普通中等专业教育机电类规划教材

微型计算机原理及其应用

（第2版）

福建机电学校　陈立周　主编

机械工业出版社

微型计算机原理及其应用

普通中等专业教育机电类规划教材

微型计算机原理及其应用

（第 2 版）

福建机电学校　陈立周　主编

机 械 工 业 出 版 社

内 容 简 介

本书是原中等专业学校试用教材《微型计算机原理及其应用》一书的修订本。修订后全书共九章，第一章至第六章为基本知识，介绍微型计算机的原理、Z80 指令系统和编程设计、接口技术和硬件方面的知识。第七章介绍 MCS-51 系列单片机的编程与应用。第八章和第九章介绍 TP-801 单板机和 SCB-1 单片单板机原理和微机应用实例。书中附有练习题、实验指导书、Z80 和 MCS-51 的指令系统表，以便于组织教学。

本书可作为中专电类专业《微型计算机原理及其应用》课程的教材。由于第一章至第六章以及第七章的一部分，均参照机械类专业的教学大纲要求进行编写，所以也可作为机械类（包括热加工）专业的教材，并可供有关培训班、职业学校师生或工程技术人员自学时选用。

微型计算机原理及其应用

（第 2 版）

（重 排 本）

福建机电学校　陈立周　主编

*

责任编辑：贡克勤　版式设计：霍永明

封面设计：郭景云　责任校对：唐海燕

责任印制：何全君

*

机械工业出版社出版（北京市百万庄大街 22 号）

邮政编码：100037

（北京市书刊出版业营业许可证出字第 117 号）

北京京丰印刷厂印刷

新华书店北京发行所发行·新华书店经售

*

开本 787×1092 1/16 · 印张 15.75 · 插页 2 · 字数 393 千字

1999 年 5 月第 2 版第 11 次印刷

印数　187 801—199 800　定价：20.50 元

*

ISBN 7-111-04402-9/TP·246（课）

前　言

本书是原中等专业学校试用教材《微型计算机原理及其应用》的修订本。原书系根据1987年原国家机械工业委员会教育局颁布的中等专业学校电类专业的教学计划，以及《微型计算机原理及其应用》课程的教学大纲组织编写的。根据当时国家机械工业委员会中专机械制造专业教材编审委员会的意见，机械类和热加工类专业也要采用本书作为教材，为此组织编写时，已经兼顾了机、电两类专业的需要，为此本书可作为机、电两类专业《微型计算机原理及其应用》课程的教材。

修订本是根据1992年原机械电子工业部中专电类专业教学指导委员会对教学改革和修订教学计划的意见，在教学指导委员会直接指导下进行修订的，修订时参照近年来微型计算机的发展和应用现状，吸收了许多老师和同学对原书提出的宝贵意见，删除了一部分次要内容，增加了第七章的单片机，同时为便于同学的自学，对各章的重点和难点，都增加了必要的说明，修订后既保留Z80作为入门基础，便于学生接受，又增加了单片机的内容，使教材有较强的实用性。

按电类专业的教学大纲，本课程讲课为61学时，实验为18学时。机类专业的课程授课时数少于电类专业，为此机类专业可以不讲第八章，并可删除第三章与第九章中较繁的实例，或改讲简单一些的实例。书中的练习题和实验指导书都可供教学中选用。

修订本由福建机电学校陈立周副教授担任主编（编写第一、二、三、七、八章），上海电机制造技术专科学校褚文奎高级讲师（编写第四、五、六章）和山东机械工业学校赵欣讲师（编写第九章和全部实验指导书）担任协编。

原书是由上海机械专科学校李浩副教授负责主审，当时的上海交通大学白英彩教授和上海机械学院的杨伟民副教授，杭州机械工业学校詹红军讲师对原书提了许多宝贵的意见。

此次修订由山东机械工业学校徐仁贵高级讲师担任主审。山东机械工业学校刘兆峰、应泉莉、巩建国，福建机电学校雷伍，福建电子学校陈秋龙，福建第五工业学校张国利，福州工业学校黄晓玲，福建邮电学校董小龙等老师参加了审稿工作。在此对他们谨致谢意。

本书虽经修订，但仍难免存在缺点与错误，敬请使用本教材的老师和广大读者予以批评指正。

编　者

1994年4月

目　　录

第一章　微型计算机的基础知识

第一节　不同进位计数制及其互换

在人们日常生活中，都习惯于使用十进制，但在数字电路和电子计算机内部，由于只能通过电位的高低表示 1 和 0 两个数码，所以计算机中不用十进制而用二进制。这样，在使用计算机的时候，就存在常用的十进制与计算机中使用的二进制进行互换的问题。又由于用二进制表示一个数，所用的数码长，不但书写和阅读不方便，而且容易出错，所以在书写时又常把二进制转换为十六进制，在开始学习微型电子计算机的时候，首先要熟练掌握这三种进位计数制的互换。

任何一种进位计数制，其位数多少决定于所表示的数的大小，位数越多所表示的数值也越大。考虑到本书所讲的微型计算机是一种字长为 8 位的计算机，所以下面讨论二进制数时都取 8 位，当然这不是说二进制数都是 8 位数，而是说本书常用的二进制数为 8 位。

一、二进制与十六进制数的互换

一个 8 位二进制数，可以写成 2 位的十六进制数，所以两种进位制数进行转换时，可以把 4 位二进制数划为 1 组，然后对每一组进行相应的变换，例如二进制转换为十六进制可写成

$$\underbrace{0110}_{6}\quad\underbrace{0110}_{6}$$

把十六进制转换为二进制可写成

$$\underbrace{4}_{0100}\quad\underbrace{3}_{0011}$$

为了不使二进制、十进制或十六进制数相混淆，规定在二进制数的后面加上符号 B。如 01100110B，在十六进制数后面加上符号 H，如 66H、43H 等。也可以不在数后面加 H，而在数的前面加符号 $，如 $66 或 $43 等等。如果数的后面没有符号，一般就指该数为十进制数。如果被转换的数不是 8 位，而且是一个小数，则分组时应以小数点为准，整数从小数点开始，从右向左每 4 位划为一组，小数部分则从小数点开始，从左向右每 4 位划为一组，如果最后一组不是 4 位，可以用零补齐，以数 1101100.11011B 为例，可在数的前后补零，即

$$\underbrace{0110}_{6}\quad\underbrace{1100}_{C}\;.\;\underbrace{1101}_{D}\quad\underbrace{1000}_{8}$$

二、二进制与十进制数的互换

对于二进制整数，各位数的位权可以用底数为 2 的 n－1 次幂来确定，n 表示该数的位数，即第 1 位的位权为 $2^0=1$，第 2 位的位权为 $2^1=2$，……若已知一个二进制数为 10101010B，对应的十进制值应为

$$10101010B = 1\times2^7+0\times2^6+1\times2^5+0\times2^4+1\times2^3+0\times2^2+1\times2^1+0\times2^0$$
$$=170$$

对于二进制小数，其小数点以后各位的位权，可以用底数为 2 的负 n 次幂来确定，n 同样表示位数。即从小数点向右算起，第 1 位的位权为 $2^{-1}=0.5$，第 2 位的位权为 $2^{-2}=0.25$……，例如求 11001100.00110011B 的十进制值则

$$11001100.00110011B = 1\times2^7+1\times2^6+1\times2^3+1\times2^2+1\times2^{-3}$$
$$+1\times2^{-4}+1\times2^{-7}+1\times2^{-8}$$
$$=204.19921875$$

反过来，要将十进制整数转换为二进制数，可以采用逐次除以 2 余数反序排列的方法，所谓反序排列，指第 1 次除以 2 的余数排在最低位。以十进制数 25 为例，逐次除以 2 列式如下

$$25\div2=12\cdots\cdots\text{余 }1$$
$$12\div2=6\cdots\cdots\text{余 }0$$
$$6\div2=3\cdots\cdots\text{余 }0$$
$$3\div2=1\cdots\cdots\text{余 }1$$
$$1\div2=0\cdots\cdots\text{余 }1$$

由于 8 位微型计算机习惯将二进制数写成 8 位，可得

$$25=00011001B$$

如果二进制数不超过 8 位，即十进制数不超过 255，也可以不必列式，直接用口算转换。

要把十进制小数转换为二进制数，可以采用小数部分逐次乘 2，其乘积的整数个位（即所谓溢出位）按正序排列的方法。以 33.6875 为例，其整数部分

$$33=00100001B$$

其小数部分

$$0.6875\times2=1.375\cdots\cdots\text{溢出数为 }1$$
$$0.375\times2=0.75\cdots\cdots\text{溢出数为 }0$$
$$0.75\times2=1.5\cdots\cdots\cdots\text{溢出数为 }1$$
$$0.5\times2=1\cdots\cdots\cdots\cdots\text{溢出数为 }1$$

可得出

$$33.6875=00100001.10110000B$$

三、十进制与十六进制数的互换

我们已经掌握了十进制与二进制的互换以及二进制与十六进制的互换。因此要把十进制数转换为十六进制数，可以先转换成二进制数，再改写成十六进制数。反此，十六进制数也可以先改成二进制数，再转换成十六进制数。

十六进制数要直接转换为十进制数也可以按各位的位权求出该数对应的十进制数值。整数部分的位权等于底数为 16 的 n－1 次幂，(n 为位数)，即第 1 位位权为 $16^0=1$，第 2 位位权为 $16^1=16$，依次类推，例如十六进制的数为 8A71H 可求出

$$8A71H = 8\times16^3+10\times16^2+7\times16^1+1\times16^0$$
$$=35441$$

对于十六进制的小数，其小数点以后各位的位权，同样可以用底数为 16 的负 n 次幂来确定，n 表示位数，即从小数点向右算起，第 1 位的位权为 $16^{-1}=0.0625$，第 2 位位权为$16^{-2}=$

0.00390625……，例如某个小数为 0.4AC9H，转换为十进制数为

$$0.4AC9H = 4\times16^{-1}+10\times16^{-2}+12\times16^{-3}+9\times16^{-4}$$
$$=0.2921295$$

反过来，要把十进制转换为十六进制数，其方法与十进制数转换为二进制相似，即整数部分采用逐次除以 16 余数反序排列的方法。小数部分则采用逐次乘 16 溢出数正序排列的方法。例如将 13562 十进制数转换为十六进制

13562÷16＝847……余 10（记作 0AH）

847÷16＝52 ……余 15（记作 0FH）

52÷16＝3……余 4

3÷16＝0……余 3

可得

13562＝34FAH

在书写十六进制数时，若打头的数为 A～F；则应在 A～F 之前再加一个 0，以表示这是一个数而不是其他符号。

十进制小数 0.359375 转换为十六进制数

0.359375×16＝5.75…溢出数为 5

0.75×16＝12.0……溢出数为 C

可得

0.359375＝0.5CH

第二节　带符号的二进制数

一、带符号二进制数与不带符号二进制数的区别

在数学运算中，表示一个数的正负，可以在数的前面冠以正号或负号。但计算机只能辨认 0 和 1 两个数码，不能辨认其他符号，因此在字长为 8 位的二进制数中，将最高位规定为符号位，最高位为 0，表示该数为正，最高位为 1，表示该位为负。例如数 01101111B 等于＋1101111B，而数 11101111B，则等于－1101111B。这种将最高位定为符号位的二进制数，称为带符号的二进制数。对于 8 位字长的带符号二进制数来说，表示数值的仅有 7 位，所能表示的范围为＋127～－128。

应该注意，同样一个二进制数，它既可以是无符号数，也可以是带符号数。例如二进制数 11001100B，既可以是无符号数 204，也可以是带符号数－76，到底它是 204 还是－76，全靠事先约定，单从数的本身无法判别它属于什么数。

例如下面要介绍的相对转移指令中的偏移量，约定必须使用带符号数，因此出现在偏移量中的二进制数，总是一个带符号数，在填写偏移量时，也只能使用带符号数，而相反程序中的地址值，约定使用无符号数，就只能使用无符号数。

总之，一个数是带符号的还是无符号的二进制数，从数的本身是无法区别的，只能根据它出现的场合，以及该场合约定使用什么数才能区分它们。

二、带符号数的表示方法

上面讲过，一个带符号数，它的最高位是符号位，其余表示数值。这种表示方式称为原

码，实际上，带符号的二进制数，除了原码表示法之外，还有反码与补码，下面分别加以介绍。

（一）原码（True form）

用原码表示一个带符号二进制数，其最高位为符号位，其余表示数值，例如

$$x=+1010101B \qquad [x]_{true\ form}=01010101B$$

$$x=-1010101B \qquad [x]_{true\ form}=11010101B$$

式中 x 为真值，$[x]_{true\ form}$表示数 x 的原码，对于数 0，其原码可以为

$$[+0]_{true\ form}=00000000B$$

$$[-0]_{true\ form}=10000000B$$

（二）反码（One′s complement）

反码也是带符号数的一种表示法，它同样规定最高位为符号位，其余则要看是正数还是负数，对于正数，其余各位表示数值，也就是正数的反码等于原码。对于负数，其余各位应将 1 换成 0，将 0 换成 1，即所谓将原码取反，例如

$$x=+1010101B \qquad [x]_{O.C}=01010101B$$

$$x=-1010101B \qquad [x]_{O.C}=10101010B$$

式中 $[x]_{o.c}$表示真值 x 的反码，对于数 0 的反码，也有两种形式

$$[+0]_{O.C}=00000000$$

$$[-0]_{O.C}=11111111$$

（三）补码（Two′s complement）

补码仍然把最高位定为符号位，对于正数，其余各位表示数值，可见正数的补码、反码与原码完全相同。对于负数，则其余各位等于真值取反后再加 1，简称为取反加 1，例如

$$x=+1010101B \qquad [x]_{T.C}=01010101B$$

$$x=-1010101B \qquad [x]_{T.C}=10101011B$$

式中 $[x]_{T.C}$为真值 x 的补码。

补码这个概念与某个具体记数器的最大容量有关，以常用的 8 位二进制数为例，扣除最高位作为符号位外，记数的最大容量为 7 位数。当计数器计至 128 时，最高位产生溢出，又由于计数器只有 7 位，溢出的数就被丢弃。这个被丢弃的值就是最大容量值，又称为模，用符号 Mod 表示。对于模等于 128 的 7 位计数器，0 与 128 的数码相同，或者说 0 与 128 等价、即

$$0=0000000B$$

$$128=\boxed{1}\ 0000000B$$

方框中的 1 就是被丢弃的数，可见，若模为 M，则 a 与 a+M 等价，例如模为 128，1 与 129 等价，2 与 130 等价。因为不计及溢出数，计数器内的示值是相同的。即

$$a=a+M\ (\text{Mod 为 M}) \tag{1-1}$$

再把这个概念推广到负数领域，例如 a=（−2），代入式（1-1）有

$$(-2)=(-2)+128$$

$$=126 \qquad (\text{M 为 128})$$

即模为 128 时，（−2）与 126 等价，或者说这两个数互为补数，同样，可推出（−3）的补码为 125，（−4）的补码为 124。

为了进一步理解这个概念，可以用时钟做为例子，时钟的钟面最大示数为12，时钟走到12点就等于0点，数12就被自动丢弃。可见时钟的钟面是一个模为12的计数器。时钟的时针向前拨3个字（a＝3），跟向前拨15个字（a＋M＝3＋12＝15）都停在同一位置上，也就是说＋3与＋15等价。

如果将时针向后拨定义为负数，那么时针向后拨3个字〔a＝（－3)〕，跟时针向前拨9个字〔a＝a＋M＝（－3）＋12＝9〕都停在相同位置上，也就是－3与9等价，因为对于模为12的钟面来讲，－3可以用其补码9表示。

要求得一个数的补码，可以用公式a＝a＋M（Mod＝M)。也可以采用正数补码等于原码，负数补码为取反加1的方法求得（注意：取反加1不包括符号位)。表1-1是8位带符号二进制数的原码、反码、补码对照表，应注意表中的

$$〔+0〕_{T,C}=00000000B$$

$$〔-0〕_{T,C}=00000000B$$

而10000000B不是（－0）的补码，而是（－128）的补码。

表1-1　二进制数原码、反码、补码对照表

十进制数	二进制数	原　码	反　码	补　码
＋0	＋0000000	00000000	00000000	00000000
＋1	＋0000001	00000001	00000001	00000001
＋2	＋0000010	00000010	00000010	00000010
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
＋126	＋1111110	01111110	01111110	01111110
＋127	＋1111111	01111111	01111111	01111111
－0	－0000000	10000000	11111111	00000000
－1	－0000001	10000001	11111110	11111111
－2	－0000010	10000010	11111101	11111110
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
－126	－1111110	11111110	10000001	10000010
－127	－1111111	11111111	10000000	10000001
－128	－10000000	无法表示	无法表示	10000000

三、带符号二进制数的运算

二进制数和十进制数的运算规则基本相同，所不同的仅仅是前者逢2进1，后者逢10进1，借位时，从高位借1到低位，前者为2，后者为10。但如果是带符号的二进制数，则情况比较复杂，因为带符号二进制数有三种表示方法，其中反码用得较少，下面主要介绍原码与补码运算中应注意的问题。

原码运算时，首先要把符号与数值分开。例如两数相加，先要判断两数的符号，如果同号，可以做加法，如果异号，实际要做减法，减后的差做为两数之和，和数的符号与绝对值较大的数的符号相同。两数相减也是一样，也要先判断符号，然后决定是相加还是相减，还要根据两数的大小与符号决定两数差的符号。

补码运算不存在符号与数值分开的问题，而且加法运算就一定是相加，减法运算就一定是相减，所以凡是需要进行加减的地方，最好使用补码。

设有 x、y 两个数，并以补码表示，即

x＝10011111B　　（－97 的补码）

y＝00001000B　　（＋8 的补码）

若求 x＋y 之和，可不用考虑两数的符号，直接相加，得出的和为 x＋y＝10100111B（－89 的补码），可见相加结果必定正确。

若求 x－y 之差，也可以直接相减，即

x＝10011111B　　（－97 的补码）

－y＝00001000B　　（＋8 的补码）

x－y＝10010111B　　（－105 的补码）

若求 y－x 之差，同样也用减法即

y＝00001000B　　（＋8 的补码）

－x＝10011111B　　（－97 的补码）

y－x＝[1] 01101001B　　（＋105 的补码）

也就是说做减法时，不论两数符号如何，其相减结果不论是数值还是符号都将是正确的。

在上述 y－x 算式中，最高位发生进位（加方框的数字表示是进位数字），只是因为在字长为 8 位的计算机中，进位无处可进，就产生自然丢弃现象。计算机在运算中，这种自然丢弃是允许的，它并不影响结果的准确性。

但要注意，有的进位是不允许的，例如得数超过 7 位二进制所允许的表示范围（即超出＋127～－128）而产生的进位，这种进位就不允许，我们称这种进位为溢出。溢出与自然丢弃是两种不同的概念。判别属于哪一种，则要看第 7 位与第 8 位的进位情况，如果第 7 位和第 8 位同时产生进位，即所谓双进位，则这种进位属于允许的自然丢弃，丢弃后并不影响结果的准确性。如果只有第 7 位或者只有第 8 位产生进位，即只有单进位，则这种进位属于溢出，溢出表示其数值超出计算机字长所能表示的范围，运算结果必然是错误的，因而也是不允许的。

例 1-1　求下列两组 x、y 之和。

x＝＋1100100B	$[x]_{T.C}$＝01100100B	$[+100]_{T.C}$
y＝＋1000011B	＋$[y]_{T.C}$＝01000011B	＋$[+67]_{T.C}$
	$[x+y]_{T.C}$＝10100111B	$[-89]_{T.C}$
x＝－1111000B	$[x]_{T.C}$＝10001000B	$[-120]_{T.C}$
y＝－0011000B	$[y]_{T.C}$＝11101000B	＋$[-24]_{T.C}$
	$[x+y]_{T.C}$＝[1] 01110000B	$[+112]_{T.C}$

例中第 1 组，只有第 7 位产生进位，第 8 位没有进位。在第 2 组中只有第 8 位产生进位，第 7 位没有进位，都是单进位，所以都属于溢出，运算结果为－89 和＋112 显然也都是错误的。

还应注意，溢出主要是指带符号二进制数进行加减运算时可能产生的一种结果。对于无符号数，不论是单进位还是双进位，进位都是得数的一个部分，一般称为进位，而不采用溢

出这个概念。

对于无符号数还应注意一点，当两个无符号数相减时，不允许用较小的数去减一个较大的数，因为这两者相减，其差一定是负数，无符号数的前提是没有符号，显然也不允许存在负数，如果这样做，必然会得出错误的结果。

第三节 BCD码及文字符号代码

一个二进制数，可以表示一个无符号数，也可以表示一个带符号的数。而且它还可以根据事先约定代表一个文字、一个符号或者代表一个特定的内容。当它代表文字或符号时称为码，例如00110100B，作为数它等于无符号数34或是带符号数+34，如果事先约定，它又能代表引号。所谓约定，可以按标准约定也可以自行约定，下面介绍两种按标准约定的常见码。

一、BCD码（Binary coded decimal）

BCD码是一种以二进制形式表示十进制数的编码，又称为二-十进制码，它貌似二进制数，实行是十进制数。

BCD码以4位为一组，选用0000B～1001B10种状态代表0～9共10个数，舍弃掉后面6种状态。当BCD码与十进制数进行互换时，可以按4位一组，逐组进行互换。

例 1-2 将84.7转换为BCD码。

$$\underbrace{8}_{1000}\ \underbrace{4}_{0100}\ .\ \underbrace{7}_{0111}\ \underbrace{0}_{0000}$$

例 1-3 将BCD码10010100.01110010B转换为十进制数。

$$\underbrace{1001}_{9}\ \underbrace{0100}_{4}\ .\ \underbrace{0111}_{7}\ \underbrace{0010}_{2}$$

要将一个二进制数转换为BCD码，通常先将它转换为十进制数，然后再转换为BCD码，同样将BCD转换为二进制数，也是先转换成十进制数，再转换为二进制数：

例 1-4 将二进制数11110111B转换为BCD码。

$$\begin{aligned}11110111\text{B} &= 1\times2^7+1\times2^6+1\times2^5+1\times2^4\\ &\quad +0\times2^3+1\times2^2+1\times2^1+1\times2^0\\ &= 247\end{aligned}$$

$$\underbrace{2}_{0010}\ \underbrace{4}_{0100}\ \underbrace{7}_{0111}$$

二进制数11110111B的BCD码为〔001001000111〕$_{BCD}$也可以按习惯写成两个字节即等于〔00000010 01000111〕$_{BCD}$。

例 1-5 将BCD码〔10001001〕$_{BCD}$转换为二进制数。

$$\underbrace{1000}_{8}\ \underbrace{1001}_{9}$$

$$89=01011001\text{B}$$

BCD码的低4位向高4位进位是遵循逢10进1的原则。这与二进制显然不同，二进制低4位要向高4位进位必须遵循逢16进1，这样两个BCD码相加，可以当做二进制数相加，但

必须以 4 位为一组，逐组相加，凡相加后和大于 9 者，应进行加 6 修正。

例 1-6 将 BCD 码〔01011000〕$_{BCD}$与〔01101001〕$_{BCD}$相加。

```
  0101  1000
+ 0110  1001
-------------
  1100  0001
进位←┘
```

由于第一组和为 17，除进位 1 外，余数应加 6 修正。第二组和为 12，也大于 9，也应加 6 修正，即将上式和数修正如下：

```
  1100  0001      检验      58
+ 0110  0110                +69
-------------               ----
0001  0010  0111            127
进位←┘
```

两个 BCD 码相减，凡低 4 位有向高 4 位借位者，都要进行减 6 修正，无借位的可以不必修正。

例 1-7 将 BCD 码〔10000101〕$_{BCD}$与〔0010 1000〕$_{BCD}$相减。

```
  1000  0101
- 0010  1000
-------------
  0101  1101
      ↑
      └──借位
```

```
0101  1101      检验      85
-       0110              -28
---------------           ---
0101  0111                57
```

二、ASCII 码

ASCII 码是美国信息交换标准代码的简称，由 American Standard Code Information Interchang 的第一个字母组成。

计算机只能辨认或存储 0 和 1 两个数码，也就是计算机内部只能使用二进制数。但在编制计算机程序或处理信息时，还会碰到许多文字符号，这就需要把文字符号改为一串二进制数码来代替，即把文字符号数码化，现在国际通用的文字符号代码是 ASCII 码。

ASCII 码共 128 个，用 00000000～01111111 代表，其中英文大小写字母共 52 个，0～9 数字 10个，常用书写符号（如！ %等等）和常用数学运算符号（如＋、－、＞、＜等）32 个，另有控制符号 34 个，共计 127 个。例如英文大写字母 A 的编码为 01000001，或写成十六进制的 41H，数码 5 的编码为 00110101 或写成十六进制的 35H。ASCII 码表见书后附录。

ASCII 码实际只占用一个字节中的 7 位，余下最高位可留作奇偶校验位。

1.2 逻辑电路
1.3 集成门电路

非门（反相器） A —□— Y $Y=\bar{A}$

或门 A, B —⊃1— Y. $Y=A+B$

与门 A, B —& — Y. $Y=A\cdot B$

第四节　微型计算机系统的组成

电子计算机自 1946 年问世以来，经历了从电子管、晶体管、中小规模集成电路、大规模集成电路和超大规模集成电路的五代演变。由于采用了大规模集成电路，使整机体积大幅度缩小，出现了体积小、价格低廉的微处理器，用微处理器组成的电子计算机就称为微型计算机，配上相应的外围设备和系统软件之后，就称为微型计算机系统。可见组成一个微型计算机系统包括硬件和软件两大部分。硬件是指组成计算机的所有机械、磁性和电子的部件或装置，它是组成计算机的物质基础。软件则是各种程序的总称，它属于信息的东西，以数字形式存在于硬件之中。下面分别加以介绍。

一、微型计算机系统的硬件

微型计算机的硬件，如图 1-1 所示，包括 CPU、存储器、输入输出接口三大部分，在图中用虚线框表示。作为微型计算机系统，则还要包括电源和外围设备。图中的显示器、键盘、打印机都是常见的外围设备。外围设备通过输入输出接口与计算机相连。

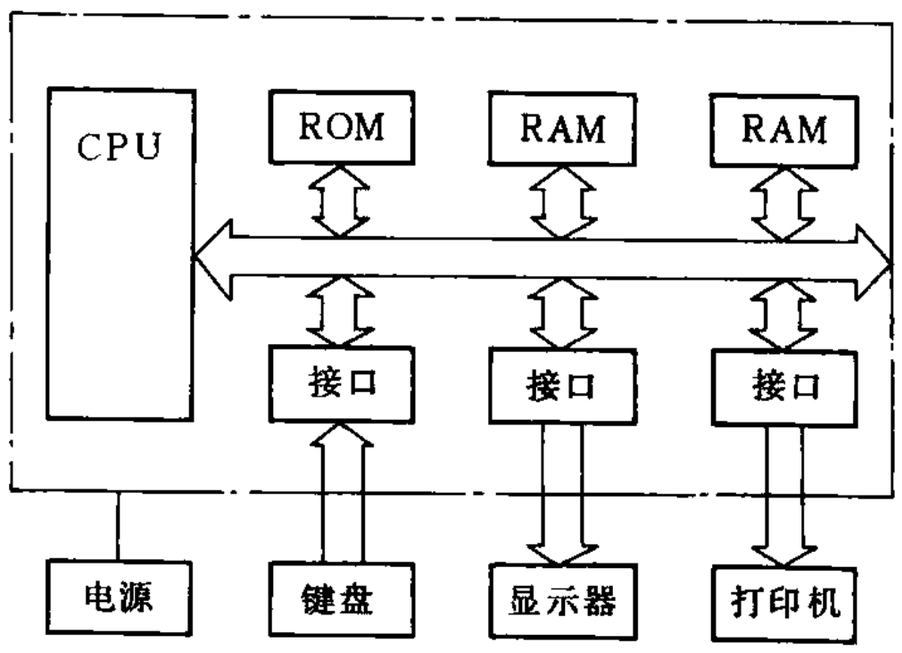


图 1-1　微型计算机系统的硬件组成

（一）微处理器

微处理器是微型计算机的中央处理单元，中央处理单元简称为 CPU（Central Processing Unit）它是电子计算机的核心部件，微型计算机的 CPU 是制作在一块大规模集成电路的芯片上，所以又称为微处理器，简称 μp 或 Mp（Microprocessor）采用微处理器是微型计算机和普通计算机的主要区别。

微处理由运算器、控制器和内部寄存器三大部分构成。

运算器对输入的数据进行算术运算、逻辑运算或移位操作。控制器是计算机的指挥中心，通常讲的电子计算机指的是数字电子计算机它是根据程序存储和程序控制的原理进行工作的。程序本身由一系列指令所组成，计算机必须从程序中逐条取出指令、分析指令、执行指令，并做好取出下一条指令的准备工作。这一系列的操作要求严格按一定顺序进行，控制器任务就是控制这些操作的顺序，以保证计算机能快速而又有条不紊地工作。内部寄存器是计算机在运算过程中用来存储或转移数据及指令的场所。因此在运算之前，控制器要把数据调入内部寄存器。

（二）存储器

存储器是存放和记忆数据及程序的部件。微型计算机内的存储器简称为内存，它由许许多多存储单元组成。微型计算机（简称微机）常用的存储器是半导体存储器。一个芯片内包含有几千个存储单元，为了辨认这些单元，需要将存储单元按顺序加以编号，这种编号称为存储器地址。一般 8 位机的地址用一个 16 位二进制数表示，地址编号可以从 0000H 编至 0FFFFH，共 65536 个单元，简称为 64K。每个地址内可以存一组 8 位二进制数。存在内存中的数称为该单元的内容，图 1-2 是半导体存储器的外形。

存储器按它的工作方式分为随机存取存储器（简称 RAM）和只读存储器（简称 ROM）两

2. 其它　或非　$Y=\overline{A+B}$　　异或 $Y=A\oplus B=\bar{A}B+A\bar{B}$

与非　$Y=\overline{A\cdot B}$　　缓冲门 A —□—□— $Y=\bar{\bar{A}}=A$，可提供驱动电流，提高负载能力

种。RAM 可以随意存入数据或取出数据，存入又称为写入，取出又称为读出。要向存储器进行写入或读出操作，首先要指定读或写的地址，地址经译码后接通所指定的存储单元，并向存储器发出读写请求信号，然后，CPU 就可以向所指定的存储单元进行读或写。在这里要注意不要把地址和内容两个概念混淆起来。地址是存储单元的编号，通过地址选择某个存储单元，以便对该单元进行操作，地址是一组 16 位二进制数。而存储单元中所存的数据则称为内容，内容可以是指令代码，也可以是数据，包括运算前的原始数据，运算过程的中间数据，以及运算的最后结果，内容总是一组 8 位二进制数。

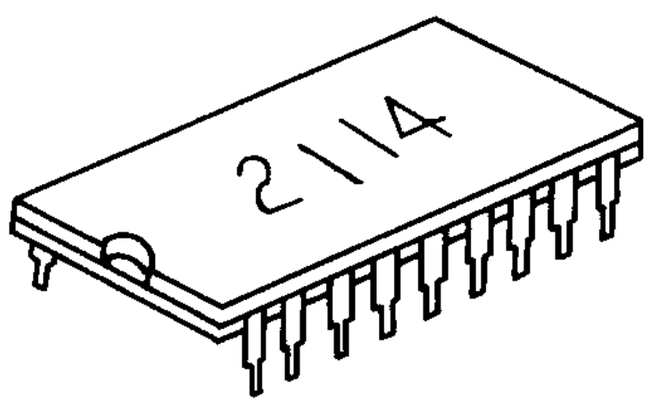


图 1-2 半导体存储器的外形

ROM 是事先通过写入设备存入数据，写入之后就不能随便更改，以后只能读出不能写入，所以称为只读。通常 ROM 用于存放不必随时更改的程序或数据。

一台计算机需要配置多少存储单元，是根据其用途和需要来确定的。一般通用计算机配置的存储器需要多一些，专用计算机则根据需要配到够用即可。

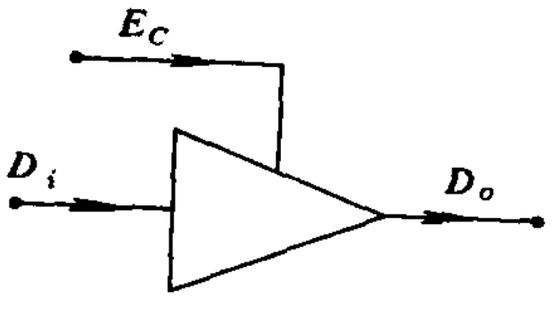
（三）输入输出接口

输入输出接口简称 I/O 接口，上面讲的微处理器和存储器可以组成主机，作为微型计算机系统，则除了主机之外，还需要配置不同的输入输出设备，或统称为外围设备（简称外设）。例如输入数据用的键盘、乒乓开关、纸带机等输入设备，显示运算结果用的数码管、显示器、打印机等输出设备。而 I/O 接口的任务就是作为主机与外设之间的连接电路。

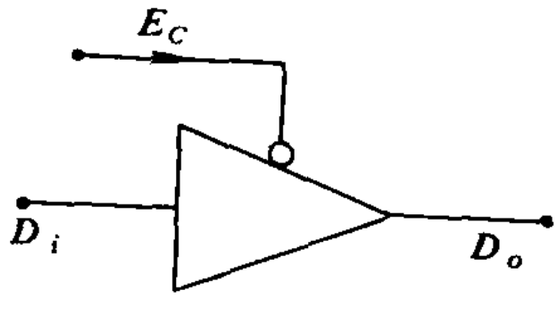
现在电子计算机都采用总线结构，CPU 与各个部件间的数据传送都是通过总线进行的。微机中的总线，根据它们所传送的信息不同，分别称为数据总线 DB (Data Bus)，地址总线 AB (Address Bus) 和控制总线 CB (Control Bus)。由于某一时刻 CPU 只能与一个存储单元或一个外围设备互相传送信息，因此不允许所有外围设备都直接挂在总线上，而要通过 I/O 接口与总线相连，这样，CPU 在某一时刻只能通过地址总线选通一个接口，将其他外围设备的接口阻断，以便与总线隔离起来，不至于相互干扰，这是要使用 I/O 接口的一个原因。其次，外设的工作速度快慢不一，有的外设速度比较慢，例如打印机，打一个字不是几个微秒所能完成的。而 CPU 执行一个操作命令只需要几个微秒，这就产生了 CPU 与外设的速度匹配问题。为了解决这个矛盾，就可以采用 I/O 接口，先把 CPU 送来的数据保存在接口中，然后由外设进行处理，只有处理完毕之后，才允许再次打开接口，接受新的数据，这是使用 I/O 接口的另一个原因。除此之外为了电平匹配，信息形式的转换都需要使用接口。

要实现外设与 CPU 的隔离，可以用三态门作为 I/O 接口，三态门又称为三态驱动缓冲电路，它是一种具有控制端的门电路，可以通过控制端电平高低控制三态门的开或闭，在图 1-3 中，图 a 是高电平控制的三态门，只有控制端为高电平时，输出端才受输入端控制。图 b 是低电平控制的三态门，与图 a 相反，只有控制端为低电平时，输出端才受输入端控制。当输出端不受输入端控制时，其输出端呈高阻状态，相当于三态门将总线与外设隔离起来。利用三态门作为接口的电路如图 1-4 所示。由高电平控制的三态门真值表如表 1-2 所示。

如果要把总线送来的数据锁存，则可用 D 触发器作为 I/O 接口，通过时钟脉冲输入端，将总线数据锁存于接口中。图 1-5 是用 D 触发器作为 I/O 接口连接图。



a)



b)

图 1-3　三态门符号

a）高电平控制　b）低电平控制

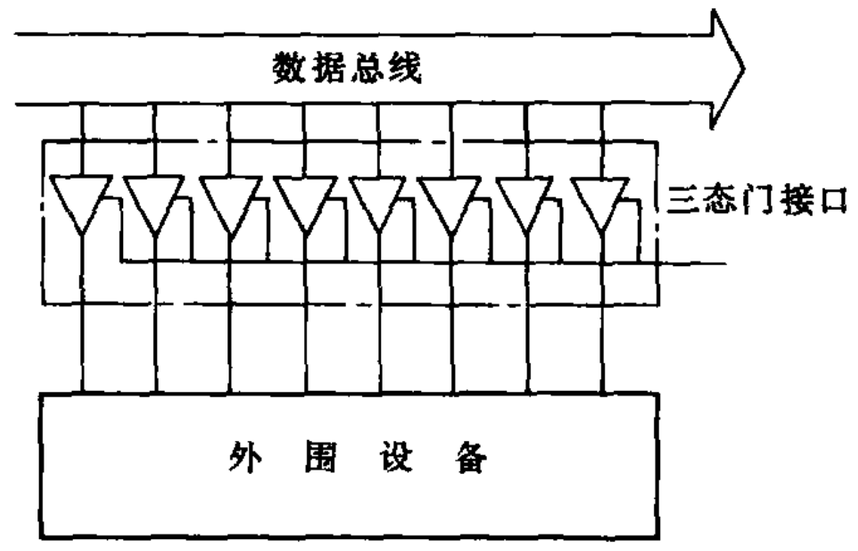


图 1-4　由三态门构成的 I/O 接口

表 1-2　三态门真值表

输入端电平	控制端电平	输出端状态
0	0	高阻
1	0	高阻
0	1	0
1	1	1

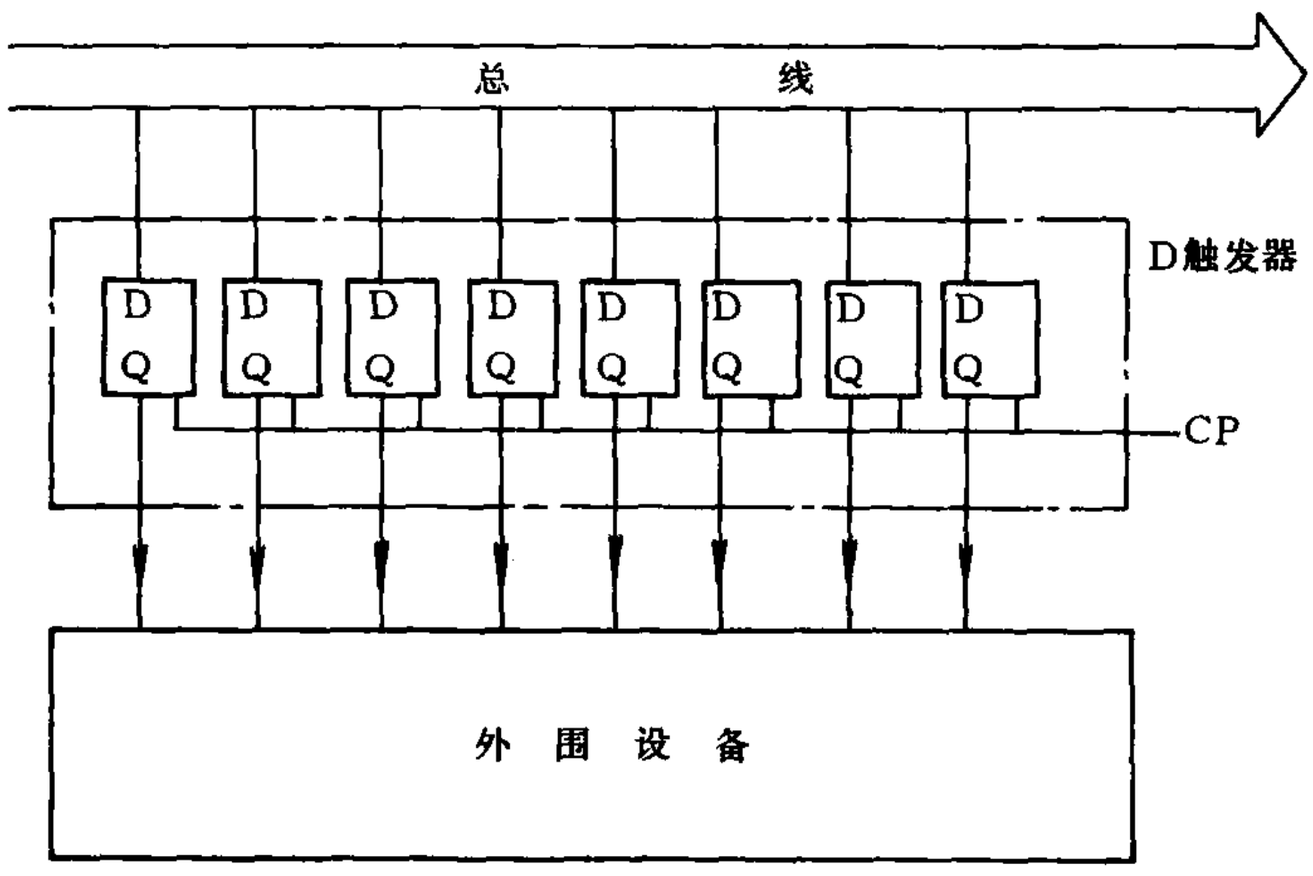


图 1-5　由 D 触发器构成的 I/O 接口

二、微型计算机系统的软件

由硬件构成的计算机，如果内存单元是“空”的，只能称为裸机。裸机不会自行工作，象空白磁带不会自己发声一样。要计算机完成某项运算任务，首先要把解题步骤按照计算机所能理解的语言编成程序，把程序连同原始数据经输入设备存入计算机的内存单元，然后，计算机才能在程序控制下，自动进行各种操作和运算，直至运算完毕，再通过输出设备把运算结果显示或打印出来。可见只有内存中存储有程序的计算机才能工作，我们把各种程序统称为软件。

软件分为应用软件和系统软件两种。应用软件是计算机用户为了解决某个具体问题而编制的程序。系统软件是计算机生产厂家设计的计算机管理程序，它包括监控程序、操作系统、编译程序等等。软件以数码形式依附于硬件之中，它可以被替代或破坏，这也是软件“软”字的含义。

编写软件可以用三种语言：

（一）机器语言

机器语言是指用一系列二进制数码表示的指令和数据的一种语言。它可以直接为 CPU