全国高等教育自学考试教材（机电一体化工程专业）

微机原理与接口技术

应锦春主编

中国科学技术出版社

全国高等教育自学考试教材

微机原理与接口技术

赵长德　刘学斌　游珂　编

中国科学技术出版社

内 容 提 要

本书共分九章。选择最通用的Z 80全面讲述了其组成原理、指令系统、汇编语言程序设计、半导体存贮器、输入输出方式、中断系统；并从应用的角度讲述了常用的接口芯片和接口技术；还介绍了ＳＴＤ总线标准，ＩＥＥＥ-488，ＲＳ-232Ｃ接口标准。本书根据机电一体化应用的需要，还讲述了ＭＣＳ-51单片机、可编程控制器（ＰＣ）的原理和应用，最后介绍了微型机的典型应用实例。本书为高等教育机电一体化专业自学考试教材，也适于高等学校机械类有关专业作为教学参考书，同时也可供在职工程技术人员参考。

全国高等教育自学考试教材
微机原理与接口技术
赵长德　刘学斌　游珂　编
*
中国科学技术出版社出版（北京海淀区学院南路86号）
新华书店北京发行所发行　各地新华书店经售
北京星月印刷厂印刷
*
开本：787×1092 毫米 1/16 印张：25 字数：480 千字
1990年9月第1版　1991年7月第2次印刷
印数：1—20,000册　定价：28.00元
ISBN 7-5046-0220-5/TP·11

出　版　前　言

高等教育自学考试教材建设是高等教育自学考试工作的一项基本建设。经国家教育委员会同意，我们拟有计划、有步骤地组织编写一些高等教育自学考试教材，以满足社会自学和适应考试的需要。《微机原理与接口技术》是为高等教育自学考试机电一体化工程专业组编的一套教材中的一种。这本教材根据专业考试计划，从造就和选拔人才的需要出发，按照全国颁布的《微机原理与接口技术课程自学考试大纲》的要求，结合自学考试的特点，组织高等院校一些专家学者集体编写而成的。

机电一体化工程专业《微机原理与接口技术》自学考试教材，是供个人自学、社会助学和国家考试使用的。无疑也适用于其他相同专业方面的学习需要。现经审定同意予以出版发行。我们相信，随着高教自学考试教材的陆续出版，必将对我国高等教育事业的发展，保证自学考试的质量起到积极的促进作用。

编写高等教育自学考试教材是一种新的尝试，希望得到社会各方面的关怀和支持，使它在使用中不断提高和日臻完善。

全国高等教育自学考试指导委员会

1990年5月

前　言

近年来，计算机科学与技术取得了惊人的飞速发展。特别是微型计算机（简称微机），由于其具备成本低、体积小、可靠性高等特点，因此，才真正使计算机渗透和占领各个技术领域，为计算机的普及和应用开创了现实的可能性。

机电一体化就是在这种形势下产生的一种复合化技术。它是机械学、微电子学和信息科学三者进行有机地结合而构成的一种优化技术。应用这种技术生产出来的产品，即机电一体化的产品，其机械结构大为简化，并具有节能、自动化操作及多功能的特点。因此机电一体化是当今机械工业技术和产品发展的主要趋向之一。微机原理与接口技术课程就是为适应这一要求而开设的。

微型计算机在机械电子工业中的检测、控制方面的应用非常广泛，在数控技术，传感技术，智能化仪器仪表，数据采集系统，工业机器人，自动生产线中都大量地应用着不同型号的单片计算机、单板机、ＳＴＤ总线工业控制机，以及其它各种微机系统。可以说，微型计算机的开发和应用，是现有企业进行技术改造和新产品设计的技术手段之一。

本书是根据国家教委批准试行的机电一体化工程专业本科段“微机原理与接口技术”课程自学考试大纲而编写的。

本书选择国内最通用的八位机Ｚ80，讲述其组成原理、指令系统、汇编语言程序设计，常用接口芯片，Ｚ80中断系统以及常用接口技术，在此基础上讲述了51系列单片机的原理和组成，简单地介绍了其指令系统和应用，讲述了普遍应用的可编程控制器（ＰＣ）的原理和应用，为学习机电一体化技术打下基础。书中还列举了一些应用实例，可供读者参考。本书力求深入浅出，适合自学，偏难的内容用小字以便今后参考。为配合本书的学习，还编写了实验指导书。

本书第一、二、五、六、七、九章由清华大学赵长德副教授编写，第三、四章由刘学斌副教授编写，第八章由机电部北京自动化所游珂高级工程师编写。本书由赵长德主编，北京理工大学焦振学副教授主审。在编写过程中还得到北京工业大学陈锡璞副教授，机电部自动化所陈瑜、余禄高级工程师，北京机械工业管理学院朱耀祥教授、杨庆东讲师，北京理工大学陆士毅副教授，清华大学沈钊教授的指导和帮助，清华大学仪器系微机实验室曹志锦同志参加了绘图工作，在此一并表示衷心的感谢。

由于编写时间仓促，错误和不足之处在所难免，敬请读者批评指正。

编　者

1990年3月

目　　录

第一章　微型计算机基础知识

第一节　微型计算机的发展和应用

自从1946年世界上第一台电子计算机问世以来，计算机科学和技术获得了日新月异的飞速发展。一般可分为下列几个发展阶段：

第一代是电子管计算机，发展年代大约为1946～1957年。这一阶段计算机的逻辑元件采用电子管，存贮器采用磁鼓和磁芯。软件主要使用机器语言，开始使用汇编语言。当时这种计算机还是庞然大物，例如第一台电子管计算机，占地150平方米，重30吨，耗电150千瓦。可其字长只有12位，加法运算速度5000次/秒，被称为电子数字积分器和计算器（ENIAC）。虽然它与今天的微型计算机相比不可同日而语，但它却奠定了计算机科学和技术的发展基础。

第二代是晶体管计算机，其发展年代大约为1958～1964年。计算机的逻辑元件为晶体管，主存贮器仍用磁芯，外存贮器已开始用磁盘。软件也有较大发展，出现了各种高级语言。计算机除了用于各种事务的数据处理外，也开始用于工业控制。

第三代是固体组件计算机，其逻辑元件开始采用中小规模集成电路，发展年代为1965～1971年。这一期间，小型计算机也随着集成电路规模的增大而迅速发展起来，操作系统，会话式高级语言等软件发展更快，计算机开始广泛应用。

第四代是在大规模集成电路高速发展起来之后的产物，即微型机和巨型机同时得到发展。这是1972年以后发展起来的。所谓大规模集成电路（LSI）或超大规模集成电路（VLSI）是指在单个硅片上集成1000～20000个甚至集成6万～10万个晶体管的集成电路。由于LSI的体积小、耗电少、可靠性高，因而促使微型计算机以很快的速度得到发展，现在广泛使用单板机，单片机，各种型号的个人计算机。同时，相继出现用于科学计算和尖端技术中的巨型机。

第五代计算机是人工智能计算机，它是综合了计算机科学和控制论而发展的一门新技术，它能模拟人的智能，如识别图形、语言、物体等，它将对社会的发展带来不可估量的影响，目前第五代计算机仍处于研究之中。

本书主要讲述微型计算机，它的特点在于中央处理器（CPU）是集成在一小块硅片上，而大型或小型计算机的CPU则是由相当多的集成电路组成的。

微型计算机除了配有单片微处理器外，还有用LSI技术制成的主存贮器和输入/输出接口电路，这三者之间采用总线结构联系起来。如果再配上相应的外部设备，如屏幕显示器（CRT）、键盘、打印机和磁盘，以及配套软件，就构成了完整的微型机系统。

微型计算机是70年代以后开始发展起来的，到现在不过20年，但也经历了以下几个发展阶段。

1．第一代微处理器　1971年美国Intel公司生产了4004芯片，它本来只是为高级计算器设计的，但生产出来后却获得了意外的成功。经过改进，生产了4位的微处理器4040，1972年生产出8位微处理器8008。这就是第一代微处理器。

2．第二代微处理器　其发展年代为1973～1974年。当时，由于第一代微处理器的出现，引起各厂家极大兴趣，于是又先后推出 Intel 8080，Motorola 6800，Signetics 2650，等等。这些微处理器已开始用于控制和仪表中。

3．第三代微处理器　其发展年代为1975～1977年。它的标志是出现了集成度更高的八位微处理器，其代表性产品为Zilog公司的Z80和Intel公司的8085等。另外也出现了一系列单片微计算机，Motorola公司经过改进也推出了MC 6809，此时，一个硅片上可以容纳一万个以上的晶体管，16K位和64K位的存贮器已生产出来，进入了超大规模集成电路(V-LSI)的时代。

4．第四代微处理器　是从1978年开始生产的可与中档小型机相比拟的16位微处理器，如Intel的8086，Zilog的Z8000以及Motorola的M68000。同时也出现了16位单片机。IBM公司利用8088，80286CPU研制成IBM-PC/XT及IBM-PC/AT个人计算机，使微型机的应用又向前迈进了一大步，其应用已深入到各个领域。

5．第五代微处理器　是从1981年以后发展起来的。其典型产品为Intel公司先后推出的iAPX 43201和80386，Motorola公司推出的MC 68010和MC 68020。这些微处理器字长达32位，集成度在10～20万器件/片，时钟频率高达16MHz。以这种高档微处理器为中心构成的高档微型计算机系统，已完全达到了传统的超级小型机水平。

微型机的出现和发展开拓了计算机广泛应用和普及的新纪元。在机械工业技术改造、机电一体化工程方面更是不可缺少的技术手段。例如机床数控系统，工业机器人，传感技术，智能化仪表，工业对象的遥控遥测，故障断诊及在工业过程的实时控制、数据采集系统中无一不应用微机技术。现在工业企业中已使用大量的微机系统，及可编程序控制器（PC）、STD总线工业控制机、单板机、单片机。本书主要从应用的角度讲述微型机工作的原理和接口技术。至于在管理上和办公室自动化方面的应用不在本书进行讨论。

第二节　计算机的数和编码系统

本节重点介绍进位计数制，不同进位计数制之间的相互转换以及常用的各种编码系统。

一、进位计数制

人们在日常生活中，采用多种进制的数字系统。最常用的是十进制。例如

$$1991 = 1\times10^3 + 9\times10^2 + 9\times10^1 + 1\times10^0$$

任何一个十进制数都可表示为

$$X = x_m\times10^m + \cdots + x_0 +$$

$$x_{-1}\times10^{-1} + \cdots + x_{-n}\times10^{-n} = \sum_{i=-n}^{m} x_i\times10^i$$

其中，10称为十进制的基。而所在数位 i 的权为 10^i。对任意进制来说，所谓基数 J，就是用来表示数时可以选用的不同数字的个数。所以十进制的 J 为10，用来表示数时可以选用的数 x_i 为0～9十个数字，且逢十进1。而小数点左面各位的权依次为 $J^0, J^1, \cdots, J^m$，小数点右面的权依次为 $J^{-1}, J^{-2}, \cdots, J^{-n}$。而且，小数点左移一位等于减小 J 倍，小数点右移一位等于增大 J 倍。

在计算机内如果要使用十进制数时，每位数就要用0～9共十个数字，电路上难以实现。

而仅有两个不同的稳定状态，且可相互转换的器件，就可表示一位二进制数，所以在计算机内都使用二进制。其特点是：基为2，只需要两个数字符号0和1，逢二进位。

对任意二进制数X都可展开为

$$X=x_m\times 2^m+\cdots+x_0\times 2^0+$$

$$x_{-1}\times 2^{-1}+\cdots+x_{-n}\times 2^{-n}=\sum_{i=-n}^{m}x_i\times 2^i$$

第i位数是0或1，对应的数值为$x_i\times 2^i$，并称2^i为第i位的权；

例如二进制数1101.1011可展开为

$$1101.1011=1\times 2^3+1\times 2^2+0\times 2^1+1\times 2^0$$
$$+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}+1\times 2^{-4}$$

因为使用二进制需用较多的位数，书写很繁，所以还经常使用十六进制，它的特点是：基为16，使用16个数字符号，它们是0～9，A，B，C，D，E，F；对于十六进制的加法计算是逢16进位。

任意一个十六进制数都可表示为

$$X=x_m\times 16^m+\cdots+x_0\times 16^0+$$

$$x_{-1}\times 16^{-1}+\cdots+x_{-n}\times 16^{-n}=\sum_{i=-n}^{m}x_i\times 16^i$$

第i位数是上述16个符号之一，对应的数值为$x_i\times 16^i$，并称16^i为第i位的权。

为了避免使用多种进位制的混乱，使用后缀表明数的进制：

后缀B——表示二进制；

后缀H——表示十六进制；

后缀D——表示十进制（也可不加后缀）。例如：

10011011 B——使用二进制

9 B H——使用十六进制

155 D——使用十进制

这些数都表示同一数值，即十进制的155，只是所用的进制不同。

二、各种进制数的转换

尽管有不同的进制，但在计算机中的数仍然只能用二进制表示，十六进制是适应于读写方便的需要，而十进制则是日常生活所必需的。因此，就要掌握各种进制的转换关系。

由于$2^4=16$，一位十六进制数可用四位二进制数表示，它们之间存在着直接而又唯一的对应关系，如表1.1所示。可见二进制数和十六进制数之间的转换是十分简便的。

（一）二进制和十六进制数的相互转换

1．十六进制转换为二进制　不论是十六进制的整数或小数，只要把每一位十六进制的数用相应的四位二进制数代替，就可以转换为二进制数。例如

9	B	.	A	6
↓	↓		↓	↓
1001	1011		1010	0110

即　9 B．A 6 H＝10011011．10100110 B

表 1.1　二进制，十进制，十六进制数码对照表

十　进　制	十六进制	二　进　制
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	10000

2．二进制转换为十六进制　这种转换可分两步进行。对整数部分，从小数点向左数，每四位二进制数一组，最后不足四位的前面补 0。对小数部分，从小数点向右数，每四位一组，最后不足四位的后面补 0。然后把每四位二进制数用相应的16进制数代替，即可转换为16进制数。例如

1011　0111．0101　0100

↓　↓　↓　↓

B　7　5　4

即10110111．010101 B＝B7.54H

（二）二进制、十六进制与十进制数的相互转换

1．十六进制数转换为十进制数　对给定的十六进制数，只要按前述公式展开，即可得到对应的十进制数。例如

$$\begin{aligned}\text{ABCDH} &= 10\times16^3+11\times16^2+12\times16^1+13\times16^0\\ &= 10\times4096+11\times256+12\times16+13\\ &= 43981\text{D}\end{aligned}$$

八位二进制数称为一个字节，若用十六进制表示，其数的范围为00H～FFH，而FFH＝255。

16位二进制数(两个字节)可用4位十六进制数表示，其最大的数为FFFFH＝65535D 。

2．二进制数转换为十进制数　对所给的二进制数，只要按前述的公式展开，即可得到对应的十进制数。例如

$$\begin{aligned}1011.1010\text{B} &= 1\times2^3+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-3}\\ &= 11.625\end{aligned}$$

对给定的一个字节或两个字节的二进制数也可先化成十六进制数，再转换成十进制数。例如

$$\begin{aligned}1011101001110101\text{B} &= \text{BA75H}\\ &= 11\times16^3+10\times16^2+7\times16^1+5\times16^0\\ &= 47733\text{D}\end{aligned}$$

3．十进制整数转换为二进制数　把十进制整数转换为二进制数，一般采用除 2 取余法。例如

$$215\text{D}=x_m\times2^m+\cdots+x_0\times2^0$$

只要决定x_m，x_{m-1}，…，x_1，x_0的值，就可写出二进制数。因$2^0=1$，所以$(215-x_0)$一定是 2 的整数倍，215÷2 所得的余数即为x_0。其转换过程为

$$\begin{aligned}&215\div2=107\ (\text{商}),\ \text{余数}=1=x_0;\\ &107\div2=53\ (\text{商}),\ \text{余数}=1=x_1;\\ &53\div2=26\ (\text{商}),\ \text{余数}=1=x_2;\\ &26\div2=13\ (\text{商}),\ \text{余数}=0=x_3;\\ &13\div2=6\ (\text{商}),\ \text{余数}=1=x_4;\end{aligned}$$

$6 \div 2 = 3$ （商），余数 $= 0 = x_5$；

$3 \div 2 = 1$ （商），余数 $= 1 = x_6$；

$1 \div 2 = 0$ （商），余数 $= 1 = x_7$；商为 0，转换结束。

故215D＝11010111B。

4．十进制整数转换为十六进制数　同转换为二进制数的道理一样，也可采用除16取余法。例如215D转换为十六进制的过程为

$215 \div 16 = 13$ （商），余数 $= 7 = x_0$；

$13 \div 16 = 0$ （商），余数 $= 13 = x_1$；商为 0，转换结束。

故215D＝D7H。通常写成0D7H，D前面的 0 字说明D不是英文字符D而是数字13。

又如12345D的转换过程为

$12345 \div 16 = 771$（商），余数 $= 9 = x_0$；

$771 \div 16 = 48$ （商），余数 $= 3 = x_1$；

$48 \div 16 = 3$ （商），余数 $= 0 = x_2$；

$3 \div 16 = 0$ （商），余数 $= 3 = x_3$，商为0，转换结束。

故12345D＝3039H。再化成二进制，要比除 2 取余法快。

5．十进制纯小数转换成二进制小数　采用乘 2 取整法。设 x 是十进制数的小数部分，步骤为

(1) 取 $i = -1$，x 作为被乘数；

(2) 用2乘 x，取整数部分为 x_i，取其小数部分为新的被乘数 x；

(3) 使 i 减1；

(4) 重复(2)、(3) 步骤，直至小数部分为零，则结束转换。但也有可能小数部分永不为零，可根据计算机的字长或精度要求，在适当的时候结束转换。

例如，把0.625转换成二进制数

2×0.625	
1.25	整数部分 $= 1 = x_{-1}$，
2×0.25	取小数部分为新被乘数；
0.50	整数部分 $= 0 = x_{-2}$，
2×0.50	取小数部分为新被乘数；
1.00	整数部分 $= 1 = x_{-3}$，
0.00	小数部分为零，转换结束。

$a_{-1}\ a_{-2}\ a_{-3}$

故0.625＝0.101B。

分别把整数部分和小数部分转换成二进制，然后合并就可将一个任意十进制数转换成二进制数。例如

$$215.625 = 11010111.101\,B$$

三、带符号数的表示方法

（一）机器数的概念

前面讲的二进制数，没有提到符号问题，实际上是一种无符号数的表示。但在计算机中，数显然可能有正、有负，通常一个有符号数的最高位为符号位，即数的符号在机器中也数码化了。我们把一个数放在计算机中的表示形式叫机器数，而这个数本身就称为这个机器数的真值。一个有符号数，由于编码不同，可有几种机器数。反之，一个机器数，由于解释方

法不同，又可代表几种真值，见表1.2所示。

表 1.2 数的表示法

机器数		真值（十进制）			
二进制数码	16进制表示	无符号数	原 码	反 码	补 码
00000000	00	0	+0	+0	+0
00000001	01	1	+1	+1	+1
00000010	02	2	+2	+2	+2
⋮	⋮	⋮	⋮	⋮	⋮
01111110	7E	126	+126	+126	+126
01111111	7F	127	+127	+127	+127
10000000	80	128	-0	-127	-128
10000001	81	129	-1	-126	-127
⋮	⋮	⋮	⋮	⋮	⋮
11111110	FE	254	-126	-1	-2
11111111	FF	255	-127	-0	-1

更广义地说，在计算机内（存在内存或寄存器中）的数就是机器数，它可以代表无符号数，也可代表有符号数，有时还可代表字符，它究竟代表什么是由编程者确定的，下面通过一些示例将会理解机器数的真正概念。

（二）原码

如上所述，正数的符号位用 0 表示，负数的符号位用 1 表示，符号位之后表示数值的大小，这种表示方法称为原码。如

$$x=+114,\quad [x]_{原}=01110010\,B$$
$$x=-114,\quad [x]_{原}=11110010\,B$$

这里，后面的 7 位表示数值部分，最高位为符号位。如果字长为16位二进制时，那么后面的15位为数值部分。

原码的表示方法很简单，缺点是原码的“0”有两种：

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

另外，当两个异号数相加，就要做减法，为了把减法运算改为加法运算，引进了反码和补码。

（三）反码

正数的反码与原码相同。最高位一定为 0，代表符号。其余位为数值位。

负数的反码其符号位为 1，与原码相同，数值位则将其负数的原码的数值位按位求反。如

$$x=-4,\quad [x]_{反}=11111011\,B$$
$$x=-0,\quad [x]_{反}=11111111\,B$$
$$x=-127,\quad [x]_{反}=10000000\,B$$

所以，反码中“0”仍有两种表示方法。

（四）补码

正数的补码表示与原码相同，即最高位为符号位，用“0”表示正，其余位为数值位。

而负数的补码为其反码，且在最低位加 1 形成。如

$$x=-4,[x]_{补}=[x]_{反}+1=11111100\,B$$

$$x=-127,[\ x\ ]_{补}=[\ x\ ]_{反}+1=10000001\,B$$

$$x=-0,[\ x\ ]_{补}=[\ x\ ]_{反}+1=00000000\,B$$

表1.2给出了不同的机器数，当它作为无符号数、原码、反码、补码时所代表的真值。

八位二进制补码有以下特点：

(1) $[\ +0]_{补}=[-0]_{补}=00000000\,B$；

(2) 8位二进制补码所能表示的数值为+127～-128；

(3) 当求补码的真值时，如最高位为0时，其余7位即为此数的二进制值。但当最高位为“1”时（即负数），需把其余7位求反以后最低位加1，才是它的二进制值。例如

$$[\ x\ ]_{补}=10010100$$

$$x=-(\ 1101011+1)=-1101100=-108D$$

(4) 当负数采用补码时，就可把减法转换为加法。例如

$$x=24-10=14$$

也可用

$$[\ x\ ]_{补}=[24]_{补}+[-10]_{补}$$

而

$$[24]_{补}=00011000$$

$$[-10]_{补}=11110110$$

$$\begin{array}{r} 00011000 \\ +\ 11110110 \\ \hline \boxed{1}\ 00001110 \end{array}$$

自然丢失

故$[\ x\ ]_{补}=00001110$，$x=14$，同减法结果一致。

在日常生活中，有不少补数的例子，如手表，若标准时间为7点，而现在表的实际指示为9点，要拨到7点，一种是倒拨2小时，一种是顺拨10小时，这里的顺拨（+10）与倒拨（-2）对模12互为补数，即

$$9-2=9+10\quad(\ \mathrm{Mod}\ 12)$$

意思是，等号两边同除以12，它们的余数相同，其中12称为模（Mod 12）。

根据同样原理，可得出十进制数的补码。例如一个圆形两位十进制里程表，初始位置0，里程表转一周时为100km，此时又回到初始位置。当往前走1km时里程表指示为01，而往回倒走1km时里程表指示为99，所以99表示-1。或者说-1的补码是99，其模为100。

同理，采用八位二进制时，其模为100H，因此-1的补码是（100H-01H）=FFH，-10的补码是（100H-0AH）=F6H，…，直到-128的补码是（100H-80H）=80H。这可使求补码的计算加快。

现再举例说明原码、补码、反码的求法，例如已知真值（十进制数），求其原码、反码和补码，这些十进制数为+126、-126、+110、-110。

把十进制正数转换成二进制数可用除2取余法，但直接变成16进制更快，书写也简单。上述四个数均在-128～+127之内，都可用8位二进制表示。先求原码，+126=7EH，而-126的原码应为80H+7EH=FEH，因为只需在最高位D_7位改为1。+110=6EH，故-110的原码为EEH，也是在符号位（D_7位）改为1。再求反码、补码。但对于正数来说，其反码=原码=补码。而对于负数来说，可用对应的符号位不动，其余各位按位求反，即可得反码，反码加1即得补码。故-126的反码为81H，补码为82H，而-110的反码为91H，补码为92H。

另一种求补码的办法更为简单，因-1的补码为FFH，即100H-01H=FFH。注意，减1时不够减向高位借1为低位的16，所以求-126的补码可用100H减去+126的原码7EH

得到，即100H－7EH＝82H，再求反码为82H－1＝81H，可将其结果列表如表1.3所示。

表 1.3 十进制数的原码、反码和补码

真值	+126		−126		+110		−110	
原码	01111110	7EH	11111110	FEH	01101110	6EH	11101110	EEH
反码	01111110	7EH	10000001	81H	01101110	6EH	10010001	91H
补码	01111110	7EH	10000010	82H	01101110	6EH	10010010	92H

四、数的定点与浮点表示

在计算机中，数可以用定点和浮点表示。所谓定点表示，就是小数点在数中的位置是固定不变的；所谓浮点表示，就是小数点在数中的位置是浮动的。

（一）数的定点表示

对于任意一个二进制数总可以表示为纯整数（或纯小数）和一个2的整数次幂的乘积。例如，二进制数N可写成

$$N=2^P \times S$$

其中，S为N的尾数，P称为数N的阶码，2称为阶码的底。尾数S表示了数N的全部有效数字，阶码P指明了小数点的位置。而S，P都是用二进制表示的数。

当阶码为固定值时，称这种表示方法为数的定点表示法，这样的数称为定点数。

当$P=0$，且尾数S为纯整数时，这时定点数只能表示整数。如$P=0$，且尾数S为纯小数时，这时定点数只能表示小数。本书中，均约定采用前一种方法。

在计算机中，定点数的表示方法如下：假设一个单元可存放一个8位二进制数，其中最高位称为符号位，其余7位用来表示数的尾数部分。这与前面讲的有符号数的表示方法是一致的。

（二）数的浮点表示

如果阶码可取不同的数值，称这种表示方法为数的浮点表示法，这样的数，称为浮点数。其中阶码P用整数表示，可为正或负。用一位二进制数P_f表示阶码的符号位，当$P_f=0$时，表示阶码为正数；当$P_f=1$时，表示阶码为负数。尾数S中，用S_f表示尾数的符号，$S_f=0$表示尾数为正数；$S_f=1$表示尾数为负数。在计算机中表示形式为

P_f	阶　码	S_f	尾　数

也就是说，在机器中表示一个浮点数，要分为阶码和尾数两个部分。设8位表示阶码，其值为$P=00000011$，尾数部分为8位，$S=00001101$，如S为纯整数，则

$$N=2^3\times 13=104\text{D}$$

如S为纯小数，则

$$N=2^3\times(2^{-4}+2^{-5}+2^{-7})=0.8125\text{D}$$

在工程实际中，浮点运算通常采用4个字节，阶码用一个字节，尾数占3个字节。显然，当S表示纯整数部分，浮点数的范围为

$$+2^{127}\times(2^{23}-1)\sim-2^{127}\times(2^{23}-1)$$

可见，浮点数的范围远远超过定点数，运算精度也相应提高。缺点是浮点数的运算远比定点数复杂，相应的程序也很不容易编制，因此除非必要时，一般还是用定点数。

五、二进制编码的十进制数（二-十进制数）

采用二进制，易于实现数据传送和运算，但二进制数不直观，于是在计算机的输入和输出时，通常还是采用十进制数表示。不过这样的十进制数要用二进制编码来表示。一位十进制数要用四位二进制编码来表示。编码方式又有多种，常用的是8421 BCD码。十进制字符共有十个，而四位二进制可有16种不同的编码，故舍去A～F的编码，分别用0000～1001表示。例如2457的BCD码为

0010 0100 0101 0111

而数

1000 0111 0110 1001

是用8421编码方法表示的十进制数8769。

六、字符编码

当把用汇编语言或高级语言编写的程序送入计算机时，就要键入很多字母、数字、标点，及其它特殊符号。这就需要对字符进行编码，在微型计算机中，常用ASCII码，它总共有128个元素，其中包括32个通用控制字符，10个十进制数码，52个英文大小写字母和34个专用符号。例如字符“A”的ASCII码为41H，0～9的ASCII码为30H～39H，详见附录1。

第三节　微型计算机的基本组成和结构特点

一、微型计算机的基本组成

微型计算机（以下称微机）与一般计算机无本质区别。它也是由运算器、控制器、存贮器和输入/输出设备等几部分组成。其不同点在于，由于半导体工艺的发展，把线路复杂庞大的运算器和控制器集成到5×5 mm²大小的单片集成电路中，并称之为微处理器（MPU）或中央处理器（CPU）。以CPU为中心，再加上存贮器芯片、输入/输出（I/O）接口芯片，以及配置上键盘、显示器、打印机等外部设备就构成了一台微机。图1.1是微机的基本组成方块图。

图1.1是微型机的硬件系统，只有同时配备了软件系统和硬件系统的微型计算机系统才具有计算机的效能，配置的软件越丰富，这个微机系统发挥的效能就越强。没有软件系统的微机称为裸机系统，是没有什么用途的。

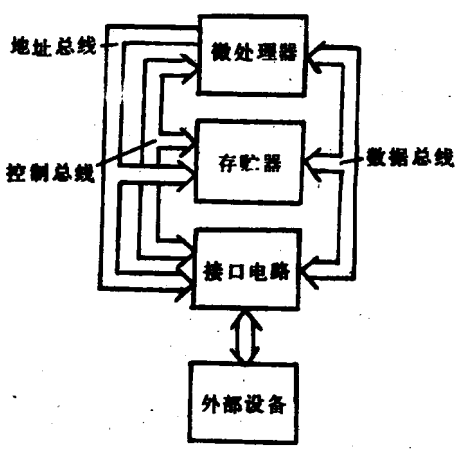

图 1.1　微型计算机组成方块图

所谓软件系统，系指微机系统所使用的各种程序的集合。软件系统是人与计算机进行信息交换，通信对话，对计算机进行控制与管理的工具。它包括系统中配置的各种系统软件和为满足用户需要编制的各种应用软件。系统软件包括操作系统、各种高级语言编译程序、诊断程序、监控程序等。操作系统、监控程序主要用于管理计算机；编译程序用来把高级语言翻译成机器语言；诊断程序用于诊断机器故障。

程序设计语言分为高级语言、汇编语言和机器语言。机器语言是机器唯一能够识别和执行的语言，它使用二进制代码，不易记忆，编程复杂，但它的执行速度快。而汇编语言使用

与机器语言一一对应的字母或代码来表示，这些字母和代码，称为助记符。使用汇编语言编制程序，提高了编程速度，也可编出质量很高的程序。对微型机来说，汇编语言使用得极为广泛。但是计算机也不能直接执行使用汇编语言编制的程序，必须通过汇编程序的翻译，变成机器语言才能执行。另外，各种CPU如Z80、6502、8085等，都有自己的汇编语言，本书选择目前八位机中最通用的Z80来讲述其汇编语言。实践证明，掌握了这种机种，也很容易掌握其它机种。

高级语言可以用于各种类型的计算机，不受机种限制。但是编译工作复杂，降低了程序的执行速度。

二、微机的结构特点

图1.1给出了微机组成方块图，下面就其结构和特点做进一步说明。

（一）微处理器

它是微机的核心部件，图1.2给出了微处理器的结构框图。

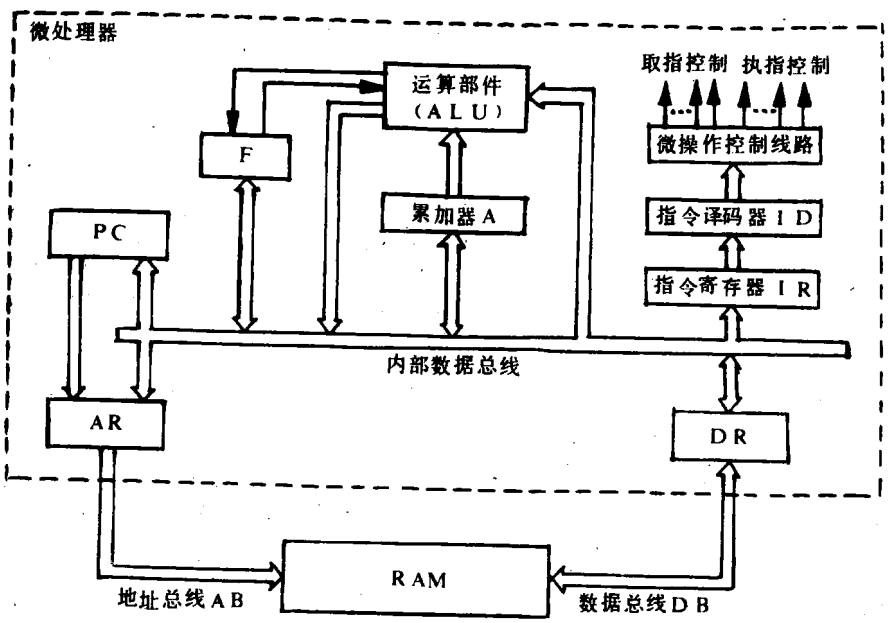

图 1.2 微处理器的结构框图

从图中可看出，主要包括运算部件ALU，即算术逻辑单元，参加运算的两个操作数，一个来自累加器A，另一个来自数据寄存器DR。运算结果通常又送到累加器A保存起来。数据寄存器DR暂存从存贮器取出的指令或数据，存入存贮器的数据也在此暂存。

程序计数器PC用于控制程序执行的顺序，它存放的是指令地址。而地址寄存器AR用来存放指令地址或操作数地址。在取指令时，将PC中存放的指令地址送AR，以便根据AR给出的地址，从对应的存贮器中取出指令；在取操作数时，可将操作数地址通过内部数据总线送到AR，以便从存贮器中取出操作数；如向存贮器存入数据，也要给出地址送AR保存，以便存入数据。

指令寄存器IR用来存放从存贮器取出的将要执行的指令。指令译码器ID用来翻译来自IR的指令，以便控制微操作控制线路发出各种控制信号，完成规定的操作。状态标志寄存器