hint: One major disadvantage can be implied from the following:

$$x+y+Bias = (x+Bias)+(y+Bias)-Bias.$$

§3.3. Complement representations

There are two alternatives for complement methods:

- Radix complement (also called 2's complement in the binary system)
- Diminished-radix complement (called 1's complement in the binary system)

The idea of complement methods is to represent a negative number, $-Y$, as $(R-Y)$ where R is a constant, while representing a positive number in the same way as the sign-magnitude method.

The consistant R can be chosen as follows

Representation	Consistant R
Radix complement	$R = r^k$
Diminished-radix complement	$R = r^k - ulp$

For the binary number system it follows

Representation	Consistant R
Two's complement	$R = 2^k$
One's complement	$R = 2^k - ulp$

For a given number, a method to find its complement representations can be derived directly from the definition: If the given number has a positive value (including zero), simply add zero(s) to its most significant end so that the representation has k digits in length. Otherwise, when the given number has a negative value, say, $-Y$, then the complement representation is $R - Y$.

There is another method to obtain complement representation, especially for obtaining 1's and 2's complement forms, which is commonly used in practice. Note that this method differentiates itself from the previous one only by the way converting negative numbers. Assume the given negative number is $-Y$. Then its diminished radix complement form can be obtained by writing Y in k-digit and then complementing every digit in Y. For its radix complement form, one first obtaining the diminished complement form and then adding ulp (one at the last position).

In the following we will show that the two methods discussed above are equivalent. Let $X = x_{k-1} \cdots x_{-m}$ and let $\overline{X} = \overline{x_k} \cdots \overline{x_m}$ be obtained by complementing every digit of X, where $\overline{x}_i = (r-1) - x_i$. Then we perform the following additions:

$$\begin{array}{cccccc}
 & & x_{k-1} & \cdots & x_{-m} & \\
+ & & \overline{x}_{k-1} & \cdots & \overline{x}_{-m} & \\
\hline
 & & (r-1) & \cdots & (r-1) & \\
+ & & & & 1 & \\
\hline
 & 1 & 0 & \cdots & 0 & = \; r^k
\end{array}$$

So, from

$$X + \overline{X} + ulp = r^k,$$

when $R = r^k$, it follows

$$R - X = r^k - X = \overline{X} + ulp, \tag{4}$$

and when $R = r^k - ulp$, we have

$$R - X = r^k - ulp - X = \overline{X}, \tag{5}$$

Obviously, the radix complement and diminished-radix complement representations of a negative number, $-X$, can be obtained with (4) and (5), respectively.

Sequence	Unsigned Binary	Machine Representations			
		Signed-magnitude	Biased (8)	2's Complement	1's Complement
0000	0	0	-8	0	0
0001	1	1	-7	1	1
0010	2	2	-6	2	2
0011	3	3	-5	3	3
0100	4	4	-4	4	4
0101	5	5	-3	5	5
0110	6	6	-2	6	6
0111	7	7	-1	7	7
1000	8	-0	0	-8	-7
1001	9	-1	1	-7	-6
1010	10	-2	2	-6	-5
1011	11	-3	3	-5	-4
1100	12	-4	4	-4	-3
1101	13	-5	5	-3	-2
1110	14	-6	6	-2	-1
1111	15	-7	7	-1	-0
Range	$[0, 15]$	$[-7, 7]$	$[-8, 7]$	$[-8, 7]$	$[-7, 7]$

Table 1: Several representation methods of binary systems with $l = 0$ and $n = k = 4$.

A table of the three representation methods of binary systems with k = n = 4 is given as follows. Given $X = (x_{n-1}, \cdots, x_0)$ in two's complement representation, we have the following procedure to find its value.

1. If $x_{n-1} = 0$, then the representation represents a positive number and its value is given by

$$X = \sum_{i=0}^{n-1} x_i 2^i. \tag{6}$$

2. If $x_{n-1} = 1$, then X is a negative integer and it can be evaluated with

$$\begin{aligned} X &= -\left(\sum_{i=0}^{n-2} \overline{x}_i 2^i + 1\right) \\ &= -\left(\sum_{i=0}^{n-2} (1 - x_i) 2^i + 1\right) \\ &= -2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i. \end{aligned} \tag{7}$$

Put the equations (7) and (6) together, we have

$$X = -x_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i.$$

Given $X = (x_{n-1}, \cdots, x_0)$ in one's complement representation, we can also obtain the following expression to evaluate X.

$$X = -x_{n-1} \cdot (2^{n-1} - ulp) + \sum_{i=0}^{n-2} x_i 2^i.$$

§3.4. Addition/subtraction with complement representations

Example 2 *Add X and Y, where $X = 13_{10} = (01101)_2$ and $Y = -8_{10} = (11000)_2$.*

$$\begin{array}{llcccccccc}
X+Y & : & & & & & & & & \\
 & & & & 0 & 1 & 1 & 0 & 1 & X = 13 \\
 & + & & & 1 & 1 & 0 & 0 & 0 & Y = -8 \\
\hline
 & & & 1 & 0 & 0 & 1 & 0 & 1 & 5
\end{array}$$

There is a carry-out but the result is correct (no overflow).

Example 3 *Add X and Y, where $X = -7_{10} = (11001)_2$ and $Y = -10_{10} = (10110)_2$.*

$$\begin{array}{llcccccccc}
X+Y & : & & & & & & & & \\
 & & & & 1 & 1 & 0 & 0 & 1 & X = -7 \\
 & + & & & 1 & 0 & 1 & 1 & 0 & Y = -10 \\
\hline
 & & & 1 & 0 & 1 & 1 & 1 & 1 & 15
\end{array}$$

There is a carry-out but the result is incorrect (overflow).

Example 4 *Add X and Y, where $X = 7_{10} = (00111)_2$ and $Y = 10_{10} = (01010)_2$.*

$$\begin{array}{llccccccc}
X+Y & : & & & & & & & \\
 & & & 0 & 0 & 1 & 1 & 1 & X = 7 \\
 & + & & 0 & 1 & 0 & 1 & 0 & Y = 10 \\
\hline
 & & & 1 & 0 & 0 & 0 & 1 & -15
\end{array}$$

we do not have any carry-out but the result is incorrect (overflow).

- If X and Y have opposite signs, no overflow can occur regardless whether there is a carry-out or not.

- If X and Y have the same sign and the sign of the result is different from that of the two operands, then an overflow occurs.