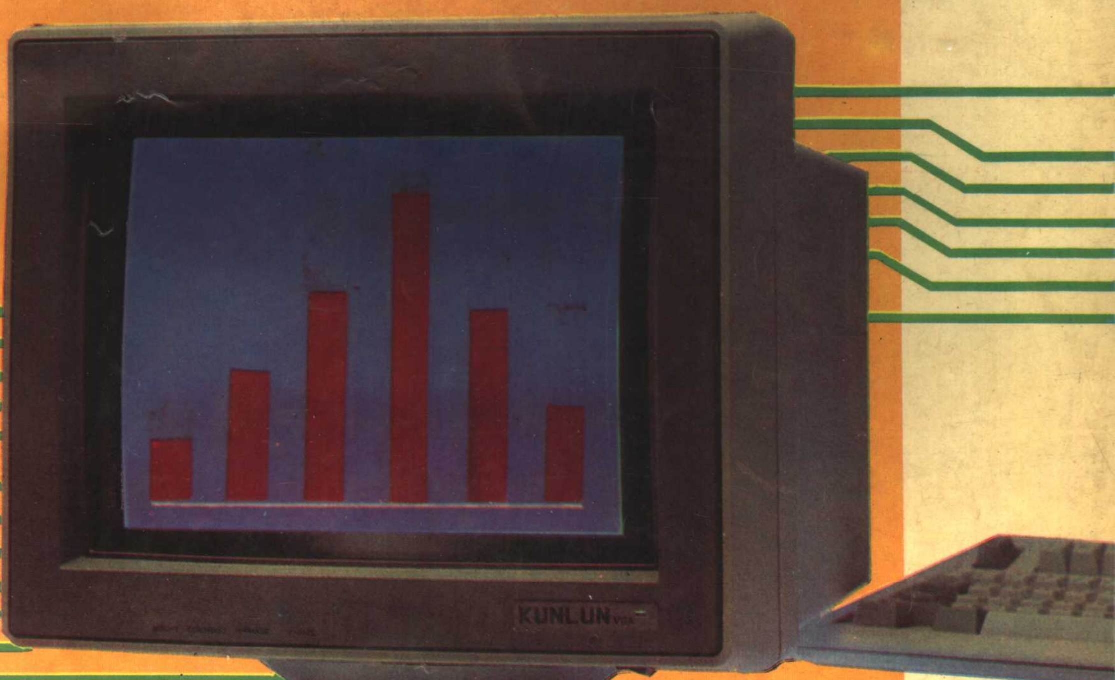
# 8088/8086 汇编语言程序设计

杨振生　单洪　编著






中国科学技术大学出版社

(皖)新登字 08 号

8088/8086 汇编语言程序设计
杨振生　单　洪　编著
*
中国科学技术大学出版社出版
(安徽省合肥市金寨路 96 号,邮政编码:230026)
安徽经纬激光照排中心排版　合肥丰航彩印厂印刷
安徽省新华书店发行
*
开本:787×1092/16　印张:26.25　字数:640 千
1994 年 1 月第 1 版　1994 年 1 月第 1 次印刷
印数:1—10000 册
ISBN 7-312-00539-X/TP·74　定价:13.60 元

# 前　言

目前IBM PC机及其兼容机在我国各行各业广为流行，在生产、科研、教学、管理以及应用开发等方面正在发挥着重要作用，并对我国计算机事业的普及和发展产生了深远影响。

为了提高工作效率和充分利用机器的性能，人们大都喜欢使用汇编语言解决各自面临的问题，而且许多人也都想学习和掌握汇编语言。因为汇编语言具有高速高效的特点，能够利用计算机所有软、硬件资源，直接控制硬件工作，所以赢得了广大用户的青睐。

“汇编语言程序设计”是高等院校计算机及其应用专业学生必修的核心课程之一，对于培养学生掌握程序设计方法、提高程序设计水平、加强上机实践环节和熟悉程序调试技术具有重要作用。鉴于目前国内高等院校大量采用了IBM PC系列机为教学服务，且在社会的各个领域又得到了广泛应用，我们特编写了这本书。

本书共分十一章，第一章介绍了计算机中的数制与编码系统；第二章介绍了IBM PC机的基本结构；第三章介绍了8088/8086的寻址方式和指令系统；第四章介绍了8088/8086的伪指令、汇编语言程序的风格及上机过程；第五章介绍了汇编语言程序设计的方法与技巧；第六章介绍了高级汇编技术；第七章介绍了输入/输出与中断；第八和第九章介绍了图形显示、音乐及文件管理的基本技术；第十章介绍了模块化程序设计方法；第十一章简要介绍了80386/80286微处理器的性能特点。

本书第一、三、四、五、六、八、九、十一章由杨振生编写，第二、七、十章由单洪编写。

王世明教授认真审阅了全书，提出了宝贵意见和建议。郝炳焜工程师对本书部分章节进行了复核，并提出了修改意见。周以海老师精心绘制了本书的全部插图，在此一并表示感谢。

限于编者水平，难免有不妥之处，敬请读者指正。

编　者

1993年5月于合肥

# 目　　录

# 第一章　计算机的数制与编码系统

**本章要点**：1. 计算机中的数制系统
2. 进制间的转换
3. 数的算术运算和逻辑运算
4. 计算机中的编码系统

**本章目的**：通过本章学习，使读者了解计算机中的数制与编码系统，掌握不同进制间的转换和运算规律。

## 第一节　计算机中的数制系统

在人们的日常生活和工作中，最常用的是十进制数。在计算机中，为了便于存储和计算，普遍采用二进制数，这是根据构成计算机的电子元件只有两种稳定状态这一物理特性而决定的。为了便于阅读和书写，人们还经常使用八进制数或十六进制数来表示二进制数。由于数的进制不同，便形成了不同进制的数制系统。

### 一、　十进制数制系统

十进制数遵循逢十进一的规则。十进制数的基数为10，任何基数的数制系统所需要的符号数目恰等于其基数的值。十进制数的符号数字为0，1，2，3，4，5，6，7，8，9，这十个符号数字构成了十进制数制系统。

十进制数的每位数字都有相应的权与之对应。例如

387.56

小数点左边自右至左各位数字的权依次为$10^0, 10^1, 10^2, \cdots$，小数点右边自左向右各位数字的权依次为$10^{-1}, 10^{-2}, \cdots$。

一个十进制数的值等于它的各位数字与其权的乘积之和。因此，387.56可表示为

$$387.56=3\times10^2+8\times10^1+7\times10^0+5\times10^{-1}+6\times10^{-2}$$

任何一个十进制数X可表示为

$$\begin{aligned}X &= X_nX_{n-1}\cdots X_1X_0.X_{-1}X_{-2}\cdots X_{-m}\\ &= X_n\times10^n+X_{n-1}\times10^{n-1}+\cdots+X_1\times10^1+X_0\times10^0+X_{-1}\times10^{-1}+X_{-2}\times10^{-2}+\cdots+X_{-m}\times10^{-m}\\ &= \sum_{i=-m}^{n}X_i\times10^i\end{aligned}$$

## 二、 二进制数制系统

二进制数遵循逢二进一的规则。二进制数的基数为 2，因此构成二进制数制系统的只有符号数字 0 和 1。

二进制数的每位数字都有相应的权与之对应。例如

1011.101

小数点左边自右向左各位数字的权依次为 $2^0,2^1,2^2,2^3,\cdots$，小数点右边自左向右各位数字的权依次为 $2^{-1},2^{-2},2^{-3},\cdots$。

一个二进制数的值等于它的各位数字与其权的乘积之和。因此，1011.101 可表示为

$$(1011.101)_2=1\times2^3+0\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}=(11.625)_{10}$$

任何一个二进制数 X 可表示为

$$\begin{aligned}(X)_2 &= X_nX_{n-1}\cdots X_1X_0.X_{-1}X_{-2}\cdots X_{-m}\\ &= X_n\times2^n+X_{n-1}\times2^{n-1}+\cdots+X_1\times2^1+X_0\times2^0+X_{-1}\times2^{-1}+X_{-2}\times2^{-2}+\cdots+X_{-m}\times 2^{-m}\\ &= \sum_{i=-m}^{n}X_i\times2^i\end{aligned}$$

n 位二进制数可以表示 $2^n$ 个数。例如 3 位二进制数可以表示 8 个数：

| 二进制数 | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| 对应的十进制数 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

而 4 位二进制数可以表示 16 个数，且它们与十进制数的对应关系如下：

| 二进制数 | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 对应的十进制数 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

依此类推，可知 8 位二进制数可以表示 $2^8$ 或 256 个数，即 0～255。

## 三、 八进制数制系统

八进制数遵循逢八进一的规则。八进制数的基数为 8，构成八进制数制系统的是以下八个符号数字：

0，1，2，3，4，5，6，7

八进制数的每位数字都有相应的权与之对应。例如

$(237.56)_8$

小数点左边自右向左各位数字的权依次为 $8^0,8^1,8^2,\cdots$，小数点右边自左向右各位数字的权依次为 $8^{-1},8^{-2},\cdots$。

一个八进制数的值等于它的各位数字与其权乘积之和。因此，$(237.56)_8$ 可表示为

$$(237.56)_8=2\times8^2+3\times8^1+7\times8^0+5\times8^{-1}+6\times8^{-2}=(159.71875)_{10}$$

任何一个八进制数 X 可表示为

$$\begin{aligned}(X)_8 &= X_nX_{n-1}\cdots X_1X_0.X_{-1}X_{-2}\cdots X_{-m}\\ &= X_n\times8^n+X_{n-1}\times8^{n-1}+\cdots+X_1\times8^1+X_0\times8^0+X_{-1}\times8^{-1}+X_{-2}\times8^{-2}+\cdots+X_{-m}\times 8^{-m}\\ &= \sum_{i=-m}^{n}X_i\times8^i\end{aligned}$$

## 四、 十六进制数制系统

十六进制数遵循逢十六进一的规则。十六进制数的基数为16,构成十六进制数制系统是以下16个符号数字:

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

这十六个符号数字与十进制数和二进制数之间的对应关系如表1-1所示。

**表1-1 二进制、十进制、十六进制数码对照表**

| 十进制 | 十六进制 | 二进制 |
|---|---|---|
| 0 | 0 | 0000 |
| 1 | 1 | 0001 |
| 2 | 2 | 0010 |
| 3 | 3 | 0011 |
| 4 | 4 | 0100 |
| 5 | 5 | 0101 |
| 6 | 6 | 0110 |
| 7 | 7 | 0111 |
| 8 | 8 | 1000 |
| 9 | 9 | 1001 |
| 10 | A | 1010 |
| 11 | B | 1011 |
| 12 | C | 1100 |
| 13 | D | 1101 |
| 14 | E | 1110 |
| 15 | F | 1111 |

十六进制数的每位数字都有相应的权与之对应。例如

3AB.11

小数点左边自右向左各位数字的权依次为$16^0, 16^1, 16^2, \cdots$,小数点右边自左向右各位数字的权依次为$16^{-1}, 16^{-2}, \cdots$。

一个十六进制数的值等于它的各位数字与其权的乘积之和。因此,3AB.11可表示为

$$(3AB.11)_{16}=3\times16^2+A\times16^1+B\times16^0+1\times16^{-1}+1\times16^{-2}=(939.0664)_{10}$$

任何一个十六进制数X可表示为

$$\begin{aligned}(X)_{16}&=X_nX_{n-1}\cdots X_1X_0.X_{-1}X_{-2}\cdots X_{-m}\\&=X_n\times16^n+X_{n-1}\times16^{n-1}+\cdots+X_1\times16^1+X_0\times16^0+X_{-1}\times16^{-1}+X_{-2}\times16^{-2}+\cdots+\\&\quad X_{-m}\times16^{-m}\\&=\sum_{i=-m}^{n}X_i\times16^i\end{aligned}$$

为了避免使用多种进制所引进的混乱,在8086/8088汇编语言中,使用后缀表明数的进制:

后缀B——表示2进制;

后缀H——表示16进制;

后缀Q或O——表示8进制;

后缀 D 或无后缀——表示 10 进制。

例如：

1011.1010B——表示 2 进制数

0B.AH——表示 16 进制数

13.5Q——表示 8 进制数

11.625——表示 10 进制数

它们都表示同一数值——10 进制的 11.625，只是所用的进制不同。

## 第二节　数制间的转换

### 一、　二进制数与十进制数间的相互转换

1.　二进制数转换为十进制数

二进制数各位数字与其所对应的权的乘积之和即为该二进制数所对应的十进制数。例如

$$10101.101B = 1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$$
$$=21.625$$

2.　十进制数转换为二进制数

任意十进制数转换为二进制数可按以下步骤进行：

第一步，先把十进制数的整数部分转换为二进制数 $x_m x_{m-1} \cdots x_0$。转换过程如下：

1)使 $i=0$，十进制数的整数部分 X 为被除数；

2)用 2 去除被除数，余数为 x，商作为新的被除数；

3)令 $x_i=x$；

4)使 i 增 1；

5)检验 2)步得到的商，若不为 0，转至 2)；否则转换结束。

第二步，把十进制数的小数部分 $X^*$ 转换为二进制数 $x_{-1}x_{-2}\cdots x_{-n}$。转换过程如下：

1)使 $i=-1$，$x^*$ 作为被乘数；

2)用 2 乘 $x^*$，其整数部分为 X，取其小数部分作为新的被乘数 $X^*$；

3)取 $x_i=x$；

5)检验 2)所得的新被乘数。若 $X^*$ 不为 0，则转至 2)；否则转换结束。

第三步，把整数部分和小数部分所转换成的二进制数结果合并，即完成其转换任务。

例如，把十进制数 21.625 转换为二进制数。

第一步，先把它的整数部分 21 转换成二进制数。

```
2|______21    (x0=1)
  2|____10    (x1=0)
    2|___5    (x2=1)
      2|_2    (x3=0)
       2|1    (x4=1)
        |0    转换结束
```

所以，21＝10101B

第二步，把小数部分转换成二进制数。

$0.625\times2=1.250$　　　$(x_{-1}=1)$

$0.250\times2=0.500$　　　$(x_{-2}=0)$

$0.500\times2=1.000$　　　$(x_{-3}=1)$

所以，0.625＝0.101B

第三步，把上述两部分合并就是转换后的结果。即

21.625＝10101.101B

## 二、 二进制数与八进制数间的相互转换

### 1. 二进制数转换为八进制数

任意一个二进制数，对于小数点左边自右向左每三位二进制数化为一位八进制数；对于小数点右边自左向右每三位二进制数化为一位八进制数，这样就可完成转换任务。

例如，将二进制数 10110100111.10110111 用八进制数表示。

| 10 | 110 | 100 | 111 | · | 101 | 101 | 110 |
|---|---|---|---|---|---|---|---|
| ← | ← | ← | ← | | → | → | → |
| 2 | 6 | 4 | 7 | | 5 | 5 | 6 |

即 $(10110100111.10110111)_2=(2647.556)_8$

### 2. 八进制数转换为二进制数

任意一个八进制数，对于每位八进制数化为三位二进制数即可完成转换任务。

例如，把八进制数 $(735461.14)_8$ 化为二进制数，即可做以下直接转换：

$(735461.14)_8=(111011101100110001.0011)_2$

## 三、 二进制与十六进制数间的相互转换

### 1. 二进制数转换为十六进制数

任意一个二进制数，对于小数点左方自右向左每四位二进制数化为一位十六进制数；对于小数点右方自左向右每四位二进制数化为一位十六进制数，即可完成转换任务。

例如，把二进制数 10110100111.10110111 化为十六进制数。

| 101 | 1010 | 0111 | · | 1011 | 0111 |
|---|---|---|---|---|---|
| ← | ← | ← | | → | → |
| 5 | A | 7 | | B | 7 |

即 $(10110100111.10110111)_2=(5A7.B7)_{16}$

### 2. 十六进制数转换为二进制数

任意一个十六进制数，对于每位十六进制数化为四位二进制数即转换为二进制数。

例如，把十六进制数 32DF.4BC 化为二进制数。则

$(32DF.4BC)_{16}=(0011001011011111.010010111100)_2$

## 四、 十进制数转换为八进制数和十六进制数

### 1. 十进制数转换为八进制数

任意一个十进制数，对于整数部分 N 采用“除 8 取余法”化为八进制数。即用 8 去除 N，得一整数商和一余数。此余数是八进制数的最低位。用 8 再去除所得的商，又得一整数商和一余数。此余数是八进制数的次低位。这样一直除下去，直到商为 0 为止。对于小数部分 M 采用“乘 8 取整法”。即用 8 去乘 M，得一乘积，其整数部分就是八进制小数的最高位；再用 8 乘上面所得积的小数部分，又得一乘积，其整数部分是八进制小数的次高位，这样一直乘下去，直到乘积的小数部分为 0 止。

例如，将十进制数 477.451171875 化为八进制数。

对整数部分用除 8 取余法：

$477 \div 8 = 59$　　余 5，　即 $x_0 = 5$

$59 \div 8 = 7$　　余 3，　即 $x_1 = 3$

$7 \div 8 = 0$　　余 7，　即 $x_2 = 7$

故有　$(477)_{10} = (735)_8$

对于小数部分采用“乘 8 取整”法：

$0.451171875 \times 8 = 3.609375$，即 $x_{-1} = 3$

$0.609375 \times 8 = 4.875$，　　即 $x_{-2} = 4$

$0.875 \times 8 = 7.0$，　　即 $x_{-3} = 7$

故有　$(0.451171875)_{10} = (0.347))_8$

最后转换结果：

$(477.451171875)_{10} = (735.347)_8$

2.　十进制数转换为十六进制数

转换方法与十进制转换为八进制数完全类似，只是将“除 8 取余法”变为“除 16 取余法”，将“乘 8 取整法”变为“乘 16 取整法”。

例如，将十进制数 477.451171875 化为十六进制数。

对十进制数的整数部分 N 采用“除 16 取余法”。

$477 \div 16 = 29$　　余 13，　即 $x_0 = D$

$29 \div 16 = 1$　　余 13，　即 $x_1 = D$

$1 \div 16 = 0$　　余 1，　即 $x_2 = 1$

即　$(477)_{10} = (1DD)_{16}$

对于十进制数的小数部分采用“乘 16 取整法”。

即　$0.451171875 \times 16 = 7.21875$　　即 $x_{-1} = 7$

$0.21875 \times 16 = 3.5$　　即 $x_{-2} = 3$

$0.5 \times 16 = 8.0$　　即 $x_{-3} = 8$

故有　$(0.451171875)_{10} = (0.738)_{16}$

最后转换结果：

$$(477.451171875)_{10} = (1DD.738)_{16}$$

# 第三节　数的算术运算与逻辑运算

## 一、 数的算术运算

1.　二进制的算术运算

| 加法规则： | 乘法规则： |
| --- | --- |
| $0+0=0$ | $0\times 0=0$ |
| $0+1=1$ | $0\times 1=0$ |
| $1+0=1$ | $1\times 0=0$ |
| $1+1=0$(进位 1) | $1\times 1=1$ |

二进制数减法与十制数减法类似，够减时是直接相减，不够减时服从向高位借 1 为 2 的规则。

例 1-1　二进制数的算术运算

$$
\begin{array}{r}
11001\\
+\,10011\\
\hline
101100
\end{array}
\qquad
\begin{array}{r}
11001\\
-\,10011\\
\hline
00110
\end{array}
\qquad
\begin{array}{r}
11001\\
\times\,10011\\
\hline
11001\\
11001\phantom{0}\\
00000\phantom{00}\\
00000\phantom{000}\\
+\,11001\phantom{0000}\\
\hline
111011011
\end{array}
$$

2.　八进制数的算术运算

八进制数加法：当两个一位数之和 S 小于 8 时，与十进制数同样处理；若两个一位数之和 S≥8 时，则应该用进位 1 和 S−8 取代。

八进制数减法和十进制数减法类似，够减时则直接相减，不够减时服从向高位借 1 为 8 的规则。

八进制数乘法可用十进制数乘法规则来计算，但结果必须用八进制数表示。

**例** 1-2　八进制数的算术运算

$$
\begin{array}{r}
735Q\\
+\,356Q\\
\hline
1313Q
\end{array}
\qquad
\begin{array}{r}
735Q\\
-\,356Q\\
\hline
357Q
\end{array}
\qquad
\begin{array}{r}
735Q\\
\times\,356Q\\
\hline
5456\phantom{Q}\\
4521\phantom{0Q}\\
+\,2627\phantom{00Q}\\
\hline
335566Q
\end{array}
$$

3. 十六进制数的算术运算

十六进制数加法：当两个一位数之和 S 小于 16 时，与十进制数同样处理；当两个一位数之和 S≥16 时，则应该用进位 1 及 S−16 取代。

十六进制数减法也和十进制数减法类似，当够减时，则可直接相减；当不够减时服从向高位借 1 为 16 的规则。

十六进制数乘法可用十进制数的乘法规则，但结果必须用十六进制数表示。

**例** 1-3　十六进制数的算术运算

```
  2D34H          2D34H           2D34H
+05CBH         -05CBH          ×05CBH
  32FFH          2769H         -------
                                 1F13C
                                21E70
                              +E204
                              -------
                              105DC3CH
```

## 二、 数的逻辑运算

1.　“与”运算(AND)

“与”运算又称逻辑乘，运算符为・或∧。“与”运算的规则如下：

| X | Y | X ∧ Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

2.　“或”运算(OR)

“或”运算又称逻辑加，运算符为＋或∨，“或”运算的规则如下：

| X | Y | X ∨ Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

3.　“非”运算(NOT)

“非”运算又称逻辑非，变量 A 的“非”运算记作 $\overline{A}$。“非”运算的规则如下：

| X | $\overline{X}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

4.　“异或”运算(XOR)

“异或”运算的运算符为$\veebar$或⊕，其运算规则如下：

| X | Y | X $\veebar$ Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**例** 1-4　设变量 A＝537Q，B＝256Q，求 $C_1=A\wedge B$，$C_2=A\vee B$，$C_3=\overline{A}$，$C_4=A\veebar B$。

因为　A＝101011111，B＝010101110

故有　$C_1=A\wedge B=000001110=016Q$

$C_2=A\vee B=111111111=777Q$

$C_3=\overline{A}=010100000=240Q$

$C_4=A\veebar B=111110001=761Q$

**例** 1-5　设变量 X＝23CFH，Y＝3545H，试求 $Z_1=X\wedge Y$，$Z_2=X\vee Y$，$Z_3=\overline{X}$，$Z_4=X\veebar Y$。

因为　X＝0010001111001111

Y＝0011010101000101

$Z_1=X\wedge Y=0010000101000101=2145H$

$Z_2=X\vee Y=0011011111001111=37CFH$

$Z_3=\overline{X}=1101110000110000=DC30H$

$Z_4=X\veebar Y=0001011010001010=168AH$

## 第四节　计算机的编码系统

### 一、　符号编码

根据计算机的特定规则，通常用二进制数表示人们常用的字母、数字、标点及其它特殊符号，即所谓“编码”。在微机系统中，最常用的是 ASCII 编码，规定 7 位 2 进制数表示一个符号。其中包括数字符号、字母、特殊符号等可打印的字符，其范围为：

20H～7EH

还有若干个控制代码，其范围为：

0～1FH 及 7FH

全部 ASCII 编码见附录一。

### 二、　有符号数的编码

在计算机内，有符号数的编码称为机器数。机器数所表示的数值称为真值。

在机器中，有符号数有三种表示法——原码、反码和补码。

1.　原码

在有符号数的绝对值前面附加一个符号位就称为原码。在计算机中，正号用 0 表示，负号用 1 表示。设 m＋1 位 2 进制数的绝对值为 $X_m\cdots X_0$，则原码的定义为：

$$[X]_{原}=\begin{cases}0X_m\cdots X_0，若\ X\ 的真值\geqslant 0\\ 1X_m\cdots X_0，若\ X\ 的真值<0\end{cases}$$

例如，$(+37)_{10}$和$(-37)_{10}$在计算机(设机器字长为 16 位)中分别表示为：

$[+37]_{原}=0000000000100101$

$[-37]_{原}=1000000000100101$

2. 反码

一个正数的反码，等于该数的原码。一个负数的反码，等于该负数所对应的正数的原码按位取反，即 0 变 1，1 变 0。

例如，$(+37)_{10}$和$(-37)_{10}$的反码分别为：

$[+37]_{反}=0000000000100101$

$[-37]_{反}=1111111111011010$

3. 补码

一个正数的补码等于该数的原码。一个负数的补码等于该数的反码在末位上加 1。

例如，$(+37)_{10}$和$(-37)_{10}$的补码分别表示为：

$(+37)_{补}=0000000000100101$

$(-37)_{补}=1111111111011011$

补码的加法运算规则：

$[X+Y]_{补}=[X]_{补}+[Y]_{补}$

补码的减法运算规则：

$[X-Y]_{补}=[X]_{补}+[-Y]_{补}$

根据补码的运算规则可以得出以下结论：

(1)任何两个数相加，无论其正负号如何，只要对它们的补码实施加法运算即可得到正确的结果，只不过这结果是补码的形式。

(2)减法求补变加法。若两个数相减，对减数求补，则变成被减数与$[减数]_{补}$相加则可得到正确结果，只不过这结果是补码形式。

n 位 P 进制数 X 的补码定义为：

$[X]_{补}=P^n-X$

因此，n 位二进制 X，则$[X]_{补}=2^n-X$

求补运算具有下列重要性质：

$[X]_{补}\xrightarrow{求补}[-X]_{补}\xrightarrow{求补}[X]_{补}$

**例** 1-6 设 A=25DFH，B=327BH，求 A+B，A−B。

因为 $A=0010010111011111=[A]_{补}$

$B=0011001001111011=[B]_{补}$

$[A]_{补}+[B]_{补}=0101100001011010$

$[A+B]_{补}=[0101100001011010]_{补}=0101100001011010$

故有 $[A+B]_{补}=[A]_{补}+[B]_{补}$

$[A]_{补}=0010010111011111$

$[-B]_{补}=1100110110000101$

$[A-B]_{补}=[A]_{补}+[-B]_{补}=1111001101100100$

$[A-B]_{补}\xrightarrow{求补}[-(A-B)]_{补}=1000110010011100$

故有 $A-B=1000110010011100=-0C9CH$

**例** 1-7 设 X=00DAH，Y=00A8H，求 X−Y.



2

与

因为 X＝0000000011011010，Y＝0000000010101000

$[X-Y]_{补}=[X]_{补}+[-Y]_{补}=0000000011011010+1111111101011000$

$=\boxed{1}\quad 0000000000110010$

即　　$X-Y=32H$

从最高有效位向高位的进位由于机器字长限制自动丢失。

在日常生活中，有许多“补”数的事例，就钟表而言，若现在的标准时间为 8 点正，而有一只表却为 10 点，要拨到 8 点，可以有两种拨法：

(1)10－2＝8　　倒拨；

(2)10＋10＝8　　顺拨。

所以，这里 10 加 10 与 10 减 2 是相同的。当然是有条件的，因为在钟表中

$10+10=12+8$

把 12 称为模，即一个系统的量程或此系统所能表示的最大的数，它自然会丢掉。即

$10-2=10+10 \pmod{12}$

意思是，等号两边同除以 12，它们的余数相同，12 为模。或称(10－2)与(10＋10)对模 12 是同余的，也就是说，(＋10)与(－2)对模 12 互为补数。

一般来说，若数 Z、Y、K 满足下列关系：

$Z=nk+Y$　　（n 为整数）

则称 Z 与 Y 对模 K 是同余的。

在上例中，K＝12，因此，(－2)与(＋10)，(－5)与＋7，(－6)与(＋6)等对于模 12 都是同余的，即是说，它们对模 12 来说互为补数。

由此可以得出结论：对于某一确定的模，某数减去小于模的一个数，总可以用加上该数的负数与其模数之和(即补数)来代替。所以，引进补码以后，减法就可以转换为加法了。

例如，在字长为 8 位的二进制数字系统中，其模为 $2^8=(256)_{10}$。若有

$76-12=76+(-12)=76+(256-12)=76+244=256+64=64$

$$\begin{array}{r} 76 \\ -12 \\ \hline 64 \end{array} \qquad \begin{array}{r} 01001100 \\ -00001100 \\ \hline 01000000 \end{array} \qquad \begin{array}{r} 76 \\ +244 \\ \hline 64 \end{array} \qquad \begin{array}{r} 01001100 \\ +11110100 \\ \hline \boxed{1}\,01000000 \end{array}$$

自然丢失

由此可见，在字长为 8 位(模 $2^8$)的情况下，(76－12)与(76＋244)的结果是相同的。所以，(－12)与 244 互为补数。

由于计算机的字长有一定限制，因此一个带符号数是有一定范围的，在字长为 8 位用补码表示时其范围为：＋127～－128。当运算结果超过这个范围时，称为溢出。

补码表示有符号数时应注意，一般来说，n 位补码表示有符号数的范围是：

$-2^{n-1}\leqslant N\leqslant 2^{n-1}-1$

例如，当 n＝16 时，$-32768\leqslant N\leqslant +32767$

4.　8421 编码—BCD(Binary—Coded Decimel)码

采用 2 进制，易于数据传送、存储和运算。但 2 进制数不直观，所以，还经常使用 2 进制编码的 10 进制数，简称 BCD 数或 BCD 码。