

可下载教学资料

<http://www.tup.tsinghua.edu.cn>



高等学校教材
计算机科学与技术

微机原理、汇编 与接口技术学习指导

朱定华 编著

清华大学出版社



高等学校教材
计算机科学与技术

微机原理、汇编 与接口技术学习指导

朱定华 编著

清华大学出版社
北京

内 容 简 介

本书是同作者所编著的《微机原理、汇编与接口技术》的配套用书,是为了配合“微机原理与接口技术”课程的教学而编写的辅导教材。本书对该课程教学中的重点和难点进行了较详细的论述,并通过典型的例题对该课程中较难理解的内容作了详细的分析,同时还对精选的习题作了较详细的解答。旨在帮助读者拓宽思路,掌握解题方法,从而加深理解,灵活运用。

本书可作为大专院校的本科和专科学生学习微型计算机原理、汇编语言程序设计、微机接口技术等课程的辅助教材,也可以作为报考研究生的人员系统复习上述课程的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

微机原理、汇编与接口技术学习指导/朱定华编著. —北京:清华大学出版社,2006.12

(高等学校教材·计算机科学与技术)

ISBN 7-302-14068-5

I. 微… II. 朱… III. ①微型计算机—理论—高等学校—教学参考资料 ②汇编语言—程序设计—高等学校—教学参考资料 ③微型计算机—接口—高等学校—教学参考资料 IV. ①TP36 ②TP313

中国版本图书馆 CIP 数据核字(2006)第 126678 号

责任编辑:魏江江 顾 冰

责任印制:何 芊

出版发行: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

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 刷 者:北京市昌平环球印刷厂

装 订 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:13.25 字 数:327 千字

版 次:2006 年 12 月第 1 版 印 次:2006 年 12 月第 1 次印刷

印 数:1~3000

定 价:19.00 元

本书如存在文字不清、漏印、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)62770177 转 3103 产品编号:023776-01

编审委员会成员

(按地区排序)

清华大学	周立柱	教授
	覃 征	教授
	王建民	教授
	刘 强	副教授
	冯建华	副教授
北京大学	杨冬青	教授
	陈 钟	教授
	陈立军	副教授
北京航空航天大学	马殿富	教授
	吴超英	副教授
	姚淑珍	教授
中国人民大学	王 珊	教授
	孟小峰	教授
	陈 红	教授
北京师范大学	周明全	教授
北京交通大学	阮秋琦	教授
北京信息工程学院	孟庆昌	教授
北京科技大学	杨炳儒	教授
石油大学	陈 明	教授
天津大学	艾德才	教授
复旦大学	吴立德	教授
	吴百锋	教授
	杨卫东	副教授
华东理工大学	邵志清	教授
华东师范大学	杨宗源	教授
	应吉康	教授
东华大学	乐嘉锦	教授
上海第二工业大学	蒋川群	教授
浙江大学	吴朝晖	教授
	李善平	教授
南京大学	骆 斌	教授
南京航空航天大学	秦小麟	教授
南京理工大学	张功萱	教授

南京邮电学院	朱秀昌	教授
苏州大学	龚声蓉	教授
江苏大学	宋余庆	教授
武汉大学	何炎祥	教授
华中科技大学	刘乐善	教授
中南财经政法大学	刘腾红	教授
华中师范大学	王林平	副教授
	魏开平	副教授
	叶俊民	副教授
国防科技大学	赵克佳	教授
	肖 侬	副教授
中南大学	陈松乔	教授
	刘卫国	教授
湖南大学	林亚平	教授
	邹北骥	教授
西安交通大学	沈钧毅	教授
	齐 勇	教授
长安大学	巨永峰	教授
西安石油学院	方 明	教授
西安邮电学院	陈莉君	副教授
哈尔滨工业大学	郭茂祖	教授
吉林大学	徐一平	教授
	毕 强	教授
长春工程学院	沙胜贤	教授
山东大学	孟祥旭	教授
	郝兴伟	教授
山东科技大学	郑永果	教授
中山大学	潘小轰	教授
厦门大学	冯少荣	教授
福州大学	林世平	副教授
云南大学	刘惟一	教授
重庆邮电学院	王国胤	教授
西南交通大学	杨 燕	副教授

改革开放以来,特别是党的十五大以来,我国教育事业取得了举世瞩目的辉煌成就,高等教育实现了历史性的跨越,已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上,高等教育规模取得如此快速的发展,创造了世界教育发展史上的奇迹。当前,教育工作既面临着千载难逢的良好机遇,同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾,是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001年8月,教育部下发了《关于加强高等学校本科教学工作,提高教学质量的若干意见》,提出了十二条加强本科教学工作提高教学质量的措施和意见。2003年6月和2004年2月,教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件,指出“高等学校教学质量和教学改革工程”是教育部正在制定的《2003—2007年教育振兴行动计划》的重要组成部分,精品课程建设是“质量工程”的重要内容之一。教育部计划用五年时间(2003—2007年)建设1500门国家级精品课程,利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放,以实现优质教学资源共享,提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展、顺应并符合新世纪教学发展的规律、代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻

性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括:

(1) 高等学校教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。

(4) 高等学校教材·软件工程——高等学校软件工程相关专业的教材。

(5) 高等学校教材·信息管理与信息系统。

(6) 高等学校教材·财经管理与计算机应用。

清华大学出版社经过近 20 年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

E-mail: dingl@tup.tsinghua.edu.cn

微型计算机现已应用于国民经济的各个领域,它的普及与应用使人们传统的工作、学习、生活乃至思维方式发生了巨大变化。“微机原理与接口技术”课程也成为理工科大专院校学生的一门必修课。为了帮助大学本科和专科学生及自学者学习和掌握本门课程,也为了满足报考研究生的人员系统复习“微机原理与接口技术”课程的需要,特编写本书。

本书是为配合“微型计算机原理及应用”的教学,指导学生理解、领会教学内容,增强分析和解决问题的能力而编写的辅导教材。本书共有 8 章和 3 个附录,书中给出了每章的内容提要、重点和难点、难点详析以及习题解答。第 4 章和第 7 章的内容涉及的概念较多,这两章没有给出概念性习题的解答,第 4 章还没有精选例题。建议初学者暂不学习本课程的以下内容:地址传送指令、查表转换指令、BIOS、串处理程序设计和宏功能程序设计。为了帮助读者尽快地掌握“微机原理与接口技术”课程的基础内容,本书对上述内容也作了相应的处理:不使用地址传送指令和查表转换指令,而用其他指令代替;对于串操作,同时给出使用串操作和不使用串操作两种解题的方法。根据作者多年来进行“微型计算机原理”、“汇编语言程序设计”、“微机接口技术”和“微机应用系统设计”等课程教学的实践和体会,对“微机原理与接口技术”课程教学内容的重点和难点进行了详细的论述,并精选了较为典型的例题,对难度较大、难以理解的例题进行了详细的分析,同时,精选了适量的习题,并对这些习题作了详细的解答,以帮助读者拓宽思路,加深对本课程的学习、理解,并尽快地掌握“微机原理与接口技术”课程的基础内容,帮助读者在本课程的结业考试中获得优异成绩。

本书由朱定华编写。参加本书编写工作的人员还有饶志强、翟晟、周斌、林巍、张德芳、陈萍、刘伏新、林卫、吕建才、李志文、梁宗善等。

由于编著者水平和经验所限,书中不足和错误之处在所难免,恳请读者批评指正。衷心欢迎读者随时提出意见和建议,来信请寄:武汉市华中科技大学喻园小区 41 单元 902,邮政编码 430074。或发送电子邮件至 zhu4361@163.com。

朱定华

2006 年 8 月 8 日于武昌南湖

第1章 微型计算机的基础知识	1
1.1 内容提要	1
1.2 重点	2
1.3 难点分析	2
1.4 例题与详析	5
1.5 习题与详解	7
第2章 汇编语言与汇编程序	15
2.1 内容提要	15
2.2 重点	15
2.3 难点分析	16
2.4 例题与详析	20
2.5 习题与详解	23
第3章 程序设计的基本技术	34
3.1 内容提要	34
3.2 重点	34
3.3 难点分析	34
3.4 例题与详析	35
3.5 习题与详解	49
第4章 8086/8088 的总线与时序	107
4.1 内容提要	107
4.2 重点	107
4.3 难点分析	107
第5章 半导体存储器	111
5.1 内容提要	111

5.2 重点	111
5.3 难点分析	111
5.4 例题与详析	114
5.5 习题与详解	118
第6章 输入输出和接口技术	123
6.1 内容提要	123
6.2 重点	123
6.3 难点分析	123
6.4 例题与详析	125
6.5 习题与详解	130
第7章 中断技术	153
7.1 内容提要	153
7.2 重点	153
7.3 难点分析	154
7.4 例题与详析	155
7.5 习题与详解	159
第8章 常用可编程接口芯片	163
8.1 内容提要	163
8.2 重点	164
8.3 难点分析	164
8.4 例题与详析	166
8.5 习题与详解	173
附录 A 8086 指令系统表	191
附录 B 8086 算术逻辑运算指令对状态标志位的影响	195
附录 C 8086 指令按字母顺序查找表	196

微型计算机的基础知识

本章从学习微机原理、汇编语言与接口技术的需要出发,首先介绍了计算机中使用的数及符号在计算机内的表示方法,然后介绍了微型计算机的内部结构、存储器和堆栈以及计算机的工作原理。通过本章的学习,应熟练地掌握数及符号在计算机内的表示方法、Intel 8086/8088 中的寄存器组及作用,主存储器的编址方式,堆栈及堆栈的使用规定等,为本课程的学习打下基础。

本章的内容是学习本课程所需要的最基本的知识,其掌握程度的好坏直接影响到后面各章的学习效果,因此一定要重视本章的学习。

其中逻辑运算、逻辑电路、逻辑单元和逻辑部件等内容是对没有学习数字电路与逻辑设计的读者的最低要求,有数字电路与逻辑设计基础的读者可以阅读有关正逻辑和负逻辑的内容。

1.1 内容提要

二进制数及其编码是计算机运算的基础。计算机唯一能识别的数是二进制数,计算机的指令、数据、字符、地址均用二进制数表示。所以掌握二进制数是非常重要的。为了书写方便、读数直观、引入了十六进制数,这仅是一种手段。由于人们习惯用十进制数,所以又有这3种数制之间的转换和BCD数。

在微型计算机中,可以实现二进制数的加、减、乘、除等基本运算。为了简化电路、降低成本、增加运算速度,引入了补码表示法。利用补码表示法,可以将二进制数的减法运算变为加法运算,还可以将符号数和无符号数统一起来。

微型计算机不仅能进行算术运算,而且能进行逻辑运算。基本的逻辑运算有“与”、“或”、“非”、“异或”4种。所有的逻辑运算都是按位操作的。

本章的内容为学习微型计算机建立了一些基本的重要的概念:位、字、字节、指令、程序、微处理器、微型计算机和微型计算机系统等,还介绍了80x86微处理器、存储器的结构和堆栈。

8086/8088CPU内部由执行单元和总线接口单元两部分组成。应用时直接使用的是它们中的寄存器组。通用寄存器是CPU内部的存储器,使用率最高,应掌握它们的使用方法。段寄存器是用来存放存储器的段地址的,存储器的物理地址是由段寄存器提供的段地址和指令指定的偏移地址组成的。标志寄存器中的状态标志位反映了执行单元执行算术和逻辑运算的结果,供后面指令的执行来判别。

堆栈是用来存放信息的,对栈中的信息存取,采用“先进后出”或“后进先出”的原则。除

本章介绍的 PUSH 和 POP 指令外,还可以用后面介绍的任何访问存储器的指令访问堆栈,堆栈常用于子程序调用、子程序嵌套和中断控制等。

1.2 重 点

- (1) 补码及求补方法;
- (2) 机器数及其真值;
- (3) 位、字、字节、指令和程序等的概念;
- (4) 8086/8088 的寄存器;
- (5) 存储器的逻辑地址和物理地址,存储器的结构;
- (6) 堆栈和栈操作指令。

1.3 难 点 分 析

1. 二进制数与十进制数的转换

二进制数及其编码是计算机运算的基础。计算机唯一能识别的数是二进制数,计算机的指令(机器指令)、数据、字符、地址都是用二进制数来表示的,所以掌握二进制数是非常重要的。为了书写方便、读数直观,引入了十六进制数,这仅是一种手段。由于人们习惯用十进制数,所以要掌握十进制数与二进制数的转换方法和 BCD 数。

把一个十进制数转换为二进制数,可以先把该数转换为十六进制数,然后再转换为二进制数,这样可以减少计算次数;反之,要把一个二进制数转换为十进制数,也可以采用同样的办法。若记住 2^n (写成二进制数就是 1 后面加 n 个 0) 和十六进制数、十进制数的对应关系(见表 1-1)以及个别十进制整数(如 50, 80, 100 等)和十六进制数的对应关系,转换起来更为方便,具体实现方法请见本章例题详析和习题详解。

表 1-1 部分二进制数与十进制数及十六进制数的对应关系

2^n	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
十六进制数	20	40	80	100	200	400	800	1000	2000	4000	8000	10000
十进制数	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536

需要说明的是,在汇编语言源程序中可以使用二进制数、十六进制数,也可以使用十进制数,不需要编程作数制的转换,它们的转换工作由宏汇编程序完成。但是,若需将键盘输入的十进制数参与运算或需将运算的结果以十进制数形式输出,则要编写程序来实现将输入的十进制数字串(ASCII 码)转化为二进制数或将二进制数转化为十进制数字串。

2. 补码

符号数的最高位为符号位,其值为 0 即表示符号为正;为 1 表示符号为负。符号数在计算机内都是采用补码表示的,计算机中一个字节和一个字所能表示的补码数的范围分别

是 80H~7FH 和 8000H~7FFFH。字节 80H 和字 8000H 的真值分别是 $-128(-80H)$ 和 $-32768(-8000H)$ 。补码数 80H 和 8000H 的最高位既代表了符号为负又代表了数值为 1。

要搞清楚一个数的补码和为了求一个数的补码而进行的求补操作的区别。

补码是表示符号数的一种码制,求补是一种操作。求一个十进制数的补码,要视该十进制数是负数还是正数来确定是否要执行求补操作。若该十进制数是正数,则只需将它转换为二进制数,而不需执行求补操作;若是负数,则要先用二进制数写出该负数的绝对值,然后对写出的二进制数执行求补操作。例如 68 的补码 44H,就只需将 68 转换为二进制数;而 -68 的补码 BCH,就是先写出 -68 相应的绝对值 68 的二进制数 01000100B(44H),然后对 01000100B 执行求补操作得到的。

需要注意的是,负数的补码不能直接由其原码来求。若已知负数的原码要求其补码,则需先写出该原码的绝对值,即将该原码的符号位 1 变为 0,再对写出的绝对值执行求补操作。例如 -347 的原码为 815BH,该原码的绝对值是 015BH,对 015BH 执行求补操作得 FEA5H。FEA5H 就是 -347 的补码。

求补操作即是求补指令 NEG 所执行的操作,可以对任何数(即不论数是正还是负,也不论数是何种码)执行求补操作。对一个补码数求补,其结果是该数的补数。对一个补码表示的正数求补,其结果是与该正数的绝对值相等的负数的补码;对一个补码表示的负数求补,其结果是该负数的绝对值。一般有两种情况需要执行求补操作:一个是求一个负数的补码——对正数求补;另一个是求补码表示的负数的真值——对负数求补。例如 -68 的补码 BCH 是对 44H 执行求补操作得到的;对 -68 的补码 BCH 执行求补操作得到 44H, $-44H$ (即 -68)即是机器数 BCH 的真值。

3. 二进制数的扩展

一个二进制补码数的符号位向左扩展若干位后,所得到的补码数的真值不变。符号位向左扩展即符号扩展是指一个数从位数较少扩展到位数较多,如从 8 位(字节)扩展到 16 位(字),或从 16 位扩展到 32 位(双字)。对于用补码表示的数,正数的符号扩展应该在其前面补 0,而负数的符号扩展,则应该在前面补 1。例如:68 用 8 位表示为 44H,用 16 位表示为 0044H; -68 用 8 位表示为 BCH,用 16 位表示为 FFBCH。

无符号数是正数,无符号数的扩展是在其前面补 0。如无符号数 255 用 8 位表示为 FFH,用 16 位表示为 00FFH。原码数的最高位是符号位,其余诸位均为数值。原码正数和负数仅符号位这一位相反,数值部分相同。所以,一个二进制原码数的扩展是将其符号位向左移至最高位,最高位与原数值部分间的所有位均为 0。例如:68 用 8 位表示为 44H,用 16 位表示为 0044H; -68 用 8 位表示为 C4H,用 16 位表示为 8044H。

4. BCD(binary coded decimal)数

二进制编码的十进制数简称为 BCD 数,BCD 数是用 4 位二进制数编码来表示 1 位十进制数,所以 8 位二进制数可以表示 2 位 BCD 数。例如十进制数 29 的 BCD 数的二进制码形式为:0010 1001B;为了书写方便,再将二进制数的形式写成十六进制数的形式,那就变成了 29H。这样,十六进制数 29H 与 BCD 数 29H 在形式上就完全相同。当程序中出现

29H 时,它到底是十六进制数呢,还是 BCD 数?这就只有程序设计者知道,读程序者就要根据程序的上下关系来判断了。

对于 BCD 数的运算,例如,执行两个 BCD 数 29H 和 18H 相加,既可以先将两个加数分别转为二进制数(1DH 和 12H)之后,再进行加法计算 $1DH + 12H$,最后再将结果(2FH)转换成 BCD 数(47H);也可以先将两个 BCD 数(29H 和 18H)进行加法计算 $29H + 18H$,再用调整指令将结果(十六进制数 41H)调整为 BCD 数 47H。计算机特意设置了调整指令,使得 BCD 数的运算可以把 BCD 数作为二进制数进行运算和调整的方法来实现。

5. Intel 8086/8088 中的寄存器组

Intel 8086/8088 内部由执行单元 EU 和总线接口单元 BIU 两部分组成。执行单元负责指令的执行,而总线接口单元负责与内存储器、外部设备接口,执行所有的总线操作。应用时直接使用是它们中的寄存器组。通用寄存器是 CPU 内部的存储器,使用最多,应掌握它们的使用方法。通用寄存器组的专门用途应在今后的学习中逐渐理解后掌握,切勿现在就死记硬背。标志寄存器中的 6 个状态标志位(CF、ZF、SF、OF、PF 和 AF)反映了执行单元执行算术或逻辑运算的结果,供后面的执行来判别。其中最主要的是要搞清楚溢出标志位 OF 和进位标志位 CF 的置位方式及它们的区别。参与运算的数不论无符号数还是符号数,计算机都按符号数置位溢出标志,按无符号数置位进位标志。运算结果超出了字或字节所能表示的范围时,即字运算结果超出了 $8000H \sim 7FFFH$,字节运算结果超出了 $80H \sim 7FH$ 时溢出,OF 置 1,否则 OF 置 0。对加运算来说,如最高位向前产生了进位,则进位标志置 1,否则置 0;对减运算来说,最高位向前产生了借位即不够减,则进位标志置 1,不产生借位即够减进位标志置 0。

6. 存储器

在用汇编语言进行程序设计的过程中,需要自己定义数据存储区并安排数据存取的细节,因此,必须了解存储器的编址方式,这样才能正确地使用存储器。

存储器是按字节编址的,即每一个字节都有一个确定的地址号,称为此字节的物理地址,并约定字地址是由两个字节地址中较小的一个确定。字数据在存储器中的存放方式是:两个连续地址单元中,较小地址中存放的是字的低 8 位,高 8 位存放在较大地址单元中,一定不能弄颠倒。双字或多字节数据的存放原则是数据的低位在低地址,数据的高位在高地址;且其中的最低地址是该数据的地址。

8086/8088 中的存储器容量最大可达 1MB,因而物理地址必须用 20 位二进制数表示。但 CPU 内部的寄存器都是 16 位的,因此在程序中不直接使用物理地址而使用逻辑地址,即段地址:偏移地址。段地址存放在 4 个段寄存器中,偏移地址可以存放在 6 个寄存器(IP, SP, BP, SI, DI, BX)中,在形成物理地址时应该使用哪个寄存器是总线接口单元根据操作的要求来确定的。

CPU 在当前某时刻最多只能访问 4 个段,但并不限制一个源程序中只能定义 4 个段,可以根据需要定义少于或多于 4 个段。当程序中需要使用多个段时,只要改变相应段寄存器的内容即可。

偏移地址为段中某一存储单元到该段首地址的字节距离。

7. 堆栈

堆栈由存储区和一个指示器组成。堆栈的存储区一端封闭、一端开口,数据的存取(进出)只能在开口端进行,存(进)取(出)原则为“先进后出”。8086/8088 允许建立的堆栈段最大可达 64KB,堆栈只能作字操作而不能作字节操作。由于堆栈也是存储器的一个区域,所以不对堆栈区域的存储器进行栈操作就没有“先进后出”和字操作的限制。

堆栈存储区的位置由堆栈段寄存器 SS 给定,并自动使用 SP 作指示器。堆栈指示器 SP 所指的单元称为栈顶,执行栈操作存取数据均在栈顶进行。应特别注意栈顶是随着堆栈指示器 SP 的变化而变化的,因此,在使用堆栈时,应特别注意堆栈指示器的内容,不可随意改变,否则会造成混乱。

1.4 例题与详析

例 1-1 将十六进制数 1F3DH 转换为十进制数。

解: 方法 1 将十六进制数按权展开相加。

$$\begin{aligned} 1F3DH &= 1 \times 16^3 + 15 \times 16^2 + 3 \times 16^1 + 13 \times 16^0 \\ &= 4096 + 3840 + 48 + 13 \\ &= 7997 \end{aligned}$$

方法 2 用 2ⁿ 和十六进制数、十进制数的对应关系写出十进制数再相加(减)。

$$\begin{aligned} 1F3DH &= 2000H - 100H + 3DH \\ &= 2000H - 100H + 40H - 3 \\ &= 8192 - 256 + 64 - 3 \\ &= 7997 \end{aligned}$$

例 1-2 将十进制数 38974 转换为十六进制数。

解: 方法 1 利用除 16 取余法。

$$\begin{array}{r} 16 \overline{) 38947} \quad 3 \\ \underline{16 \overline{) 2434}} \quad 2 \\ \underline{16 \overline{) 152}} \quad 8 \\ \underline{16 \overline{) 9}} \quad 9 \\ 0 \end{array}$$

转换结果为: 38947 = 9823H

方法 2 利用 2ⁿ 和十六进制数、十进制数的对应关系求解。

$$\begin{aligned} 38947 &= 32768 + 6179 \\ &= 32768 + 4096 + 2048 + 35 \\ &= 8000H + 1000H + 800H + 20H + 3 \\ &= 9800H + 23H \\ &= 9823H \end{aligned}$$

例 1-3 求下列十进制数的补码: ±105, ±68, ±10。

解: 求十进制正数的补码的方法是将其转换为十六进制数;而求十进制负数的补码的

方法是先将其相应的正数即绝对值转换为 8 位、16 位或 32 位二进制数,然后对写出的二进制数求补。

$$(1) 105 = 100 + 5 = 64H + 5H = 69H$$

(2) -105 的绝对值为 $01101001B$, 对其求补得

$$10010111B = 97H$$

$$(3) 68 = 64 + 4 = 40H + 04H = 44H$$

(4) -68 的绝对值为 $01000100B$, 对其求补得

$$10111100B = BCH$$

$$(5) 10 = 8 + 2 = 0AH$$

(6) -10 的绝对值为 $00001010B$, 对其求补得

$$11110110B = F6H$$

例 1-4 求下列补码数的真值: $11FH, 9BH, C9H, DEH$ 。

解: 求一个补码数的真值,要先查看该补码数是正数还是负数。正数只需将其转换为十进制数;负数则要先将该数转换为二进制数,然后对该数求补得到该负数的绝对值并在绝对值前添加一负号得到一个带负号的 8 位、16 位或 32 位二进制数,最后再将二进制数转换为十进制数。查看补码数是正数还是负数的方法是看用 2 位、4 位或 8 位表示的十六进制数的最高位是否小于 8,小于 8 为正数,大于等于 8 则为负数。

(1) $11FH$ 为正数,只要将其转换为十进制数即可。

$$11FH = 100H + 10H + 0FH = 256 + 16 + 15 = 287$$

(2) $9BH, C9H, DEH$ 为字节负数,要先将其转换为带负号的二进制数,再将其转换为十进制数。

$$9BH = 10011011B = -01100101B = -65H = -(64H + 1) = -101$$

$$C9H = 11001001B = -00110111B = -37H = -32H - 5 = -55$$

$$DEH = 11011110B = -00100010B = -22H = -34$$

例 1-5 算式 $11 \times 4096 + 7 \times 512 + 6 \times 64 + 5 \times 8 + 3$ 的结果以二进制数表示时含有多少个 1?

解: 先看 5 个乘数

$$4096 = 2^{12} = 1000000000000B$$

$$512 = 2^9 = 1000000000B$$

$$64 = 2^6 = 1000000B$$

$$3 = 2^3 = 1000B$$

$$1 = 2^0 = 1B$$

5 个被乘数 11, 7, 6, 5 和 3 用二进制数表示分别为 $1101, 0111, 0110, 0101$ 和 0011 , 每个被乘数中 1 的位数都不超过 3 位。而这 5 个乘数用二进制数表示都只含有 1 个 1, 且每个乘数的最高位为 1, 1 后面的 0 的个数依次较其相邻的较大乘数少 3 个。5 个加数相乘后, 后面的加数中 1 位的最大权值均小于前面的加数中 1 位的最小权值。所以 5 个加数的加法没有 1 和 1 相加, 只有 $1+0$ 和 $0+0$ 。因此, 不用作乘法, 也不用作加法就可以确定, 该算式的结果以二进制数表示时, 含有 1 的个数等于这 5 个被乘数 11, 7, 6, 5 和 3 含有 1 的个数之和。11 和 7 用二进制数表示都含有 3 个 1, 6, 5 和 3 用二进制数表示都含有 2 个 1。所以 1

的个数等于 $12(3+3+2+2+2)$ 。

1.5 习题与详解

1-1 将下列十进制数转换为十六进制数: 10, 64, 78, 80, 93, 100, 125, 255。

解:

$$10 = 0AH$$

$$64 = 40H$$

$$78 = 80 - 2 = 10 \times 8 = 1010B \times 1000B - 2$$

$$= 01010000B - 2 = 50H - 2 = 4EH$$

$$80 = 50H$$

$$93 = 100 - 7 = 64H - 7 = 5DH$$

$$100 = 64H$$

$$125 = 128 - 3 = 80H - 3 = 7DH$$

$$255 = FFH$$

1-2 将下列十六进制无符号数转换为十进制数: 2CH, 64H, D5H, 100H, 378H, 4FEH, CADH。

解:

$$2CH = 20H + 0CH = 32 + 12 = 44$$

$$64H = 100$$

$$D5H = 80H + 40H + 15H = 128 + 64 + 21 = 213$$

$$100H = 256$$

$$378H = 400H - 88H = 400H - 80H - 8H = 1024 - 128 - 8$$

$$= 888$$

$$4FEH = 500H - 2H = 400 + 100H - 2H = 1024 + 256 - 2$$

$$= 1278$$

$$CADH = 1000H - 353H = 1000H - 400H + ADH$$

$$= 1000H - 400H + 80H + 20H + 0DH$$

$$= 4096 - 1024 + 128 + 32 + 13 = 3245$$

1-3 写出下表所列十进制数的原码和补码, 用 2 位或 4 位或 8 位十六进制数填入表中。

十进制数	原码	补码	十进制数	原码	补码
18			928		
-18			-928		
30			8796		
-30			-8796		
347			65530		
-347			-65530		

解: 求原码略。

求补码的计算如下。