

21 世纪 高职高专通用教材

华东高校计算机基础教育研究会推荐教材

微型计算机原理 及其应用

陈卜锁 主编

WEI XING JI SUAN JI YUAN LI JI QI YING YONG



上海交通大学出版社
东南大学出版社
中国水利水电出版社

华东高校计算机基础教育研究会推荐教材

21 世纪高职高专通用教材

微型计算机原理及其应用

陈卜锁 主 编

张建平 副主编

上海交通大学出版社
东南大学出版社
中国水利水电出版社

内 容 简 介

本书是计算机基础系列课程教材之一。本书以 16 位 8086 微处理器的微机系统为主,系统地介绍了计算机基础知识、8086/8088 微处理器及其系统结构、指令系统、汇编语言程序设计、存储器与存储器系统、输入输出与接口芯片、中断接口技术、D/A 和 A/D 转换器接口以及人机交互设备接口等。全书共 12 章,每章附有习题。

本书适用于高职高专电类、计算机类、机电一体化专业教学用书,也可作为工程技术人员参考用书。

图书在版编目(CIP)数据

微型计算机原理及其应用/陈卜锁主编. —上海:上海交通大学出版社,2001(2004 重印)
ISBN 7-313-02612-9

I. 微… II. 陈… III. 微型计算机-基本知识
IV. TP36

中国版本图书馆 CIP 数据核字(2000)第 83116 号

微型计算机原理及其应用

陈卜锁 主编

上海交通大学出版社出版发行

(上海市番禺路 877 号 邮政编码 200030)

电话:64071208 出版人:张天蔚

上海交大印务有限公司印刷 全国新华书店经销

开本:787mm×1092mm 1/16 印张:18.5 字数:460 千字

2001 年 1 月第 1 版 2004 年 1 月第 2 次印刷

印数:5051~7100

ISBN 7-313-02612-9/TP·448 定价:26.00 元

版权所有 侵权必究

华东高校计算机基础教育研究会教材编委会

编委会名誉主任

潘云鹤 中国工程院院士 浙江大学校长

顾冠群 中国工程院院士 东南大学校长

编委会主任 张 森

编委会副主任 李文忠

编委会秘书长 赵民德

编委会委员 (以姓氏笔划为序)

石 冰	石文俊	东鲁红	宁正元	张 森
张 煦	张钧良	冉榴红	李文忠	李正凡
吕 刚	朱 敏	陈 轩	陈凤兰	周金辉
杭必政	赵民德	俞俊甫	秦 军	徐安东
詹国华				

序

为了适应我国高校面向 21 世纪计算机基础教育的发展和需要,华东高校计算机基础教育研究会于 1998 年 11 月在浙江金华召开了理事扩大会,对高校计算机基础教育的教材建设问题进行了专题研讨。会议认为,华东地区经济发达、科教先进,高校多达 300 余所,而现有的计算机基础教育的教材建设与现有的地区优势极不相符。80 年代中期华东高校计算机基础教育研究会曾组织出版过一批深受读者欢迎的计算机教材。面对当前计算机科学与技术的飞速发展,计算机基础教育已成为理、工、农、医、商、经贸、政治、文化、艺术等各行各业的公共基础教育。培养大批掌握计算机科学知识与应用技能的跨世纪高级人才,已成为历史赋予高校的一项重要任务。为此,加强高校计算机基础教材建设已提到重要议事日程,学会决定组织力量,编写一套面向 21 世纪的、适应高校计算机基础教学需要的新教材,推动华东高校计算机教育事业的发展。学会于 1999 年 1 月在南京召开了华东地区高校计算机基础教育教材编委会第一次会议,编委会由浙江大学、上海交通大学、东南大学、同济大学、华东理工大学等知名高校的专家学者及上海交通大学出版社、东南大学出版社、中国水利水电出版社的代表共同组成。学会特邀中国工程院院士、浙江大学校长潘云鹤教授和中国工程院院士、东南大学校长顾冠群教授担任编委会名誉主任;由学会会长张森教授任编委会主任,学会副会长李文忠教授任编委会副主任,学会秘书长赵民德兼编委会秘书长。编委会汇集了浙江大学、上海交通大学、东南大学、复旦大学、华东师范大学等数十所院校长期从事高校计算机基础教育、有丰富教学实践经验的资深教师共同研讨,确定编写“华东高校计算机基础教育教材”。第一批教材计 21 种,由上海交通大学出版社、东南大学出版社、中国水利水电出版社分别负责出版发行,并作为华东高校计算机基础教育研究会的推荐教材面向大专院校。教材是教学过程中的“一剧之本”,是当前高校计算机教学的首要问题。在编委会的领导下,经过参编教师的辛勤劳动和三家出版社的共同努力,编写及出版工作进展顺利,预计 2000 年可全部推出。第二批教材的组织准备工作正在进行中。三家出版社联合策划、分工协作、联合出版、联合发行,在华东乃至全国还是首创,得到了教师和同行们的赞赏。教材建设是一项长期艰巨的系统工程,尤其是计算机科学技术发展迅速,更新快,因此,教学内容就要不断更新。为使教材更新跟上科学技术的发展,本会将密切注视计算机科学技术的发展新动向,使我们的教材编写不断推陈出新,逐步与国际接轨,不断提高教材质量,为华东高校计算机基础教育的教材建设作出应有的贡献。

华东高校计算机基础教育研究会

2000 年 8 月

前 言

本书根据 21 世纪高职高专通用教材编审委员会“微型计算机原理及其应用”编写小组审定的编写大纲进行编写,是计算机基础系列课程教材之一,其内容以 8086 微处理器的微机系统为主。全书共分 12 章,第 1 章介绍计算机基础知识;第 2 章介绍 8086/8088 微处理器及其系统结构;第 3 章介绍 8086CPU 指令系统;第 4 章介绍 8086 汇编语言程序设计;第 5 章介绍存储器与存储器系统;第 6 章介绍微机系统的输入输出;第 7 章介绍中断接口技术;第 8 章介绍并行输入/输出的接口芯片;第 9 章介绍串行通信及接口电路;第 10 章介绍可编程定时器/计数器接口芯片 8253;第 11 章介绍 D/A 和 A/D 转换器接口;第 12 章介绍人机交互设备接口。书后有附录,每章还配有适量的习题。

本书的编写力求适应高职高专教学的需要,以培养高素质的技术应用型人才为根本目的,基础理论以“实用、通用、够用”为度,不贪大求全。虽然 Intel 公司生产的 CPU 芯片经历了 8086, 80X86,但应用编程的寄存器结构只有 16 位与 32 位之分,没有本质的区别;指令系统 80X86 与 8086 80%以上是完全相同的,只是在 80386 以上的芯片上增加了一些保护虚地方式下的指令,80486 以上的芯片增加了一些绝大部分与 8087 相同的协处理器指令。因此,8086 与 80X86 是一个系列,是完全兼容的,要深入掌握 80X86 系列芯片,必须从学好 8086 入手。本书介绍的存储器、中断管理系统、I/O 接口芯片等均是高档 PC 机中常用的接口原理、接口方法和接口芯片,编写时还尝试着将“微型计算机接口技术”课程内容溶入本书。为了加强实践环节,本书还有配套实训教材《微型计算机原理实训教程》。本书基本理论建议教学时间为 72 学时。

本书由南通职业大学陈卜锁主编,张建平任副主编,参加编写的有陈卜锁(编写第 3, 4, 6, 8, 9, 10, 11 章,第 5 章 5.3, 5.4 节和附录),张建平(编写第 1, 2 章),刘其和(编写第 7, 12 章),刘旭东(编写第 5 章 5.1, 5.2 节)。全书由陈卜锁统稿。成行洁、宋建娟等同志为本书的出版付出了大量精力,在此表示衷心的感谢。

本书可作为高职高专电类、计算机类、机电一体化专业教学用书,也可作为工程技术人员参考用书。

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

2001 年 1 月 1 日

目 录

1 计算机基础知识	1
1.1 计算机中的运算基础	1
1.1.1 各种进位制数及其特点	1
1.1.2 各不同进位制数之间的转换	3
1.1.3 二进制数的运算规则	5
1.2 计算机中带符号数的表示法及其运算	6
1.2.1 微机中常用的基本术语	6
1.2.2 机器数与真值	6
1.2.3 原码、反码和补码	7
1.2.4 补码的加减运算	10
1.2.5 无符号数的运算	11
1.2.6 溢出的概念及判别	11
1.2.7 定点数及浮点数	12
1.3 计算机中数据的编码	13
1.3.1 8421BCD 码 (Binary Coded Decimal 二进制编码的十进制数)	13
1.3.2 ASCII 码 (美国信息交换标准码)	14
1.4 微机系统的基本组成	15
1.4.1 概述	15
1.4.2 微机的硬件系统	16
1.4.3 微机的软件系统	20
1.5 微机的指令	21
1.5.1 指令、指令系统、程序	21
1.5.2 微机的指令格式	21
1.5.3 微机的指令助记符	22
1.5.4 微机指令的执行过程	22
2 8086/8088 微处理器及其系统结构	26
2.1 8086/8088 CPU 的结构	26
2.1.1 8086/8088 CPU 的内部结构	26
2.1.2 8086/8088 CPU 芯片的引脚及其功能	30
2.2 8086/8088 存储器的结构	35
2.2.1 物理地址的确定	35
2.2.2 分段存储及分段寻址	36
2.3 总线结构和总线周期	37

2.3.1	锁存器、总线控制器、收发器	37
2.3.2	系统总线结构	41
2.3.3	总线周期时序	43
3	8086CPU 指令系统	47
3.1	8086 指令的一般格式	47
3.2	8086 寻址方式	48
3.2.1	立即寻址	48
3.2.2	寄存器寻址	48
3.2.3	存储器寻址	48
3.2.4	端口寻址	50
3.3	8086 指令系统	51
3.3.1	数据传送类指令	51
3.3.2	算术运算指令	54
3.3.3	位操作指令	59
3.3.4	程序控制指令	61
3.3.5	串操作指令	65
3.3.6	处理器控制指令	67
4	8086 汇编语言程序设计	70
4.1	程序设计语言概述	70
4.2	8086 汇编语言的基本语法	71
4.2.1	8086 汇编源程序实例	71
4.2.2	8086 汇编语言语句的种类	72
4.2.3	指令性语句格式	73
4.3	汇编语言中的表达式	76
4.3.1	常量和变量	76
4.3.2	表达式中的各类运算符	77
4.4	伪指令语句	81
4.4.1	符号定义伪指令	82
4.4.2	变量定义伪指令	83
4.4.3	段定义伪指令	84
4.4.4	过程定义伪指令	89
4.4.5	模块定义与通信伪指令	90
4.5	宏指令语句	92
4.5.1	宏指令的使用	92
4.5.2	用于宏定义的其他伪指令	94
4.5.3	重复块宏指令	95
4.5.4	宏指令与过程的比较	96
4.6	DOS 功能调用简介	96
4.6.1	基本的输入与输出	97

4.6.2	文件管理	98
4.6.3	其他	102
4.7	汇编语言程序设计举例	103
4.7.1	顺序程序设计	103
4.7.2	分支程序设计	104
4.7.3	循环程序设计	107
4.7.4	子程序设计	109
5	存储器与存储器系统	114
5.1	概述	114
5.1.1	存储器的分类	114
5.1.2	存储器的性能指标	114
5.1.3	存储器的层次结构	115
5.2	半导体存储器	117
5.2.1	半导体存储器的分类	117
5.2.2	随机存储器 (RAM)	117
5.2.3	只读存储器 (ROM)	120
5.3	常用半导体存储器芯片	121
5.3.1	静态 RAM 存储器 (SRAM)	121
5.3.2	EPROM 电路	122
5.4	存储器与微处理器的连接	125
5.4.1	主存储器的组织	125
5.4.2	8086 系统中存储器的连接举例	127
6	输入和输出	130
6.1	概述	130
6.1.1	外设接口的一般结构	130
6.1.2	外设接口的功能	131
6.1.3	I/O 端口的编址方式和寻址方式	131
6.2	输入/输出数据传送方式	132
6.2.1	程序控制传送方式	132
6.2.2	中断传送方式	137
6.2.3	DMA 方式	138
6.3	8086 CPU 的输入 / 输出	139
6.4	DMA 控制器 8237	140
6.4.1	主要技术特性	140
6.4.2	8237 的工作周期	141
6.4.3	8237 的引脚	141
6.4.4	8237 的工作模式	144
6.4.5	8237 的寄存器组和编程	145
6.4.6	8237 的工作时序	150

6.4.7 编程举例	152
7 中断接口技术	155
7.1 中断的基本概念	155
7.1.1 为什么要用中断	156
7.1.2 中断的类型	156
7.1.3 中断系统的功能	156
7.1.4 中断处理过程	157
7.1.5 矢量中断与中断矢量	160
7.2 8086 的中断系统	160
7.2.1 8086 系统的中断源	160
7.2.2 8086 系统的矢量中断	161
7.3 8259A 中断控制器	162
7.3.1 8259A 的主要功能与结构特点	163
7.3.2 8259A 的编程	166
7.4 PC 机中断电路的编程与工作原理	170
7.4.1 PC 机中断与初始化命令	170
7.4.2 PC 机中断响应周期	172
7.4.3 PC 机的中断嵌套	173
7.4.4 中断接口的应用设计	175
7.5 应用举例	177
8 并行输入 / 输出接口芯片	180
8.1 可编程并行接口芯片 8255A	180
8.1.1 8255A 的内部结构	180
8.1.2 8255A 的引脚功能	182
8.1.3 8255A 的工作方式	183
8.1.4 8255A 的状态字	187
8.1.5 8255A 的控制字	189
8.1.6 8255A 应用举例	190
8.2 用 8212 作为一个输入输出接口	191
8.2.1 8212 简介	191
8.2.2 8212 的工作模式	191
9 串行通信及接口电路	196
9.1 串行通信概述	196
9.1.1 串行通信方式	196
9.1.2 数据传送方向	199
9.1.3 信号的调制与解调	199
9.1.4 串行输入/输出的实现	200
9.2 EIA RS-232C 接口	202
9.2.1 引脚规定	202

9.2.2	电气性能规定	203
9.2.3	电平转换	204
9.2.4	RS-232C 的应用	204
9.3	可编程串行接口芯片 8251A	206
9.3.1	8251 A 的基本性能	206
9.3.2	8251 A 的内部结构	206
9.3.3	8251 A 的引脚功能	208
9.3.4	8251 A 的初始化编程和状态字	210
9.3.5	8251 A 的应用举例	213
10	可编程定时器 / 计数器接口芯片 8253	216
10.1	8253 的内部结构	216
10.2	8253 的引脚功能	218
10.3	8253 的工作方式	218
10.4	8253 的编程	223
10.5	8253 的应用举例	225
11	D/A 与 A/D 转换器接口	228
11.1	D/A 转换器接口	228
11.1.1	D/A 转换器的性能指标	228
11.1.2	8 位 D/A 转换芯片 0832 及其接口	229
11.1.3	12 位 D/A 转换器 DAC1208 及其接口	233
11.2	A/D 转换器接口	236
11.2.1	A/D 转换器的主要参数	236
11.2.2	A/D 转换器与 CPU 的接口方法	236
11.2.3	A/D 转换芯片 ADC0809 及其接口	237
11.2.4	AD574A 及其接口	241
12	人机交互设备接口	245
12.1	概述	245
12.2	键盘及其接口电路	245
12.2.1	按键的结构与特点	246
12.2.2	键开关矩阵	246
12.2.3	键盘编码器	246
12.2.4	简单键盘接口的扫描原理	247
12.2.5	消抖动	247
12.2.6	重键处理	248
12.2.7	有关程序	249
12.2.8	非编码键盘接口	250
12.2.9	PC 机编码键盘的工作原理	251
12.3	CRT 显示器及其接口	254
12.3.1	CRT 显示原理	254

12.3.2 显卡的显示模式	256
12.3.3 字符显示原理	256
12.3.4 字符显示接口电路	258
12.3.5 图形及汉字显示原理	259
12.3.6 PC 机显卡简介	260
12.4 发光二极管显示器接口	260
12.5 LCD 液晶显示器	263
附录	267
附录 1 ASCII 字符表 (7 位码)	267
附录 2 80X86 指令系统	268
参考文献	281

1 计算机基础知识

本章主要介绍微型计算机（以下简称微机）中的常用数制，包括十进制数、二进制数和十六进制数的表示方法及它们之间的相互转换；还介绍了有符号数的原码、反码和补码3种表示形式，以及补码的运算规则、溢出概念和判别方法；此外还介绍了微机的基本组成，硬、软件的基本概念，微机各部件的功能及指令的执行过程。

1.1 计算机中的运算基础

各种类型计算机的处理对象都是数。因此，在学习微机原理之前，首先对计算机中数的表示方法及运算规则作简要介绍。

1.1.1 各种进位制数及其特点

各种进位制数（为了叙述方便，设为N进制数），均有以下特点。

1) N进位制数的组成

任意大小的N进位制数最多由N个数码组成。

例如：十进制数有10个数码，即0, 1, 2, 3, 4, 5, 6, 7, 8, 9。

二进制数有2个数码，即0, 1。

十六进制数有16个数码，即0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F，其中A~F对应于十进制数的10~15。

2) N进制数的基础

N进制数的基数为N，它的加法计数规则是“逢N进一”。所谓基数是指该数制中所使用的数码个数。

例如：十进制数的基数是10，二进制数的基数是2，十六进制的基数是16。

3) N进制数的表示

各位数码在N进制数中所处位置的不同，其所对应的“权”也不同。所谓“权”是与各位数码在数中所代表的数值大小相关的以N为底的幂。

例如：十进制数：(Decimal number)

	9	6	5	.	7	8
	↓	↓	↓		↓	↓
各位的权：	10^2	10^1	10^0		10^{-1}	10^{-2}

该数按权展开可表示为

$$965.78 = 9 \times 10^2 + 6 \times 10^1 + 5 \times 10^0 + 7 \times 10^{-1} + 8 \times 10^{-2}$$

一般地说，任意一个十进制数D都可以表示为：

$$D = \pm (D_{n-1} \times 10^{n-1} + D_{n-2} \times 10^{n-2} + \dots + D_1 \times 10^1 + D_0 \times 10^0 + D_{-1} \times 10^{-1} + D_{-2} \times 10^{-2} + \dots + D_{-m} \times 10^{-m})$$

$$= \pm \sum_{i=-m}^{n-1} D_i \times 10^i \quad (1)$$

其中 D_i 表示第 i 位数码，可以是 0~9 中的任一个具体的数，具体由 D 确定。 m, n 为正整数， n 为整数部分的位数， m 为小数部分的位数。 $10^{n-1}, 10^{n-2}, 10^1, 10^0, 10^{-1}, \dots, 10^{-m}$ 为十进制数各位的“权”。

对于任意进位制数 K ，基数可以用正整数 R 来表示。这时，数 K 可表示为：

$$K = \pm \sum_{i=-m}^{n-1} K_i R^i \quad (2)$$

式中 m, n 均为正整数； K_i 则是 0, 1, $\dots$, $(R-1)$ 中的任一个数码， R 是基数，采用“逢 R 进一”的原则进行计数，各位的权为 R^i 。

例如：二进制数：(Binary number)

$$\begin{array}{ccccccc} 1 & 0 & 1 & 0 & 1 & . & 1 & 1 \\ \downarrow & & & & & & & \\ \downarrow & 125 & & & & & & \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \\ \text{各位的权:} & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} \end{array}$$

根据式 (2)，该数按权展开为

$$\begin{aligned} (10101.11)_2 &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 16 + 0 + 4 + 0 + 1 + 0.5 + 0.25 \\ &= (21.75)_{10} \end{aligned}$$

它是逢 2 进位的。如

$$\begin{aligned} (10)_2 + (11)_2 &= (101)_2 \\ (101)_2 - (11)_2 &= (10)_2 \end{aligned}$$

例如：十六进制数：(Hexadecimal number)

根据式 (2)，十六进制数 A5.8 可展开成下式：

$$(A5.8)_{16} = A \times 16^1 + 5 \times 16^0 + 8 \times 16^{-1} = 160 + 5 + 0.5 = (165.5)_{10}$$

它是逢 16 进位的。如：16 进制数 $39 + 98 = D1$ ， $100 - 39 = C7$ 。

十进制数，二进制数，十六进制数，它们之间存在着直接惟一的对应关系，如表 1.1 所示。

表 1.1 十进制、二进制、十六进制数对照表

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

一般用两种方法，区别所表示数的数制：一是在数的括号右下角用数字注明该数数制，

如二进制数 1010.1 可写成 $(1010.1)_2$ 、十六进制数 C8.5 可写成 $(C8.5)_{16}$ 。

另一种是在数字后面跟一个英文字母。通常用 B(Binary)表示二进制数, H(Hexadecimal)表示十六进制数, D (Decimal) 表示十进制数。所以上面两个数也可写成: 1010.1B, C8.5H。十进制数的后跟字母“D”一般可省略不写。

4) 二进制数的特点

(1) 状态简单, 凡是具有两个状态的元件都可用来表示二进制数的 0 和 1, 所以物质基础广泛, 容易实现。

(2) 运算规则简单, 大大地简化了计算机中的运算线路。

(3) 可用布尔代数这一数学工具对计算机电路进行设计和分析, 便于对计算机结构的优化。

由于它具有上述特点, 使得二进制数特别适用于计算机。但由于二进制数书写冗长, 阅读不便, 为此人们常用十六进制数来书写。此外人们习惯于十进制数, 这就需要学习不同数制之间的相互转换。

1.1.2 各不同进位制数之间的转换

1) 二进制数与十进制数之间的转换

(1) 二进制数转换成十进制数。

只要将二进制数按权展开相加即可, 例如:

$$\begin{aligned} 1010.101 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 8 + 0 + 2 + 0 + 0.5 + 0 + 0.125 \\ &= (10.625)_{10} \end{aligned}$$

(2) 十进制数转换成二进制数。

整数和小数分开转换。

① 十进制整数的转换。

方法一: 可用“除 2 取余”法, 即将十进制整数反复除以 2, 直至商等于 0 为止。然后将所得的一系列余数按逆序排列即为所求的二进制整数。

例如: $(38)_{10} = (100110)_2$

2	38	余数为 0	$K_0 = 0$	最低位 ↑ 最高位
2	19	余数为 1	$K_1 = 1$	
2	9	余数为 1	$K_2 = 1$	
2	4	余数为 0	$K_3 = 0$	
2	2	余数为 0	$K_4 = 0$	
2	1	余数为 1	$K_5 = 1$	
	0			

方法二: 可用“减幂”法, 即将十进制整数不断减去 2 的最高次幂, 直至差值等于 0 为止。若该十进制数中包含一个 2^N , 则该位为 1, 否则为 0。然后将每次减得的 2 的最高次幂排列起来即为所求的二进制整数。

例如: $(38)_{10} = (100110)_2$

② 十进制小数的转换。

0.625			
× 2			
1.250	整数部分	1	$K_{-1}=1$ 最高位
× 2			↓
0.500	整数部分	0	$K_{-2}=0$
× 2			
1.000	整数部分	1	$K_{-3}=1$ 最低位

可用“乘2取整”法，即将十进制小数反复乘以2，直至所剩小数部分等于零为止，然后将每次所得整数按顺序排列即为所求的二进制小数。如果所求的结果永不为零，可根据精度要求用“0舍1入”的方法进行取舍。

例：(0.625)₁₀ = (0.101)₂

38		
- 32	2 ⁵	
6		
- 4	2 ²	
2		
- 2	2 ¹	
0		

0.75	
- 0.5	2 ⁻¹
0.25	
- 0.25	2 ⁻²
0	

对于既有整数部分又有小数部分的十进制数，只要按上述方法分别转换，然后合并起来即可。例如：十进制数 38.75 转换结果为：(38.75)₁₀ = (100110.11)₂

2) 二进制数与十六进制数之间的转换

(1) 二进制数转换为十六进制数。

由于数 16 与数 2 之间的关系为 2⁴=16，因此 4 位二进制数对应 1 位十六进制数。根据这个关系，对于二进制数的整数部分的转换，只要从小数点开始依次向左每 4 位为一节，最后不足 4 位的前面补 0；小数部分由小数点向右，每 4 位为一节，最后不足 4 位的后面补 0。然后把每 4 位二进制数用所相应的一位十六进制数代替，即转换成十六进制数。

例：(10001111011011.111010)₂ = (23DB.E8)₁₆

0010	0011	1101	1011	. 1110	1000
↓	↓	↓	↓	↓	↓
2	3	D	B	. E	8

(2) 十六进制数转换成二进制数。

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 4 位二进制数代替，就可以转成二进制数。

例：(5BD.6A)₁₆ = (10110111101.0110101)₂

5	B	D	.	6	A
↓	↓	↓	↓	↓	↓
0101	1011	1101	.	0110	1010

1.1.3 二进制数的运算规则

1) 算术运算规则

(1) 加法规则。

① $0+0=0$

② $1+0=1+0=1$

③ $1+1=10$ (向高位进位 1)

(2) 减法规则。

① $10-0=0$

② $1-1=0$

③ $1-0=1$

④ $0-1=1$ (向高位有借位)

(3) 乘法规则。

① $0\times 0=0$

② $0\times 1=0$

③ $1\times 0=0$

④ $1\times 1=1$

二进制的乘法运算是十分简单的, 只有当两个 1 相乘时积才为 1, 否则积为 0。

(4) 除法规则。

它是乘法的逆运算, 规则与十进制除法类似, 这里不再赘述。

2) 逻辑运算法则

(1) “或”运算 (逻辑加), 运算符为 “+” 或 “ $\vee$ ”。

① $0\vee 0=0$

② $1\vee 0=1$

③ $0\vee 1=1$

④ $1\vee 1=1$

按位加是不考虑进位的加法运算。

(2) “与”运算 (逻辑乘), 运算符为 “ $\cdot$ ” 或 “ $\wedge$ ”。

① $0\wedge 0=0$

② $0\wedge 0=0$

③ $1\wedge 0=0$

④ $1\wedge 1=1$

(3) “非”运算 (逻辑否定)。

$$\bar{0}=1, \bar{1}=0$$

(4) 异或运算 (按位加), 运算符为 $\oplus$ 。

① $0\oplus 0=0$

② $0\oplus 1=1$

③ $1\oplus 0=1$

④ $1\oplus 1=0$