

Φ 高 科 教 育

微型计算机原理及应用

任仕纯 主编

北京理工大学出版社

微型计算机原理及应用

任仕纯 刘光复 徐 虎 编

北京理工大学出版社

(京)新登字149号

内 容 简 介

本书是机械电子工业部教育司组织编写的,以Z80汇编语言程序设计和接口技术为主要内容的简明教材,是高等专科系列教材之一。

本书主要内容为:微型计算机基础、Z80 CPU指令系统、汇编语言程序设计、接口技术、微型计算机应用等。内容简明,重点突出,通俗易懂,并从应用角度出发,删繁就简,联系实际,便于自学。

本书可作为高等专科学校非计算机类专业的教材,也可供工程技术人员参考。

微型计算机原理及应用

任仕纯 刘光复 徐 虎 编

北京理工大学出版社出版发行

各地新华书店经售

国防出版社印刷厂印刷

787×1092毫米 16开本 17.75印张 418千字

1992年8月第一版 1992年8月第一次印刷

ISBN 7-81013-535-X/TP·51

印数: 1—11000册

定价: 8.45元

出版说明

在机械电子工业部教育司的组织和指导下，由二十二所高等院校的百余名专家、教授编写和审定的高等专科机械制造专业系列教材即将由国防工业出版社、机械工业出版社和北京理工大学出版社陆续出版。这套教材是根据机电部教育司领导制定的《高等专科机械制造专业人才培养规格》的总要求，按照相应的《教学计划和课程大纲》编写的。参加教材编写的人员都具有丰富的教学经验，并且经过申报和严格审批。各门课程的教材编写大纲，课程之间的衔接和配合，都经过专家、教授集体讨论和审议。在编写过程中，各课程编写组在主编负责下对书稿进行了反复讨论和修改，最后经具有高水平的专家教授审查定稿。这就保证了这套教材具有相当的水平 and 较高的质量。这套系列教材适用于高等专科教育机械制造专业，亦适合成人高等专科学校教育，体现出为生产第一线培养德、智、体全面发展的应用型高级技术人才的总要求。各教材在保证基本内容的系统性和科学性的同时，注意理论联系实际和能力的培养，适当反映现代科学技术的新成就。这套教材中的基础课和技术基础课教材，也适用于其他有关专业。

这套教材的编写和出版，在机电部教育司的关心和领导下，在机械工业出版社、国防工业出版社和北京理工大学出版社的大力支持下，依靠全体编审人员的辛勤工作和有关院校的协助，得以顺利完成。在此，谨向他们表示衷心的感谢。

我们奉献的这套教材，希望能够得到读者的欢迎。但是，由于组织这样的系列教材编写工作尚属首次，这套教材难免有不足之处，欢迎广大师生和其他读者批评指正。

机电部高等专科机制专业教材编委会

1989年9月

前 言

本书是根据机电部教育司制定的、高等专科机械制造专业《微型计算机原理及应用》教学大纲编写的。它是机电部教育司组织编写的高等专科系列教材之一。

《微型计算机原理及应用》是机械制造专业的一门重要的技术基础课。微型计算机的应用日益广泛,发展非常迅速。微型计算机技术与机械技术的紧密结合,促进了机电一体化迅速发展,在我国科学技术现代化中占有重要地位。本课程的任务和作用使学生获得微型计算机硬件和软件必要的基础知识,了解微机在机械行业中的应用。本书的主要内容包括:微型计算机基础,Z80CPU指令系统、汇编语言程序设计、接口技术、微型计算机应用等。

在本书的编写过程中,力求突出专科教育的特点,在组织教学内容时,既注意教材的系统性,又突出教材的实践性。编写中力求概念清晰,文字通俗,由浅入深,便于自学。书中安排了较多的例题、思考题和习题及测验作业题,以帮助读者思考和检查学习的效果。

参加本书编写的有汕头大学任仕纯副教授(绪论、第三章、第四章),合肥工业大学刘光复副教授(第一章、第五章),南京机械专科学校徐虎副教授(第二章),由任仕纯副教授任主编。

本书主审北京理工大学陆士毅副教授仔细审阅了书稿,提出了很多宝贵意见。编者根据这些意见,进行了认真的修改。

由于我们水平所限,书中难免有不当或错误之处,恳切希望读者批评、指正。

编者

1990年4月

目 录

绪 论	1
第一章 微型计算机基础	4
§ 1-1 计算机中数的表示方法	4
一、二进制	4
二、八进制和十六进制	8
三、有符号数的表示方法	9
四、二——十进制编码 (BCD码)	12
五、字符代码	13
思考练习题	14
§ 1-2 微型计算机的基本组成及总线	14
一、微型计算机的基本组成	14
二、三种总线	14
三、三态门	15
思考练习题	17
§ 1-3 存储器	17
一、概述	17
二、RAM (随机存储器)	18
三、ROM (只读存储器) 和 EPROM (可擦除的可编程只读存储器)	20
思考练习题	22
§ 1-4 微处理器	22
一、寄存器组	23
二、运算器	25
三、控制器	26
思考练习题	27
§ 1-5 简单的微型计算机硬件系统	28
一、Z80CPU 引脚信号	28
二、译码器	30
三、CPU和存储器的连接	30
四、I/O 接口及和CPU的连接	32
五、微型计算机最小系统	33
六、含有 ROM、RAM、I/O接口系统的例子	33
思考练习题	34
本章小结	34
习题一	35
自我测验题 I	36
第二章 Z80指令系统	38
§ 2-1 指令格式	38
一、助记符	38

二、指令格式	39
三、寻址方式	39
思考练习题	42
§ 2-2 数据传送指令	42
一、八位传送指令	44
二、十六位传送指令	45
三、交换指令	47
四、数据块传送指令	48
五、输入输出指令	49
思考练习题	50
§ 2-3 数据操作指令	51
一、算术运算指令	51
二、比较指令	55
三、逻辑运算指令	56
四、循环与移位指令	57
五、位操作指令	60
六、通用算术指令	63
思考练习题	64
§ 2-4 程序控制指令	65
一、无条件转移指令	66
二、条件转移指令	66
三、循环控制指令	67
四、CPU 控制指令	68
思考练习题	68
本章小结	69
习题二	70
自我测验题 I	71
第三章 汇编语言程序设计	73
§ 3-1 汇编语言概述	73
一、机器语言、汇编语言和高级语言	73
二、汇编语言语句结构	74
三、Z80 常用伪指令	75
四、汇编方法	76
思考练习题	78
§ 3-2 简单程序的设计	78
一、程序设计的一般步骤	78
二、顺序程序	79
思考练习题	83
§ 3-3 分支程序	83
一、简单分支	84
二、多向分支	86
思考练习题	89
§ 3-4 循环程序	90
一、循环程序的基本结构	90

二、单重循环·····	92
三、多重循环·····	94
思考练习题·····	98
§ 3-5 堆栈和子程序·····	98
一、堆栈·····	98
二、子程序的调用和返回·····	99
三、子程序应用举例·····	101
思考练习题·····	108
本章小结·····	108
习题三·····	109
自我测验题Ⅱ·····	112
第四章 接口技术·····	115
§ 4-1 输入输出方式·····	115
一、概述·····	115
二、Z80 CPU 的输入输出指令·····	116
三、输入输出的控制方式·····	118
思考练习题·····	119
§ 4-2 中断·····	120
一、中断的控制及中断指令·····	120
二、中断的处理过程·····	122
三、中断的优先级及嵌套·····	125
思考练习题·····	126
§ 4-3 PIO (并行接口电路) 及其应用·····	126
一、概述·····	126
二、PIO的结构和功能·····	128
三、直接输入输出·····	130
四、中断控制的输入输出·····	133
五、联络控制的输入输出·····	140
思考练习题·····	144
§ 4-4 CTC (计数定时器) 及其应用·····	144
一、CTC 的结构和功能·····	144
二、CTC 用作计数器·····	149
三、CTC 用作定时器·····	151
思考练习题·····	154
§ 4-5 模拟通道的接口·····	154
一、概述·····	154
二、DAC与CPU的连接和应用·····	155
三、ADC与CPU的连接和应用·····	157
思考练习题·····	162
本章小结·····	163
习题四·····	163
自我测验题Ⅳ·····	165
第五章 微型计算机应用及单片机简介·····	167
§ 5-1 Z80单板机及其对外设的管理·····	167

一、单板机的组成	167
二、显示器及其显示程序	171
三、键盘及键盘处理程序	178
四、监控程序简介	189
五、打印机	193
§ 5-2 微型计算机在机械中的应用	195
一、执行机构及其与微型计算机的连接	195
二、传感器及其与微型计算机的连接	198
§ 5-3 工业顺序控制	203
一、过程顺序控制	203
二、时间顺序控制	208
§ 5-4 步进电机的微机控制	214
一、步进电机工作原理简介	214
二、步进电机的控制方法	215
三、简易数控车床直线加工的开环控制	220
§ 5-5 单片计算机简介	223
一、概述	223
二、MCS—51单片机片内结构	226
三、MCS—51功能简介	232
本章小结	237
习题五	238
附录	239
附录一 Z80CPU 助记符总表	239
附录二 Z80CPU指令表(摘要)	240
附录三 MCS—51系列单片机指令编码表(按字母顺序排列)	249
部分习题参考答案	254
参考资料	274

绪 论

微型计算机是在 70 年代初, 随着计算机应用的普及及大规模集成电路技术的发展而出现的高新技术。微型计算机在高新技术中最活跃, 最富冲击力、渗透力最强。微型计算机的飞速发展, 使计算机的应用普及成为现实, 它带动着各行各业的技术进步。例如, 在机械工业中, 从机电一体化新产品的开发, 传统设备的改造, 新的高度自动化的加工系统的建立, 无一不以微型计算机为基础在进行。离开了微型计算机, 机电一体化也就成了一句空话。

一、微型计算机的一般概念

微型计算机的核心是微处理器 (μP 或 MP)。利用大规模集成电路技术, 将计算机的中央处理单元制作成一块大规模集成电路, 这块大规模集成电路就叫微处理器, 也叫 CPU。它主要包括运算器、控制器和寄存器阵列等部件。

以微处理器为核心, 配上只读存储器 (ROM)、读写存储器 (RAM)、输入输出 (I/O) 接口电路, 并用系统总线联结起来所组成的计算机, 称为微型计算机, 简称微机 (μC 或 MC)。

以微机为中心, 配以相应的外部设备 (如键盘、CRT 显示器、打印机、磁盘机等)、电源、以及管理计算机工作的系统软件和应用软件所构成的系统称为微型计算机系统 (简称 μCS 或 MCS)。

微处理器、微型计算机、微型计算机系统间的关系如图 0-1 所示。

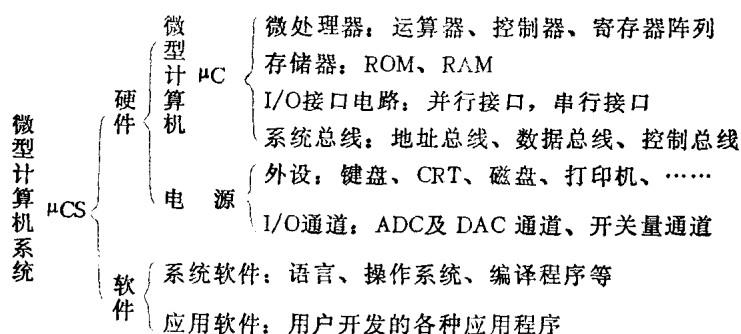


图0-1 微型计算机系统的组成

二、微型计算机的发展概况

微型计算机的发展与微处理器的发展密切相关。自 1971 年第一个微处理器问世以来, 微处理器技术的发展很快, 已经由 4 位、8 位、16 位发展到 1984 年的 32 位, 到 1989 年, Intel 公司又推出了 64 位的微处理器。微处理器的发展概况如表 0-1 所示。

随着半导体集成技术的进展, μP 的集成度不断提高, 4 位、8 位 μP 的集成度约几千个晶体管/片, 16 位 μP 已发展到几万个晶体管/片, 而 32 位微处理器每片则集成了二十多万个晶体管。随着微处理器的进展, 不仅数据处理字长位数成倍增加, 它所配的内存容量和时钟频率也相应增加, 例如 8080 的时钟频率为 2MHz, 内存容量为 64KB,

表0-1 典型微处理器的发展简况

推出年份	位 数	典型 μ P型号	制 造 公 司
1971	4	4004	Intel
1974	8	8080	Intel
	8	6800	Motorola
	8	Z 80	Zilog
1978	16	8086	Intel
	16	68000	Motorola
	16	Z 8000	Zilog
1984	32	Z 80000	Zilog
	32	68020	Motorola
	32	80386	Intel
1989	48	SCAMP	Unisys
1989	64	80486	Intel

8086 的时钟为 5MHz, 内存容量为 1MB, 80386 的时钟为 25MHz, 内存容量达 8MB。由此可见, 微处理器的位数愈大, 由微处理器组成的微型计算机的内存容量愈大, 因而功能愈强; 时钟频率愈高, 工作速度也愈快。32 位微机已被人们称为超级微机, 其功能足以同高档小型机匹敌。

近年来, 微型计算机在性能上不断更新换代, 在产量上更是与日俱增。表 0-2 给出了各种微处理器与微型计算机产量的统计数据。

表0-2 全世界 μ P 与 μ C 产量统计 (单位: 10万台)

年份	1977	1978	1979	1980	1981	1982	1983	1984	1985
μ P	100	300	568	1028	1475	1964	2390	2660	3100
μ C	0.6	1.8	2.6	5.4	10	20	28	37	46

由表可见, 80 年代以来, 微型计算机的产量正以每年递增 40% 的速度在增长, 其发展速度是空前的。目前, 微型计算机的种类越来越多, 价格越来越低, 应用越来越广, 已经深入到人类生活的各个领域。可以相信, 随着计算机技术的日益发展和应用的广泛普及, 计算机的发展前景是无限美好的。

三、计算机的类型和用途

微型计算机的种类、型号很多, 但按照用途, 可将微型计算机大致分为两类: 通用微型计算机和控制用微型计算机。

1. 通用微型计算机

通用微型计算机主要用于信息处理, 如科学计算、事务管理、情报检索、办公自动化等等。

通用机的主要特点是存储容量大, 软件丰富, 但外设比较简单, 主要外设是键盘、显示器、软硬磁盘、打字机等。常用的通用微机有 8 位和 16 位的, 32 位的超级微机也已进入市场。

通用微机主要使用面向问题的高级语言。

2. 控制用微型计算机

控制微机主要用于生产过程的控制，如生产现场的巡回检测，生产机械的自动控制等。它可以直接与机械部件紧密结合，构成新的智能产品，如智能仪器、仪表，各种机电一体化产品，如机器人、数控机床等。

控制微机的主要特点是：有完善的输入、输出通道，可与各种形式的现场设备连接，能实现多种形式的信息（开关量、数字量、模拟量）传递和数据采集；有较高的适时性，以便在规定的时间内完成控制对象所规定的操作；有高度的稳定性和强的抗干扰能力，以便能在较恶劣的工业现场稳定可靠地运行。控制微机通常也有4位、8位、16位和32位之分。

控制微机中，大量使用面向过程的汇编语言，以提高机器的适时性。

本书在讲清微型计算机原理的基础上，将侧重把微型计算机当作微控制机看待。因此，在内容安排上以汇编语言程序设计和接口技术为重点，并有大量的应用例题，目的全在于帮助读者掌握微机在工业控制中的应用。

第一章 微型计算机基础

微型计算机的应用目前已深入到商业、工业、服务行业及日常生活领域，微型计算机为什么应用如此广泛，它是由哪几个部分组成的，它是如何识别1、2、3、…、9和A、B、C、…、Y、Z这样一些数字和字母的，又是如何实现自动计算的，本章将向不熟悉这方面的读者介绍微型计算机的一些基础知识和部分硬件基础。

§ 1-1 计算机中数的表示方法

不管应用在什么场合，计算机都要处理代表各种量（或值）的信号，而量（或值）只有用数才能准确说明。因此，数的表示方法是首先要解决的问题。

我们日常生活使用的是十进制，十进制有0~9共十个数码，每位满十向高位进一，即按“逢十进一”“借一当十”的原则运算。同一个数码在不同的位置代表不同的数值。例如：3333.44这个数，小数点前从右往左数，第一个3代表3本身，第二个3代表 $3 \times 10^1(30)$ ，第三个3代表 $3 \times 10^2(300)$ ，第四个3代表 $3 \times 10^3(3000)$ ，小数点后从左往右数，第一个4代表 $4 \times 10^{-1}(0.4)$ ，第二个4代表 $4 \times 10^{-2}(0.04)$ ，因此，这个数可以写成：

$$3333.44 = 3 \times 10^3 + 3 \times 10^2 + 3 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 4 \times 10^{-2}$$

式中 10^3 ， 10^2 ， 10^1 ， 10^0 ， 10^{-1} ， 10^{-2} 称为各位的“权”，每一位的数码与该位“权”的乘积就是该位数的大小。

人们习惯于十进制数，但十进制数对于计算机来说并不合适。因为一般的电信号是用电压来表示的，如果采用十进制，十个数码就需要十个电平或十种电压值来代表。而二进制只有两个数码，即1和0，只需要两个电平来代表。两个电平是很容易用开关来实现的。晶体管的导通和截止就可以作为非常可靠的开关使用。例如，可设开关的闭合为1，断开为0，高电平为1，低电平为0。这样两种相反的物理状态简单、可靠，因此，计算机采用二进制的计数方法。

数制中所用数码的个数称为基数，为了避免混淆，我们常把二进制和十进制的基数2或10注在下标处，例如： 1101_2 ， 1101_{10} 。

一、二进制

1. 二进制到十进制的转换

二进制只有两个数码0和1，数码0不论在哪一位，代表的数值都为0，数码1是根据它在数中的位置来确定它所代表的数值的大小。例如： 1111.11_2 ，小数点前从右往左数，第一个1代表1本身，第二个1代表 $1 \times 2^1(2)$ ，第三个1代表 $1 \times 2^2(4)$ ，第四个1代表 $1 \times 2^3(8)$ ，小数点后从左往右，第一个1代表 $1 \times 2^{-1}(1/2)$ ，第二个1代表 $1 \times 2^{-2}(1/4)$ ，这个数可写成：

$$1111.11_2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

式中 $2^3, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}$, 称为“权”, 即小数点左边第一位的权是 1, 小数点左边第二位的权是 2, 第三位的权是 4 ..., 小数点右面第一位的权是 $1/2$, 右边第二位的权是 $1/4$..., 依此类推可知, 当逐位往左移动时, 每位的权增加一倍, 逐位向右移动时, 每位的权缩小一倍, 二进制数权的分布表示在图 1-1 中, 每位的权表示在该位的上方。一个二进制数最右边一位称为最低有效位 (LSB), 最左边一位称为最高有效位 (MSB)。

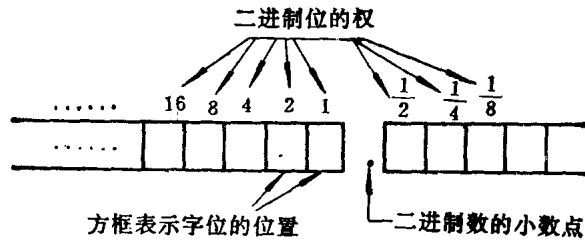


图1-1 二进制数的分布

由于二进制数中只有 1 和 0 两个数, 上面二进制到十进制按权展开的方法可以简化为: 把二进制数中每个 1 上方的权相加, 结果就是这个数的等值十进制数。

例1-1 把下面二进制数转换成等值十进制数:

解

$\frac{8}{1}$	$\frac{4}{1}$	$\frac{2}{1}$	$\frac{1}{0_2}$	← 权分布	$= 8 + 4 + 2 = 14$
$\frac{16}{1}$	$\frac{8}{1}$	$\frac{4}{0}$	$\frac{2}{0}$	$\frac{1}{0_2}$	← 权分布
$\frac{2}{1}$	$\frac{1}{1}$	$\frac{1/2}{1}$	$\frac{1/4}{1}$	← 权分布	$= 2 + 1 + 1/2 + 1/4 = 3.75$
$\frac{32}{1}$	$\frac{8}{0}$	$\frac{1}{1}$	$\frac{1}{0}$	← 权分布	$= 32 + 8 + 1 = 41$

2. 十进制到二进制的转换

将十进制整数转换为二进制数的方法是“除 2 取余”即用 2 不断地去除要转换的十进制数, 直至商为 0, 所得的余数从后往前排, 即得到等值的二进制数。

例 1-2 将十进制数 13、35、235 转换为等值二进制数。

解 (1) $2 \overline{) 13}$

$2 \overline{) 6} \cdots \text{余 } 1$	┌───┐ ├───┤ ├───┤ └───┘	
$2 \overline{) 3} \cdots \text{余 } 0$		
$2 \overline{) 1} \cdots \text{余 } 1$		
$0 \cdots \text{余 } 1$		
	1 1 0 1	$\therefore 13_{10} = 1101_2$

(2)
$$\begin{array}{r} 2 \overline{) 35} \\ 2 \overline{) 17} \cdots \text{余} 1 \\ 2 \overline{) 8} \cdots \text{余} 1 \\ 2 \overline{) 4} \cdots \text{余} 0 \\ 2 \overline{) 2} \cdots \text{余} 0 \\ 2 \overline{) 1} \cdots \text{余} 0 \\ 0 \cdots \text{余} 1 \end{array}$$

1 0 0 0 1 1

$\therefore 35_{10} = 100011_2$

(3)
$$\begin{array}{r} 2 \overline{) 235} \\ 2 \overline{) 117} \cdots \text{余} 1 \\ 2 \overline{) 58} \cdots \text{余} 1 \\ 2 \overline{) 29} \cdots \text{余} 0 \\ 2 \overline{) 14} \cdots \text{余} 1 \\ 2 \overline{) 7} \cdots \text{余} 0 \\ 2 \overline{) 3} \cdots \text{余} 1 \\ 2 \overline{) 1} \cdots \text{余} 1 \\ 0 \cdots \text{余} 1 \end{array}$$

1 1 1 0 1 0 1 1

$\therefore 235_{10} = 11101011_2$

将十进制小数化为等值的二进制小数的方法是“乘2取整”，即用2不断去乘十进制数的小数部分，将每次所得的整数从前往后依次取出，直到积的小数部分是零为止。

例1-3 将十进制小数0.375化为等值的二进制小数。

解

0.375	$\rightarrow$ 0.75	$\rightarrow$ 0.50
$\times 2$	$\times 2$	$\times 2$
0.750	1.50	1.00

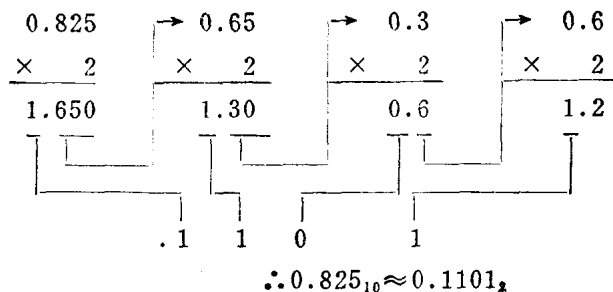
. 0 1 1

$\therefore 0.375_{10} = 0.011_2$

不断用2去乘，不一定都能使尾数部分等于零，这时只要依精度要求，二进制小数取足够的位数即可。

例1-4 将十进制小数0.825转换成二进制小数。

解



3. 二进制的运算法则

加法：从最低位开始相加，低位向高位的进位原则为“逢2进1”，即： $1 + 1 = 10$ ， $1 + 0 = 1$ 。例1-5是加法的例子，把例中的每个二进制数转换为十进制数，就可以验证其答案的正确性。

例1-5 做下列数的二进制加法运算。

$$\begin{array}{l} \text{解 (1)} \quad \begin{array}{r} 11100 \\ + 11001 \\ \hline 110101 \end{array} \left\{ \begin{array}{l} 28_{10} \\ + 25_{10} \\ \hline 53_{10} \end{array} \right. \quad (2) \quad \begin{array}{r} 010101 \\ + 101110 \\ \hline 1000011 \end{array} \left\{ \begin{array}{l} 21_{10} \\ + 46_{10} \\ \hline 67_{10} \end{array} \right. \end{array}$$

减法：从最低位开始运算，低位向高位的借位原则是“借1为2”。例1-6是减法的例子，把例中的每个二进制数转化为十进制数就可以验证其答案的正确性。

例1-6 做下列数的二进制减法运算。

$$\begin{array}{l} \text{解 (1)} \quad \begin{array}{r} 11100 \\ - 11001 \\ \hline 00011 \end{array} \left\{ \begin{array}{l} 28_{10} \\ - 25_{10} \\ \hline 3_{10} \end{array} \right. \quad (2) \quad \begin{array}{r} 101110 \\ - 010101 \\ \hline 011001 \end{array} \left\{ \begin{array}{l} 46_{10} \\ - 21_{10} \\ \hline 25_{10} \end{array} \right. \end{array}$$

乘法：运算法则和十进制类似，即 $0 \times 0 = 0$ ， $1 \times 0 = 0$ ， $1 \times 1 = 1$ 。

除法：是乘法的逆运算，运算方法也同十进制类似。

例1-7 做下列数的二进制乘、除法运算。

$$\begin{array}{l} \text{解 (1)} \quad \begin{array}{r} 1011 \\ \times 101 \\ \hline 1011 \\ 0000 \\ 1011 \\ \hline 110111 \end{array} \left\{ \begin{array}{l} 11_{10} \\ \times 5_{10} \\ \hline 55_{10} \end{array} \right. \\ \\ (2) \quad \begin{array}{r} 1011 \\ 101 \overline{) 110111} \\ \underline{101} \\ 111 \\ \underline{101} \\ 101 \\ \underline{101} \\ 0 \end{array} \rightarrow \begin{array}{r} 11 \\ 5 \overline{) 55} \\ \underline{5} \\ 5 \\ \underline{5} \\ 0 \end{array} \end{array}$$

二、八进制和十六进制

用二进制表示数值时，书写很长，很不形象，给人们带来很大的不便，所以有的小型机中采用八进制，微型机中通常采用十六进制。了解二进制后，这两种数制很容易理解，它们只不过是二进制的一种压缩形式，把每 3 位二进制写成 1 位，就是八进制数，每 4 位二进制数写成 1 位就是十六进制数。比如： $101\ 110\ 011_2 = (563)_8$ ， $1000\ 1001_2 = (89)_{16}$ 。采用这两种进制编制的程序，书写较容易，计算机也容易识别。因为在微型机中，八进制用的较少，我们只做一简单介绍，十六进制和二进制的转换应熟练掌握。

1. 八进制

八进制数用 0 ~ 7 共 8 个数码表示，运算时的原则为“逢 8 进 1”，“借 1 为 8”，数码在不同的数位表示不同的数值。例如： $(555)_8 = 5 \times 8^2 + 5 \times 8^1 + 5 \times 8^0$ 。

2. 十六进制

十六进制数用 0、1、2、…、9、A、B、C、D、E、F 等 16 个数码表示，运算法则为“逢 16 进 1”，“借 1 为 16”。数码在不同的数位代表不同的数值，例如：

$$(555)_{16} = 5 \times 16^2 + 5 \times 16^1 + 5 \times 16^0$$

每 4 位二进制数可用 1 位十六进制数表示，8 位二进制数可用 2 位十六进制数表示，十六进制数在汇编语言程序设计中应用十分普遍。表 1-1 是十进制数，8 位二进制数和 2 位十六进制数的对照表。

表 1-1 三种不同进制数对照表

十进制	二进制数 (8 位)	十六进制	十进制	二进制数 (8 位)	十六进制
0	0 0 0 0 0 0 0 0	0 0	9	0 0 0 0 1 0 0 1	0 9
1	0 0 0 0 0 0 0 1	0 1	10	0 0 0 0 1 0 1 0	0 A
2	0 0 0 0 0 0 1 0	0 2	11	0 0 0 0 1 0 1 1	0 B
3	0 0 0 0 0 0 1 1	0 3	12	0 0 0 0 1 1 0 0	0 C
4	0 0 0 0 0 1 0 0	0 4	13	0 0 0 0 1 1 0 1	0 D
5	0 0 0 0 0 1 0 1	0 5	14	0 0 0 0 1 1 1 0	0 E
6	0 0 0 0 0 1 1 0	0 6	15	0 0 0 0 1 1 1 1	0 F
7	0 0 0 0 0 1 1 1	0 7	16	0 0 0 1 0 0 0 0	1 0
8	0 0 0 0 1 0 0 0	0 8	17	0 0 0 1 0 0 0 1	1 1

从表 1-1 可见，二进制转换为十六进制时，将二进制数从最低整数位开始，每 4 位划为一组，用等值的十六进制数代替各组的 4 位二进制数，即采用“4 位聚 1 位”的方法。十六进制数转换为二进制数时，只要把每 1 位十六进制数用相当的 4 位二进制数来代替即可，也就是采用“1 位拉 4 位”的方法。

例 1-8 将二进制数 10001100_2 和十六进制数 $2F3A_{16}$ 分别转换为等值的十六进制数和二进制数。

解 $1000\ 1100_2 = (8C)_{16}$

$(2F3A)_{16} = 0010\ 1111\ 0011\ 1010$ (舍去有效数字前的 0) $= 10111100111010$

采用“4 位聚 1 位”方法转换时，如果最后一组不足 4 位时，可在最高位前加“0”。如：把二进制数 11101011011001_2 转换成十六进制数时，应在最高位前加 2 个“0”，然后再进行分组和转换。即：