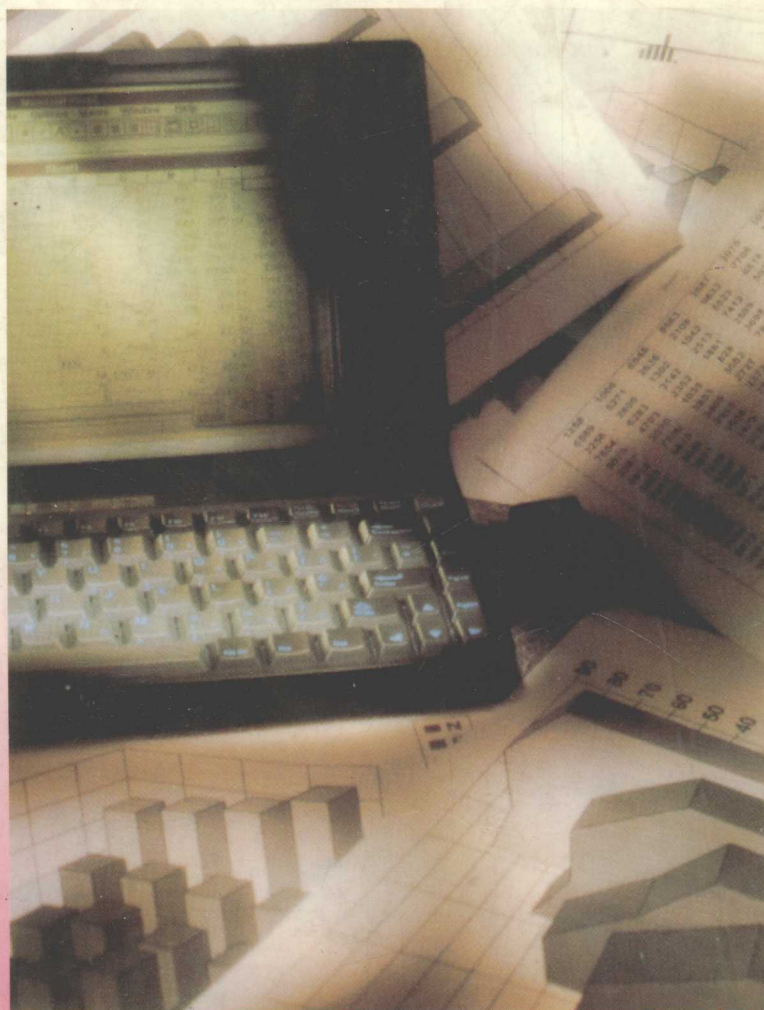


微型计算机原理及应用

尹朝庆 杨国勋 编著
胡家宝 杨 青



大连海事大学出版社

微型计算机原理及应用

尹朝庆 杨国勋 编著
胡家宝 杨 青

大连海事大学出版社

图书在版编目(CIP)数据

微型计算机原理及应用/尹朝庆等编著. —大连:大连海事大学出版社,
1999.1

ISBN 7-5632-1042-3

I. 微… II. 尹… 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 03057 号

大连海事大学出版社出版

(大连市凌水桥 邮政编码 116026 电话 4684394)

大连海事大学印刷厂印刷

大连海事大学出版社发行

1999 年 1 月第 1 版

1999 年 1 月第 1 次印刷

开本:787×1092 1/16

印张:15.75

字数:403 千

印数:0~1300 册

责任编辑:田立华

封面设计:王 艳

定价:20.50 元

前 言

《微型计算机原理及应用》是高等学校工科各专业普遍开设的课程。根据国家教委有关高等学校工科专业“微型计算机原理及应用”课程教学的要求,本课程的任务应是使学生从理论和实践上掌握微型机的基本组成、工作原理、接口电路及硬件的连接,建立微机系统的整体概念,使学生具有应用微机系统软硬件开发的基本技能。微型计算机自问世以来发展很快,尤其是近几年来,新技术、新芯片、新机型、新应用等层出不穷。因此,课程教学内容需要不断更新和充实。

本书结合大量实例,讲述微型计算机系统的基本组成、工作原理、接口电路和汇编语言程序设计的基本知识。全书共分六章:第一章讲述计算机中的数制和编码,IBM PC/XT 的配置,8086/8088、8087、80286、80386、80486、Pentium 等微处理器,以及多媒体个人计算机的配置及有关基本知识;第二章讲述 8086/8088 指令系统,80286 和 80386 对指令的功能扩充和增加的指令;第三章讲述汇编语言程序设计;第四章讲述各类半导体存储器,IBM PC/XT 的 RAM 和 ROM 子系统,以及高速缓冲存储器;第五章讲述 IBM PC/XT 的中断系统,以及 80386/80486 的中断;第六章讲述输入/输出系统,包括 IBM PC/XT 的系统总线与 I/O 端口,可编程计数器/定时器、并行输入/输出接口、异步串行输入/输出接口、DMA 控制器、多功能 I/O 接口 82380 等。各章附有习题,反映各章的基本要求,用以培养学生分析问题和解决问题的能力。

本书结构清晰完整,系统性强,精选各章内容,注重微型机系统的基本知识和基本技术,力求反映微型机的新技术、新机型、新应用。叙述深入浅出、循序渐进、逐级提高,实例丰富,利教利学。

本书可作为高等学校工科各专业的本科生学习有关微型机系统原理及应用方面课程的教材,参考学时 50~60 学时。也可供从事微型计算机硬件或软件技术工作的工程技术人员参考。

本书第一章和第五章由尹朝庆编写;第二章由杨国勋编写;第三章由杨青编写;第四章和第六章由胡家宝编写。尹朝庆担任主编,负责全书内容的修改和最后定稿。以上作者均长期从事研究生和本科生的微机课程的教学工作。

这里,要特别感谢本书的主审人李腊元教授,他进行了认真细致的审稿,并提出了宝贵的修改意见。本书在编写中得到武汉交通科技大学教材科等单位的大力支持。凡在本书编写过程中提出过意见和建议的同志,在此一并表示衷心的感谢。

恳请读者对本书的缺点和错误予以指正。

编 者

1997 年 9 月

目 录

第一章 微型计算机系统基础	1
1.1 计算机中的数制和编码	1
1.1.1 无符号数的表示及运算	1
1.1.2 带符号数的表示及运算	5
1.1.3 二进制编码	7
1.2 微型计算机系统的组成	8
1.2.1 微型计算机硬件	8
1.2.2 微型计算机软件	9
1.2.3 IBM PC/XT 的配置	12
1.3 微处理器	16
1.3.1 Intel 8086/8088	16
1.3.2 Intel 8087 协处理器	24
1.3.3 Intel 80286	29
1.3.4 Intel 80386	32
1.3.5 Intel 80486	34
1.3.6 Pentium	36
1.4 多媒体个人计算机	37
1.4.1 MPC 的系统总线与内存扩展	37
1.4.2 显示系统	41
1.4.3 光存储设备	44
1.4.4 音频卡与视频卡	48
习题一	50
第二章 指令系统	52
2.1 寻址方式	52
2.1.1 立即寻址	52
2.1.2 寄存器寻址	53
2.1.3 直接寻址	53
2.1.4 寄存器间接寻址	53
2.1.5 变址寻址	54
2.1.6 基址寻址	54
2.1.7 基址—变址寻址	55
2.1.8 操作数的类型	55
2.2 8086/8088 指令系统	57
2.2.1 数据传送指令	58

2.2.2	算术运算指令	64
2.2.3	逻辑运算和移位指令	74
2.2.4	串操作指令	80
2.2.5	控制转移指令	83
2.2.6	处理器控制指令	90
2.3	80286 对指令的功能扩充和增加	91
2.3.1	80286 对指令功能的扩充	91
2.3.2	80286 增加的指令	93
2.4	80386 对指令的功能扩充和增加	96
2.4.1	80386 对指令功能的扩充	96
2.4.2	80386 增加的指令	97
	习题二	100
第三章	汇编语言程序设计	105
3.1	概述	105
3.2	汇编语言源程序的格式	105
3.3	语句行的构成及伪运算符	107
3.3.1	语句行的构成	107
3.3.2	伪运算符	109
3.4	伪操作	112
3.4.1	处理器方式伪操作	113
3.4.2	数据定义伪操作	114
3.4.3	符号定义伪操作	117
3.4.4	段定义伪操作	119
3.4.5	过程定义伪操作	122
3.4.6	模块定义与连接伪操作	123
3.4.7	宏处理伪操作	125
3.4.8	其他伪操作	127
3.5	DOS 和 BIOS 中断调用	127
3.5.1	DOS 中断调用及系统功能调用	127
3.5.2	BIOS 中断调用	129
3.6	汇编语言程序设计及举例	129
3.6.1	顺序程序设计	129
3.6.2	循环程序设计	131
3.6.3	分支程序设计	135
3.6.4	子程序设计	137
	习题三	145
第四章	半导体存储器	149
4.1	概述	149
4.2	随机读写存储器(RAM)	151
4.2.1	静态随机读写存储器(SRAM)	151

4.2.2 动态随机读写存储器(DRAM)	154
4.3 只读存储器(ROM)	158
4.3.1 掩膜只读存储器	158
4.3.2 可擦除只读存储器(EPROM)	158
4.3.3 电擦除只读存储器(EEPROM)	161
4.4 CPU 与存储器的连接	162
4.4.1 连接时应注意的问题	162
4.4.2 典型的 CPU 与存储器的连接	162
4.5 IBM PC/XT 的存储器	164
4.5.1 存储器空间的分配	164
4.5.2 ROM 存储器空间	165
4.5.3 RAM 存储器空间	166
4.6 存储器管理	167
4.6.1 存储器管理	167
4.6.2 存储器管理软件	168
4.7 高速缓冲存储器(Cache)	170
习题四	171
第五章 中断系统	173
5.1 中断的概念	173
5.1.1 中断类型	173
5.1.2 中断响应	175
5.1.3 80386/80486 的中断	177
5.2 可编程中断控制器 Intel 8259A	178
5.2.1 8259A 的引脚及功能结构	178
5.2.2 8259A 的编程控制	181
5.3 中断系统实例分析	185
5.3.1 IBM PC/XT 的中断控制逻辑结构	185
5.3.2 IBM PC/XT 的 BIOS 对 8259A 的初始化	188
5.3.3 利用软中断方式扩展中断系统功能	188
习题五	191
第六章 数字量输入输出	192
6.1 概述	192
6.1.1 I/O 接口	192
6.1.2 I/O 接口的数据传送的方式	193
6.1.3 I/O 端口的寻址方式	193
6.1.4 I/O 接口的控制方式	194
6.2 IBM PC/XT 的系统总线与系统 I/O 端口	194
6.2.1 IBM PC/XT 的系统总线	195
6.2.2 IBM PC/XT 的系统 I/O 端口	200
6.3 可编程计数器/定时器	202

6.3.1	8253 的结构	202
6.3.2	8253 的编程	203
6.3.3	8253 的应用	207
6.4	并行输入/输出接口	210
6.4.1	可编程并行接口 Intel 8255A	210
6.4.2	8255A 的应用	215
6.5	异步串行输入/输出接口	217
6.5.1	异步串行通信格式	218
6.5.2	异步串行 I/O 接口标准	218
6.5.3	可编程串行接口 Ins 8250	220
6.6	直接存储器存取(DMA)控制器	228
6.6.1	DMA 控制器 Intel 8237	229
6.6.2	Intel 8237 的应用	237
6.7	多功能 I/O 接口电路	238
6.7.1	82380 的结构	238
6.7.2	82380 的 DMA 功能	239
6.7.3	82380 的中断功能	239
6.7.4	82380 的定时器	240
	习题六	240
	主要参考文献	241

第一章 微型计算机系统基础

计算机按其性能、价格和体积的不同，一般分为五大类：巨型机、大型机、中型机、小型机和微型机。微型机是本世纪 70 年代初研制成功的，大规模集成电路技术的不断发展为微型机的产生和发展打下了坚实的物质基础。

微处理器是微型机的核心芯片，也称为中央处理单元 (CPU)。20 多年来，微处理器和微型计算机获得极快的发展，几乎每两年微处理器的集成度翻一番，每 2~4 年更新换代一次，现已进入第五代。第一代 (1971~1973 年) 是 4 位或低档 8 位微处理器。第二代 (1974~1978 年) 是中高档 8 位微处理器。第三代 (1978~1981 年) 是 16 位微处理器，以 Intel 公司推出的 Intel 8086、Motorola 公司推出的 MC68000、Zilog 公司推出的 Z8000 为代表。Intel 公司很快又推出 8088，并以 8088 为 CPU 组成了 IBM PC、PC/XT 等准 16 位机。由于其性能价格比较高，很快占领了世界市场。与此同时，Intel 公司在 8086 基础上研制出性能更优越的 16 位微处理器芯片 80286，以 80286 为 CPU 组成 IBM PC/AT 高档 16 位机。第四代是 1985 年后推出的 32 位高档微处理器。1985 年，Intel 公司推出 32 位微处理器芯片 80386。1990 年，Intel 公司在 80386 基础上研制出新一代 32 位微处理器芯片 80486。分别以它们为 CPU，推出 386 微机和 486 微机。第五代是 1993 年后推出的 64 位高档微处理器。1993 年，Intel 公司推出 64 位微处理器芯片 Pentium，以它为 CPU 的 Pentium 机是一种 64 位高档微机。IBM、Apple 和 Motorola 三家公司合作生产的 Power PC 芯片是另一种优异的 64 位微处理器芯片，以它为 CPU 的微型机型号是 Macintosh。

新一代高性能的 P6 微处理器芯片也已推出，装有 P6 芯片的 686 微机也已问世。

计算机技术的发展使微型机的性能价格比有了很大的提高，与此同时，多媒体技术也有了长足的进步。多媒体技术大大拓宽了计算机的应用范围。普通的计算机系统可以处理文字、数据和图形等信息，多媒体计算机除以上类型的信息外，还可以综合处理图像、动画、音频、视频等信息。为了能够处理多种媒体，要求多媒体计算机具有速度快、存储容量大等特点。高档微机为多媒体技术的发展和應用提供了必要的技术手段，多媒体计算机加速了计算机进入社会各个方面和家庭的进程，对计算机技术的发展起到极大的推动作用。

1.1 计算机中的数制和编码

计算机最基本的功能是进行数的计算和处理。在计算机中，数是以电子器件的物理状态来表示的。由于用电子器件表示两种状态更为方便和可靠，因此，计算机一般采用二进制数制，所有的字母、符号也都要用二进制编码来表示。

1.1.1 无符号数的表示及运算

一、无符号数的表示

1. 十进制数的表示

十进制计数的特点是：

(1) 用 10 个数字符号 0, 1, 2, ..., 9 表示数值。

(2) 逢 10 进位。

任何一个十进制数 N_D 可以表示为：

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

其中, m 表示 N_D 的小数位的位数; n 表示 N_D 的整数位的位数; D_i 为十进制数字符号 0~9。

例如：

$$157.3D = 1 \times 10^2 + 5 \times 10^1 + 7 \times 10^0 + 3 \times 10^{-1}$$

后缀 D 表示这个数是一个十进制数 (Decimal), 通常省略十进制数的后缀 D。

2. 二进制数的表示

二进制计数的特点是：

(1) 用两个数字符号 0, 1 表示数值。

(2) 逢 2 进位。

任何一个二进制数 N_B 可以表示为：

$$N_B = \sum_{i=-m}^{n-1} B_i \times 2^i \quad (1.1.1)$$

其中, m 和 n 分别表示 N_B 的小数位和整数位的位数; B_i 为二进制数字符号 0~1。例如：

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

后缀 B 表示这个数是一个二进制数 (Binary)。

3. 十六进制数的表示

十六进制计数的特点是：

(1) 用 16 个数字符号 0, 1, 2, ..., 9, A, B, C, D, E, F 表示数值。其中 A~F 依次表示十进制数值 10~15。

(2) 逢 16 进位。

任何一个十六进制数 N_H 可以表示为：

$$N_H = \sum_{i=-m}^{n-1} H_i \times 16^i \quad (1.1.2)$$

其中, m 和 n 分别表示 N_H 的小数位和整数位的位数; H_i 为十六进制数字符号 0~F。例如：

$$3AB.11H = 3 \times 16^2 + A \times 16^1 + B \times 16^0 + 1 \times 16^{-1} + 1 \times 16^{-2}$$

后缀 H 表示这个数是一个十六进制数 (Hexadecimal)。

二、不同数制之间数的转换

1. 二进制数、十六进制数转换为十进制数

任何一个二进制数 N_B 可按 (1.1.1) 式求和计算转换成等值的十进制数。例如, $1101.1B = 13.5D$ 。

任何一个十六进制数 N_H 可按 (1.1.2) 式求和计算转换成等值的十进制数。例如, $3AB.11H = 939.0664D$ 。

2. 十进制数转换为二进制数、十六进制数

(1) 十进制整数转换为二进制整数、十六进制整数

由 (1.1.1) 式可知, 如果将一个十进制整数表示成：

$$\sum_{i=0}^{n-1} B_i \times 2^i = B_{n-1} \times 2^{n-1} + \dots + B_1 \times 2^1 + B_0 \times 2^0$$

那么, 转换成二进制整数 $B_{n-1}B_{n-2}\dots B_1B_0$ 。

采用“除 2 取余”方法将十进制整数转换为二进制整数, 即: 用 2 连续去除十进制整数, 直至商等于零为止。逆序排列余数即得相应的二进制整数。

例 1.1.1 把十进制数 13D 转换成二进制数。

用“除 2 取余”方法的转换过程为:

$2 \overline{) 13}$	
$\underline{2 \times 6}$	余数为 1, $B_0=1$
$\underline{2 \times 3}$	余数为 0, $B_1=0$
$\underline{2 \times 1}$	余数为 1, $B_2=1$
0	余数为 1, $B_3=1$

得 $13D = B_3B_2B_1B_0 = 1101B$

类似, 用“除 16 取余”的方法可将十进制整数转换成十六进制整数, 即: 用 16 连续去除十进制整数, 直至商等于零为止。逆序排列余数即得相应的十六进制整数。

例 1.1.2 把十进制数 939D 转换成十六进制数。

用“除 16 取余”方法的转换过程为:

$16 \overline{) 939}$	
$\underline{16 \times 58}$	余数为 B, $H_0=B$
$\underline{16 \times 3}$	余数为 A, $H_1=A$
0	余数为 3, $H_2=3$

得 $939D = H_2H_1H_0 = 3ABH$

(2) 十进制小数转换为二进制小数、十六进制小数

由 (1.1.1) 式可知, 如果将一个十进制小数表示成:

$$\sum_{i=-m}^{-1} B_i \times 2^i = B_{-1} \times 2^{-1} + B_{-2} \times 2^{-2} + \dots + B_{-m} \times 2^{-m}$$

那么, 转换成二进制小数 $0.B_{-1}B_{-2}\dots B_{-m}$ 。

采用“乘 2 取整”方法将十进制小数转换为二进制小数, 即: 连续用 2 去乘十进制小数, 直至乘积的小数部分等于零为止。顺序排列每次乘积的整数部分即得相应的二进制小数。若连续 m 次乘 2 取整后乘积的小数部分仍不为零, 则得到小数点后有 m 位的二进制小数是十进制小数的近似值。

例 1.1.3 把十进制数 0.8125D 转换成二进制数。

用“乘 2 取整”方法的转换过程为:

$0.8125 \times 2 = 1.625$	整数为 1, $B_{-1}=1$
$0.625 \times 2 = 1.25$	整数为 1, $B_{-2}=1$
$0.25 \times 2 = 0.5$	整数为 0, $B_{-3}=0$
$0.5 \times 2 = 1.0$	整数为 1, $B_{-4}=1$

得 $0.8125D = 0.B_{-1}B_{-2}B_{-3}B_{-4} = 0.1101B$

类似, 用“乘 16 取整”的方法可将十进制小数转换为十六进制小数。

例 1.1.4 把十进制数 0.32D 转换成十六进制数, 转换精度为小数点后保留 3 位。

用“乘 16 取整”方法的转换过程为:

$$0.32 \times 16 = 5.12 \quad \text{整数为 } 5, H_{-1} = 5$$

$$0.12 \times 16 = 1.92 \quad \text{整数为 } 1, H_{-2} = 1$$

$$0.92 \times 16 = 14.72 \quad \text{整数为 } E, H_{-3} = E$$

$$\text{得 } 0.32D = 0.H_{-1}H_{-2}H_{-3} = 0.51EH$$

一个具有整数和小数部分的十进制数，在转换为二进制数或十六进制数时，只要把它的整数和小数部分分别转换，然后用小数点把转换结果连起来就可以了。例如， $13.8125D = 1101.1101B$ ， $939.32D = 3AB.51EH$ 。

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 位二进制数用 1 位十六进制数表示即可。例如：

$$1101110.01011B = 0110\ 1110.0101\ 1000B = 6E.58H$$

二进制数书写冗长易错，转换为十六进制数也很方便，因此一般用十六进制表示数。

三、二进制数的运算

1. 二进制数的算术运算

(1) 二进制加法运算。二进制加法运算规则为：

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=0 \quad (\text{进位 } 1)$$

(2) 二进制减法运算。二进制减法运算规则为：

$$0-0=0$$

$$1-1=0$$

$$1-0=1$$

$$0-1=1 \quad (\text{有借位})$$

(3) 二进制乘法运算。二进制乘法运算规则为：

$$0 \times 0 = 1 \times 0 = 0 \times 1 = 0$$

$$1 \times 1 = 1$$

(4) 二进制除法运算。二进制除法是乘法的逆运算。

2. 二进制数的逻辑运算

一个逻辑变量只能取值 0 或 1，逻辑运算的结果也只能为 0 或 1，因此，逻辑运算在任何时候都不发生进位。下面介绍 4 种逻辑运算，运算规则见表 1.1.1。

(1) 与运算 (AND)。与运算又称为逻辑乘，运算符号可用 “ $\cdot$ ” 或 “ $\wedge$ ” 表示。

由表 1.1.1 可知，只有当 A、B 两个变量皆为 1 时，与运算结果才为 1；否则，为 0。

(2) 或运算 (OR)。或运算又称为逻辑加，运算符号可用 “ $+$ ” 或 “ $\vee$ ” 表示。

由表 1.1.1 可知，A、B 两个变量中只要有一个为 1，或运算结果就为 1；否则，为 0。

(3) 非运算 (NOT)。变量 A 的非运算结果用 $\bar{A}$ 表示。若 $A=0$ ，则 $\bar{A}=1$ ；若 $A=1$ ，则 $\bar{A}=0$ 。

(4) 异或运算 (XOR)。异或运算又称为按位加，运算符号用 “ $\oplus$ ” 表示。

由表 1.1.1 可知，A、B 两个变量取值相同时，异或运算结果为 0；取值相异时，结果为 1。或者说，一个逻辑变量和 0 异和，结果同该变量的值相同；一个变量和 1 异和，结果为该

变量的值取反。

表 1.1.1 逻辑运算规则

A	B	$A \wedge B$	$A \vee B$	$\bar{A}$	$A \oplus B$	A	B	$A \wedge B$	$A \vee B$	$\bar{A}$	$A \oplus B$
0	0	0	0	1	0	1	0	0	1	0	1
0	1	0	1	1	1	1	1	1	1	0	0

例 1.1.5 已知 $A=11110101B$, $B=00110000B$, 求 $\bar{A}$, $A \wedge B$, $A \vee B$, $A \oplus B$ 。

注意到逻辑运算是按位进行的, 根据运算规则可得

$$\bar{A}=00001010B, A \wedge B=00110000B,$$

$$A \vee B=11110101B, A \oplus B=11000101B.$$

1.1.2 带符号数的表示及运算

一、带符号数的表示

数有正数也有负数, 带有正、负号的数在机器中如何表示和运算呢? 数的符号在计算机中也要用二进制数表示, 通常用二进制数的最高位表示数的符号, 这样的数称为机器数, 它的数值称为机器数的真值。机器数的表示有原码、反码、补码三种表示法。

1. 原码

数 x 的原码记为 $[x]_{\text{原}}$, 原码的定义为:

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

其中, n 为机器字长。

例如, 当机器字长 $n=8$ 时, 有:

$$[+1]_{\text{原}}=00000001 \quad [+127]_{\text{原}}=01111111$$

$$[-1]_{\text{原}}=10000001 \quad [-127]_{\text{原}}=11111111$$

原码表示有以下特点:

(1) 最高位为符号位。正数的符号位是 0; 负数的符号位是 1。其余 $n-1$ 位表示数的绝对值。

(2) 原码表示数的范围是 $-(2^{n-1}-1) \sim +(2^{n-1}-1)$ 。

2. 反码

数 x 的反码记为 $[x]_{\text{反}}$ 。反码的定义为:

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

其中, n 为机器字长。

例如, 当机器字长 $n=8$ 时, 有:

$$[+1]_{\text{反}}=00000001 \quad [+127]_{\text{反}}=01111111$$

$$[-1]_{\text{反}}=11111110 \quad [-127]_{\text{反}}=10000000$$

反码表示有以下特点:

(1) 最高位为符号位。正数的符号位是 0; 负数的符号位是 1。

(2) 反码表示数的范围是 $-(2^{n-1}-1) \sim +(2^{n-1}-1)$ 。

(3) 正数的反码与原码相同。负数的反码可由其相应正数的反码按位取反 (连同符号

位) 得到。

3. 补码

数 x 的补码记为 $[x]_{\text{补}}$ 。补码的定义为:

$$[x]_{\text{补}} = \begin{cases} x & 0 \leq x \leq 2^{n-1} - 1 \\ 2^n - |x| & -2^{n-1} \leq x < 0 \end{cases}$$

其中, n 为机器字长。

例如, 当机器字长 $n=8$ 时, 有:

$$[+1]_{\text{补}} = 00000001 \quad [+127]_{\text{补}} = 01111111$$

$$[-1]_{\text{补}} = 11111111 \quad [-127]_{\text{补}} = 10000001$$

补码表示有以下特点:

- (1) 最高位为符号位。正数的符号位是 0; 负数的符号位是 1。
- (2) 补码表示数的范围是 $-2^{n-1} \sim + (2^{n-1} - 1)$ 。
- (3) 正数的补码与原码、反码相同。负数的补码可由其反码加 1 得到。也就是说, 负数的补码可由其相应正数的补码按位取反 (连同符号位) 再加 1 得到。
- (4) 正数的补码就是正数本身, 因此, 正数的真值 $x = [x]_{\text{补}}$ 。由负数的补码得到这个负数的真值的绝对值的方法是对这个负数的补码再求补, 即 $|x| = [[x]_{\text{补}}]_{\text{补}}$ 。

二、补码的运算

1. 补码加法

补码加法的规则是:

$$[x+y]_{\text{补}} = [x]_{\text{补}} + [y]_{\text{补}}$$

其中, x, y 为正负数皆可。

用一个例子来说明补码加法规则。

例 1.1.6 已知 x 和 y , 运用补码加法规则计算 $x+y$ 。

$$(1) x = +66 \quad y = +51 \quad (2) x = +66 \quad y = -51$$

$$(3) x = -66 \quad y = +51 \quad (4) x = -66 \quad y = -51$$

先写出 x 和 y 的补码, 有:

$$[+51]_{\text{补}} = 00110011 \quad [+66]_{\text{补}} = 01000010$$

$$[-51]_{\text{补}} = 11001101 \quad [-66]_{\text{补}} = 10111110$$

分别进行十进制加法和二进制补码加法, 以验证补码加法规则。

(1)	+ 66	01000010	$[+66]_{\text{补}}$
	+) + 51	+) 00110011	$[+51]_{\text{补}}$
	+ 117	01110101	$[+117]_{\text{补}}$
(2)	+ 66	01000010	$[+66]_{\text{补}}$
	+) - 51	+) 11001101	$[-51]_{\text{补}}$
	+ 15	①00001111	$[+15]_{\text{补}}$
(3)	- 66	10111110	$[-66]_{\text{补}}$
	+) + 51	+) 00110011	$[+51]_{\text{补}}$
	- 15	11110001	$[-15]_{\text{补}}$
(4)	- 66	10111110	$[-66]_{\text{补}}$

$$\begin{array}{r}
 +) \underline{-51} \\
 -117
 \end{array}
 \quad
 \begin{array}{r}
 +) \underline{11001101} \\
 \textcircled{1}10001011
 \end{array}
 \quad
 \begin{array}{l}
 [-51]_{\text{补}} \\
 [-117]_{\text{补}}
 \end{array}$$

可见，两个数的补码之和等于这两个数的和的补码。当然，和数不能超出补码的表示范围，否则，结果就不正确了。这种情况称为溢出。

本例中的 (2)、(4) 发生符号位向更高位的进位，由于机器字长的限制，这种进位会自动的丢失，不影响运算结果的正确。

2. 补码减法

补码减法的规则是：

$$[x-y]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}}$$

仍用一个例子来说明补码减法规则。

例 1.1.7 已知 x 和 y ，运用补码减法规则计算 $x-y$ 。

- (1) $x=+66$ $y=+51$ (2) $x=+66$ $y=-51$
 (3) $x=+51$ $y=+66$ (4) $x=-51$ $y=-66$

先写出 x 和 y 的补码，有：

$$\begin{array}{ll}
 [+51]_{\text{补}} = 00110011 & [+66]_{\text{补}} = 01000010 \\
 [-51]_{\text{补}} = 11001101 & [-66]_{\text{补}} = 10111110
 \end{array}$$

分别进行十进制减法和按补码减法规则用补码的相加来计算

(1)	$+66$	01000010	$[+66]_{\text{补}}$
	$-) \underline{+51}$	$+) \underline{11001101}$	$[-51]_{\text{补}}$
	$+15$	$\textcircled{1}00001111$	$[+15]_{\text{补}}$
(2)	$+66$	01000010	$[+66]_{\text{补}}$
	$-) \underline{-51}$	$+) \underline{00110011}$	$[+51]_{\text{补}}$
	$+117$	01110101	$[+117]_{\text{补}}$
(3)	$+51$	00110011	$[+51]_{\text{补}}$
	$-) \underline{+66}$	$+) \underline{10111110}$	$[-66]_{\text{补}}$
	-15	11110001	$[-15]_{\text{补}}$
(4)	-51	11001101	$[-51]_{\text{补}}$
	$-) \underline{-66}$	$+) \underline{01000010}$	$[+66]_{\text{补}}$
	$+15$	$\textcircled{1}00001111$	$[+15]_{\text{补}}$

本例中的 (1)、(4) 发生符号位向更高位的进位，同样会自动丢失而不影响运算结果的正确。

补码减法规则指出，在计算机中，利用补码表示可将减法运算用加法运算来求解，从而省去减法器，简化硬件电路。

1.1.3 二进制编码

在计算机中表示的数、字母、符号等等都要以特定的二进制码来表示，这就是二进制编码。

一、二进制编码的十进制数 (BCD 码)

尽管二进制数容易实现、可靠，且运算规则简单，但是人们习惯采用十进制数。因此，计

计算机的输入和输出通常是十进制数。当然，十进制数在计算机中需要用二进制编码表示，这种编码有多种形式，BCD (Binary-Coded Decimal) 码是一种常用的编码。

BCD 码有两种形式，即压缩 BCD 码和非压缩 BCD 码。

1. 压缩 BCD 码

压缩 BCD 码将一位十进制数用 4 位二进制数表示，即 0~9 分别用 0000~1001 表示。一个字节表示两位十进制数，例如，10010110B 表示十进制数 96D。

BCD 码与十进制数之间的转换是很容易的。例如，一个 BCD 编码 01111000.00010100B 很方便地转换为十进制数 78.14D。

BCD 码与二进制数之间的转换是不直接的，要先经过十进制。即：BCD 码先转换成十进制数，然后再转换成二进制数。反之亦然。

2. 非压缩 BCD 码

非压缩 BCD 码用一个字节表示一位十进制数，高 4 位总是 0000，低 4 位用 0000~1001 表示 0~9。例如，00000110B 表示十进制数 6。

二、字母和符号的编码 (ASCII 码)

字母和各种字符也必须按特定的规定用二进制编码才能在计算机中表示和处理。目前，在微型机中最普遍的是采用 ASCII (American Standard Code for Information Interchange) 码，即美国标准信息交换码。ASCII 码用 8 位二进制对字符进行编码，故可表示 256 个符号。ASCII 码表见附录一。

1.2 微型计算机系统的组成

计算机系统由硬件和软件组成。微型计算机的硬件由主机和外部设备组成，主机包括 CPU、存储器和输入/输出接口三个主要部分。软件可分为系统软件和应用软件。

1.2.1 微型计算机硬件

一、CPU

中央处理单元 CPU (Central Processing Unit) 是微型机的核心芯片，芯片上有运算器、控制器和寄存器。运算器也称为算术逻辑单元 ALU (Arithmetic and Logic Unit)。运算器的功能是完成数据的算术和逻辑运算。控制器一般由指令寄存器、指令译码器和控制电路组成，控制器根据指令的要求对各部件发出相应的控制信息，使各部件协调工作。CPU 内部寄存器用来暂存运算器要使用的数据和中间结果。

二、存储器

存储器 (Memory) 又称为主存或内存，是计算机的存储和记忆装置，用来存放数据和程序，数据和程序是以二进制数的形式存放在内存中。把一个 8 位二进制数称为一个字节 (Byte)，内存容量通常以字节为单位， $2^{10}=1024$ 字节记为 1KB， 2^{20} 字节记为 1MB， 2^{30} 字节记为 1GB。

内存由若干内存单元组成，通常每个内存单元存放一个字节的二进制信息。每个内存单元都有一个各自不同的地址，CPU 按照地址找到相应的内存单元，对内存单元中的内容进行读操作或写操作。读操作是 CPU 将内存单元中的内容读入 CPU 内，写操作是 CPU 将一个字节的二进制信息传送到内存单元保存。显然，写操作改变了被写入的内存单元的原有内容，读

操作则在该内存单元的内容被读后仍然保持。

内存通常由两部分组成，这就是随机存储器 RAM (Random Access Memory) 和只读存储器 ROM (Read Only Memory)。RAM 可以被 CPU 随机地读写，即 CPU 可对 RAM 进行读操作，也可进行写操作。RAM 用于存放用户装入的程序、数据及部分系统信息。当机器断电后，RAM 中所存信息消失。ROM 只能由 CPU 进行读操作，不能进行写操作，故而称为只读存储器。机器断电后，ROM 中所存信息仍然保留。ROM 中的信息是事先通过专用设备写入的，如基本 I/O 程序，BASIC 解释程序等。

三、输入/输出设备和输入/输出接口

常用的输入设备有键盘、鼠标器、数字化仪、扫描仪、A/D 变换器等、常用的输出设备有显示器、打印机、绘图仪等，磁盘、磁带既是输入设备，又是输出设备。

外设的种类繁多，有机械式、电动式、电子式等，它们的工作速度同 CPU 相比要低得多。外设处理的信息有数字量、模拟量、开关量等，但是，微型机只能处理数字量。另外，外设同微型机工作的逻辑时序也可能不一致。鉴于上述原因，微型机同外设之间的连接及信息的交换不能直接进行。根据不同外设的需要设计相应的接口电路作为微型机同外设之间连接和交换信息的桥梁，这种接口电路称为 I/O 接口，又称为 I/O 适配器 (I/O Adapter)。

综上所述，微型机硬件主要由 CPU、内存、I/O 接口和 I/O 设备组成。微型机各部件之间是用称为系统总线的公共导线连接的，根据传送信息的不同，可将系统总线分为三组，分别是地址总线 AB (Address Bus)、数据总线 DB (Data Bus)、控制总线 CB (Control Bus)。AB 传送 CPU 发出的地址信息，是单向总线。DB 传送数据信息，CPU 既可通过 DB 从内存或输入设备读入数据，又可通过 DB 将 CPU 内部的数据送至内存或输出设备，因此，DB 是双向总线。CB 传送控制信息，控制信息有的是 CPU 向内存或向外设发出的信息，有的是外设发送给 CPU 的信息，因此，CB 中的每一根线的传送方向是一定的。微型机的系统结构可用图 1.2.1 表示，图中，CB 表示控制总线组，用双向表示。

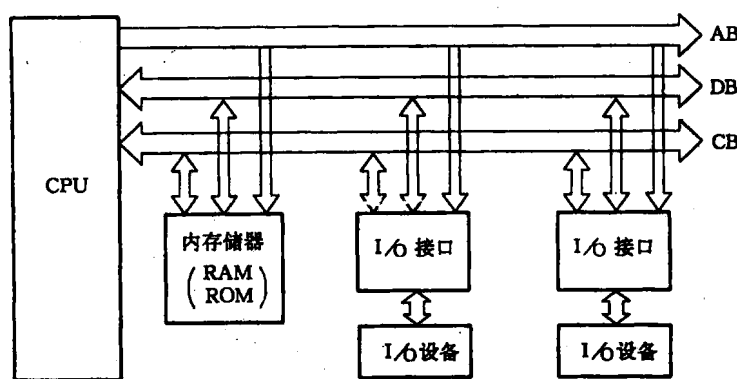


图 1.2.1 微型机的系统结构框图

1.2.2 微型计算机软件

微型计算机软件包括系统软件和应用软件。应用软件是用户为解决各种实际问题而编制的各种程序，这些程序在系统软件和硬件的支持下编制和运行。系统软件主要包括操作系统