成人高等教育教材

微机原理与接口技术

王晓军　徐志宏　编

北京邮电大学出版社

·北京·

内 容 简 介

本书是根据北京邮电大学高等函授‘微型计算机原理与接口技术’课程教学大纲编写而成。

全书共分七章：第一章概述，第二章微机组成原理，第三章 8086 指令系统，第四章 8086 汇编语言程序设计，第五章微机接口技术基础，第六章中断技术，第七章接口技术。

本书在基本原理与基本概念方面，结构严谨，由浅入深，并注重理论与实际相结合，便于教学，更适于自学。

本书除适于高等函授等成人教育使用外，还可供同类专业的学生及培训学员学习，亦可供广大科技工作者参考。

图书在版编目(CIP)数据

微机原理与接口技术/ 王晓军，徐志宏编 .—北京：北京邮电大学出版社，2001 .5

ISBN 7-5635-0497-4

Ⅰ. 微 . . .　Ⅱ. ① 王 . . . ② 徐 . . .　Ⅲ. ①微型计算机—理论② 微型计算机—接口　Ⅳ. TP36

中国版本图书馆 CIP 数据核字 (2001) 第 12019 号

微机原理与接口技术

编　者　王晓军　徐志宏

责任编辑　郑　捷

*

北京邮电大学出版社出版发行

新华书店北京发行所发行　各地新华书店经售

北京源海印刷厂印刷

*

787 mm×1092 mm　1/ 16　印张 23　字数 540 千字

2001 年 5 月第 1 版　2001 年 5 月第 1 次印刷

印数：1—65 00 册

ISBN 7-5635-0497-4/ TP・48　定价：39 .00 元

前　言

本教材以 Intel8086/ 8088 系列微型计算机为背景,全面系统地阐述了微型计算机的基本概念和基本原理。同时,介绍了从 Intel80286 到 Pentium 微处理器的最新技术发展。

为了适应教学要求,本教材解决了微机的结构、汇编语言程序设计、外部设备与主机之间的衔接等问题。其中,微机结构部分主要介绍了微机的基本组成和一些计算机的基础知识,这是学习本书内容的基础。学习汇编语言的难度相对大些,但它是学习微机原理不可缺少的一部分。通过学习它可以掌握微机的工作过程和原理,同时解决一些高级语言无法解决的问题。

至于接口部分,以接口技术为主线介绍了各种接口技术,再辅之以应用实例,这样可使读者较系统地掌握接口技术本身,以便应用于各种实际工作之中。

本教材是根据北京邮电大学高等函授‘微型计算机原理与接口技术’课程教学大纲编写而成。编者在多年教学实践的基础上,以理论联系实际为原则,力求由浅入深、循序渐进,以使读者通过本书的学习,掌握微型计算机系统的组成原理和工作原理,并具有一定应用能力。

全书共分七章:第一章概述,第二章微机组成原理,第三章 8086 指令系统,第四章 8086 汇编语言程序设计,第五章微机接口技术基础,第六章中断技术,第七章接口技术。

本书第一至四章由徐志宏编写,第五至七章由王晓军编写。

由于编者水平有限,书中难免存在缺点与错误,敬请读者批评指正。

编　者

2001 年 1 月

目　　录

第一章　概　述

自 学 指 导

本章主要介绍计算机的基础知识,这是学习计算机技术必不可缺少的内容。

本章简要介绍了微型计算机的发展、分类及其基本用途,重点介绍了计算机的数制和码制以及信息的编码方式,应该掌握数制之间的转换方法、补码的运算,了解微型计算机的概况和信息在计算机中的编码方法。

第一节　微型计算机的概况

一、微型计算机的发展与现状

电子计算机是由各种电子器件组成的,能够自动、高速、精确地进行逻辑控制和信息处理的现代化设备。从第一台电子计算机出现至今的 50 多年来,已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四个时代。现在世界上许多国家正在研制以人工智能、神经网络为主要特征的新一代计算机。

电子计算机按其功能来分,有巨型、中型、小型和微型计算机。微型计算机的核心部分是微处理器或微处理机,它是指由一片或几片大规模集成电路组成的,具有运算器和控制器功能的中央处理器(CPU)。

以微处理器为核心,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,简称微型计算机。以微型计算机为中心,配以相应的外围设备、电源和辅助电路以及指挥微型计算机工作的系统软件,就构成了微型计算机系统。

自从微处理器和微型计算机问世以来,按 CPU 字长和功能划分,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。代表产品是美国 Intel 公司的 4004 微处理机及由它组成的 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 MC68000, 美国 Zilog 公司的 Z80 等为 CPU 的微型机为典型代表。

第三代(1978～1981 年)是 16 位微机。如以 8086, Z8000 和 MC68000 为 CPU 的微型机。

第四代(1981～1992 年)是 32 位微机。典型的 CPU 产品有 80386, MC68020。之后, Intel 公司又推出了 80486 微处理器。

第五代(1993 年以后)是 64 位微机。1993 年 3 月 Intel 公司推出了微处理器芯片——64 位的 Pentium。该芯片采用了新的体系结构,其性能大大高于以前 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。

二、微型计算机的分类

微处理器(MP: Micro Processor)是集成在一片大规模集成电路芯片上的中央处理器,又称 MPU,简称 MP。

按 MPU 处理数据的位数来看,微处理器可分为 4 位、8 位、16 位、32 位和 64 位 MPU。

从制造微处理器器件的工艺来看,可分为 MOS 工艺的通用微处理器和双极型 TTL 工艺的位片式微处理器。

三、微型计算机的应用范围

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,促进了世界新产业革命的到来。微型计算机在当今信息时代是不可缺少的一种重要工具。

微型计算机按其复杂程度的不同,可适用于各种行业,从仪器仪表和家电的智能化到科学计算、自动控制、数据和事务处理、辅助设计、辅助教学和人工智能等均可应用微型计算机。

低档的微型机和单片式微型机在仪器仪表和家电的智能化方面的应用,取代了过去的硬件逻辑电路对仪器仪表和家电的控制,这种控制本身并不复杂,但却经常是重复的,从而使逻辑电路庞大而复杂。采用微型计算机之后,很容易简化其控制逻辑,用程序的重复执行以及循环控制,可以作到电路最省、控制更佳,并可通过修改程序来修改控制方案,因而灵活多变、可靠性高。

自从 20 世纪 80 年代初期 IBM PC 系列个人计算机出现以后,微型计算机在数据处理和管理方面的应用迅速推广,不仅工矿企业的各种数据和管理可应用微型计算机,而且微型计算机的应用还深入到家庭。

微型计算机另一主要应用方面是生产过程的自动控制。比起常规的自动控制仪表而言,微型计算机自动控制系统具有不可比拟的优点。

微型计算机在计算机辅助设计(CAD: Computer Aided Design)、计算机辅助教学(CAI:Computer Aided Instruction)、计算机辅助生产(CAM:Computer Aided Manufacturing)等方面的应用,是用功能很强的高档微机来实现的。它要求具有足够的存储器空间、高分辨率的显示器、丰富的系统软件和应用软件。

微型计算机在科学计算和数据处理方面的应用,是与微处理器单机功能的增强和协处理器的发展分不开的。数据处理能力增强的微型计算机(或多处理器系统)在科学计算方面能够发挥较大的作用。

微型计算机在人工智能方面有自动证明数学定理、代替专家为病人看病等广泛的

应用。

第二节　计算机中的数制

计算机的基本功能是对数据进行加工,因此要加工的数据必须送入计算机中。人们习惯用十进制数,而计算机中却采用二进制,这是因为制作具有十个物理状态的器件很困难,而制作具有两个物理状态的器件却容易得多,省器件,且有成熟的逻辑处理工具,运算、处理也方便,所以在计算机中,所用的数字、字符、指令、状态都是用二进制数来表示的。为了书写方便,计算机还采用其他进制数,如十六进制和八进制等。

一、无符号数的表示方法

1.十进制计数的表示法

十进制计数法的特点是:

以 10 为底,逢 10 进位;

需要 10 个数字符号 0,1,2,3,4,5,6,7,8,9。

任何一个十进制数 N_D 可以表示为:

$$N_D = \sum_{i=-m}^{n-1} D_i \times 10^i \qquad (1.2.1)$$

其中,m 表示小数位的位数,n 表示整数位的位数,D_i 为十进制数字符号 0～9。

例如:

$$374.53D = 3\times10^2 + 7\times10^1 + 4\times10^0 + 5\times10^{-1} + 3\times10^{-2}$$

上式中后缀 D 表示十进制数(Decimal),标识符 D 也可省略。

2.二进制计数的表示法

二进制计数法的特点是:

以 2 为底,逢 2 进位;

需要两个数字符号 0,1。

一个二进制数 N_B 可以表示为:

$$N_B = \sum_{i=-m}^{n-1} B_i \times 2^i \qquad (1.2.2)$$

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

例如:

$$1101.1B = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1}$$

上式中后缀 B 表示二进制数(Binary)。

3.十六进制计数的表示法

十六进制计数法的特点是:

以 16 为底,逢 16 进位;

需要 16 个数字符号 0,1,2,…,9,A,B,C,D,E,F。其中 A～F 依次表示 10～15。

一个十六进制数 N_H 可以表示为:

$$N_H = \sum_{i=-m}^{n-1} H_i \times 16^i \qquad (1.2.3)$$

其中，m 表示小数位的位数，n 表示整数位的位数，H_i 为十六进制数字符号 0～F。

例如：

$$E5AD.BFH = 14\times16^3 + 5\times16^2 + 10\times16^1 + 13\times16^0 + 11\times16^{-1} + 15\times16^{-2}$$

上式中后缀 H 表示十六进制数(Hexadecimal)。

一般来说，对于基数为 X 的任一数可以用多项式表示为：

$$N_X = \sum_{i=-m}^{n-1} K_i X^i \qquad (1.2.4)$$

其中：X——基数，表示 X 进位制；

i——位序号；

K_i——第 i 位的系数，可以为 0，1，2，…，$X-1$ 共 X 个数字符号中任一数字符号；

m，n——m 为小数部分位数，n 为整数部分位数；

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(1.2.2)，(1.2.3)，(1.2.4)公式展开求和即可。

2．十进制数转换为二进制数

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) 十进制整数转换为二进制整数

任何一个十进制数转换为二进制数后，都可以表示成为式(1.2.2)的形式。问题的核心在于求出 n 及 B_i。

下面通过一个简单的例子分析一下转换的方法。例如，已知

$$13D = 1101B = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$$
$$B_3 \quad B_2 \quad B_1 \quad B_0$$

上式也可以表示为：

$$13D = 1101B = (1\times2^2 + 1\times2)\times2 + 0\times2^1 + 1\times2^0$$
$$= ((1\times2+1)\times2+0)\times2+1$$
$$B_3 \quad B_2 \quad B_1 \quad B_0$$

可见，要确定 13D 对应的二进制数，只需从右到左分别确定 B_0，B_1，B_2 和 B_3 即可。显然，从上式可以归纳出以下转换方法，即用 2 连续去除十进制数，直至商等于零为止。逆序排列余数便是与该十进制数相应的二进制数各位的系数值。过程如下：

2　13

2　6　　1（商 6 余 1）——B_0

2　3　　0（商 3 余 0）——B_1

2　1　　1（商 1 余 1）——B_2

2　0　　1（商 0 余 1）——B_3

所以　　13D = 1101B

用与此类似的方法也可以完成十进制数至十六进制数的转换，不同的是用 16 连续去除而已。

(2) 十进制小数转换为二进制小数

根据式(1.2.2)

$$0.812\,5D = B_{-1} \times 2^{-1} + B_{-2} \times 2^{-2} + B_{-3} \times 2^{-3} + B_{-4} \times 2^{-4}$$

$$= 2^{-1}(B_{-1} + 2^{-1}(B_{-2} + 2^{-1}(B_{-3} + 2^{-1} \times B_{-4})))$$

由上式可以看出，十进制小数转换为二进制小数的方法是，连续用 2 去乘十进制小数，直至乘积的小数部分等于“0”。顺序排列每次乘积的整数部分，便得到二进制小数各位的系数 B_{-1}，B_{-2}，B_{-3}，…若乘积的小数部分永不为“0”，则根据精度的要求截取一定的位数即可。

0.8125D 的转换过程如下：

0.812 5D × 2 = 1.625，得出 B_{-1} = 1

0.625D × 2 = 1.25，得出 B_{-2} = 1

0.25D × 2 = 0.5，得出 B_{-3} = 0

0.50D × 2 = 1.0，得出 B_{-4} = 1

所以　　0.812 5D = 0.1101B

可见　　13.812 5D = 1101.1101B

(3) 二进制数与十六进制数之间的转换

因为 2^4 = 16，故二进制数转换为十六进制数只需以小数点为起点，向两端每 4 位(不足 4 位者补 0)二进制用 1 位十六进制数表示即可。例如，十六进制数转换为二进制数时，将每 1 位十六进制数转换为相应的 4 位二进制数即可。

1101110.01011B = 01101110.01011000B = 6E.58H

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

二进制加法运算规则为：

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0(进位 1)

(2) 二进制减法

二进制减法运算规则为：

0 - 0 = 0

1 - 1 = 0

1 - 0 = 1

0 - 1 = 1(有借位)

(3) 二进制乘法

二进制乘法运算规则为:　　0×0 = 1×0 = 0×1 = 0

1×1 = 1

(4) 二进制除法

二进制除法是乘法的逆运算。

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“ Λ ”表示。 *A*, *B* 两个逻辑变量进行“ 与 ”运算规则如下:

A	*B*	*A*Λ*B*
0	0	0
0	1	0
1	0	0
1	1	1

由上可知,只有当 *A*, *B* 变量皆为“ 1 ”时,“ 与 ”的结果才为“ 1 ”。

(2)“ 或 ”运算(OR)

“ 或 ”运算又称为逻辑加,可用符号“ + ”或“ V ”表示。 *A*, *B* 两个逻辑变量进行“ 或 ”运算规则如下:

A	*B*	*A*V*B*
0	0	0
0	1	1
1	0	1
1	1	1

由上可知, *A*, *B* 两变量中,只要有一个为“ 1 ”,“ 或 ”运算的结果就是“ 1 ”。

(3)“ 非 ”运算(NOT)

变量 *A* 的“ 非 ”运算的结果用 *A*¯ 表示,“ 非 ”运算规则如下:

A	*A*¯
0	1
1	0

(4)“异或”运算(XOR)

“异或”运算用“ " ”表示,逻辑变量 A, B 进行“异或”运算的规则如下:

A	B	A " B
0	0	0
0	1	1
1	0	1
1	1	0

由上可知, A, B 两变量只要不同,“异或”运算的结果就是 1。

四、定点数和浮点数

在计算机中,用二进制数表示实数的方法有两种,即定点数和浮点数。

1 . 定点数

所谓定点数,即小数点在数中的位置是固定不变的。以定点法表示的实数叫定点数。通常,定点数表示的数 N 范围,用原码表示是:

$$1 - 2^{-n} \geq N \geq -(1 - 2^{-n})$$

它能表示的数的最大绝对值为 $1 - 2^{-n}$,最小绝对值为 2^{-n}。

规定小数点固定在最低数值位之后,机器中能表示的所有数都是整数。 N 位数值部分所能表示的数 N 的范围,用原码表示是:

$$2^{n} - 1 \geq N \geq -(2^{n} - 1)$$

它能表示的数的最大绝对值为 $2^{n} - 1$,最小绝对值为 1。

图 1. 1 给出定点数的两种表示法。

因为实际数值很少有都是小数或都是整数的,所以定点表示法要求程序员做的一件重要工作是为要计算的问题选择“比例因子”。所有原始数据都要用比例因子化成小数或整数计算,结果又要按比例因子恢复为实际值。在计算过程中,中间结果若超过最大绝对值,机器便产生溢出,叫做“上溢”。这时必须重新调整比例因子。中间结果若超过最小绝对值,计算机只能把它当作 0 处理,叫做“下溢”。这时也必须重新调整比例因子。对于复杂计算,计算中间需多次调整比例因子。

2 . 浮点数

任意一个二进制数 N 总可以写成下面的形式:

$$N = \pm d \cdot 2^{\pm p}$$

其中: d 称为尾数,是二进制纯小数,指明数的全部有效数字。前面的符号称作数符,表示数的符号,用尾数的最高位表示。0 表示正,1 表示负; p 称为阶数,它的符号位称作阶符,用阶码最高位表示。阶符为正时,用 0 表示,阶符为负时,用 1 表示。由此可知,将尾部 d

的小数点向右(对 + p)或向左(对 - p)移动 p 位,即得数值 N。所以阶符和阶码指明小数点的位置。小数点随着 p 的符号和大小而浮动,这种数称为浮点数。浮点数的编码格式如图 1 .2 所示。

设阶码的位数为 m 位,尾数的位数为 n,则浮点数的取值范围为:

$$2^{-n} \cdot 2^{-(2^m-1)} \leq |N| \leq (1 - 2^{-n}) \cdot 2^{(2^m-1)}$$

浮点数能表示的数值范围大,是它的主要可取之处。

如果尾数的绝对值小于 1 且大于等于 0 .5,即采用原码编码的正数或负数和采用补码编码的正数,其尾数的最高位数字为 1;采用补码编码的负数,其尾数的最高位数字为 0,则该浮点二进制数被称为规格化浮点数。浮点运算后,经常要把结果规格化。规格化的操作是尾数每右移 1 位(相当于小数点左移 1 位),阶码加 1;尾数每左移 1 位,阶码减 1。

数的加减运算要求小数点对齐。对于浮点表示的数而言,就是阶码(包括阶符)相等。使阶码相等的操作称为对阶。一个浮点数的阶码改变,必须伴随着尾数的移位,才不改变数的值。即阶码加 1,尾数必须右移 1 位;阶码减 1,则要求尾数左移 1 位。两个规格化的浮点数相加或相减之前必须对阶。对阶的规则是:将两个数中阶码小的数的尾数右移,阶码增大,直到与另一个数的阶码相等为止。这样操作是合理的,因为尾数右移,只可能丢失最低有效位,造成误差较小。

可见,完成浮点数加法或减法运算,需要进行对阶、求和、规格化、舍入、判溢出等步骤,下面举例说明。

例如,设,$x = 2^{010} \times 0.11011011$, $y = 2^{100} \times (-0.10101100)$,求 $x + y$。

解:假设两数均以补码表示,且采用双符号位,则它门的浮点表示分别为:

$$[x]_{浮} = 00\ 010, 00.11011011$$

$$[y]_{浮} = 00\ 100, 11.10101100$$

(1) 求阶差及对阶

$$\Delta_E = E_X - E_Y = [E_X]_{补} + [-E_Y]_{补} = 00\ 010 + 11\ 100 = 11\ 110$$

即 Δ_E 为 - 2, x 的阶码小,应使 M_X 右移 2 位, E_X 加 2,

$$[x]_{补} = 00\ 100, 00.110110(11)$$

其中(11)表示 M_X 右移两位后移出的最低两位数。

(2) 尾数和求和

$$\begin{array}{r} 00.0\ 0\ 1\ 1\ 0\ 1\ 1\ 0(11) \\ +\ 11.0\ 1\ 0\ 1\ 0\ 1\ 0\ 0 \\ \hline 11.1\ 0\ 0\ 0\ 1\ 0\ 1\ 0(11) \end{array}$$

(3) 规格化处理

尾数运算结果的符号位与最高位数值同值,应执行左移处理,结果为11,00010101(10),阶码为 00011。

(4) 舍入处理

采用 0 舍 1 入法处理,则有:

$$\begin{array}{r} 11.0\,0\,0\,1\,0\,1\,0\,1 \\ +\qquad\qquad\qquad\quad 1 \\ \hline 11.0\,0\,0\,1\,0\,1\,1\,0 \end{array}$$

(5) 判溢出

阶码符号位 00,不溢出,故得最终结果为:

$$x + y = 2^{011} \times (-0.11101010)$$

符点数的乘除法方法比较简单,这里就不详细说明了。

第三节　计算机中的码制

一、带符号数的表示

日常生活中遇到的数除无符号数外,还有带符号数。数的符号在计算机中也用二进制数表示,通常用二进制的最高位表示数的符号。把一个数及其符号在机器中的表示加以数值化,这样的数称为机器数,而机器数所代表的数称为该机器数的真值。机器数可以用不同方法表示,常用的有原码、反码和补码表示法。

1.原码

数 x 的原码记作 $[x]_{原}$,如机器字长为 n,则原码的定义如下:

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

例如,当机器字长 $n=8$ 时,

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

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

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

当机器字长 $n=16$ 时,

$[+0]_{原} = 0000000000000000$　　$[-0]_{原} = 1000000000000000$

$[+1]_{原} = 0000000000000001$　　$[-1]_{原} = 1000000000000001$

$[+32\ 767]_{原} = 0111111111111111$　　$[-32\ 767]_{原} = 1111111111111111$

由此看出,在原码表示中,最高位为符号位,正数为 0,负数为 1,其余 $n-1$ 位表示数的绝对值。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～+127,16位二进制原码表示的整数范围是 -32767～+32767。原码表示法简单直观,但不便于进行加减运算。

2.反 码

数 x 的反码记作 $[x]_{反}$,如机器字长为 n,反码定义如下:

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

例如,当机器字长 $n=8$ 时,