

单板微型机上使用的算术运算程序

黄元洪

[计算机科学(电脑)系]

一、前言

单板微型机的应用近年已普遍起来,它作为学习微型计算机的工具、生产过程控制、智能仪器的部件等等,无疑具有价格低廉,使用方便等优点。但是如果要让它来进行最简单的算术运算就不那么方便了。单板微型机的学习者往往希望能够通过编程,实现算术运算,却又感到相当费事。为此,我们介绍一种用Z80单板微型机指令编制的多字节算术运算程序,以供使用者参考。

二、数据格式

由于单板微型机的显示幕均为六位,所以我们设定运算的数据字长为三个字节,也即十进制六位。数据是带符号的,并可为整数或小数。

在数据的格式上,用一个字节表示数据的符号和小数点,加上数据本身,须使用四个字节的存贮单元才可完整地表示一个数据。其格式如图(1)所示。符号和小数点在存贮单元中

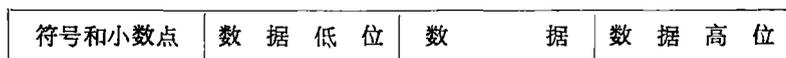


图1

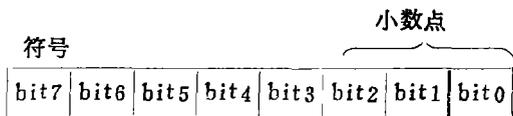


图2

的表示如图(2)所示。其中最高位表示符号,当bit7=0时,表示正数;当bit7=1时,表示负数。bit2~bit0表示小数点,因小数点在这里最大不超过5,所以用3位就可满足。其余几位bit6~bit3与数据无关。

数据在存贮单元中的地址安排如图(3)所示。

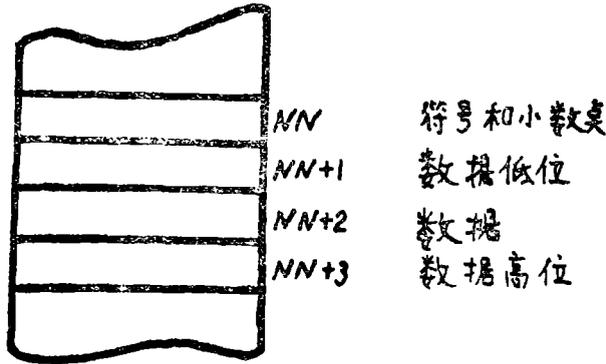


图 8

三、程 序 设 计

在程序设计中，将符号，小数点、数据三者分开进行处理，分别建立它们各自的子程序。加减法中，由于数据带有符号，因此不能简单地按运算符号去进行运算，须由两个操作数的符号和运算符综合加以决定。

下面，就加减法中决定实际运算的情况进行分析。为此，我们将情况列于下面表中。

| 被加(减)数 | 加(减)数 | 运算符 | 实际运算 | 运算结果的符号 |
|--------|-------|-----|------|--|
| 正 | 正 | + | + | 和被加(减)数相同 |
| 正 | 正 | - | - | 当进位C=1时, 和被加(减)数相反 当进位C=0时, 和被加(减)数相同 |
| 正 | 负 | + | - | 当进位C=1时, 和被加(减)数相反 当进位C=0时, 和被加(减)数相同 |
| 正 | 负 | - | + | 和被加(减)数相同 |
| 负 | 正 | + | - | 当进位C=1时, 和被加(减)数相反 当进位C=0时, 和被加(减)数相同 |
| 负 | 负 | + | + | 和被加(减)数相同 |
| 负 | 正 | - | + | 和被加(减)数相同 |
| 负 | 负 | - | - | 当进位C=1时, 和被加(减)数相反 当进位C=0时, 和被加(减)数相同 |

*注：当被减数小于减数时，进位C=1；当被减数等于大于减数时，进位C=0

以x表示被加(减)，y表示加(减)数，z表示运算符，以“0”表示x、y中的正数，以“1”表示x、y中的负数；以“0”表示z为加法运算符，“1”表示z为减法运算符。将它们之间的关系用卡诺图表示如下：

| | | | | |
|---|----|----|----|----|
| | xy | | | |
| z | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

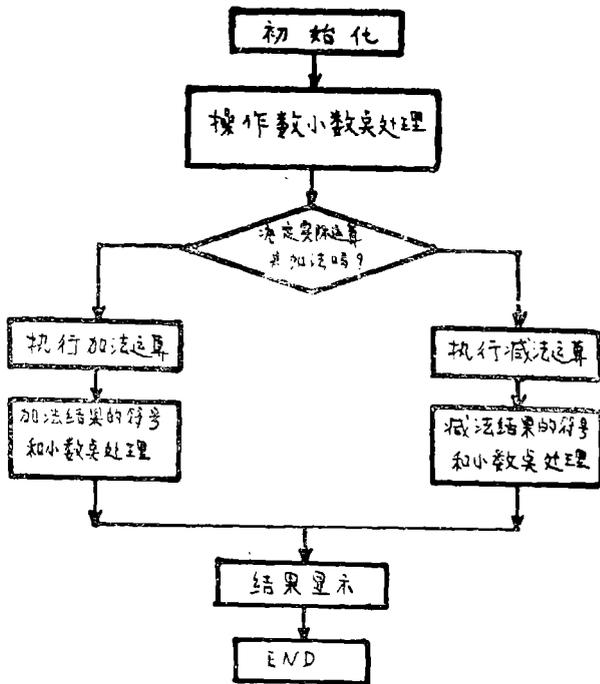
$$\text{实际加法} = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}z + xy\bar{z} = x \oplus y \oplus z$$

$$\text{实际减法} = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}z + xy\bar{z} = x \oplus y \oplus z$$

另外，从表中看出，在实际执行加法运算时，运算结果的符号和x的符号相同，在实际执行减法运算时，当被减数小于减数时，结果的符号和x的符号相反，否则结果的符号与x的符号相同。

下面，将算术运算的逻辑框图分别画出。

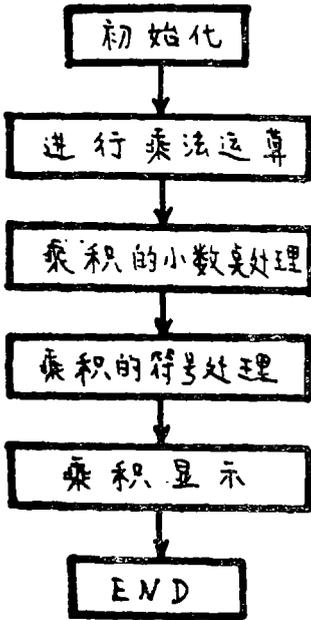
1、加减运算逻辑框图



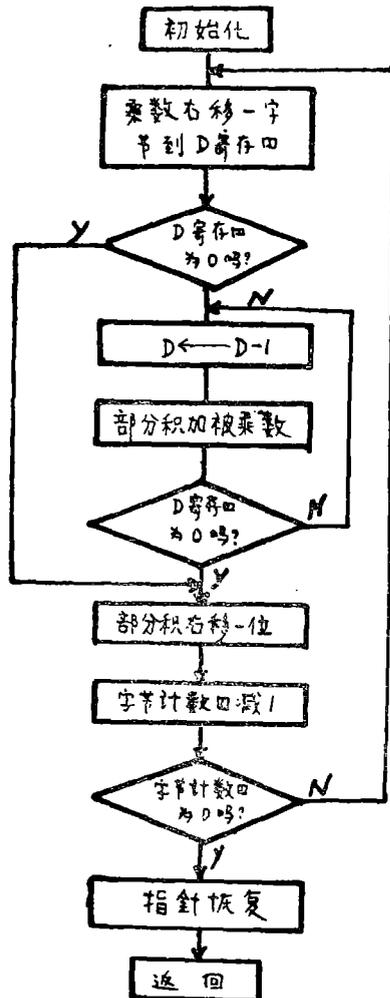
2、乘法运算逻辑框图

说明：乘法运算取双倍精度，最后运算结果仍取3字节字长。

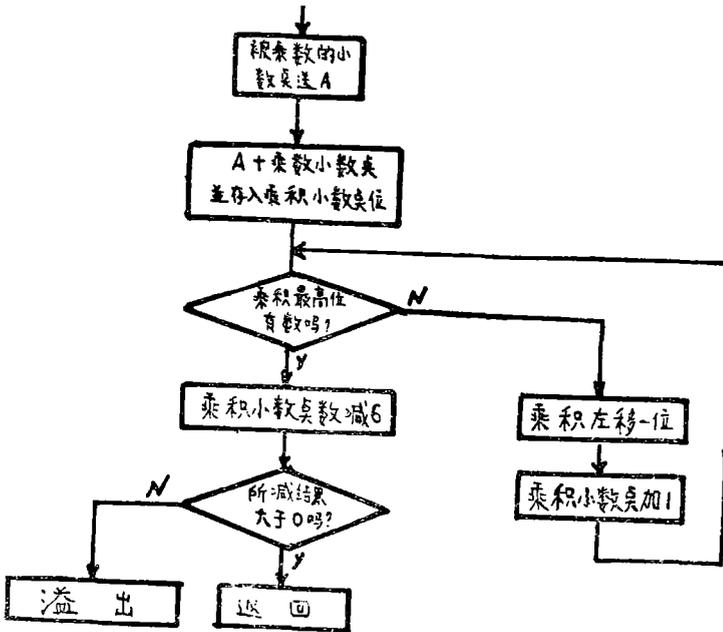
<1> 乘法总框图



<2> 乘法运算框图

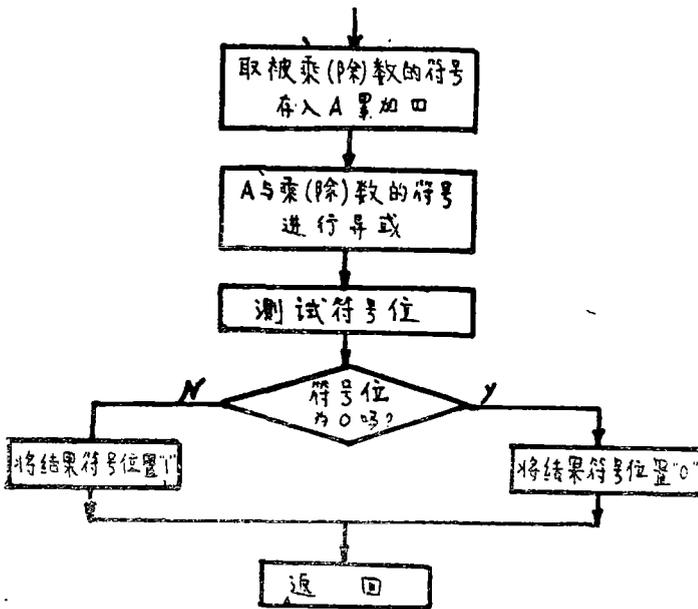


<3> 乘积小数点处理逻辑框图



乘积小数点的处理方法是：先将参加乘法两数的小数点相加，然后看乘积最高位是否有数，若无数，即将乘积左移一位，小数点加1，一直到乘积最高位有数为止。再将小数点数减6，当所减结果大于0，那么这个数即为乘积的小数点；当所减结果小于0时，说明乘积溢出。

<4> 乘积符号处理的逻辑框图

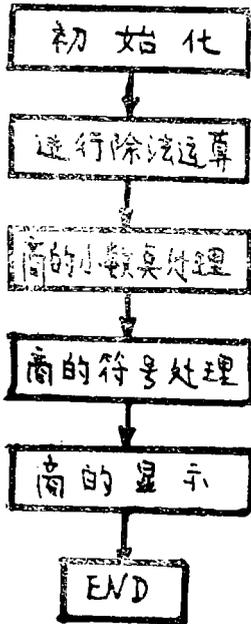


乘积和除法运算中商的符号处理方法相同，也比较简单。将被乘（除）数的符号与乘（除）数的符号进行比较，同号时结果为正，异号时结果为负。

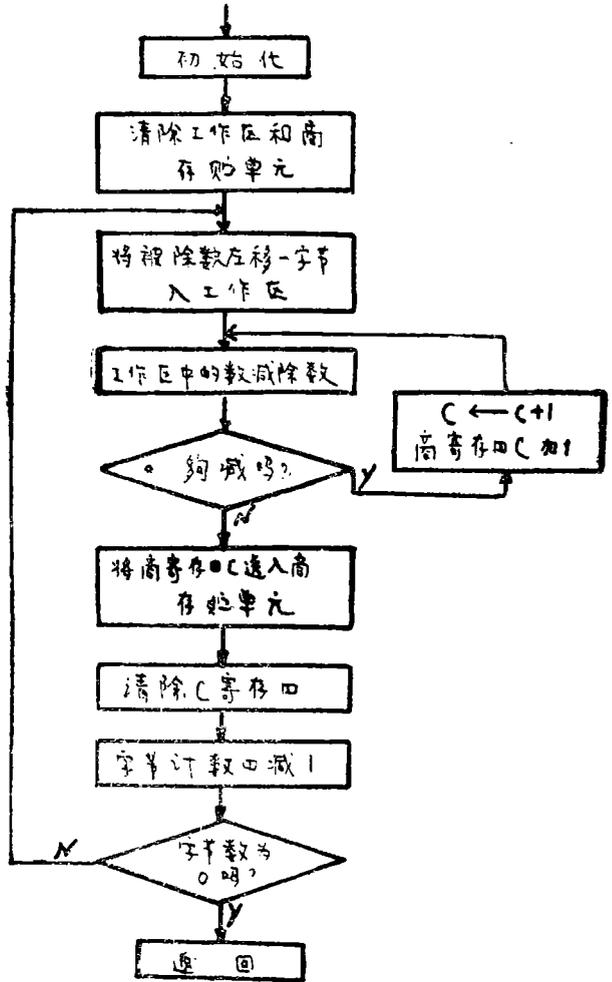
3、除法运算逻辑框图

为了使运算过程中不破坏参加运算的数，另外开辟一个工作区，以DE寄存器对为工作区的地址指针。

<1> 除法总框图



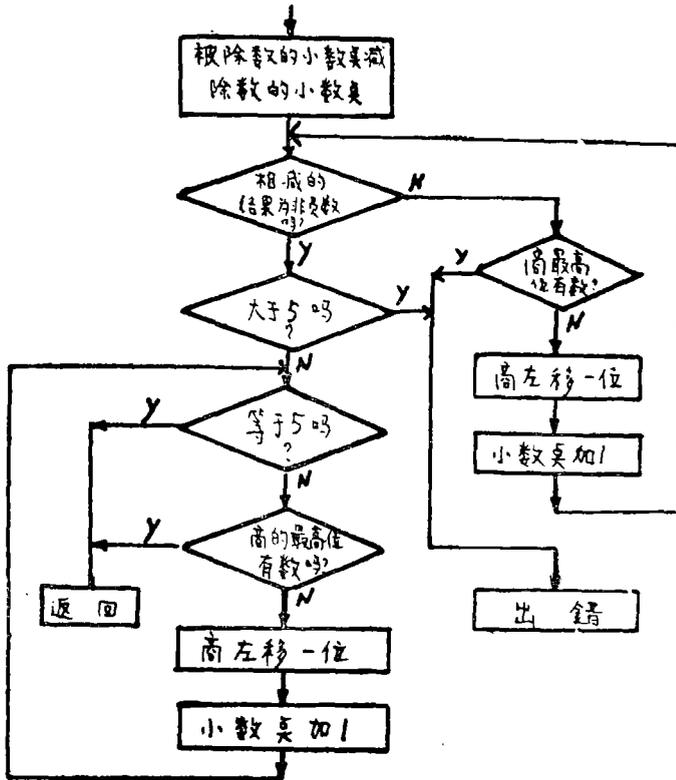
<2> 除法运算框图



<3> 商小数点处理逻辑框图

商小数点处理的方法是：将被除数小数点减去除数的小数点，若相减结果大于等于0，又大于5时则出现错误，若相减结果是 $0 \leq \text{小数点数} \leq 5$ 时，检查商最高位是否有数，有数则结束，无数时则将商左移一位，小数点加1，再检查小数点是否为5，若等于5，则结束，不等5就继续这一工作。如果相减结果小于0，检查商最高位是否有数，如有数，则出现溢出，如无数，则商左移一位，小数点加1，再检查小数点是否大于或等于0，然后按上述情

况继续进行。



〈4〉商的符号处理逻辑框图

商的符号处理与乘积的符号处理方法相同，逻辑框图也相同。

四、程序清单的说明

- 1、下面的程序清单是以“MPF—I” 单板微型机编写，程序的起始地址在 1800H，如果是其它机型，可根据实际加以改动。
- 2、程序中的结果显示部分是调用“MPF—I” 单板微型机的监控所提供的子程序来实现的，若是其它机型，也应根据实际情况加以修改。
- 3、出错程序，在清单中只是列出当出错时，程序转向 1DF7H 单元，使用者可在 1DF7H 单元中放置具体出错处理内容，本文不予列出。
- 4、加法运算的程序起动脉址为 1800H；
减法运算的程序起动脉址为 1807H；
乘法运算的程序起动脉址为 1932H；
除法运算的程序起动脉址为 19E0H。
- 5、IX表示第一操作数的起始地址，

IY表示第二操作数的起始地址,

HL表示运算结果的起始地址,

DE表示工作区的起始地址。

6、操作数一律占用四个字节字长单元,

运算结果为六个字节字长单元,

工作区为9个字节字长单元。

7、程序中赋予下列数值:

NN = IDFDH, MM = IDFBH, PP = IDF9H,

ERROR = IDF7H, CONSTANT = IDF6H。

8、程序运行实例:

首先将程序输入“MPF-I”单板微型机中,然后确定:

IX = ICOOH, IY = ICO4H, HL = ICO8H, DE = IC10H 再输入操作数:

假设两个数为000008及000003,那么它们在内存中的位置为:

1C00 = 00(符号及小数) 1C04 = 00(符号及小数点)

1C01 = 08 1C05 = 03

1C02 = 00 1C06 = 00

1C03 = 00 1C07 = 00

如果要这两个数相加,那么可以使程序从1800单元开始执行,送地址1800后按GO键,就可在显示幕上看到000011的结果。

如果要这两个数相减,可以送地址1807,然后按GO键,就可在显示幕上看到500000,检查结果的符号及小数点存贮单元1C08,则可看到1C08中的内容为05,说明显示的数据为5.0000。注意,在检查1C08单元之前,须先按RS键,其它运算也应这样。

如果要使这两个数相乘,可以送地址1932,然后按GO键,可在显示幕上看到240000,检查1C08单元,其内容为04,说明显示数为24.0000。

如果要使这两个数相除,可以送地址19E0,然后按GO键,可在显示幕上看到266666,检查1C08单元,其内容为05,说明显示数为2.66666。

参 考 文 献

- [1] Z80汇编语言程序设计手册, 烏振生等编译, 清华大学出版社, (1981)。
- [2] 微型计算机(第九集), 上海交通大学, (1979)。
- [3] 《MICROPROFESSOR》使用手册, 华侨大学电脑系微型机教研室, (1982)。