微型计算机原理、接口及应用

吕 杨 刘瑞敏 邹金慧 蔡光程 编著

重庆大学出版社

微型计算机原理、接口及应用

吕　杨　刘瑞敏　邹金慧　蔡光程　编著

重庆大学出版社

内容简介

本书以8086/8088机型为主导机型，详细介绍了16位微型计算机的基本组成：CPU，存储器，I/O系统及I/O接口的硬件结构与工作原理；基于8086/8088指令系统的汇编程序设计及在工业控制领域中的应用实例。在此基础上，较为系统地介绍了80386微机的特点、原理；并介绍了MCS—51单片机的基本结构、原理。章末附有习题和思考题。

本书可作为高等学校工科非计算机专业的本科教材，也可作为微机学习者的自学书籍及从事微机应用开发工作的科技人员的参考书。

图书在版编目(CIP)数据

微型计算机原理、接口及应用/吕杨等编著.—重庆：重庆大学出版社，2000.1

ISBN 7-5624-2098-X

Ⅰ.微…　Ⅱ.吕…　Ⅲ.微型计算机—基本知识　Ⅳ.TP36

中国版本图书馆CIP数据核字(1999)第56499号

微型计算机原理、接口及应用

吕　杨　刘瑞敏　邹金慧　蔡光程　编著

责任编辑　曾令维

*

重庆大学出版社出版发行

新　华　书　店　经　销

重庆电力印刷厂印刷

*

开本：787×1092　1/16　印张：28.75　字数：718千

2000年1月第1版　　2000年1月第1次印刷

印数：1—3000

ISBN　7-5624-2098-X/TP·244　定价：30.00元

前　言

20世纪后半叶，科学技术以前所未有的速度迅猛发展，人类将进入高科技时代。其中一个十分重要的特征就是计算机技术的突飞猛进和计算机应用领域的迅速扩展。计算机技术深入到经济、文化、科学、教育、管理、军事等各个领域，促进了整个社会的重大变革。在21世纪，计算机技术将成为人类生存的基本技能。因此，作为培养未来的建设人才的高等学校，必须把计算机技术作为新型大学生知识结构的重要组成部分，认真地、扎实地、迅速地提高学生的计算机实际操作能力和应用水平。

计算机技术是一门新的学科，计算机的设备日新月异；计算机的性能不断提高；计算机的应用领域迅速扩展；计算机软件的不断更新。这些都要求计算机教材不断更改、创新，以适应变化了的形势。为了适应计算机技术迅速发展的形势，为了反映计算机在各工业领域中的应用和将来的发展前景，为了尽快地提高工科院校非计算机专业学生的计算机操作技能和应用水平，我们编写了这本适用于工科院校非计算机专业的教材。

全书共分八章。本书以8086/8088机型为主导机型，详细介绍了微型计算机的结构和工作原理；计算机的存储系统；指令系统和汇编语言程序设计；计算机外围接口，输入/输出的基本方式，常用接口芯片及A/D、D/A转换的原理；介绍应用实例，这些例子都是我们在科研中的成功经验，并较系统地介绍了80386微型机的特点和MCS—51单片机的结构原理，对学生在将来的工作中，将有重要的参考价值和示范作用。

在教材编写中，力求贯彻基础理论与实际应用相结合的原则。在选材上注重科学性、先进性、实用性。内容组织上注意由浅入深，循序渐进，即便于组织教学，又便于学生自学。根据工科院校非计算机专业多学科、多层次的特点，采用"宽编窄用"的选材原则，本教材有较宽的适应面和灵活的选择余地，可适用于不同专业、不同层次、不同学时的教学安排。

本教材由刘瑞敏老师编写第二、七、八章；邹金慧老师编写第四章和第六章中的第5节；蔡光程老师编写第三章；吕杨老师编写第一、五章和第六章第1、2、3、4节。全书由吕杨老师统稿。

昆明理工大学计算机系杨腾祥教授对书稿进行了认真细致的审阅，并提出了十分宝贵的意见和修改建议。本书在编写过程中，得到昆明理工大学教务处周荣教授、宋焕斌教授和计算机系张怀宁教授的支持和指教。在此表示衷心地感谢！

尽管编者多年来从事微机应用方面的教学与科研工作，但由于水平和经验有限，书中难免有错误和不足之处，敬请使用本教材的专家和读者不吝赐教。

编者

1999年9月于昆明理工大学

目　录

第一章　绪　论

1.1　微型计算机发展概况

1.1.1　微型计算机的发展史

作为信息技术的基础——电子计算机是20世纪科学技术最卓越的成就之一，从第一台计算机问世至今的50多年来，它的发展之快在人类科技史上还没有哪一门学科可以与之相提并论。

在人类长河中这短暂的50多年里，电子计算机大致经历了电子管式计算机、晶体管式计算机、集成电路式(中小规模)计算机、大规模集成电路计算机几个阶段，现在世界上许多国家正在加紧研制以人工智能、神经网络为主要特征的新一代计算机。

电子计算机按其性能来分，有巨型、中型、小型和微型计算机。微型计算机是以微处理器(CPU)为核心，配上由大规模集成电路制作的存储器、输入/输出接口电路及系统总线所组成的计算机。以微型计算机为中心，配以相应外围设备和辅助电路，以及相应的系统软件，就构成了微型计算机系统。

自70年代初微型计算机问世以来，按其核心单元CPU的功能来划分，它已经历了五代的演变。

第一代(1971～1973年)是4位和低档8位微机。典型代表产品是美国Intel公司的4004、4040和8008，它们是采用PMOS工艺的4位、8位微处理器，由该微处理器构成的计算机只有串行十进制运算的功能。主要用于各类计算器中，集成度在每片2000个晶体管左右，基本指令执行时间约为10～20μs，以及由它们所组成的MCS－4微型计算机。

第二代(1974～1978年)是中高档8位微机，以Intel公司的8080和8085，Motorola公司的MC6800，Zilog公司的Z80为典型代表。它们是采用了NMOS工艺的8位微处理器，集成度已达到每片9000个晶体管，基本指令的执行时间为1～2μs，指令系统比较完善，运算功能较强，其应用范围已扩大作为接口芯片、控制器和一般的信号处理器等。

第三代(1978～1981年)是16位微机。典型代表有Intel公司的8086/8088，Motorola公司的MC68000和Zilog公司的Z8000，由于采用了HMOS高密度工艺，集成度高达每片29000个晶体管，运算速度比8位机快2～5倍，已经赶上甚至超过70年代小型机的水平。

第四代(1981～1992年)是32位微机。首先是Intel公司推出的性能更高，功能更强的高级16位微处理器80186和80286，它们与8086兼容。1985年，Intel公司又率先推出32位微处理器80386，它与8086，80186和80286兼容。1990年，Intel公司推出80486，它的平均运算速度可达每秒执行20M条指令。与此同时，Motorola公司推出32位微处理器M68020，集成度达每片

68000个晶体管,HP公司推出 μp32位微处理器芯片,集成度高达45万个晶体管,时钟频率达到18MHz。

基于第四代微处理器芯片组成的微型计算机其性能已达到70年代大中型计算机的水平。如80386微型机采用优化多任务操作系统,可直接寻找的物理存储空间为4000M字节,虚拟存储空间可达64MM字节,运算速度达每秒300~400万条指令。

第五代(1993年后)是64位微机。1993年3月Intel公司推出当前最先进的微处理器芯片——64位的Pentium,该芯片采用新的体系结构,其性能大大高于Intel系列的其他微处理器,为处理器体系结构和PC机的性能引入了全新的概念。

目前,微型计算机的发展趋势:一方面向小型化、性能更全的单片机方向发展。所谓单片机既是把微处理器、一定容量的存储器和必要的I/O接口电路集成在一片芯片上构成的微型计算机。单片机依其三高性能,即高可靠性、高控制功能及高运行速度,被广泛用于家用电器、办公机械、智能仪器仪表、实时控制等领域之中。另一方面向微巨型化方向发展。"微巨型机"是将超级微型计算机与巨型机的技术结合起来形成一种完全新型的体系结构。典型的代表有MASS COMP公司的MC 5000系列,它采用了最新的超大规模集成电路(VLSI)技术、多微处理器结构、虚拟存储器体系、流水线工作方式,在CPU与存储器之间设置了超高速缓冲存储器(Cache),内部采用多级共享总线结构,取得良好的效果。

总之,从70年代初至今,微型计算机技术得到了突飞猛进的发展;同时,它也促进了其他技术的迅速发展。

1.1.2 微型计算机的特点

建立在微电子技术加工工艺基础上的微型计算机有许多突出的优点,正是这些优点,使它从问世以来,就得到了极其迅速的发展和广泛的应用。

1.功能强

微型计算机的设计,参考了其他类型计算机的优点,使其运算速度快,计算精度高,具有记忆和逻辑判断能力,而且每种微处理器都配有一整套支持相应微型计算机工作的软件。硬件和软件的配合,相辅相成,使微型计算机的功能大大提高,适合各行各业的各种不同目的的应用。

2.可靠性高

由于微处理器及其配套系列芯片上可集成几千甚至几百万个元件,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。据某些资料估计,芯片集成度增加100倍,系统的可靠性也可增加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 ~ 10^8 小时。

3.价格低

微处理器及其配套系列芯片采用集成电路工艺,集成度高,适合工厂批量生产。因此,产品造价十分低廉。据估计集成度增加100倍,其价格也可降为同功能分立元件的百分之一。很显然,低的价格对于微型计算机的推广和普及是极为有利的。

4.适应性强

在微型计算机中,硬件扩展是很方便的,而且系统的软件是很容易改变的。因此,在相同

的配置情况下，只要对硬件和软件作些相应的变动和修改，就能适应不同用户的要求。

5.周期短，见效快

微处理器制造厂家除生产微处理器芯片外，还生产各种配套的支持芯片，同时也提供许多开发支持软件，为用户构成一个所需的微型计算机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，使研制的系统很快地投入运行，取得明显的经济效益。

6.体积小，重量轻，耗电省

微处理器及其配套支持芯片的尺寸均比较小，最大也不超过几百平方毫米。近几年在微型计算机中还大量地采用了 ASIC(大规模集成专用芯片)和 GAL(通用可编程门阵列)器件，使得微型计算机体积明显缩小。

目前微型计算机中芯片大多采用 MOS 和 CMOS 工艺，使耗电量很小。这样，使得微型计算机的应用领域得到进一步扩大，并成为高科技领域中不可缺少的研究手段和方式。

7.维护方便

现在用微处理器及其系列技术所构成的微型计算机已逐渐趋于标准化、模块化和系列化；一般都可用自检、诊断及测试发现系统故障，一旦发现故障，排除故障也就比较容易。

1.1.3 微型计算机的应用

微型计算机具有价格低廉、体积小、重量轻、功耗低、可靠性高、使用灵活等优点，且随着大规模和超大规模集成电路工艺的不断发展，其功能亦不断增强，使得微型计算机的应用日益深入各行各业，大到卫星、导弹的发射，矿产开采，石油勘探，小到儿童玩具、电子表的研制，微型计算机已成为当今信息时代不可缺少的一种重要工具。

微型计算机按其结构、规模及功能的不同，可广泛地应用于各个领域之中，下面列出几个主要的应用方面。

1.科学计算

科学计算一直是电子计算机的重要应用领域。世界上第一台计算机 ENIAC 就是用于计算弹道的。现在，包括天气预报的计算，人造卫星、原子反应堆和武器、导弹和航天飞机、大型水力枢纽、大型桥梁、高层建筑等的结构设计，飞机、轮船的外形设计都离不开大型高速计算机。在基础科学研究方面，脱氧核糖核酸和人工蛋白质的合成，人工胰岛素的合成，物质结构分析等复杂计算也都需要高速大型计算机。随着微型计算机的计算速度和容量的不断提高，虽然在运算速度上还比不上某些巨型机，但已能满足相当范围的科学计算需要，特别是微巨型机的发展，使微型机用于科学计算的前景更为广阔。

2.数据处理

电子计算机应用最广泛的领域是数据处理。所谓数据处理是指计算机用于处理生产、经济活动、社会和科学研究中获得的大量信息。例如人口普查数据的搜集、转换、分类、计算、存储、传输和输出报表；政府机关的办公文件的处理；银行的电子化，全市甚至全国同类银行联机办理存款支付；情报检索系统，通过卫星、Internet 网能查阅国内外某图书馆的资料，某行业的最新发展动态等；企业管理中的生产统计、账务管理、成本核算、库存管理等都离不开微型计算机；微型计算机在办公室自动化中所起到的作用更是引人注目。它有效地提高了办公室效率，减轻了办公室工作人员的劳动强度，根据需要打印的各种统计报表，不仅质量高，而且速度快。

3.过程控制

计算机在工业测量和控制方面的应用已十分成熟和广泛,如大型化工企业中的各项工艺参数自动采集、检验、比较以及工艺流程的控制;大型冶金企业的高炉温度控制,钢材轧制控制,铜吹炼转炉分布式监控管理系统等。在计算机集散系统中,将计算机技术与网络技术集为一体,不仅可直接对被控对象或生产流程进行控制,还可利用上位机进行监督管理。由于单片机的发展和应用,使仪器仪表向数字化、智能化、多功能和易于通信等方面发展,并具有精度高、体积小、成本低和适应性强等优点。微机的出现为实时控制开展了更为广泛的应用领域。

4.人工智能

计算机在人工智能方面的应用越来越引人注目。所谓人工智能是将人脑在进行演绎推理的思维过程、规则和所采取的策略、技巧等编成计算机程序,在计算机中存储一些公理和推理规则,然后让计算机去自动探索解题的方法,所以这种程序不同于计算机的一般应用程序。

智能机器人是人工智能各种研究课题的综合产物,有感知和理解周围环境、进行推理和操作工具的能力,并通过学习适应周围环境,完成某种动作。在不允许人进入的场所,如高温或有放射性物质等,使用机器人有特殊的意义。

尽管人工智能的研究已取得一些成果,但与建立真正的智能程序还相差甚远。

1.2 数制和编码

计算机的基本功能是对数进行加工和处理,数在机器中是以器件的物理状态来表示的。一个具有两种不同的稳定状态且能相互转换的器件实现起来较容易,就可用这样的器件来代替一位二进制数,所以在计算机中,所有数字、字符、指令都是用二进制数来表示的。

1.2.1 无符号数的表示及运算

一、无符号数的表示方法

1.二进制计数的表示方法

二进制计数的特点是:

(1)具有两个不同的数字符号,即 0 和 1。

(2)以 2 为底,逢 2 进位。

一个二进制数可以表示为如下形式:

$$N_B = \sum_{i=-m}^{n-1} B_i \times 2^i \qquad (1.2.1)$$

其中,m 表示小数位的位数,n 表示整数位的位数,B_i 为二进制数的数字符号 0 和 1。

例如:$1011.101B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

式中后缀 B 表示二进制数(Binary)。

2.十六进制计数的表示法

十六进制计数法的特点是：

(1)具有 16 个数字符号，采用 0～9 和 A～F。其中 A～F 依次表示 10～15。

(2)以 16 为底，逢 16 进位。

一个十六进制数可表示为如下形式：

$$N_H = \sum_{i=-m}^{n-1} H_i \times 16^i \tag{1.2.2}$$

例如：$3AB.15H = 3 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 + 1 \times 16^{-1} + 5 \times 16^{-2}$

式中后缀 H 表示十六进制数(Hexadecimal)。

但在机器中，十六进制数仍是用二进制表示的。由于二进制与十六进制之间存在着一特殊关系，即 $2^4 = 16$，于是一位十六进制数可用四位二进制数来表示。

一般来说，对于基数为 X 的任一数都可用多项式表示为：

$$N_X = \sum_{i=-m}^{n-1} K_i \times X^i \tag{1.2.3}$$

其中：X——基数，表示为 X 进位制；

i——位序号；

K_i——第 i 位的系数，可为 0，1，…，X－1 共 X 个数字符号中的任一数字符号；

m，n——m 为小数部分位数，n 为整数部分位数，均为正整数；

X^i——第 i 位的权。

二、各种数制之间的转换

1.任意进制数转换为十进制数

二进制、十六进制以至任意进制的数转换为十进制数的方法简单，可按式(1.2.1)、(1.2.2)、(1.2.3)公式展开求和即可得到相应的十进制数。

2.十进制数转换为任意进制数

由于整数部分与小数部分转换的规则不同，故整数部分与小数部分分别进行转换。

(1)十进制整数转换成任意进制整数

设 N 为任一十进制整数，若要将它转换成 n 位 X 进制整数，可表示为：

$$N = \sum_{i=0}^{n-1} K_i X^i$$
$$= k_{n-1} X^{n-1} + K_{n-2} X^{n-2} + \cdots + K_1 X^1 + K_0$$

显然，等式右边，除了最后一项 K_0 外，其余各项都包含基数 X 的因子，都能被 X 除尽。所以等式两边同除以基数 X，余数正是要求的 X 进制数的最低位 K_0。商为

$K_{n-1} X^{n-2} + K_{n-2} X^{n-3} + \cdots + K_2 X^1 + K_1$

同理，上式中各项除 K_1 之外，其余各项都包含基数 X 的因子，都能被 X 除尽。所以再将商除以 X，余数正是要求的 X 进制数的次低位 K_1，商为

$K_{n-1} X^{n-3} + K_{n-2} X^{n-4} + \cdots + K_3 X^1 + K_2$

依此类推，直到商等于 0 为止，所得到的一系列余数，正是所要求的 X 进制数的各位。但应注意的是：所得到余数的顺序是从低到高。

我们可以把十进制整数转换为 X 进制整数的方法总结为：用该数或商除以基数取余数，先为低位后为高位。

例 1.1： 将十进制数 301 分别转换为二进制数和十六进制数。

转换为二进制的过程如下：

$$
\begin{array}{rll}
2 \mid \underline{301} & \cdots\cdots\cdots\cdots & \\
2 \mid \underline{150} & \cdots\cdots\cdots\cdots & 1 = K_0 \\
2 \mid \underline{75} & \cdots\cdots\cdots\cdots & 0 = K_1 \\
2 \mid \underline{37} & \cdots\cdots\cdots\cdots & 1 = K_2 \\
2 \mid \underline{18} & \cdots\cdots\cdots\cdots & 1 = K_3 \\
2 \mid \underline{9} & \cdots\cdots\cdots\cdots & 0 = K_4 \\
2 \mid \underline{4} & \cdots\cdots\cdots\cdots & 1 = K_5 \\
2 \mid \underline{2} & \cdots\cdots\cdots\cdots & 0 = K_6 \\
2 \mid \underline{1} & \cdots\cdots\cdots\cdots & 0 = K_7 \\
0 & \cdots\cdots\cdots\cdots & 1 = K_8
\end{array}
$$

301 = 100101101B

转换为十六进制的过程如下：

$$
\begin{array}{rll}
16 \mid \underline{301} & & \\
16 \mid \underline{18} & \cdots\cdots\cdots\cdots & D = K_0 \\
16 \mid \underline{1} & \cdots\cdots\cdots\cdots & 2 = K_1 \\
0 & \cdots\cdots\cdots\cdots & 1 = K_2
\end{array}
$$

301 = 12DH

(2)十进制小数转换成任意进制小数

设 N 为任一十进制小数，若把它转换为 m 位 X 进制小数，

$$N = \sum_{i=-m}^{-1} KiX^i = K_{-1}X^{-1} + K_{-2}X^{-2} + \cdots + K_{-m}X^{-m}$$

等式两边同乘以基数 X，得到

$$K_{-1} + K_{-2}X^{-1} + K_{-3}X^{-2} + \cdots\cdots + K_{-m}X^{-m+1}$$

其中 K_{-1} 为整数部分，它正好是所要求的 X 进制小数的最高位，而新的小数部分为

$$K_{-2}X^{-1} + K_{-3}X^{-2} + \cdots\cdots + K_{-m}X^{-m+1}$$

再将上式乘以 X 便得到

$$K_{-2} + K_{-3}X^{-1} + K_{-4}X^{-2} + \cdots\cdots + K_{-m}X^{-m+2}$$

k_{-2} 为整数部分，正好为所要求的 X 进制小数的次高位。依此类推，直至小数部分为零时止。若乘积的小数部分始终不为 0，说明相对应的 X 进制小数为不尽小数，这时，乘到能满足精度要求为止。

综上所述，我们可以把十进制小数转换为相应 X 进制小数的方法总结为：对该小数或乘以 X 后新的小数部分乘以基数取整数，先为高位后为低位。

例 1.2： 将 0.6875 分别转换为二进制及十六进制小数。

转换为二进制的过程如下：

$0.6875 \times 2 = 1.375 \qquad K_{-1} = 1$

$0.375 \times 2 = 0.75 \qquad K_{-2} = 0$

$0.75 \times 2 = 1.5 \qquad K_{-3} = 1$

$0.5 \times 2 = 1.0 \qquad K_{-4} = 1$

$0.6875 = 0.1011B$

转换为十六进制的过程如下：

$0.6875 \times 16 = 11.0 \qquad K_{-1} = B$

$0.6875 = 0.BH$

例 1.3:将 0.734 转换为二进制小数。

转换过程如下：

$0.734 \times 2 = 1.468 \qquad K_{-1} = 1$

$0.468 \times 2 = 0.936 \qquad K_{-2} = 0$

$0.936 \times 2 = 1.872 \qquad K_{-3} = 1$

$0.872 \times 2 = 1.744 \qquad K_{-4} = 1$

$0.744 \times 2 = 1.488 \qquad K_{-5} = 1$

$0.488 \times 2 = 0.978 \qquad K_{-6} = 0$

$0.734 = 0.1011101\cdots B$

例 1.4:将十进制数 301.6875 转换为二进制数。利用例 1.1,例 1.2 的结果,得到

$301 = 100101101B \qquad 0.6875 = 0.1011B$

$301.6875 = 100101101.1011B$

3.二进制数与十六进制数之间的转换

由于 $16 = 2^4$,故 1 位十六进制数用 4 位二进制数表示。因此,十六进制数与二进制数之间的相互转换是十分简便的。

二进制数转换为十六进制数的方法:首先从小数点开始分别向左和向右把整数及小数部分每 4 位分成一组。若一组不够 4 位,则加 0 补足 4 位。然后,把每 4 位二进制数用相应的十六进制数代替,即得到对应的十六进制数。

例 1.5:二进制数 1011101001.110101 转换为十六进制数。

0010	1110	1001	.	1101	0100
↓	↓	↓		↓	↓
2	E	9	.	D	4

$1011101001.110101B = 2E9.D4H$

十六进制数转换为二进制数的方法:用相应的 4 位二进制数取代每 1 位十六进制数。

例 1.6:十六进制数 5C7A.3B 转换为二进制数。

5	C	7	A	.	3	B
↓	↓	↓	↓		↓	↓
0101	1100	0111	1010	.	0011	1011

$5C7A.3BH = 101110001111010.00111011B$

由于二进制数书写冗长易错，因此一般用十六进制来表示数，书写方便且不容易出错。

三、无符号二进制数的运算

1.二进制数的算术运算

(1)二进制加法

二进制加法规则为：0 + 0 = 0

0 + 1 = 1 + 0 = 1

1 + 1 = 0　　进位 1

1 + 1 + 1 = 1　　进位 1

例 1.7：10110101B + 1111B

进位		1 1 1 1 1 1
被加数		1 0 1 1 0 1 0 1
加数	+	0 0 0 0 1 1 1 1
和		1 1 0 0 0 1 0 0

10110101B + 1111B = 11000100B

可见，两个二进制数相加，每 1 位有 3 个数，即被加数、加数以及低位的进位，用二进制的加法规则得到本位的和及向高位的进位。

(2)二进制减法

二进制减法的规则为：0 - 0 = 0

1 - 1 = 0

1 - 0 = 1

0 - 1 = 1　　有借位

例 1.8：11000100B - 100101B

借　位		1 1 1 1 1 1
被减数		1 1 0 0 0 1 0 0
减　数	-	0 0 1 0 0 1 0 1
和		1 0 0 1 1 1 1 1

11000100B - 100101B = 10011111B

与加法相类似，两个二进制数相减时，每 1 位有 3 个数参加运算：本位的被减数和减数，以及低位来的借位。运算顺序为本位被减数—本位减数—低位来的借位，可得到本位的差及向高位的借位。

(3)二进制乘法

二进制乘法的运算规则为：0 × 0 = 0

0 × 1 = 0

1 × 0 = 0

1 × 1 = 1

例 1.9：1101B × 1011B

$$
\begin{array}{lr}
\text{被乘数} & 1101 \\
\text{乘　数} & \times 1011 \\
\hline
 & 1101 \\
 & 1101 \\
 & 0000 \\
 & +\,1101 \\
\hline
\text{积} & 10001111
\end{array}
$$

1101B × 1011B = 10001111B

从上述运算过程中可以清楚看到,二进制的乘法与十进制乘法相类似。但计算机实现起来不很方便,有的微型计算机有专门的乘法指令来完成乘法运算。在没有乘法指令的微型计算机中,是采用部分积右移的办法来实现乘法运算的。

(4)二进制除法

二进制除法运算与十进制除法运算相类似,举例说明。

例 1.10: 100110B ÷ 110B

$$
\begin{array}{r}
000110 \\
110\,\overline{)\,100110} \\
\underline{110} \\
111 \\
\underline{110} \\
10
\end{array}
$$

100110B ÷ 110B = 110B　　余 10B

有的微型计算机有专门的除法指令来完成除法运算,在没有除法指令的微型计算机中,常用“相减,左移”法编制除法运算程序。

2.二进制数的逻辑运算

(1)“与”运算(AND)

“与”运算又称为逻辑乘,可用符号“·”或“∧”表示。“与”运算的规则为:

$0 \wedge 0 = 0$

$0 \wedge 1 = 0$

$1 \wedge 0 = 0$

$1 \wedge 1 = 1$

(2)“或”运算(OR)

“或”运算又称为逻辑加,可用符号“+”或“∨”表示。“或”运算的规则为:

$0 \vee 0 = 0$

$1 \vee 0 = 1$

$0 \vee 1 = 1$

$1 \vee 1 = 1$

(3)“非”运算(NOT)

“非”运算可在某逻辑值上加“—”表示,“非”运算的规则为:

$\overline{0} = 1$　　$\overline{1} = 0$

(4)“异或”运算(XOR)

“异或”运算用“$\oplus$”表示,“异或”运算的规则为:

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$

多位二进制数进行逻辑运算时,是按位进行运算的。

例 1.11:二进制数 10010111 和 00111000 的“与”、“或”及“异或”。运算结果分别为:

$$\begin{array}{r} 10010111 \\ \wedge)\ 00111000 \\ \hline 00010000 \end{array} \qquad \begin{array}{r} 10010111 \\ \vee)\ 00111000 \\ \hline 10111111 \end{array}$$

$$\begin{array}{r} 10010111 \\ \oplus)\ 00111000 \\ \hline 10101111 \end{array}$$

1.2.2 带符号数的表示及运算

一、带符号数的表示

前面所讲述的二进制数均为无符号数,即所有二进制数位均为数值位。然而实际的数值有时是带有符号的,既可是正数,也可是负数。为了表示一个带符号数,除了数值位以外还应指定符号位,通常以这个二进制数的最高位为符号位,并用 0 表示正数的符号“+”,用 1 表示负数的符号“-”。这样,数的符号在机器中也数字化了,用符号位和数值位一起完整地表示带符号的二进制数。我们把一个数在机器中的表示形式称为机器数,把它们的实际数值称为机器数的真值。

符号位数值化之后,为了能方便地对机器数进行算术运算,提高运算速度,人们设计了多种符号位与数值位一起编码的方法,最常用的是原码、反码和补码。

1.原码表示法

如果将正数的符号位用 0 表示,负数的符号位用 1 表示,用数的绝对值表示数值部分,这种表示法就称为原码表示法。

把数 X 的原码记作$[X]_{原}$,当机器字长为 n 时,原码表示法可定义为:

$$[X]_{原} = \begin{cases} X & 0 \leqslant X \leqslant 2^{n-1} - 1 \\ 2^{n-1} + |X| & -(2^{n-1} - 1) \leqslant X \leqslant 0 \end{cases}$$

例如,当机器字长 n = 8 时,

$[+0]_{原} = 00000000$, $[-0]_{原} = 10000000$

$[+1]_{原} = 00000001$, $[-1]_{原} = 10000001$

$[+127]_{原} = 01111111$, $[-127]_{原} = 11111111$

由此看出，当 X 为正数时，$[X]_{原}$的值就是 X 自身；当 X 为负数时，$[X]_{原}$的符号位由 1 表示，而数值部分不变。

原码表示的整数范围是 $-(2^{n-1}-1) \sim +(2^{n-1}-1)$。8 位二进制原码表示的整数范围是 $-127 \sim +127$，16 位二进制原码表示整数的范围 $-32767 \sim +32767$。原码表示法简单直观，但不便于进行加减运算。

2.反码表示法

反码表示法是用机器数的最高一位代表符号位（表示方法与原码相同），数值位是对负数值各位取反的表示方法。

把数 X 的反码记作$[X]_{反}$，当机器字长为 n 时，反码表示法可定义如下：

$$[X]_{反}=\begin{cases}X & 0 \leqslant X \leqslant 2^{n-1}-1 \\ (2^n-1)-|X| & -(2^{n-1}-1) \leqslant X \leqslant 0\end{cases}$$

例如，当机器字长 n = 8 时，

$[+0]_{反}=00000000$，　　$[-0]_{反}=11111111$

$[+1]_{反}=00000001$，　　$[-1]_{反}=11111110$

$[+127]_{反}=01111111$，　　$[-127]_{反}=10000000$

从反码表示法中可见，当 X 为正数时，$[X]_{反}$的值就是 X 自身；当 X 为负数时，$[X]_{反}$的符号位用 1 表示，而数值部分是按位取反。

反码表示整数的范围是 $-(2^{n-1}-1) \sim +(2^{n-1}-1)$，与原码相同。

3.补码表示法

根据同余的概念：$a+NK=a \pmod K$

其中 K 是模，N 为任意整数。就是说，在模的意义下，数 a 与该数本身加上其模的任意整数倍之和相等。

在数 a 的无数个 a + NK 同余数中，我们感兴趣的是 N 为 1 的同余数，即补数：

$$[a]_{补数}=a+k \pmod K$$
$$=\begin{cases}a & 0 \leqslant a < K \\ K-|a| & -K < a < 0\end{cases}$$

由上式确定的两种条件下数 a 的补数，正是补码的定义和补码编码规则的基础。

在计算机运算过程中，数据的位数，即字长总是有限的，假设字长为 n，两数相加求和时，如果 n 位的最高位产生进位，进位位就会丢失，这正是在模的意义下相加的概念。相加时丢掉的进位即等于模。所以，当 n 位表示整数时（1 位为符号位，n - 1 位为数值位），它的模为 2^n。在 n 位字长的机器中，2^n 仅能以 n 个 0 来表示，而最左边的数字 1 就自动舍弃了，即模 2^n 和 0 在机器中的表示形式是完全一样的。

把数 X 的补码记做$[X]_{补}$，当机器字长为 n，补码可定义为：

$$[X]_{补}=\begin{cases}X & 0 \leqslant X < 2^{n-1}-1 \\ 2^n-|X| & -2^{n-1} \leqslant X < 0\end{cases} \pmod{2^n}$$

例如，当机器字长为 n = 8 时，

$[+0]_{补}=00000000$，　　$[-0]_{补}=00000000$

$[+1]_{补}=00000001$，　　$[-1]_{补}=2^8-|-1|=11111111$