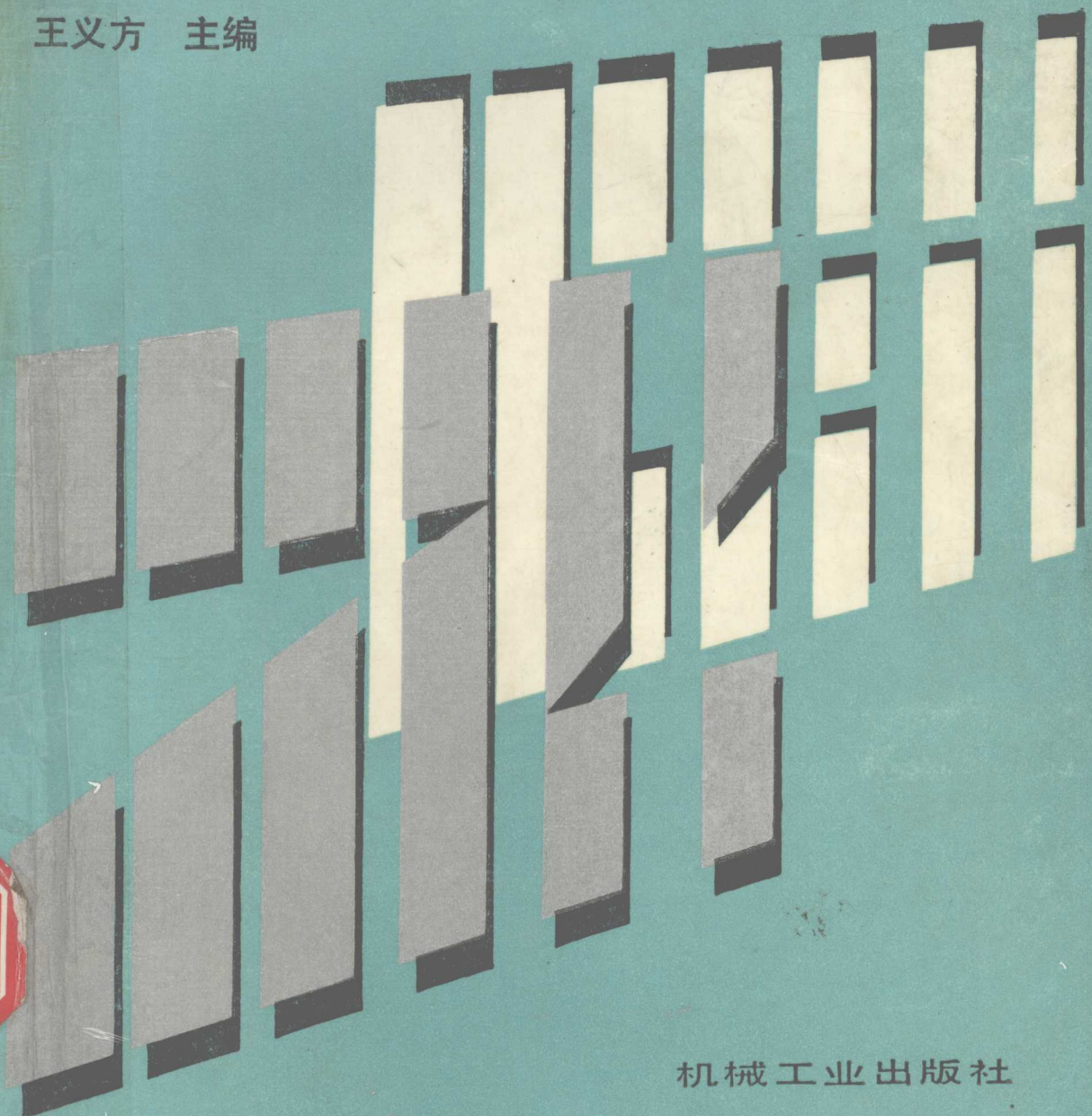
职工高等工业专科学校教材

微型计算机原理及应用

（第 2 版）

王义方　主编



机械工业出版社

职工高等工业专科学校教材

微型计算机原理及应用

（第 2 版）

王义方　主编

机械工业出版社

(京)新登字054号

本书是原《微型计算机原理及应用》一书的修订本。原书是根据教育部成人教育司、全国总工会和机械工业部教育局制订的“工业电气自动化专业”教学计划和“微型计算机原理及应用”教学大纲而组织编写的。在连续印刷7次以后，通过审定，由试用教材转为正式教材。内容包括：微型计算机的基础知识、Z80微处理器的原理及应用和MCS-51单片微机的原理及应用。

全书着眼于应用，内容简明扼要，通俗易懂，便于教学和自学。每章均附有较多数量的例题、思考题和习题，书末还附有应用实例，另有实验指导书与本书相配套。

本书适用于职工大学和业余大学，中专和各种微机培训班也可选用，并可供广大工程技术人员自学入门之用，它尤其适合于成人教学。

微型计算机原理及应用
（第 2 版）
王义方 主编
*
责任编辑：贡克勤　　责任校对：孙志鸿
封面设计：肖　晴　　版式设计：冉晓华
责任印制：王国光
*
机械工业出版社出版(北京阜成门外百万庄南街一号)
(北京市书刊出版业营业许可证出字第117号)
机械工业出版社京丰印刷厂印刷
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23 1/4·字数574千字
1985年12月北京第1版
1992年10月北京第2版·1992年10月北京第8次印刷
印数 119 546—135 545·定价：9.90元
*
ISBN 7-111-03279-9/TP·161(课)

前　言

本书是原《微型计算机原理及应用》一书的修订本。原书是1984年根据教育部成人教育司、全国总工会和机械工业部教育局制订的"工业电气自动化专业"教学计划和"微型计算机原理及应用"教学大纲而组织编写的，是职工高等工业专科学校工业电气自动化专业的正式教材。自1985年出版发行以来，得到了广大读者的关心和支持，许多大、中专学校和各种类型的微机培训班都选它作为教材，社会反映良好，因而年年印刷出版，在此，作者深表感谢。

根据许多读者来信提出的宝贵建议，现在对该书进行了修订，以进一步适应形势发展的需要。

这次修订主要增加了MCS-51单片微机（安排在第九章）的有关内容，另外对原各章内容都作了必要的修订，第二章删去了"其他微处理器简介"一节，第三章删去了部分例子，第五章全部进行了重写，简化了有关存储器芯片的内部结构的内容，增写了许多新的常用芯片的内容，第六章增写了中断的分类，第七章删去了"线路反转"技术识别按键和利用D/A转换芯片实现A/D转换的内容，第八章删去了"EPROM写入接口电路与程序"及"盒式磁带机串行接口电路与程序"等两节，增写了"Z80单板机的扩展"一节，原第九章改为第十章并删去了"保温瓶的自动检测"一节，增写了"应用系统设计方法简述"及"定长控制系统"等内容，最后还对原书中的疏漏和错误进行了修正。

本书修订后，不但继续保持了简明扼要、通俗易懂和便于教学与自学的特点，而且增加了许多新的芯片及一些设计技术的内容，在层次上有所提高，因而更加精炼、实用和先进。按本专业教学计划的规定，Z80微处理器的原理及应用部分讲课为72学时，实验为14学时；MCS-61单片微机的原理及应用部分则建议为60学时（其中带星号'*'的章节为选讲内容），实验为10学时，各校根据具体情况可酌情增减。

本书由苏州市职工业余大学王义方任主编，并编写第一、二、三、四、五、六章及第十章的§10-1、§10-2，上海市卢湾区业余大学杨振英编写第七章、第八章的§8-1、§8-2、§8-3及第十章的§10-3、§10-4、§10-5，苏州市职工业余大学丁建强编写第八章的§8-4及第十章的§10-6，苏州市职工业余大学李绍成编写第九章。

本书修订后，于1991年8月由九所高校和厂职工大学所组成的审稿会审定通过。由湖北汽车工业学院田瑞庭教授主审，桂林市职工业余大学邱振诒副教授协审。参加审稿的还有成都市职工大学聂剑和徐文芳、天津市红桥区职工大学刘风章、杭州市工人业余大学王华兴、淄博市职工大学吕昌泰、常熟市高等专科学校周伟航、德阳市第二重机厂职工大学温良玉和佛山市电梯厂李思亲等。

本书在修订过程中曾得到苏州市职工业余大学有关领导的支持，彭玲再次描画了书中的插图，在此一并表示衷心的感谢。

由于编者水平有限，书中难免存在缺点和错误，恳请读者批评指正。

编者

1991年10月

目　录

第一章　计算机的基础知识

§1-1　概　　述

本章主要介绍微处理器及微型计算机（简称微计算机或微机）的发展与应用概况，计算机的数制与码制以及初级微机的组成及其工作原理。要求能进行十进制、二进制和十六进制之间的转换，明确补码在微机中的应用，了解微机的基本结构和工作过程。

一、微处理器和微型计算机的发展概况

随着计算机技术和大规模集成电路的发展，微机应运而生。自从1971年美国Intel公司研制成功以I4004微处理器为核心的4位微机以来，短短的十几年里得到了突飞猛进的发展，微处理器（CPU）的集成度差不多每两年翻一番，且性能增长一个数量级。因此，完全可以名副其实地讲，微处理器及微计算机的发展正在日新月异。纵观其发展历史，至今已经历了四代的演变：

第一代(1971年至1972年)：

美国Intel公司首先研制成功I 4004，它是4位的微处理器。以它为基础再配以相应的RAM、ROM和I/O接口芯片就构成了MCS-4微计算机。同年，该公司还研制出I 8080，它是8位的微处理器。

第二代(1973年至1975年)：

代表产品是美国Intel公司的I8080、MCS-80和Motorola公司的6800，它们是8位机的中档机。

1976年至1977年，美国Zilqg公司研制的Z 80和Intel公司研制的I 8085，一般称之为第二代半的产品，它们是高性能的8位微处理器。

第三代(1978年至1981年)：

代表产品是美国Intel公司的I 8086、zilog公司的Z 8000和Motorola公司的M68000，它们是16位微处理器，又称第一代超大规模集成电路的微处理器。

第四代(1981年以后)：

代表产品是美国Intel公司的IAPX432、BELL研究所的MAC-32、NS公司的NS16032，它们是32位微处理器，又称超级微处理器。

当前微处理器与微计算机正朝着以下几个方向发展：

①发展高性能的16位和32位微处理器。

②发展专用化的单片微计算机。

③发展带有软件固化的微计算机。

④发展多微处理机系统和局部网络。

⑤充实和发展外围接口电路。

我国是在1974年开始研制微计算机的，于1977年制出了第一台微计算机DJS-050，同年

决定研制DJS-050和DJS-060两个系列的微计算机，它们的中央处理单元，前者相当于Intel公司的8080A微处理器系列，后者相当于Motorola公司的M6800微处理器系列。1981年又正式确定以Z80 CPU为核心组建的微计算机系统为8位微计算机发展的系列之一，其型谱谱号定为DJS-040系列（注：现在已有新的命名法）。总之，随着形势的发展，我国微计算机的研制与生产发展很快，1位机、4位机以及16位机等相继涌现。目前已有100多种型号的产品，主要有紫金Ⅱ号微机（与APPLEⅡ兼容，其CPU为6502微处理器）、ZD-2000汉字终端（其CPU为Z 80微处理器）、0520微型机（与IBM PC兼容，其CPU为8088微处理器）以及与TRS-80和CROMEMCO相当的8位微计算机。

二、微计算机的特点

微计算机和普通计算机没有本质上的差别，同样具有快速、精确、记忆、逻辑判断能力以及程序控制等特点。

此外，它还具有下列特点：

1．价格低、体积小、重量轻、功耗低和可靠性高　这个特点使得一些中、小型的或廉价的设备都能用上微处理器和微计算机，从而使它深入到过去计算机所无法深入的领域。

2．方便灵活，通用性强　由于微计算机的体系结构采用总线结构形式，因而它十分机动灵活，容易构成各种各样的系统。亦即能根据需要来选择总体布局，能大能小。小到将微机作为一个部件组装在应用设备中，使设备电脑化；大到与大型计算机和外部设备互连成一个网络，彼此进行通信并共享资源。总之，它能方便地进行扩展与集散，因此适应面广，可以面向各行各业。

另外，目前构成微计算机的基本部件大多数已经标准化，而且机器结构是大同小异的，所以它们特别适合用户的需要。

更重要的是微计算机具有通用性和灵活性的特点，当任务改变时，勿需对硬件进行重新设计，只需通过重新编制程序就能执行不同的任务。

当然，微计算机也有其缺陷，主要是低档机的处理速度较低，存储容量较小，指令系统比较简单，配套的外部设备的种类和数量较少，支持软件不够丰富等。但是，随着近年来性能更强的新一代微计算机的出现，上述缺陷将会逐步得到克服。

三、微计算机的应用

微处理器与微计算机以其上述显著的特点，迅速地得到了极其广泛的应用，它正在渗透到各个部门，并深入到家庭日常生活之中。

目前，微机主要应用在过程的自动控制、机床的数值控制、智能终端、智能仪表以及过去计算机无法深入的其他领域如家用电器的控制和教育用装置等。

下面，分几个方面介绍微机的典型应用。

1．微机促进和加速了产品的更新换代，使一大类产品朝着数字化、智能化、多功能和易使用的方向发展。

例如，各种仪器仪表由于装入了微机，使其结构和功能发生了根本性的变革，从而具有精度高、自动控制功能强、结构规范化和接口简单的特点。

又如在通信设备中使用了微机技术后，将从原来的音频通信转变为音频与数字之间的混频通信，从而使通信质量大大提高。

微机在消费类产品中的应用尤为引人注意，从照相机、录音机到缝纫机、电冰箱等正在

形成新一代的智能消费电子产品，与此同时，电子游戏机、语言学习机等新的消费产品也正在不断出现。

2．微机在生产过程的控制、检测和监视中的应用　目前正在从使用传统小型机为主的集中控制趋向使用多台微机的集中分散型控制。即将生产过程分段由微机进行监视和控制，而各微机之间又用小型机或高性能的微机来集中指挥，从而实现生产过程的自动控制，这样比集中系统具有更好的灵活性和可靠性。

3．微机在企、事业事务管理中的应用　在企、事业单位中，数据信息的收集、处理、存储、综合和检索是管理工作中最基本的要求，而这些工作又最适合于计算机来完成，依靠计算机及时提供管理信息，以便提高决策过程中业务处理的正确性和迅速性，从而获取巨大的经济效益。目前使用微机来进行事务处理的内容大致有库存管理、销售管理、总帐会计和分析决策等。

4．微机在教育部门和家庭中的应用　教育部门利用微机可以来辅导解题、辅助教学、辅助实验模拟和辅助教学管理等，目前不仅是大学而且各级中、小学甚至幼儿园也竞相购买微机，用来改革教学方法，提高教育质量。

家用计算机除了供家庭文化娱乐（如电子游戏）外，还可辅导儿童学习，进行日常家庭财务管理等。

5．微机在工程计算、产品设计和科学实验等方面的应用　由于16位、16/32位及32位微计算机的研制成功和投入使用，微机在这些方面的应用也正在逐步得到推广。

我国微机的应用虽然时间不长，但是已经充分显示出它强大的生命力，应用的领域已涉及到工业、农业、商业、交通、运输、教育和邮电等部门，而且各方面都有很好的应用事例。特别是根据我国的国情，对汉字信息处理进行了很多研究并取得了显著的成绩。现在，可利用微机辅导向用户提供汉字系统，有关部门还正在组织力量开发汉字应用软件包。

本书从应用的角度出发，主要介绍在我国使用最广泛的Z 80微处理器的原理及应用和MCS-51单片微型计算机的原理及应用。在叙述方法上注意到微机硬件和软件结合十分紧密的特点，从使用的角度去讲解必须掌握的硬件知识。课程的重点是汇编语言程序设计、接口电路及其编程，有关计算机的基础知识是围绕着这些重点而进行介绍的。

§1-2　计算机的数制和码制

本节主要介绍计算机中有关数值运算的基础知识，如进位计数制、不同进位计数制之间的转换、数与字符的编码方法以及数的符号与小数点的表示方法等。

一、二进制、十六进制和十进制之间的互相转换

在日常生活中，人们采用着各种进位计数制，如六十进制（1min等于60s，1h等于60min），十二进制（1ft等于12in，1年等于12个月）和十六进制（1市斤等于16老两）等，但最为熟悉和常用的是十进制，然而在计算机中通常并不采用十进制，而是采用二进制。这是因为二进制数容易实现、可靠、而且运算简单。

考虑到这部分内容其中有些已在先修课中学过，因此，我们主要通过对比与举例的方法来进行介绍。

（一）二进制数

我们已经知道一个十进制数有两个主要特点：

① 它有十个不同的数字符号，即0、1、2、…、9。

② 低位向高位的进位是逢十进一。

因此，同一个数字符号在不同的位置（或叫数位）所代表的数值是不同的。如353.3其中三个3分别代表不同的数值300、3和0.3，这个数可以写成：

$$353.3=3\times10^{2}+5\times10^{1}+3\times10^{0}+3\times10^{-1}$$

通常称上式中的10为十进制的基数，即基数就是所用数字符号（或称数码）的个数，而称10^{2}、10^{1}、10^{0}、10^{-1}、…等为各数位的位值（或权）。

二进制数与十进制数类似，它也有两个主要特点：

① 它有且只有两个不同的数字符号，即0和1。

② 低位向高位的进位是逢二进一。

因此，同一个数字符号在不同的位置所代表的数值也是不同的。

例如：$(101.1)_2=1\times2^{2}+0\times2^{1}+1\times2^{0}+1\times2^{-1}$

$$=(4+1+0.5)_{10}=(5.5)_{10}$$

亦即二进制就是基数为2的进位记数法，其各数位上的位值（权）分别为2^{n}、…、2^{1}、2^{0}…、2^{-m}。

表1-1列出了一些十进制数与二进制数的对照关系。

表1-1 十进制与二进制对照表

十进制数	二进制数	十进制数	二进制数	十进制数	二进制数
0	0	10	1010	16	10000记为2^{4}
1	1	11	1011	32	2^{5}
2	10	12	1100	64	2^{6}
3	11	13	1101	128	2^{7}
4	100	14	1110	256	2^{8}
5	101	15	1111	512	2^{9}
6	110	16	10000	1024记为1K	2^{10}
7	111	0.5	0.1	2K	2^{11}
8	1000	0.25	0.01	4K	2^{12}
9	1001	0.125	0.001	64K	2^{16}

一个十进制数要转换为二进制数，可以用两种方法。

第一种方法：利用上述对照表，先搞清已知的十进制数中包含哪些二的方幂（数位的位值），出现的方幂所对应的二进制数的数位上取1，不出现的方幂所对应的二进制数的数位上取0。

例1 $(133.75)_{10}=128+4+1+0.5+0.25$

$$=1\times2^{7}+0\times2^{6}+0\times2^{5}+0\times2^{4}+0\times2^{3}+1\times2^{2}$$
$$+0\times2^{1}+1\times2^{0}+1\times2^{-1}+1\times2^{-2}$$
$$=(10000101.11)_2$$

第二种方法：将十进制整数部分和小数部分分别采用“除2取余法”及“乘2取整法”。即一个十进制整数要转换为二进制整数，只需将它除以2并且记下余数，把所得的商再除以2，并记下余数，然后再把所得的商再除以2并记下余数，如此不断继续下去，直到结果得0为止，然后收集余数，即为二进制整数，见例2。

例2 将25转换为二进制数。

$$25 \div 2 = 12 \quad \text{余数 } 1 = K_0$$
$$12 \div 2 = 6 \quad 0 = K_1$$
$$6 \div 2 = 3 \quad 0 = K_2$$
$$3 \div 2 = 1 \quad 1 = K_3$$
$$1 \div 2 = 0 \quad 1 = K_4$$

$\therefore (25)_{10} = K_4K_3K_2K_1K_0 = (11001)_2$

一个十进制小数要转换为二进制小数，只需将它一次又一次地乘2，取其整数，即为二进制小数（这个过程重复继续，直到小数得0为止或转换到所要求的精度为止），见例3。

例3 将0.90625转换为二进制数。

$$0.90625 \times 2 = 1.8125 \quad \text{整数 } 1 = K_{-1}$$
$$0.81250 \times 2 = 1.6250 \quad 1 = K_{-2}$$
$$0.6250 \times 2 = 1.2500 \quad 1 = K_{-3}$$
$$0.2500 \times 2 = 0.5000 \quad 0 = K_{-4}$$
$$0.5000 \times 2 = 1.0000 \quad 1 = K_{-5}$$

$\therefore (0.90625)_{10} = 0_0K_{-1}K_{-2}K_{-3}K_{-4}K_{-5} = (0.11101)_2$

（二）十六进制数

使用二进制数当数据较大时，书写与阅读很容易出错，记忆又困难。因此，通常使用八进制或十六进制来作为二进制的缩写。在微计算机中，目前通用的字长为8位，这恰巧可用2位十六进制表示，因此十六进制应用十分普遍，它已经成为微处理机工业的标准。

1．十六进制数的表示　它也有两个主要特点：①它有16个不同的数字和字母符号0～9以及A、B、C、D、E、F；②数位之间是逢十六进一。因此，在不同的数位，数码所表示的值是不同的。

例如：$(4AC.5E)_{16} = 4\times16^2 + 10\times16^1 + 12\times16^0 + 5\times16^{-1} + 14\times16^{-2} = (1197.1875)_{10}$

即十六进制就是基数为16的进位记数法，它的各数位上的位值（权）分别为$16^n \cdots 16^1 \cdots 16^{-m}$。表1-2列出了部分十进制、十六进制和二进制的对照关系。

2．十六进制与二进制之间的转换　由于4位二进制数恰好表示16个数的组合（$2^4=16$），即一个十六进制数与4位二进制数是完全一一对应的，所以十六进制和二进制间的转换，是十分简单的。

（1）十六进制转换为二进制　只要将每一位十六进制数用相应的4位二进制数代替即可。

例4 将$(5EA.7B4)_{16}$转换为二进制数。

5	E	A	.	7	B	4
↓	↓	↓		↓	↓	↓
0101	1110	1010		0111	1011	0100

$\therefore (5EA.7B4)_{16} = (101\quad 1110\quad 1010.0111\quad 1011\quad 01)_2$

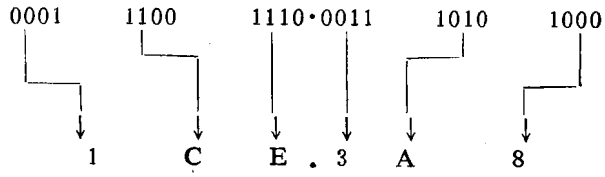
由于在二进制中，最高位的一个0和小数点后最低位的两个0无意义，所以最后从结果中将它们舍去不写。

表1-2 十进制、十六进制和二进制对照表

十进制数	十六进制数	二进制数	十进制数	十六进制数	二进制数
0	0	0	12	C	1100
1	1	1	13	D	1101
2	2	10	14	E	1110
3	3	11	15	F	1111
4	4	100	16	10	10000
5	5	101	17	11	10001
6	6	110	18	12	10010
7	7	111	19	13	10011
8	8	1000	20	14	10100
9	9	1001	21	15	10101
10	A	1010	0.0625	0.1	0.0001
11	B	1011	0.06640625	0.11	0.00010001

（2）二进制转换为十六进制　从小数点开始，分别向左和向右将每4位二进制数(不足4位时，添0补足之)，用1位十六进制数表示即可。

例5　将$(1\ 1100\ 1110\ .\ 0011\ 1010\ 1)_2$转换为16进制数。



$\therefore (1\ 1100\ 1110\ .\ 0011\ 1010\ 1)_2=(1CE\cdot 3A8)_{16}$

由此可见，十六进制在相当大的程度上简化了二进制的书写，因而给人们的学习和交流带来了方便。

3．十进制转换为十六进制　一般有下述两种方法：

（1）直接法　类似于十进制数转换为二进制数的方法，整数部分采用“除16取余法”以及小数部分采用“乘16取整法”。

例6　将$(47632.136)_{10}$转换为16进制数。

整数部分		小数部分	
47632÷16=2977	余 0	0.136×16=2.176	整数 2
2977÷16=186	1	0.176×16=2.816	2
186÷16=11	10	0.816×16=13.056	13
11÷16= 0	11	0.056×16=0.896	0
		0.896×16=14.336	14

$\therefore (47632.136)_{10}\approx(BA10.22D0E)_{16}$

小数部分的转换总有误差，一般转换到所要求的精度为止。

（2）间接法　先将十进制数转换为二进制数，再由二进制数转换为16进制数。

例7　将十进制数$(543.625)_{10}$转换为十六进制数。

先采用“除2取余法”及“乘2取整法”将它转换为二进制数$(1000011111.101)_2$，然后将每4位二进制数用1位十六进制数代替，就可得到十六进制数$(21F.A)_{16}$。

两点说明：

①今后为了便于区别不同数制表示的数，规定在数字后面用一个H表示十六进制数，用Q表示八进制数（这在小型机中通常使用，其表示方法及有关转换都是与十六进制类似的），用B表示二进制数，用D（或不加标志）表示十进制数。如64H，754Q，1101B，369D分别表示十六进制、八进制、二进制和十进制数。另外，规定当十六进制数以字母开头时，为了避免与其他字符相混，在书写时前面加一个数0。如十六进制数B9H，应写成OB9H，（有时又为了避免数0与字母O相混，数0常用"ϕ"代替）。

②几个术语

字　在计算机术语中，一个字是一组二进制数，即作为一个单一的单元对待的一组二进制数，它是计算机中信息使用的基本单元。

字长　指字的二进制数的位长（即位的个数）。

字节　它是作为一个单位看待的8位二进制数。在8位微处理器中，每个字由一个字节组成；在16位微处理器中，每个字由2个字节组成，见图1-1。

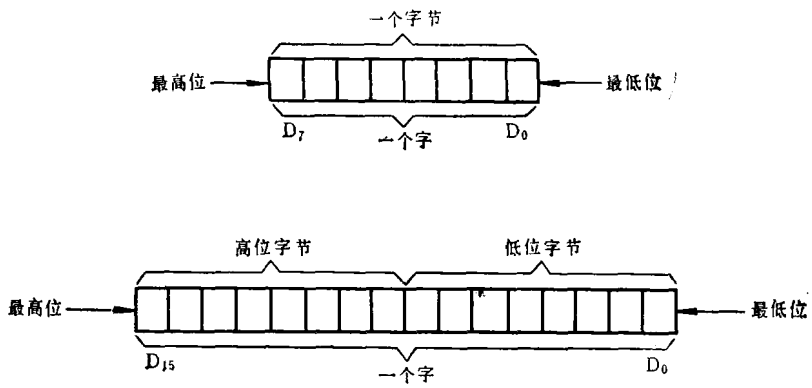


图1-1　字和字节

二、二进制编码

在计算应用中，除了经常使用阿拉伯数字0～9之外，尚需使用一些字母如A、B、…、Z，符号如+、-、*、/…以及一些控制信号等来跟计算机打交道。然而，在计算机内部所有信息都是采用二进制代码的，因而就存在着如何利用二进制代码来表示数、字母以及符号等，这就是所谓二进制编码。

（一）二进制编码的十进制数（BCD码）

1、8421 BCD码　在计算机中使用二进制代码工作，尽管有许多机器设备方面的优点，但是人们要认出和转换二进制数的大小是颇花功夫的。因而提出了一个比较适合于十进制系统的二进制代码的特殊形式，即用4位二进制代码来表示1位十进制数，一般称之为十进制数的二进制代码表示法，简称BCD码。

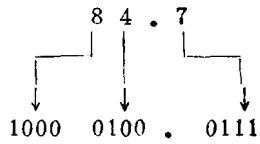
由于4位二进制数从0000到1111可以表示16个数，因此原则上可任意选择其中的10种来编码0到9的10个数字，但通常采用与0～9各数字所对应的二进制数作为代码，这称为8421 BCD码，见表1-3。

可见，这种BCD码与十进制数的关系相当直观，它们之间的转换也是十分简单和直截了当的，只需将十进制数的各位数字用与其对应的一组（4位）二进制数代替即可。

表1-3 8421 BCD码

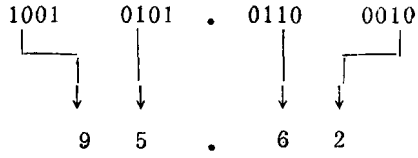
十 进 制 数	8421 BCD码	十 进 制 数	8421 BCD码
0	0000	9	1001
1	0001	10	0001 0000
2	0010	11	0001 0001
3	0011	12	0001 0010
4	0100	13	0001 0011
5	0101	14	0001 0100
6	0110	15	0001 0101
7	0111	16	0001 0110
8	1000	17	0001 0111

例8 将十进制数84.7转换为BCD码



$\therefore (84.7)_{10}=(1000\ \ 0100.0111)_{BCD}$

例9 将BCD码1001 0101.0110 0010 转换为十进制数。



$\therefore (1001\ \ 0101.0110\ \ 0010)_{BCD}=(95.62)_{10}$

要注意BCD码与真正的纯二进制是不同的，它貌似二进制，实为十进制，常用在电子计算机的输入输出设备中，作为计算机用的二进制与人们日常用的十进制之间的一种过渡性的编码，以简化人机联系。

2．二—十进制调整 由于BCD码仅是将每个十进制数用4位二进制数来表示，而每组（4位二进制）之间仍然是逢十进一的。因此，若将这种BCD码直接交给计算机去运算，由于计算机总是将数作为二进制数来处理的，因而结果就可能出错。

例10 求BCD码的7＋6。

$$\begin{array}{rll} & 0111 & 7\text{的BCD码} \\ + & 0110 & 6\text{的BCD码} \\ \hline & 1101 & \end{array}$$

1101为非法的BCD码，因此结果出错，而7＋6＝13的正确的BCD码应为$(0001\ \ 0011)_{BCD}$。

例11 求BCD码的9＋8。

$$\begin{array}{rll} & 1001 & 9\text{的BCD码} \\ + & 1000 & 8\text{的BCD码} \\ \hline & 10001 & \end{array}$$

10001是11的BCD码，结果同样出错，而9＋8＝17的正确的BCD码应为$(0001\ \ 0111)_{BCD}$。

为了得到正确的BCD码的运算结果，必须进行二—十进制调整，规则如下：

①4位（二进制）一组，两个BCD数相加结果大于9（即1001）时，则需对该4位进行"加6修正"。

②4位（二进制）一组，两个BCD数相加结果等于和大于16（即10000）时，则需对该4位进行"加6修正"。

下面通过具体例子来说明其正确性。

例12 用BCD码求38+46，并进行调整

	0011	1000	38
+	0100	0110	46
	0111	1110	低4位大于9
+		0110	"加6修正"
	1000	0100	84

例12按规则1调整后，显然结果是正确的。因为BCD码相加，是4位（二进制）一组按十进制加法规则逢十进一。但是，对于二进制数来说，第4位向高位（第5位）进位时必须要满16才能进位，这样两个进位相差6，因此，为了由10得到一个进位，需将BCD码的和再加上一个6，以帮助产生正确的进位，从而去跳过六个非法的BCD码（1010～1111）。

例13 用BCD码求84+85，并进行调整。

	1000	0100	84
+	1000	0101	85
	10000	1001	高4位等于16
+	0110		"加6修正"
	10110	1001	169

例13按规则2修正后，显然结果是正确的。因为当BCD码的和大于、等于16时，将产生一个进位，这个进位对二进制数来说，相当于16，但是，在这里是作为BCD码的进位，它只相当于10，这样一来，得到的BCD的和比真正的和数少了一个6，所以必须加6修正。

例14 用BCD码求99+3。

	1001	1001	99
+		0011	3
	1001	1100	低4位大于9
+		0110	"加6修正"
	1010	0010	高4位大于9
+	0110		"加6修正"
	10000	0010	102

这里为了说明问题，所以举了较多的加法例子，实际在计算机中有一条十进制调整指令，无论对于加法或减法，机器都能按照规则自动进行调整（加法时，"加6修正"；减法时，"减6修正"），人们只管放心使用就是了。

另外，BCD码除了采用上述方法调整以外，也可以在交付计算机运算以前，先将BCD码转换为二进制(称为"十翻二")，然后让计算机对二进制运算，运算以后再将二进制转换为BCD码(称为"二翻十")。这种"十翻二"、"二翻十"的转换，可以通过编制程序后交给计算机本身去完成（详见第四章）。

总之，采用BCD码可简化人机联系，而一些繁琐的调整或转换工作，可由计算机自身去完成。

（二）字符的ASCII码

在计算机应用中，通常起码有下列字符需要用二进制代码来编码，如26个英文字母A、B、…、Z；10个阿拉伯数字0、1、…、9；运算符号+、-、*、/…；标点符号，、；。？…以及一些数据控制的特殊字符，如换行、换页、回车等。

若采用6位、7位或8位二进制代码来表示，那么可以分别编码64、128或256个不同的字符。

在微计算机系统中最常用的一种编码是ASCII码，ASCII是美国信息交换标准代码的缩写，它采用7位二进制代码来对字符进行编码（第8位可用作奇偶校验位），见附录Ⅰ。

表示数字、字母或控制功能的7位ASCII码是由3位和4位一组组成的，其格式如下(其中右边的0为最低位)：

6 5 4	3 2 1 0
3位组	4位组

例如，阿拉伯数字0～9的ASCII码分别为30H～39H，英文大写字母A、B、…、Z的ASCII码是从41H开始依次往下编排。

当使用6位ASCII代码时，3位一组的减到2位一组，它去掉了26个英文小写字母。

三、带符号数的定点表示法

实际的数值是带有符号的，即可能是正数，也可能是负数；而数值本身可以包括整数也可包括小数。那么，在计算机中是如何表示数的正负号以及确定小数点的位置呢？

（一）数的定点表示法

在计算机中，数有两种表示方法，即定点表示和浮点表示。定点表示，就是指小数点在数中的位置是固定不变的；浮点表示，就是指小数点在数中的位置不是固定的而是浮动的。在本书中，我们只考虑数的定点表示法。

定点表示也有两种方法：

方法1规定小数点固定在数据的末尾，即认为参加运算的数据全部均为整数。

方法2规定小数点固定在符号位与数据位之间，即认为参加运算的数据全部均为小数。

下面用8位二进制来说明定点数的两种表示，见图1-2。

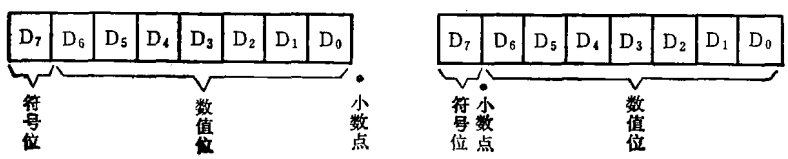


图1-2　定点数的两种表示方法

其中，最高位D_7为符号位，D_6～D_0为数值位，而D_0为最低位。

定点数的这两种表示方法在计算机中均有采用，究竟采用哪种方法，完全是事先约定的。我们则约定采用第一种方法，即参加运算的数全部视为整数。为此，在运算之前，必须对数据进行适当的处理。

（二）数的符号表示法

由于在计算机中，二进制数码是用双稳态元件来表示的，因此，对于数的符号“＋”或“－”很容易想到也用数码来表示。即：

用数码“0”表示正数的符号“＋”。

用数码“1”表示负数的符号“－”。

例15 二进制数$x_1=+0001101$及$x_2=-0001011$在定点机中的表示分别为：

x_1：00001101 ↓ 表示“＋”号　　x_2：10001011 ↓ 表示“－”号

这就是说，数的符号在机器中也数码化了。我们把一个数在机器中的表示形式叫机器数，而把原来的实际数本身叫做机器数的真值。

例16 已知定点机中的两个机器数为01001001及10101001，则它们的真值分别为＋1001001与－0101001。

（三）原码、反码和补码

既然一个数的数值与符号全都是数码，那么当对这种机器数进行运算操作时，符号位该如何处理？能不能也同数值位一道参加运算操作呢？为了妥善处理这些问题，引出了机器数的三种不同的编码形式，即原码、反码及补码。

下面我们将一个数x的原码、反码和补码分别记作：$[x]_{原}$、$[x]_{反}$和$[x]_{补}$。

1。原码　设$x=x_1x_2\cdots\cdots x_{n-1}$，其中$x_i$为一位二进制数，$i=1,\cdots,(n-1)$。

则
$$[x]_{原}=\begin{cases}0\ x_1\ x_2\cdots x_{n-1} & \text{当 } x\geqslant 0\\ 1\ x_1\ x_2\cdots x_{n-1} & \text{当 } x\leqslant 0\end{cases}$$

即一个数的原码，就是数的数值部分不变，而仅仅用0及1分别来表示数的符号“＋”及“－”的机器数。

例17 设　$x_1=+100111$　　$x_2=-0001001$

则 $[x_1]_{原}=0100111$　　$[x_2]_{原}=10001001$

例18 根据定义，数“0”的原码有两种不同形式：

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

显然，原码表示法简单易懂，而且与真值转换方便，但是，却使机器进行加法和减法运算变得复杂了。例如，当两个数相加时，如果两数的符号相异，那么，它们之间的运算，实际上是数值部分相减；而在做减法时，还必须首先比较两个数中哪个数的绝对值大，然后才能从绝对值较大的数中减去绝对值较小的数，而差值的符号则与绝对值较大的数的符号一致。这就是说，采用原码表示后，势必将使机器的结构相应地复杂化或增加机器的运算时间。因此，原码只流行了很短暂的时间就被抛弃了。为了解决这个问题，就引进了数的补码，它可使正、负数的加法或减法运算简化为单纯的相加运算，而为了求补码的方便又引入了反码的概念。

2．反码　设 $x=x_1x_2\cdots x_{n-1}$，其中x_i为一位二进制数，$i=1,\cdots,(n-1)$。

则
$$[x]_{反}=\begin{cases}0\ x_1\ x_2\cdots x_{n-1} & \text{当 } x\geqslant 0\\ 1\ \bar{x}_1\ \bar{x}_2\cdots\bar{x}_{n-1} & \text{当 } x\leqslant 0\end{cases}$$

其中
$$\bar{x}_i=\begin{cases}0 & \text{当}x_i=1\\ 1 & \text{当}x_i=0\end{cases}$$

即对于正数，反码与原码相同，符号位为0，其余位为数值位本身；对于负数，反码的符号位为1，其余位为数值位按位取反（即将1变为0，0变为1）。

例19　设　$x_1=+1000001$　　$x_2=-0101010$

则 $[x_1]_{反}=01000001$　　$[x_2]_{反}=11010101$

例20　数0，根据定义，反码有两种不同形式：

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

例21　已知 $[x_1]_{反}=01111001$　　$[x_2]_{反}=10000111$

则　　$x_1=+1111001$　　$x_2=-1111000$

3．补码　上面已经谈到，引入补码的概念，目的在于将加、减运算简化为单纯的相加运算。

（1）同余的概念和补码　设有两个数 $a=17$，$b=27$，若用10去除 a 和 b，则它们的余数均为7，我们称17和27在以10为模时是同余的，并记以：

$$17=27 \pmod{10}$$

或者说17和27在以10为模时是相等的。由同余的概念，不难得出：

$$a+M=a \pmod{M}$$
$$a+2M=a \pmod{M}$$

因此，当 a 为负数时，如 $a=-4$，在以10为模时，有：

$$-4+10=-4 \pmod{10}$$
$$6=-4 \pmod{10}$$

即在以10为模时，-4与$+6$是相等的。我们称$+6$为-4的补码，或者说$+6$与-4对模10来说互为补数。同理，7为-3的补码，8为-2的补码等。有了补码的概念，就可将减法转化为加法（加补码）来进行。如：

$$7-4=7+6 \pmod{10}$$

即在以10为模时，7减4可以通过7加-4的补码6来进行，而所得的结果是一样的(只要在加补码6时，将所产生的进位10舍弃即可，这也正是以10为模的含意)。

更通俗的例子是校对手表，例如当我们听到广播中北京时间7点整的时候，发现自己的手表停在9点的地方，这时可用两种方法来校正自己的手表。见图1-3。

① 倒拨2h　　$9-2=7$

② 顺拨10h　　$9+10=7$

这是因为手表是以12为模的（即超过12点就丢掉

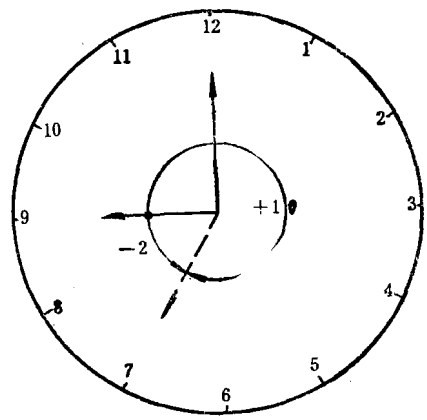


图1-3　校表