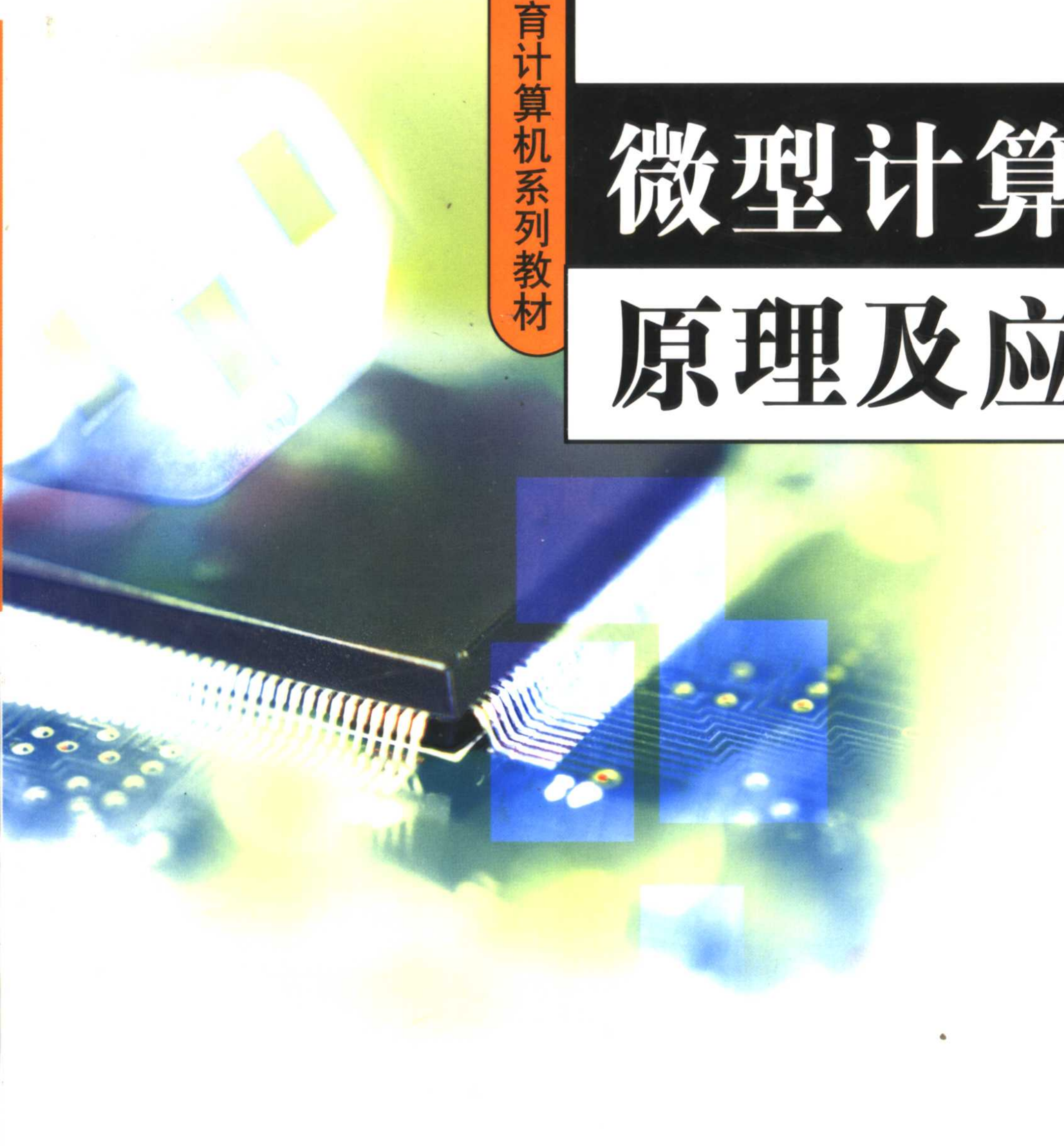
21世纪中等职业技术教育计算机系列教材

史建华 主编

微型计算机原理及应用



清华大学出版社

21世纪中等职业技术教育计算机系列教材

史建华 主编

史建华 赵丽艳
刘妍东 史嘉松 编著

微型计算机原理及应用

清华大学出版社

北京

内 容 简 介

本书主要是面向中等职业学校的学生，为他们提供一本比较合适的微型计算机原理教材。

考虑到读者的基础知识状况，本书由浅入深(比较通俗)地先介绍微型计算机的基础知识及基本逻辑部件，接着以模型机的形式通俗地介绍微型计算机的基本工作原理，使学生对现代微机的概貌有一个初步的认识。在这个基础上，以与现代微机机型相近的 8086/8088 为样本讲解一般工作原理。考虑到学是为了用和中等职业教育的要求，叙述以最小模式为主，重点放在应用上。

“应用”必须有接口，有接口就必须会编程应用，故在介绍基本原理之后，又用一定的篇幅介绍 8086 汇编语言源程序的基础知识及简单程序设计。最后用较多的章节介绍微机应用中常用到的多种接口芯片，为学生从事该项工作打下一定基础。

本书可作为中等职业技术教育的计算机教材，也可供从事该项工作的工程技术人员参考。

图书在版编目(CIP)数据

微型计算机原理及应用/史建华主编. —北京：清华大学出版社，2003
(21 世纪中等职业技术教育计算机系列教材)
ISBN 7-302-07702-9

Ⅰ. 微… Ⅱ. 史… Ⅲ. 微型计算机—专业学校—教材 Ⅳ. TP36

中国版本图书馆 CIP 数据核字(2003)第 108573 号

出 版 者：清华大学出版社　　**地　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
责任编辑：王敏稚
印 刷 者：北京人民文学印刷厂
装 订 者：北京鑫海金澳胶印有限公司
发 行 者：新华书店总店北京发行所
开　本：185×260　**印张**：23　**字数**：526 千字
版　次：2004 年 2 月第 1 版　2004 年 5 月第 2 次印刷
书　号：ISBN 7-302-07702-9/TP · 5640
印　数：3001～5000
定　价：29.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770175-3103 或(010)62795704。

21世纪中等职业技术教育计算机系列教材

丛 书 编 委 会

丛书前言

在新世纪里，职业教育逐渐成为我国国民教育的一种重要形式，职业应用型人才日益为各行各业所急需。继中等职业教育、高等职业教育之后，本科职业教育、研究生职业教育也已提到了议事日程并得到了开展。多年来，中等职业教育已为社会输送了大批的应用型人才，为经济建设和社会发展做出了很大的贡献。可以说，中等职业教育是整个职业教育体系的基础。因此，大力开展中等职业教育教学工作，建设适用的中等职业技术教育教材，是一项非常有意义的工作。

由清华大学出版社出版的《中等职业技术教育计算机教材》丛书自1997年问世以来，得到了全国各地广大师生的广泛使用，产生了很好的社会影响。在我国加入WTO后，对职业学校毕业生计算机知识和技能的掌握程度，尤其是计算机操作能力，提出了更高的要求。为了搞好中等职业教育计算机课程系列教材建设，清华大学出版社从1999年初开始就对各地的职业教育教研部门和职业学校进行了广泛的调研，征求了广大用户的意见和建议，根据中华人民共和国教育部最新颁发的计算机系列课程的教学大纲，在原丛书成功的基础之上，编写了这套《21世纪中等职业技术教育计算机系列教材》。

本套教材在编写理念上首先注意借鉴国外的先进教学方法和手段，以“学”为中心进行教学设计，有利于互动教学和任务驱动教学；注重中等职业教育计算机教学的特点，在兼顾知识科学性和系统性的基础上，突出实用性和操作性；教材的结构有利于教师采用建构主义模式教学，在内容选择、语言表达、例题设计上力求符合学生的认知特点，做到深入浅出、层次分明、步骤翔实、易学易用。

侧重上机(或实习)指导，将上机(或实习)指导作为主要内容之一是本系列教材的又一特色。书中提供的上机(或实习)指导，内容通俗易懂，操作循序渐进。部分操作练习提供了详细的参考步骤，其目的是为了举一反三；另一部分操作练习不提供参考步骤，鼓励学生自己动手实践，以便牢固地掌握计算机实用技术。

本套教材首批将推出十余种，还将陆续为使用本教材的教师提供配套的“电子教案”、“教学素材”、“题库和模拟考试系统”等作为教学参考资料。

本系列教材的主要作者均为从事计算机教育10年以上的计算机高级和特级教师，来自全国部分职业学校计算机教学的第一线，其中既有负责全国中等职业教育计算机教育科研的专家，也有市、区级的计算机名师、学科带头人以及市、区计算机教研员，他们有丰富的计算机教育、教学经验，并出版过多部计算机教育的书籍。

衷心希望广大师生、教育科研人员和业内人士关心这套教材的建设，多提宝贵意见。如果教材的结构、内容有不妥之处，请及时反映给我们，也可与清华大学出版社的编辑联系，以便再版时作必要的修改和补充。

21世纪中等职业技术教育计算机系列教材

丛书编委会

2002年1月

前　言

学习计算机，一要学计算机基本组成原理，二要学计算机的操作，学习内容包括理论和实践操作两个方面。计算机是一门应用型学科，操作性强。随着计算机在社会各个领域的应用越来越广泛，对计算机操作能力的要求也越来越高。计算机课的教学要面向社会、面向市场，既要让学生学习计算机知识，又要对学生进行计算机操作技能的训练，重点是侧重操作和技能性方面的训练。

选好教材、用好教材是搞好计算机教学的重要保证。出版一本适合各类职业高中、中专使用的教材，就是我们编写本书的初衷。

根据职业高中、中专各专业计算机教学的特点，本书在注重系统性、科学性的基础上重点突出了实用性和操作性，重点讲述计算机的基本概念和基本操作方法。按照由浅入深的教学原则，采取循序渐进的教学方法，力求通俗而不肤浅，深入而不玄奥。对重点概念、重要的操作技能，力争讲深讲透。

侧重上机操作，将上机指导作为主要内容之一是本教材的又一特色。每章后的上机指导内容通俗易懂，操作循序渐进。每个上机指导包括目的与要求、软硬件环境和操作步骤三部分。每章的上机指导配合小结、习题，使学生在动脑、动手过程中牢固地掌握计算机实用技术。

本书再版时，对第 2 章中的存储器 DRAM 部分作了适当删节。因为现在的动态 RAM 刷新过程对职高及中专学生来讲有一定难度，而目前市场上的 DRAM 芯片，大多数又将刷新电路做在芯片内，可视同 RAM 一样与 CPU 连接，所以将重点放在 RAM 与 CPU 的连接上，这有利于学生掌握这部分内容。

本书再版时对第 9 章的键盘接口、LED 显示接口也作了改动，更利于学生掌握和应用。

本书共分 10 章，并附有习题。在第 10 章专门安排了上机操作及接口实验的内容，由选用本教材的学校根据所具备的条件进行选用。本教材建议 96 学时，上机操作和实验的学时不包括在内。

本书修改时，孔繁雄、侯燕平、常维东、茹小康、卞国华、郑雪鹏等参与了修改工作，编者在此谨表谢意。

对教材内容中不妥或需要改进之处，殷切希望广大师生向我们指出，以便修改和补充。

编　者

2003 年 11 月于北京

目　　录

第1章　计算机的基础知识 …… 1

1.1　数制 …… 1

1.2　二进制数的特点 …… 4

1.3　数制间的转换 …… 4

1.4　二进制数的运算规则 …… 7

1.5　计算机中数的定点与浮点表示 …… 9

1.6　原码、补码和反码 …… 11

1.7　补码的加减运算 …… 13

1.8　不带符号数的加减运算 …… 16

1.9　溢出判断 …… 18

1.10　常用编码 …… 20

1.11　二进制数的运算及其加法电路 …… 22

习题1 …… 27

第2章　微型计算机的基本逻辑部件 …… 28

2.1　寄存器 …… 28

2.2　三态输出电路 …… 33

2.3　总线结构 …… 35

2.4　半导体存储器 …… 36

习题2 …… 57

第3章　微型计算机的基本工作原理 …… 59

3.1　微型计算机结构的简化形式 …… 59

3.2　指令系统 …… 63

3.3　程序设计 …… 64

3.4　执行指令的例行程序 …… 67

3.5　控制部件 …… 70

3.6　微型计算机功能的扩展 …… 73

3.7　初级程序设计举例 …… 77

3.8　控制部件的扩展 …… 83

3.9　现代技术在微型计算机中的应用 …… 85

习题3 …… 88

第 4 章　微处理器 …… 89
4.1　微处理器的概述 …… 89
4.2　8086/8088 CPU 结构 …… 90
4.3　8086 CPU 引脚信号和工作模式 …… 96
4.4　8086/8088 的主要操作功能 …… 110
习题 4 …… 124

第 5 章　微型计算机的指令系统 …… 126
5.1　概述 …… 126
5.2　指令格式 …… 126
5.3　8086/8088 的寻址方式 …… 127
5.4　数据传送指令 …… 132
5.5　算术运算指令 …… 137
5.6　逻辑运算指令和移位指令 …… 142
5.7　串操作指令 …… 147
5.8　控制转移指令 …… 152
习题 5 …… 161

第 6 章　微型计算机的程序设计 …… 163
6.1　简单程序设计步骤 …… 163
6.2　伪指令 …… 166
6.3　系统功能调用 …… 175
6.4　汇编语言程序结构 …… 178
6.5　简单程序设计 …… 179
6.6　分支程序设计 …… 181
6.7　循环程序设计 …… 186
6.8　子程序设计 …… 193
习题 6 …… 206

第 7 章　输入/输出接口 …… 208
7.1　微型计算机的输入/输出接口 …… 208
7.2　并行通信与并行接口 …… 212
7.3　可编程并行通信接口芯片 8255A …… 214
7.4　串行通信及串行接口 …… 227
7.5　可编程串行通信接口芯片 8251A …… 231
习题 7 …… 241

第8章 中断控制器、计数/定时控制器及DMA控制器 …… 243
8.1 可编程中断控制器8259A …… 243
8.2 可编程计数/定时控制器8253 …… 262
8.3 可编程DMA控制器8257 …… 275
习题8 …… 285

第9章 A/D,D/A及简易键盘、显示接口设计 …… 287
9.1 D/A转换器的工作原理 …… 287
9.2 D/A转换器的性能参数和术语 …… 291
9.3 DAC0832芯片 …… 293
9.4 A/D转换器的工作原理 …… 298
9.5 A/D的性能参数和技术术语 …… 304
9.6 ADC0809和AD570 A/D芯片 …… 304
9.7 简易键盘接口和LED显示器接口的设计 …… 311
习题9 …… 320

第10章 汇编语言的上机操作及实验 …… 321
10.1 汇编语言程序的上机过程 …… 321
10.2 汇编语言程序设计实验 …… 331
10.3 接口实验 …… 346

第 1 章

计算机的基础知识

本章重点要求掌握各种数制之间的转换方法，理解补码的概念与常用进制数的补码求法，以及求补的概念和求$[-y]_{补}$的方法，了解溢出的概念和判别有符号数及无符号数的溢出方法。

1.1 数　　制

按照进位的方法进行计数，称进位计数制，简称数制。在计算机中常用的进制有十进制、二进制、八进制和十六进制。

1. 十进制数(decimal number)

在日常生活中人们常用的是十进制数。十进制数的数值部分是用 10 个不同的数字符号 0,1,2,3,4,5,6,7,8,9 来表示的，我们把这些数字符号叫做数码。在数中一个数码所代表的意义与它所处的位置有关。例如，78.42 这个数，小数左边的第一位代表个位，表示它本身的数值是 8；左边的第二位是十位，表示 7×10^1；而小数点右边的第一位 4 表示 4×10^{-1}；第二位是 2，表示 2×10^{-2}。因此这个数可以写成：

$$78.42=7\times10^1+8\times10^0+4\times10^{-1}+2\times10^{-2}$$

一般，对任意一个正的十进制数 S，可以表示为：

$$S=K_{n-1}(10)^{n-1}+K_{n-2}(10)^{n-2}+\cdots+K_0(10)^0+K_{-1}(10)^{-1}+K_{-2}(10)^{-2}+\cdots+K_{-m}(10)^{-m}$$

或
$$S=\sum_{j=-m}^{n-1}K_j(10)^j$$

其中，K_j 可以是 0,1, …,9 这十个数码中的任意一个，它由数 S 决定；m,n 为正整数，n 为小数点左边的位数，m 为小数点右边的位数。括号内的 10 称为计数制的基数，表示逢“十”进位。K_j 为权系数，$K_j(10)$为本位的值。

一般地说，若 P 是大于 1 的整数，则任一数 N 总可以用下式表示：

$$N=K_{n-1}(P)^{n-1}+K_{n-2}(P)^{n-2}+\cdots+K_0(P)^0+K_{-1}(P)^{-1}+K_{-2}(P)^{-2}+\cdots+K_{-m}(P)^{-m}$$

或
$$S=\sum_{j=-m}^{n-1}K_j(P)^j$$

其中，K_j 可以是$(P-1)$中的任意一个数码；m,n 为正整数；P 为基数。当 P 取不同的数值时，N 为不同进制的数。

$P=10$ 就是十进制的表示形式，N 称为十进制数；

$P=8$ 就是八进制的表示形式，N 称为八进制数；

$P=2$ 就是二进制的表示形式，N 称为二进制数；

为区别不同进制的数，十进制数后缀为 D，或无后缀；二进制数后缀为 B；八进制数后缀为 O(Octal)。这些后缀为该进制的第一个英文字母，因 0(zero)与 O(Octal)容易相混，常用形状相近的 Q 作八进制数的后缀，十六进制数的后缀为 H。

2. 二进制数(binary number)

主要特点是：

(1) 它只有两个不同的数码，即“0”和“1”。

(2) 它是逢“二”进位的。如对十进制数 1+1=2，而对二进制数 1+1=10B。二进制数可通过按权展开相加法，化为十进制数，如：

$$\begin{aligned}1111.11\text{B} &= 1\times 2^3+1\times 2^2+1\times 2^1+1\times 2^0+1\times 2^{-1}+1\times 2^{-2}\\ &= 8+4+2+1+0.5+0.25\\ &= 15.75\text{D}\end{aligned}$$

一般地说，任意一个二进制数 N(正的，或负的)，可以表示为：

$$\begin{aligned}N &= \pm(K_{n-1}\times 2^{n-1}+K_{n-2}\times 2^{n-2}+\cdots+K_0\times 2^0+K_{-1}\times 2^{-1}\\ &\quad +K_{-2}\times 2^{-2}+\cdots+K_{-m}\times 2^{-m})\\ &= \pm\sum_{j=-m}^{n-1}K_j(2)^j\end{aligned}$$

其中，K_j 只能取 1 或 0，由具体的数 N 确定。m,n 为正整数，n 为小数点左边的位数，m 为小数点右边的位数；“2”是二进制的基数，表示“逢 2 进 1”，故称二进制，见表 1-1。

3. 八进制数(octal number)

主要特点是：

(1) 它有 8 个不同的数码，即 0,1,3,4,5,6,7。

(2) 它是逢“八”进位的。

如上所述，任意一个八进制数 N，可以表示为：

$$\begin{aligned}N &= \pm(K_{n-1}\times 8^{n-1}+K_{n-2}\times 8^{n-2}+\cdots+K_0\times 8^0+K_{-1}\times 8^{-1}\\ &\quad +K_{-2}\times 8^{-2}+\cdots+K_{-m}\times 8^{-m})=\pm\sum_{j=-m}^{n-1}K_j(8)^j\end{aligned}$$

其中，K_j 可以是 0～7 中的任何一个数，取决于数 N；m,n 为正整数，n 为小数点左边的位数，m 为小数点右边的位数；8 为基数，故称八进制。

由于数 8 与数 2 有以下关系：$8^1=2^3$，因此，一位八进制数相当于三位二进制数，它们之间的关系见表 1-1。

根据这种对应关系，二进制与八进制之间的转换十分简单。只需从小数点向左和向

右每三位分为一组，每组用一位八进制数表示即可。注意，不足三位者应补 0，凑成三位一组。如：

```
010 100 101 . 010 111 010
 |   |   |     |   |   |
 2   4   5  .  2   7   2
```

若要将八进制数转换为二进制数，只需用三位二进制数代换为一位八进制数即可。例如，八进制 367.505Q，可转换为二进制数 011110111.101000101B。

表 1-1 各种数制对应表

十进制	十六进制	八进制	二进制	十进制	十六进制	八进制	二进制
0	0	0	0000	9	9	11	1001
1	1	1	0001	10	A	12	1010
2	2	2	0010	11	B	13	1011
3	3	3	0011	12	C	14	1100
4	4	4	0100	13	D	15	1101
5	5	5	0101	14	E	16	1110
6	6	6	0110	15	F	17	1111
7	7	7	0111	16	10	20	10000
8	8	10	1000				

4. 十六进制数(hexadecimal number)

主要特点是：

(1) 它有 16 个不同的数码，即 0～9，A、B、C、D、E、F。它与十、二、八进制数之间的关系见表 1-1。

(2) 它是逢“十六”进位的。

如上所述，任意一个十六进制数 N，可以表示为：

$$
\begin{aligned}
N &= \pm (K_{n-1} \times 16^{n-1} + K_{n-2} \times 16^{n-2} + \cdots + K_0 \times 16^0 + K_{-1} \times 16^{-1} \\
&\quad + K_{-2} \times 16^{-2} + \cdots + K_{-m} \times 16^{-m}) \\
&= \pm \sum_{j=-m}^{n-1} K_j (16)^j
\end{aligned}
$$

其中，K_j 可以是 0～F 之间的任意一个，取决于数 N；m，n 为正整数；“16”为基数，故称十六进制。

由于数 16 与数 2 之间的关系为：$16^1 = 2^4$，因此，一位十六进制数相当于四位二进制数。只要我们了解这种关系，十六进制数与二进制数之间的转换也十分简单。例如，二进制数(1111111000111.100101011B)可用下述方法转换为十六进制数，即二进制数以小数点为界向左、向右每四位数分为一组，不足四位者用 0 补齐四位，然后每组的四位二进制数用一位十六进制数表示即可。如：

111111000111.100101011B = 1FC7.958H

又如，十六进制数是 3AB.4AH，可方便地转换成二进制数：

001110101011.01001010B

目前，微机汇编语言中，常用十六进制，小型机中汇编语言常用八进制，目的都是为书写较短，便于阅读程序。至于八进制数与十六进制数之间的转换，通过二进制数也可方便

地进行。

1.2 二进制数的特点

从前节可知，同一个数用二进制表示比用十进制表示位数多得多。粗略估计，前者约为后者的 3 倍左右。既然人们习惯于用十进制数，书写又方便，而二进制数书写起来位数太长，又不便于阅读，那么为什么在计算机中要采用二进制数呢？这是由二进制数本身的特点决定的。因为计算机惟一能识别的是二进制数，这是问题的实质。八进制、十六进制数的引入，主要是为了书写方便，仅仅是一种手段而已。

二进制数制与其他数制相比有以下特点：

(1) 数制的状态简单，容易表示

二进制只有“0”，“1”两种状态，可以用具有两个稳态的元件表示，如晶体管导通或截止，电平的高与低，脉冲的有和无等，均可分别用来表示“1”和“0”状态。这种简单的状态工作可靠，抗干扰能力强。

(2) 运算规则简单

二进制运算的规则极为简单（以后介绍），故在计算机中表现二进制运算的线路也大大简化了。

(3) 可以节省设备

如果采用十进制数制表示 0～9 之间的数，需要 1 位，这一位共需十个设备状态。若采用二进制数制表示，需要 4 位，每位只需两个状态，总共八个设备状态。而且这八个设备状态所能表示的数的范围可达 0000～1111，即 0～15，这说明二进制数制可以节省设备。

(4) 可以选用逻辑代数这一数学工具对计算机逻辑线路进行分析和综合

便于机器结构的简化。

在计算机内部采用二进制操作时，为解决书写冗长、阅读不便，常用八进制和十六进制表示。但人们习惯用十进制，故需介绍一下十进制转换为二进制的问题。

1.3 数制间的转换

1. 二进制数与十进制数之间的相互转换

(1) 二进制数转换为十进制数

这种转换十分简单，只要将二进制数按“权”展开相加即可。

例如：11001.1001B

$$=1\times2^4+1\times2^3+0\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+0\times2^{-3}+1\times2^{-4}$$

$$=16+8+1+0.5+0.0625=25.5625\text{D}$$

二换十的规则就是要算出二进制数每一位为“1”时，分别代表的十进制数，然后将这些数相加，即按“权”相加。

(2) 十进制数转化为二进制数

十进制数转换为二进制数时，要把整数部分和小数部分分别换算，然后再相加即可。

① 整数换二利用除 2 取余法

【例 1-1】 求 215 的二进制数。

2 | 215	
2 | 107	……余 1(最低位)
2 | 53	……余 1
2 | 26	……余 1
2 | 13	……余 0
2 | 6	……余 1
2 | 3	……余 0
2 | 1	……余 1
0	……余 1(最高位)

所以,215D=11010111B

【例 1-2】 求 92D 的二进制数(用推算法)。

如果你对十换二之间的关系很熟悉,也可以用推算法进行十换二,即找一个与所要转换的十进制数最接近的 2 的乘方不断地进行试减,直至减到小于 1 为止。

92	
−64	$\rightarrow 2^6$
28	
−16	$\rightarrow 2^4$
12	
−8	$\rightarrow 2^3$
4	
−4	$\rightarrow 2^2$
0	

所以,$92D = 64+16+8+4 = 1\times2^6+0\times2^5+1\times2^4+1\times2^3+1\times2^2+0\times2^1+0\times2^0$

$= 1011100B$

② 小数的十换二

采用乘 2 取整法,即用 2 不断地去乘要转换的十进制数,直到小数部分为 0 或满足所要求的精度为止。把每次乘积的整数部分(不参加下次乘),以初整数为最高位(没有整数的取 0),依次排列,即得所转换的二进制小数。

【例 1-3】 求 0.6875D 的二进制数。

0.6875	
× 2	
1.3750	……整数部分为 1,相当于 $K_{-1}=1$
0.375	
× 2	
0.750	……整数部分为 0,相当于 $K_{-2}=0$
× 2	
1.500	……整数部分为 1,相当于 $K_{-3}=1$
0.5	
× 2	
1.0	……整数部分为 1,相当于 $K_{-4}=1$

所以，0.6875D=0.1011B

有的数在十换二时，整个计算过程会无限制地进行下去，这时可以根据精度的要求，选取适当的位数。

对于具有整数和小数部分的十进制数，只要对整数和小数分别进行转换为二进数制数，然后合并起来即是整个的结果。

2. 八进制数和十进制数之间的相互转换

一般地说，任意进制数和十进制数之间的转换原理和方法，同二进制数与十进制数之间的转换相似，区别仅在于基数 2 换成相应的基数（如 8，16 等）。

（1）八换十

它与二换十相类似，即将八进制数转按"权"展开，然后相加即可。

例如：

$$
\begin{aligned}
51.6Q &= 5\times 8^1+1\times 8^0+6\times 8^{-1}\\
&=40+1+0.75\\
&=41.75D
\end{aligned}
$$

（2）十换八

要把整数和小数分别进行转换。

【例 1-4】 将十进制数 75.6875D 转换为八进制数。

① 整数部分采用除 8 取余法

$$
\begin{array}{r|l}
8 & 75 \\ \hline
8 & 9 \quad \cdots\cdots 余 3(K_0=3) \\ \hline
8 & 1 \quad \cdots\cdots 余 1(K_1=1) \\ \hline
 & 0 \quad \cdots\cdots 余 1(K_2=1)
\end{array}
$$

所以，75D=113Q

② 小数部分乘 8 取整法

$$
\begin{array}{rl}
0.6875 & \\
\times\quad\ 8 & \\ \hline
5.5000 & \cdots\cdots 整数部分为 5，相当于 K_{-1}=5 \\
0.5000 & \\
\times\quad\ 8 & \\ \hline
4.0000 & \cdots\cdots 整数部分为 4，相当于 K_{-2}=4
\end{array}
$$

所以，0.6875=0.54Q

最后结果为：75.6875D=113.54Q

3. 十六进制数与十进制数之间的相互转换

（1）十六换十

同二换十、八换十相类似，即将十六进制数按"权"展开相加即可。

例如：

$$
\begin{aligned}
F30H &= 15\times 16^2+3\times 16^1+13\times 16^0\\
&=3840+48+13\\
&=3901D
\end{aligned}
$$

(2) 十换十六

① 整数的十换十六(除 16 取余法)

【例 1-5】 求 3901D 的十六进制数。

$$
\begin{array}{r|l}
16 & 3901 \\
\hline
16 & 243 \quad \cdots\cdots 余 13(K_0=D) \\
\hline
16 & 15 \quad \cdots\cdots 余 3(K_1=3) \\
\hline
 & 0 \quad \cdots\cdots 余 15(K_2=F)
\end{array}
$$

所以,3901D=F3DH

② 小数的十换十六(乘 16 取整法)

【例 1-6】 求 0.9032D 的十六进制数。

$$
\begin{array}{r}
0.9032 \\
\times \quad 16 \\
\hline
14.4512
\end{array}
\quad \cdots\cdots 整数部分为 14,相当于 K_{-1}=E
$$

$$
\begin{array}{r}
0.4512 \\
\times \quad 16 \\
\hline
7.2192
\end{array}
\quad \cdots\cdots 整数部分为 7,相当于 K_{-2}=7
$$

$$
\begin{array}{r}
0.2192 \\
\times \quad 16 \\
\hline
3.5072
\end{array}
\quad \cdots\cdots 整数部分为 3,相当于 K_{-3}=3
$$

$$
\begin{array}{r}
0.5072 \\
\times \quad 16 \\
\hline
8.1152
\end{array}
\quad \cdots\cdots 整数部分为 8,相当于 K_{-4}=8
$$

所以,0.9032D=0.E738H

1.4 二进制数的运算规则

二进制数只有 0,1 两个数码,它的加、减、乘、除运算规则要比十进制数的运算规则简单得多。

1. 加法规则

(1) 0+0=0

(2) 0+1=1+0=1

(3) 1+1=0,进位 1

(4) 1+1+1=1,进位 1

例如:将两个二进制数 1111 与 1011 相加,其过程如下:

$$
\begin{array}{rl}
1111 & 进位 \\
1111 & 被加数 \\
+\ 1011 & 加数 \\
\hline
11010 & 和
\end{array}
$$

可见,两个二进制数相加,每一位有三个数——即相加的两个数以及低位的进位,用

二进制的加法规则得到末位的和以及向高位的进位。

2. 减法规则

(1) 0－0＝0

(2) 1－0＝1

(3) 1－1＝0

(4) 0－1＝1　　向相邻高位借位，1当作2

例如：求二进制数10100减去1001的结果

10100	被减数
－1001	减数
1011	差

3. 乘法规则

(1) 0×0＝0

(2) 0×1＝0

(3) 1×0＝0

(4) 1×1＝1

除了1×1外，其他情况乘积均为0，不需像十进制乘法运算时那样背“九九”乘法口诀。

例如：

```
       1101      被乘数
     × 1010      乘数
   ----------
       0000  ⎫
      1101   ⎪
     0000    ⎬   部分积
    1101     ⎭
   ----------
    10000010     乘积
```

从上例可知，在乘法运算时，若乘数为1，则把被乘数照抄一遍，只是它的最后一位与相应的乘数位对齐；若乘数为0，则部分积为0；当所有的乘数位都乘过之后，再把各部分积相加，便得到最后乘积。因而二进制数乘法实质上是由“加”(加被乘数)和“移位”两种操作实现的。

4. 除法规则

除法是乘法的逆运算。与十进制相类似，可以从被除数的最高位检查，并定出需要超过除数的位数。找到这个位数时，商记1，并且将选定的被除数去减除数。然后，将被除数的下一位移位到余数上；若余数够减，则商为1，余数减去除数，这样反复进行，直至全部被除数的位都下移完为止。

例如：