微型计算机原理

WEI XING JI SUAN JI YUAN LI

李伯成 侯伯亨 编

西北电讯工程学院出版社

微型计算机原理

李伯成　侯伯亨　编

西北电讯工程学院出版社

1988

内容简介

本书以Z80微处理器为例，较系统地介绍了微型计算机的基本原理。内容包括：Z80微处理器；Z80指令系统及汇编语言程序设计；Z 80微处理器的时序；存贮器；输入输出技术；输入输出接口芯片；微型计算机的开发系统及几种常见微处理器介绍。

本书为《微处理机应用系统工程设计及其实例》一书的基础篇，可作为一般工程技术人员的自学用书，也可作为高等院校师生的教科书或教学参考书。

微型计算机原理

李伯成　侯伯亨　编

西北电讯工程学院出版社出版发行

西北电讯工程学院印刷厂印刷

新华书店经销

开本 787×1092 1/16　印张 16　字数 387 千字

1988 年 3 月第 1 版　1988 年 3 月第 1 次印刷　印数 1—6 000

ISBN 7-5606-0055-7/TP·0017　　定价：2.70 元

前　言

当前，在我国推广和应用微型计算机技术方兴未艾，许多有志于四化建设的工程技术人员、工人、学生都想尽快掌握微型计算机的基础知识，跟上当代技术发展的步伐，成为本单位的技术革新者。另外，在教学实践中我们深深感到，现有的教材和有关参考书的内容过于庞杂，头绪太多，使读者难于掌握微型计算机技术的基本要点。为此，我们编写了该书，力图使全书重点突出，线条清楚；删掉了次要的枝节，以提高本书的可读性。

本书是《微处理机应用系统工程设计及其实例》一书的基础篇。全书共分九章：第一章介绍计算机发展概况及数制与编码；第二章详述了Z 80微处理器的原理与结构；第三章对Z80的指令系统及汇编语言进行了解释和举例；第四章讨论了Z80微处理器的时序；第五章介绍了存贮器的原理和使用；第六章介绍了微型计算机中所使用的输入输出技术；第七章详述了输入输出接口芯片的功能及正确使用方法；第八章介绍了微型计算机开发系统的组成；第九章介绍了单片机8051和十六位微处理器8086。

本书第一、二、三、四、五章由李伯成执笔编写；第六、七、八、九章由侯伯亨执笔编写；习题由龚补编写。在本书编写过程中，我们参考了几本有关教材，充实了本书的内容。

由于时间仓促，加上编者水平有限，错误及不当之处在所难免，敬请广大读者批评指正。

编　者

1987年4月于西安

目　录

第一章　概　述

§1-1　计算机的发展概况

一、计算机的发展

电子数字计算机是高速度自动进行算术和逻辑运算的电子机器，它的发明和应用标志着人类文明进入了一个新的历史阶段。可以说在人类发展史上，电子数字计算机的发明引起了一场极为深刻的工业革命，其意义超过了蒸汽机发明所产生的影响。

从 1946 年世界上第一台电子计算机问世，到今天已有 40 年的历史了，在这不长的时间里，计算机的发展经历了四代。目前，各国正加紧研制第五代计算机。

从1946年到1959年是电子管计算机时代。那时的计算机是原始的、功能很弱。从1959年到1965年是晶体管计算机时代，这一代计算机以半导体晶体管为主要元件，其性能比第一代计算机大为提高。从1965年到七十年代初，数字集成电路的出现使计算机再次出现重大进步，产生了以中、小规模集成电路为基础，配置更完善软件的第三代计算机。七十年代以来，随着大规模、超大规模集成电路的诞生，电子计算机更是突飞猛进地向前发展，形成了第四代计算机。

第四代计算机发展的一个显著趋向就是向两极发展。一方面是研制运算速度极高、功能极强的大型和巨型计算机，以适应军事及尖端科学的需要；另一方面是研制价格低廉的微型计算机。

二、微型计算机的发展

微型计算机属于第四代计算机。从1971年 INTEL 公司制成第一个微处理器4004开始，在短短的十几年时间里，微处理机如雨后春笋大量地涌现出来，开发出了四代产品。

第一代产品出现在 1971 年到 1973 年间，以 INTEL 公司的 8008 为代表，采用 PMOS 工艺，基本指令执行时间为 20 μs 到 50 μs，基本指令有48条，时钟频率在 500 kHz 以下，集成度为 2300 元件/片。

此后，在 1973 年到 1975 年，又出现了多种微处理器，例如 MOTOROLA 的 6800，INTEL的8080等等。基本指令的执行时间比第一代产品提高十倍左右，达到 2 μs 到10 μs，时钟频率大于 1 MHz，基本指令有七十多条，性能也大大优于第一代产品，这就是第二代微处理机。以上两代产品都是 8 位微处理机。

在 1975 年到 1977 年间出现了集成度更高、性能更好的微处理器。ZILOG 公司的Z80、INTEL 公司的 8085以及各种单片微型计算机和位片式微处理器，它们被认为是第三代微处理机。

1978 年以后，各厂家相继生产出了第四代微处理机。其代表产品是 8086， M68000 和 Z8000 。它们都是16位的微处理器，具有多种灵活的寻址方式和强有力的指令系统，运算速度提高很大，直接寻址的内存空间大幅度扩充。第四代微处理机的出现直接冲击小型计算机

市场，而且在性能上已经赶上甚至超过一般小型计算机。

随着超大规模集成电路工艺的发展，八十年代初，集成度已达到10万元件/片，而且32位处理机已投放市场。在提高硬件性能的同时，各种新的功能更强的软件系统不断研制出来，使整个微型计算机日新月异。现在，正向着网络化、智能化的方向发展。

三、微处理机的特点

建立在微细加工工艺基础上的微处理机有许多优点，正是由于它的这些优点，使它从问世以来就得到极其迅速的发展和广泛的应用。

1. 功能强

微处理机的设计，参考并集中了其它类型计算机的一些优点。与别的电子设备比较，它运算速度快、计算精度高，具有记忆和逻辑判断能力，而且，每种微处理机都配有一整套支持软件。硬件和软件的配合，相辅相成，使微处理机的功能大为增强，在各行各业中得到广泛应用。

2. 可靠性高

由于微处理器及其系列芯片都是采用微细加工工艺进行制造，在一块芯片上做出几千、几万甚至更多的元件，这就减少了大量的焊点、连线、接插件等不可靠因素，使可靠性大大增加。据认为集成度增加100倍，可靠性也增加100倍。目前，微处理器及其系列芯片的平均无故障时间可达 $10^7 \sim 10^8$ 小时。

3. 价格低

微处理器及其系列芯片采用集成电路工艺，集成度高，适合工厂大批量生产，因此，产品的造价十分低廉。据认为集成度增加100倍，价格可降为同功能分立元件的百分之一。很显然，低的价格对于微处理机的推广和普及是极为有利的。

4. 适应性强

在微处理机中可以使用ROM存贮系统的监控程序和用户程序，只要更换存贮不同程序的ROM，在不改变系统硬件或只部分地改变某些硬件，就能适应不同应用任务的要求。

同时，由于微处理机具有强的功能，在适当的硬件和软件支持下，使微处理机既能适应各种工程上的应用又能适应各种事务性管理等方面的应用。

5. 周期短、见效快

微处理器制造厂家生产各种支持芯片，同时也提供许多完美的支持软件，这就为我们构成一个微处理机应用系统创造了十分有利的条件，从而节省研制时间，缩短研制周期，使研制的系统很快投入运行，取得明显的经济效益。

6. 体积小、重量轻、耗电省

微处理器及其系列支持芯片的尺寸均比较小，最大不过几百平方毫米，而且它们大都以MOS工艺制成，耗电很少，目前，还有CMOS系列产品，其功耗就更低。这对那些对体积、重量、功耗要求比较严格的使用者来说，是具有重要意义的。一些在过去用小型计算机无法实现的应用领域，例如，要求这些方面特别严格的航空、航天等部门中的某些应用，利用微处理机就可以实现。

7. 维护方便

现在的微处理机产品逐渐趋于标准化、模块化、系列化、智能化，从硬件结构到软件配

置均考虑了这些问题。一方面是通过自检诊断、在线检测及其它现代化手段，及时发现系统故障；另一方面是发现故障很容易排除，例如迅速更换标准化模板或芯片。微处理机应用的发展，使人们开始研究和使用单机及多微处理机的容错系统。这对提高系统的可靠性，增加系统的可维护性都是十分有利的。

总之，由于微处理机具备这样一些优点，从一开始出现就受到人们的重视，在科学研究、工业、农业、国防及社会生活的各领域中得到越来越广泛的应用。而且微处理机普及应用的势头正在进一步扩大，可以预料，在今后微处理机必将渗透到所有的行业和部门并将进入家庭，其发展前景不可限量。

§1-2 数制与码制

本节作为预备知识介绍书中用到的数制及信息编码问题。

一、十进制记数法

十进制记数法是使用最广、人们最熟悉的一种记数方法。

在十进制记数中，用 0,1,2,…,9 这十个符号来表示数量，无论多大的数，都是用这十个符号的组合来表示。

在这种记数法中采用位值法则，即对每个数位赋予一定的位值，又称权。不同位的权是不一样的，上一位的权是下一位的十倍，通常我们所说的个位、十位、百位、千位等等就是说的权。个位的权是 10^0、十位的权是10^1、百位的权是 10^2，依次类推。利用这种法则，就可以表示任意大小的数。例如，十进制数 3758 可用上面的法则来表示：

$$(3758)_{10}=3\times10^3+7\times10^2+5\times10^1+8\times10^0$$

式中的 10^3、10^2、10^1、10^0 分别表示各位数的权。可见，表示一个数的值可用每位数乘以该位的权而后相加得到。

根据同样的法则，也可以表示十进制小数，小数点的右边各位的权为10^{-1},10^{-2},10^{-3},…。例如，十进制数 275.368 可以用上述法则写成：

$$(275.368)_{10}=2\times10^2+7\times10^1+5\times10^0+3\times10^{-1}+6\times10^{-2}+8\times10^{-3}$$

二、二进制记数法

二进制记数法用来表示数量的符号只有两个，就是 0 和 1。二进制数中的任何一个 0 或 1称为比特(bit)。

与十进制记数法类似，一个二进制数可利用位值记数法来表示，每一位具有不同的权，权的大小是以 2 的幂来表示。例如，二进制数110101可以表示为：

$$(110101)_2=1\times2^5+1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0$$

从上式中可以看到，每一位的位置不同，它们的权也不同，从最低位开始，权分别是2^0，2^1，2^2，2^3，……。

同理，上述方法能够用来表示二进制小数。在表示中所不同的是从小数点向右，每位的权分别是 2^{-1}，2^{-2}，2^{-3}，……。例如，二进制小数 0.101101 可以表示为：

$$(0.101101)_2=1\times2^{-1}+0\times2^{-2}+1\times2^{-3}+1\times2^{-4}+0\times2^{-5}+1\times2^{-6}$$

三、二进制数与十进制数的相互转换

在微处理机的学习和应用中，经常要用到二进制数和十进制数的相互转换问题，读者应能熟练地完成它们之间的相互转换。

1. 二进制数转换成十进制数

如上所述，只要将二进制数的每一位乘上它的权然后加起来就可以求得二进制数的十进制数值。例如，二进制数 101101.11 换算成十进制数为：

$$
\begin{aligned}
(101101.11)_2 = {} & 1\times 2^5+0\times 2^4+1\times 2^3+1\times 2^2+0\times 2^1 \\
& +1\times 2^0+1\times 2^{-1}+1\times 2^{-2}=(45.75)_{10}
\end{aligned}
$$

2. 十进制数转换成二进制数

十进制数转换为二进制数的方法分两步来说明，首先说明十进制整数的转换方法。

十进制整数转换为二进制整数的法则就是“除 2 取余”。即对十进制整数连续除以 2，每次相除所得的余数就构成了所要转换的二进制数，而每次所得的整数商继续被 2 除，直到商为零为止，最后所得的余数就是所转换的二进制数的最高位。例如，欲将十进制数 175 转换为二进制数，其过程如下：

$175\div 2=87$	余数为 1
$87\div 2=43$	1
$43\div 2=21$	1
$21\div 2=10$	1
$10\div 2=5$	0
$5\div 2=2$	1
$2\div 2=1$	0
$1\div 2=0$	1

得到结果：$(175)_{10}=(10101111)_2$

现在，再来说明十进制小数的转换过程，十进制小数转换为二进制小数的法则叫做“乘二取整”。就是将十进制小数连续乘 2，每乘一次取出乘积的整数部分上的 0 或 1，并将小数部分继续乘 2，直到相乘结果的小数部分为零或达到一定的精度要求为止，这时所取出的整数就构成了要转换的二进制小数。开始取出的整数为二进制的高位，最后取得的整数为二进制小数的最低位。例如，将十进制小数 0.71875 转换成二进制小数，其过程如下：

$0.71875\times 2=1.4375$	整数部分 1
$0.4375\times 2=0.875$	0
$0.875\times 2=1.75$	1
$0.75\times 2=1.5$	1
$0.5\times 2=1.0$	1

于是，得到结果为：$(0.71875)_{10}=(0.10111)_2$

综上所述，一个十进制整数的二进制转换方法就是“除 2 取余”；而一个十进制小数的二进制转换方法就是“乘 2 取整”。若一个十进制数既包含整数部分又包含小数部分，它的二进制转换就是将它的整数部分和小数部分用上述方法分别进行转换，最后将转换好的两部分结

合在一起形成要转换的二进制数。例如，

$$(175.71875)_{10}=(10101111.10111)_2$$

四、八进制记数法

八进制记数法采用 0,1,2,……,7 这样 8 个符号来表示数量。同前面所说的十进制、二进制记数法一样，八进制数的不同位具有不同的权，权是用 8 的幂来表示的。例如，八进制数 372.01，根据各位的权不同可以写成：

$$(372.01)_8=3\times8^2+7\times8^1+2\times8^0+0\times8^{-1}+1\times8^{-2}$$

将上式中各位与其权相乘而后加到一起，就可以得到八进制数 372.01 的十进制数为：

$$(372.01)_8=(250.015625)_{10}$$

这也表明了八进制数转换为十进制数的过程。

十进制数转换为八进制数的方法是：对于十进制整数采用“除 8 取余”的方法转换为八进制整数；对于十进制小数则采用“乘 8 取整”的方法转换为八进制小数。例如，十进制数 194.46875 转换成八进制数应将整数部分和小数部分分别转换，最后再合到一起就得到要转换的八进制数。

$194\div8=24$	余数为 2		$0.46875\times8=3.75$	整数部分 3
$24\div8=3$	0		$0.75\quad\times8=6.0$	6
$3\div8=0$	3			

所以，可以得到：

$$(194.46875)_{10}=(302.36)_8$$

二进制数转换成八进制数的方法就是从小数点起，把二进制数每三位分成一组，然后写出每一组的等值八进制数，顺序排列起来就得到所要求的八进制数。例如，将二进制数 11101111010.1011 转换为八进制数。

$$(011\ 101\ 111\ 010.101\ 100)_2$$
$$(3\quad5\quad7\quad2\ .\ 5\quad4)_8$$

在转换过程中，为了构成三位一组，允许在最高位和最低位上补零。

依据同样的思想，即一位八进制数可用三位二进制数表示，就可以直接将八进制数转换成二进制数。例如，将八进制数 712.46 转换为二进制数，其过程如下：

$$(7\quad1\quad2\ .\ 4\quad6)_8$$
$$(111\ 001\ 010.100\ 110)_2$$

五、十六进制记数法

十六进制记数法是微处理机中最常用的一种数制，顾名思义，十六进制记数法采用0,1,2，3……，9，A，B，C、D，E，F 这十六个符号来表示数量。同样，一个十六进制数的每一位都有自己的权，权是由 16 的幂来表示的。这样，一个十六进制数就能够用各位与它们相应的权来表示。例如。十六进制数 E5 D7 . A3 可以表示为：

$$(E5D7.A3)_{16}=E\times16^3+5\times16^2+D\times16^1+7\times16^0$$
$$+A\times16^{-1}+3\times16^{-2}$$

同前所述，一个十进制数可以转换成十六进制数。其方法就是十进制的整数部分采用“除

16取余”的方法得到十六进制数的整数部分，而十进制的小数部分则采用“乘16取整”的方法获得十六进制数的小数部分。例如，将十进制数 47632.78125 转换成十六进制数，其转换过程为：

整数部分：

$47632 \div 16 = 2977$	余数 0 ⟶	16 进制数 0
$2977 \div 16 = 186$	1 ⟶	1
$186 \div 16 = 11$	10 ⟶	A
$11 \div 16 = 0$	11 ⟶	B

小数部分：

$0.78125 \times 16 = 12.5$	整数 12 ⟶	C
$0.5 \times 16 = 8.0$	8 ⟶	8

最后得到$(47632.78125)_{10} = (BA10 \cdot C8)_{16}$

十六进制数转换为十进制数，由前面的叙述可以很方便地实现，只要将十六进制数各位与它们对应的权相乘，再加到一起就可得到。

由于一位十六进制数可以用四位二进制数来表示，因此二进制数与十六进制数的相互转换就比较容易。二进制数到十六进制数的转换是由小数点开始，每四位二进制数为一组，将每一组用相应的一位十六进制数来表示，即可得到正确的十六进制数。例如：

$$(1\ 1101\ 0100\ 1011\ 0111 \cdot 0101\ 1110\ 1010)_2$$

$$(1\quad D\quad 4\quad B\quad 7\ \cdot\ 5\quad E\quad A)_{16}$$

相反，将十六进制数转换成二进制数，只要将十六进制数的每一位用其等值的四位二进制数代替，连在一起就得到了我们所需要的二进制数。例如，十六进制数 E 7 FF 转换成二进制数的过程如下：

$$(E\,7FF)_{16} = (1110\ 0111\ 1111\ 1111)_2$$

$$(E\quad 7\quad F\quad F)_{16}$$

到现在为止，我们介绍了几种常用的数制以及它们之间的相互转换，有一些转换做起来比较容易，也有一些比较麻烦。例如十六进制数与八进制数的转换以及它们与十进制数的相互转换等。遇到这种情况，可以直接进行转换，也可以通过二进制数进行中间转换。总之，希望读者能比较熟炼地掌握这些数制，对学好本书是有益的。

六、BCD 码

转换十进制数为其等值的二进制数称之为编码，前面所提到的二进制数称为纯二进制码。微处理器只能识别用高低电平表示的 0 或 1，就其工作来说，纯二进制码用于微处理机是十分方便的，这一点后面还将进一步说明。但就人们的习惯来说，对十进制数更熟悉，用起来也很直观，遗憾的是微处理机却无法直接进行操作。为此，人们发明一种特殊的二进制编码，它兼有二进制和十进制记数的特点，既符合人们的习惯，计算机又能直接进行运算。人们将这种编码叫做二-十进制码，简称 BCD 码。

用二进制编码来表示十进制数的方案有多种，都是用四位二进制编码来表示一位十进制数。经常使用的一种编码方案是所谓的 8421 码，其编码原则是每位十进制数用四位等值的二进制数来表示，从左至右各位二进制数的权为 8421，故名 8421 码。表 1-1 分别列出了十

进制数、纯二进制码和 8421 BCD 码的对应关系。

从表 1-1 中可以看到，BCD 码仅仅利用了四位二进制编码的十种组合，而代表十进制数 10～15 的二进制编码，对于 BCD 码来说是非法的。很明显，BCD 码只利用了二进制中对应 0～9 的十种码组，而且只用这十种就已足够，剩下的六种二进制码组是不允许使用的。

表 1-1　BCD 码与其它数制的对应关系

十进制数	8421 BCD 码	纯二进制码
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 0 0
5	0 1 0 1	0 1 0 1
6	0 1 1 0	0 1 1 0
7	0 1 1 1	0 1 1 1
8	1 0 0 0	1 0 0 0
9	1 0 0 1	1 0 0 1
10	0 0 0 1 0 0 0 0	1 0 1 0
11	0 0 0 1 0 0 0 1	1 0 1 1
12	0 0 0 1 0 0 1 0	1 1 0 0
13	0 0 0 1 0 0 1 1	1 1 0 1
14	0 0 0 1 0 1 0 0	1 1 1 0
15	0 0 0 1 0 1 0 1	1 1 1 1

根据上述说明，一个十进制数，能够很方便地用 BCD 码来表示。例如，十进制数 834用 BCD 码表示为：

$$(834)_{10} = (1000\ 0011\ 0100)_{BCD}$$

为了避免与二进制编码混淆，在 BCD 码表示中每位 BCD 码(四位二进制数)写成一组，中间留有空隙，而且要标明此数为 BCD 数，如上例中所写的那样。

只要熟记十进制数 0～9 与 BCD 码的对应关系，则它们之间的相互转换是十分方便的。例如：

$$(0110\ 1001\ 0101.0010\ 0111\ 1001)_{BCD} = (695.279)_{10}$$

就是将每一位 BCD 码所表示的十进制数直接写出来，就得到相应的十进制数。反之，将十进制数用其相应的 BCD 码代替，就可获得相应的 BCD 码。

二进制数与BCD码的相互转换要略微麻烦一点，一般要通过一次向十进制转换的中间步骤，再由十进制数转换成所要求的编码。例如，要将二进制数 1011.01 转换成 BCD 码，则首先将它转换成十进制数 11.25，而后再将此十进制数转换成 BCD 码，即 $(0001\ 0001.0010\ 0101)_{BCD}$。

同样，由 BCD 码转换成二进制数亦然。

BCD 码在计算机中常用两种表示方法，一种是如上所述，用四位二进制编码来表示一位十进制数，有时称这种表示为压缩十进制编码或压缩 BCD。这种表示法，用八位二进制数即

可以表示二位十进制数，例如，$(78)_{10}=(0111\ 1000)_{BCD}$。另一种称为扩展 BCD，是利用八位二进制数表示一位十进制数。例如，$(78)_{10}=(00000111\ 00001000)_{BCD}$

前一种方法利用率高，占用内存少；但后一种方法，在某些场合下又比较方便。

七、ASCII 码

ASCII 码是美国标准信息交换码的简称，现在为各国所广泛采用。

通常，ASCII码由 7 位二进制编码来表示，用于微处理机与它的外部设备之间进行数据交换以及通过无线或有线进行数据传送。

7 位二进制编码共有 128 种不同的组合，用以表示 128 种不同的字符或功能。它们是十进制数字 0～9，大写和小写的 26 个英文字母，标点符号，一些特殊符号及由两三个大写英文字母组成的特殊控制功能字符。详细情况见附录 A。

代表上述字符或控制功能的 ASCII 码是由一个四位组和一个三位组构成，形成七位二进制编码，其格式为：

$$\underbrace{6\ \ 5\ \ 4}_{\text{3 位组}}\ \ \overbrace{3\ \ 2\ \ 1\ \ 0}^{\text{4 位组}}$$

根据 ASCII 码的构成格式，可以很方便地从附录 A 中 ASCII 表查出每一个字符或特殊控制功能的编码。例如，大写英文字母 A，从表中查出其 3 位组为$(100)_2$，4 位组为 $(0001)_2$，故构成字母 A 的ASCII 编码为$(1000001)_2$或$(41)_{16}$。

除了上述的 7 位 ASCII 码外，还有简单的 6 位 ASCII 码。这种情况下，ASCII 码的构成格式是一个 2 位组和一个 4 位组构成。即低 4 位构成 4 位组，上述 3 位组变为 2 位组。

由于六位 ASCII 码仅有64 种组合，因此，它只能表示七位 ASCII 表中所示的第 2、3、4、5 列字符。在一些简单应用的场合下，可以使用六位 ASCII 码。

有时，在七位 ASCII 码的基础上附加上一位放在最高位的左边，形成八位 ASCII 码，而新附加上的这一位是八位中的最高位。附加这一位常用于奇偶校验，用来表征数据传送过程中是否有一位出现了错误。

偶校验的含意就是包括这一位在内八位二进制码中为 1 的位数之和为偶数。例如，字母 A 的 ASCII码为$(1000001)_2$ 变为八位 ASCII 表示时，使八位中为 1 的各位之和为偶数，故偶校验位应为 0，而形成具有偶校验位的 A 的 ASCII 码就变为$(01000001)_2$。

奇校验的含义就是包括校验位在内，所有为 1 的位数之和为奇数。根据此原则，上面所提到的字母 A 若用奇校验 ASCII 码表示就为$(11000001)_2$

八、二进制算术运算

1. 二进制加法

二进制加法与十进制加法相类似，所不同的是在二进制加法中是“逢二进一”，其法则为：

$$0+0=0$$

$$1+0=1$$

$$0+1=1$$

$$1+1=0 \text{ 并进位}$$

例如，两个二进制数相加：

$$\begin{array}{r} 10110101 \\ +\ 10001110 \\ \hline 101000011 \end{array}$$

2. 二进制减法

在二进制减法中，同样有如下法则：

$$0-0=0$$

$$1-0=1$$

$$1-1=0$$

$$0-1=1 \text{ 有借位}$$

当不够减时需要借位，高位的一等于下一位的二，即"借一当二"。例如，两个二进制数相减：

$$\begin{array}{r} 10110100 \\ -\ 01010111 \\ \hline 01011101 \end{array}$$

3. 二进制乘法

二进制乘法与十进制乘法是一样的。但因为二进制数只有0和1构成，因此，二进制乘法更简单。其法则如下：

$$0\times0=0$$

$$1\times0=0$$

$$0\times1=0$$

$$1\times1=1$$

例如，二进制数1101.1与101.1相乘之积

$$\begin{array}{r} 1101.1 \\ \times\ 101.1 \\ \hline 11011 \\ 11011 \\ 00000 \\ +\ 11011 \\ \hline 1001010.01 \end{array} \qquad \begin{array}{l} \text{被乘数}(13.5)_{10} \\ \text{乘数}\ \ (5.5)_{10} \\ \\ \\ \\ \\ \text{乘积}\ \ (74.25)_{10} \end{array}$$

再看下面的例子：$(1011)_2\times(1101)_2$

$$\begin{array}{r} 1011 \\ \times\ 1101 \\ \hline 1011 \\ 0000 \\ 1011 \\ +\ 1011 \\ \hline 10001111 \end{array} \qquad \begin{array}{l} \text{被乘数}(11)_{10} \\ \text{乘数}\ \ (13)_{10} \\ \\ \\ \\ \\ \text{乘积}\ \ (143)_{10} \end{array}$$

从上述乘法运算中我们可以看到，从乘数的最低位起，凡遇到1，就相当于在最终结果上加上一个被乘数，而遇到零则不加。但必须注意相加的位置，第0位乘数是1，被乘数直接加在结果的最右面；而第1位，应左移一位后再加；第2位，左移两位后再加；……，以此类推。然后将这些经过移位的和没有移位的被乘数加到一起，就可得到两数乘积。这一概念在

今后实现二进制数在微处理机中的乘法运算时将会用到。

4．二进制除法

二进制除法是乘法的逆运算，其方法与十进制除法是一样的，而且二进制数只有0,1构成，做起来更简单。例如，求二进制数100111除以110的商：

```
          1 1 0.1
    ---------------
110 ) 1 0 0 1 1 1
        1 1 0
    ---------------
          1 1 1
          1 1 0
    ---------------
              1 1 0
              1 1 0
    ---------------
                  0
```

二进制除法也是采用试商的方法求出商数，将此商数与除数相乘，并从被除数中减去此乘积，得到的余数加上被除数的后面一位再试，直到求得最后的结果，其过程与十进制除法相同。

九、符号数的表示及其运算

前面讨论的是不带符号的二进制数，但微处理机中经常要处理带符号的二进制数。为此我们必须知道微处理机中带符号的数是如何表示的，它们的运算规则又是怎样的。

1．符号数的表示方法

表示一个带符号的二进制数有三种方法。

(1) 原码法。在这种表示法中，一个带符号的二进制数由数的符号(正或负)和数的值构成，而数的符号和数的值均由二进制的0和1来表示，一般用0表示正而用1表示负，数的值由多位二进制数表示。这样一来，符号和数值在外表上就没有什么区别。为了避免混淆，在符号数的表示中还必须明确规定符号的位置。大多数计算机中，包括我们将要讨论的微处理机，都用二进制数的最高位表示符号。例如，8位二进制符号数$(+45)_{10}$和$(-45)_{10}$，可以如下写出：

$$(+45)_{10}=(\underset{\substack{\uparrow\\ \text{符号位}}}{0}\ \underbrace{0101101}_{\text{数值}})_2$$

$$(-45)_{10}=(\underset{\substack{\uparrow\\ \text{符号位}}}{1}\ \underbrace{0101101}_{\text{数值}})_2$$

可以看出，用原码来表示一个符号数就是由符号和数值凑到一起来实现，这种表示正负数的方法很好理解，但计算机在实现这种符号数的运算时却很麻烦。因此，这种表示方法在微处理机问世前就不为人们所用了。

(2) 反码法。在计算机的早期，曾采用反码法来表示带符号的数，其规则如下：

表示正数与原码法相同。例如：

$$(+45)_{10}=(00101101)_2$$

也就是正数用符号加上数值凑到一起来表示。

表示负数时用相应正数的原码各位取反表示，包括将符号位取反。取反的含义就是将0