

新世纪高等学校计算机学科系列教材

# 汇编语言程序设计

徐建民摇等编著

王凤先摇审

马建民摇等编著 王凤先摇审 徐建民摇等编著 王凤先摇审

北京·机械工业出版社



若书店售缺,请与本社发行部联系调换。电话摇边园苑

# 前 摇 摇 言

汇编语言程序设计是计算机科学与技术专业的一门重要课程,也是其他相关专业的一门必修或选修课。利用汇编语言可以编写出时空效率高的程序,在某些领域,汇编语言仍然是必不可少的编程语言之一。

目前,整个中国高等教育正处于教学改革的年代,新的教学思路、新的课程体系和教学内容正在形成。根据全国高等学校计算机教育研究会课程与教材建设委员会的指导意见,结合面向 21 世纪计算机科学与技术专业课程改革的基本思路,在几年教学实践的基础上,我们编写了这本《汇编语言程序设计》。

考虑到国内广泛使用的微型计算机都是以 Intel 的 8086 系列微处理器或者兼容的微处理器为载体的,所以本书以 8086 系列微处理器为基础,系统地介绍汇编语言程序设计的基础知识、程序设计方法和应用技术。

全书共分 10 章。前两章概括地介绍数在计算机中的表示,汇编语言运行的硬件环境——8086 和 8088 微处理器以及存储器的组成和结构,它是后续内容的基础。对于已经了解微型计算机原理的读者,这部分内容可以跳过。第 3 章、第 4 章介绍寻址方式、指令系统和汇编语言的程序结构。第 5 章是本书的核心部分,详细地介绍了 8086 和 8088 微型计算机汇编语言程序设计的基本方法和技巧。第 6 章介绍汇编语言和高级语言的混合编程方法。第 7 章概要介绍保护模式下汇编语言程序设计的基本方法和应用技术。考虑到新教学计划的课时安排,本书语言尽可能简练,故关于浮点数编程的内容未包括在内。

本书第 1 章至第 3 章、第 4 章、第 5 章由徐建民编写,第 6 章至第 7 章由袁方编写,第 8 章和第 9 章由杨晓辉编写。全书由徐建民统稿。

本书的编写得到北京工业大学李大友老师的大力支持,王凤先教授在百忙中审阅了全书并且提出了宝贵的修改意见,在此对他们的帮助表示衷心的感谢。本书的初稿曾在河北大学试用,得到了有关老师和同学的许多帮助,在此一并表示感谢。

由于编者能力所限,加之编写时间仓促,书中不妥甚至错误在所难免,恳切希望读者批评指正。

作 摇 者  
袁 方 袁 月

# 序 摇 摇 言

摇摇这套教材是面向 21 世纪计算机学科系列教材。为什么要组织这套教材？根据什么编写这套教材？这些都是在这篇序言中要回答的问题。

计算机学科是一个飞速发展的学科,尤其是近十年来,计算机向高度集成化、网络化和多媒体化发展的速度一日千里。但是,从另一个方面来看,目前高等学校的计算机教育,特别是教材建设,远远落后于现实的需要。现在的教材主要是根据《教学计划 1985》的要求组织编写的。这个教学计划,在制定过程中主要参照了美国 1983 年和 1985 年的《教学计划 1985》。

近年来,计算机学科已有了长足发展,这就要求高等学校计算机教育必须跟上形势发展的需要,在课程建设和教材建设上做出相应调整,以适应面向 21 世纪计算机教育的要求。这是组织这套教材的初衷。

为了组织好这套教材,全国高等学校计算机教育研究会课程与教材建设委员会在天津召开了“全国高等学校计算机学科课程与教材建设研讨会”,在北京召开了“教材编写大纲研讨会”。在这两次会议上,代表们深入地研讨了全国高校计算机专业教学指导委员会和中国计算机学会教育委员会制定的《计算机学科教学计划 1993》以及美国 1983 年和 1985 年的《计算机学科教学计划 1985》,这是这套教材参照的主要依据。

1983 年和 1985 年的《计算机学科教学计划 1985》是在总结了从《计算机学科教学计划 1985》到现在,计算机学科十年来发展的主要成果的基础上诞生的。它认为面向 21 世纪计算机学科应包括 9 个主科目,其中 6 个主科目为核心主科,它们是:算法与分析(1)、体系结构(2)、离散结构(3)、计算科学(4)、图形学、可视化、多媒体(5)、网络计算(6)、人机交互(7)、信息管理(8)、智能系统(9)、操作系统(10)、程序设计基础(11)、程序设计语言(12)、软件工程(13)、社会、道德、法律和专业问题(14)。其中除 4 和 8 为非核心主科目外,其他 6 项均为核心主科目。

将 1993 年教学计划与 1985 年教学计划比较可看出:

(1) 在 1985 年计划中,离散结构只作为数学基础提出,而在 1993 年计划中,则作为核心主科目提出,显然,提高了它在计算机学科中的地位。

(2) 在 1985 年计划中,未提及网络计算,而在 1993 年计划中,则作为核心主科目提出,以适应网络技术飞速发展的需求。

(3) 图形学、可视化与多媒体也是为适应发展要求新增加的内容。

除此之外,1993 年计划在下述 3 个方面做调整:

将程序设计语言引论调整为程序设计基础,将人-机通信调整为人机交互,将人工智能与机器人学调整为智能系统,将数据库与信息检索调整为信息管理,将数值与符号计算调整为计算科学。

显然,这些变化使 1993 年计划更具有科学性,也更好地适应了学科发展的需要。

在组织这套教材的过程中,充分考虑了这些变化和调整,在软件和硬件的课程体系、界面划分方面均做了相应的调整,使整套教材更具有科学性和实用性。

另外,还要说明一点,教材建设既要满足必修课的要求,又要满足限选课和任选课的要求。

因此,教材应按系列组织,反映整个计算机学科的要求,采用大拼盘结构,以适应各校不同的具体教学计划,使学校可根据自己的需求进行选择。

这套教材包括:《微机应用基础》、《离散数学》、《电路与电子技术》、《电路与电子技术习题与实验指南》、《数字逻辑与数字系统》、《计算机组成原理》、《微机接口技术》、《计算机体系结构》、《计算机网络》、《计算机网络实验教程》、《通信原理》、《计算机网络管理》、《网络信息系统集成》、《多媒体技术》、《计算机图形学》、《计算机维护技术》、《数据结构》、《计算机算法设计与分析》、《计算机数值分析》、《汇编语言程序设计》、《~~汇编语言~~语言程序设计》、《~~汇编语言~~程序程序设计》、《~~汇编语言~~悦语言程序设计》、《~~汇编语言~~悦垣垣语言程序设计》、《~~汇编语言~~悦垣垣语言程序设计》、《~~汇编语言~~操作系统原理》、《~~汇编语言~~操作系统原理与应用》、《~~汇编语言~~操作系统》、《软件工程》、《数据库系统原理》、《编译原理》、《编译方法》、《人工智能》、《计算机信息安全》、《计算机图像处理》、《人机交互》、《计算机伦理学》。对于~~陈永~~和~~粤坛~~的《计算机学科教学计划~~规划~~》中提出的~~五个~~个主科目,这套系列教材均涵盖,能够满足不同层次院校、不同教学计划的要求。

这套系列教材由全国高等学校计算机教育研究会课程与教材建设委员会主任李大友教授精心策划和组织。编者均为具有丰富教学实践经验的专家和教授。所编教材体系结构严谨、层次清晰、概念准确、论理充分、理论联系实际、深入浅出、通俗易懂。

教材组织过程中,得到了哈尔滨工业大学蒋宗礼教授,西安交通大学董渭清副教授,武汉大学张焕国教授,吉林大学张长海教授,福州大学王晓东教授,太原理工大学余雪丽教授等的大力支持和帮助,在此一并表示衷心感谢。

李大友  
圆垣垣年 远月

# 目 录

第 1 章 基础知识 .....	(1)
1.1 数据表示方法 .....	(1)
1.1.1 数与数制 .....	(1)
1.1.2 计算机中的数据表示 .....	(1)
1.1.3 基本数据类型 .....	(2)
1.2 汇编语言程序设计 .....	(2)
1.2.1 程序设计语言 .....	(2)
1.2.2 如何学习汇编语言 .....	(2)
本章小结 .....	(2)
习题一 .....	(2)
第 2 章 微处理器的结构及存储器组成 .....	(3)
2.1 8086 和 8088 微处理器的结构 .....	(3)
2.1.1 8086 和 8088 微处理器的结构 .....	(3)
2.1.2 8086 和 8088 微处理器的寄存器结构 .....	(3)
2.2 存储器的组织 .....	(3)
2.2.1 实模式存储器寻址 .....	(3)
2.2.2 保护模式存储器寻址 .....	(3)
本章小结 .....	(3)
习题二 .....	(3)
第 3 章 寻址方式和指令系统 .....	(3)
3.1 寻址方式 .....	(3)
3.1.1 数据寻址方式 .....	(3)
3.1.2 程序存储器寻址方式 .....	(3)
3.2 指令系统 .....	(3)
3.2.1 数据传送指令 .....	(3)
3.2.2 算术运算指令 .....	(3)
3.2.3 十进制算术运算指令 .....	(3)
3.2.4 逻辑运算指令 .....	(3)
3.2.5 处理机控制指令 .....	(3)
本章小结 .....	(3)
习题三 .....	(3)
第 4 章 伪指令及汇编语言源程序结构 .....	(3)
4.1 汇编语言语句格式 .....	(3)
4.1.1 语句种类 .....	(3)

摇摇摇摇	源源源源语句格式	(源)
摇摇摇摇	源源源源伪指令	(缘)
摇摇摇摇	源源源源符号定义伪指令	(缘)
摇摇摇摇	源源源源数据定义伪指令	(缘)
摇摇摇摇	源源源源段定义伪指令	(缘)
摇摇摇摇	源源源源简化段定义伪指令	(缘)
摇摇摇摇	源源源源程序开始和结束伪指令	(缘)
摇摇摇摇	源源源源指令集选择伪指令	(缘)
摇摇摇摇	源源源源过程定义伪指令	(缘)
摇摇源源	源源源源汇编语言源程序结构	(缘)
摇摇源源	源源源源完整段定义结构	(缘)
摇摇源源	源源源源简化段定义结构	(远)
摇摇源源	源源源源程序段前缀结构	(远)
摇摇源源	源源源源源文件结构	(远)
摇摇源源	源源源源汇编语言的上机过程	(远)
摇摇源源	源源源源概述	(远)
摇摇源源	源源源源建立汇编语言的工作环境	(远)
摇摇源源	源源源源用 源源建立 源源文件	(远)
摇摇源源	源源源源用 源源产生 源源文件	(远)
摇摇源源	源源源源用 源源产生 源源文件	(远)
摇摇源源	源源源源程序的调试和执行	(远)
摇摇	本章小结	(远)
摇摇	习题四	(远)
第 缘章	源源基本结构程序设计	(苑)
摇摇源源	源源源源汇编语言程序设计概述	(苑)
摇摇源源	源源源源汇编语言程序设计的基本步骤	(苑)
摇摇源源	源源源源流程图的画法规定	(苑)
摇摇源源	源源源源顺序结构程序设计	(苑)
摇摇源源	源源源源分支程序设计	(苑)
摇摇源源	源源源源转移指令	(苑)
摇摇源源	源源源源双分支程序设计	(苑)
摇摇源源	源源源源多分支程序设计	(苑)
摇摇源源	源源源源循环结构程序设计	(愿)
摇摇源源	源源源源循环指令	(愿)
摇摇源源	源源源源循环程序的结构	(愿)
摇摇源源	源源源源循环程序设计方法	(愿)
摇摇源源	源源源源多重循环程序设计	(愿)
摇摇源源	源源源源串操作程序	(愿)
摇摇源源	源源源源循环程序设计举例	(愿)
摇摇	本章小结	(愿)

摇摇习题五 .....	(页码)
第 远章摇子程序设计 .....	(页码)
摇摇远章摇子程序的概念与特性 .....	(页码)
摇摇远章摇子程序调用和返回指令 .....	(页码)
摇摇摇摇 远章摇调用指令 .....	(页码)
摇摇摇摇 远章摇返回指令 .....	(页码)
摇摇远章摇子程序的结构形式 .....	(页码)
摇摇摇摇 远章摇子程序调用方法说明 .....	(页码)
摇摇摇摇 远章摇现场保护和现场恢复 .....	(页码)
摇摇摇摇 远章摇子程序的定义 .....	(页码)
摇摇远章摇子程序的设计和调用 .....	(页码)
摇摇摇摇 远章摇子程序的设计 .....	(页码)
摇摇摇摇 远章摇子程序的调用 .....	(页码)
摇摇远章摇子程序的参数传递方法 .....	(页码)
摇摇摇摇 远章摇通过寄存器传递参数 .....	(页码)
摇摇摇摇 远章摇通过堆栈传递参数 .....	(页码)
摇摇摇摇 远章摇用存储单元传递参数 .....	(页码)
摇摇远章摇子程序的嵌套与递归 .....	(页码)
摇摇摇摇 远章摇子程序的嵌套调用 .....	(页码)
摇摇摇摇 远章摇子程序的递归调用 .....	(页码)
摇摇远章摇子程序设计举例 .....	(页码)
摇摇摇摇 远章摇输入输出子程序 .....	(页码)
摇摇摇摇 远章摇数制转换子程序 .....	(页码)
摇摇摇摇 远章摇多位数运算子程序 .....	(页码)
摇摇本章小结 .....	(页码)
摇摇习题六 .....	(页码)
第 苑章摇高级汇编技术 .....	(页码)
摇摇苑章摇宏汇编 .....	(页码)
摇摇摇摇 苑章摇宏指令的定义、调用和展开 .....	(页码)
摇摇摇摇 苑章摇宏操作符 .....	(页码)
摇摇摇摇 苑章摇强制伪指令 .....	(页码)
摇摇摇摇 苑章摇宏嵌套 .....	(页码)
摇摇摇摇 苑章摇宏程序库 .....	(页码)
摇摇摇摇 苑章摇宏指令与子程序的区别 .....	(页码)
摇摇苑章摇重复汇编和条件汇编 .....	(页码)
摇摇摇摇 苑章摇重复汇编 .....	(页码)
摇摇摇摇 苑章摇条件汇编 .....	(页码)
摇摇本章小结 .....	(页码)
第 愿章摇中断和输入输出程序设计 .....	(页码)
摇摇愿章摇中断概述 .....	(页码)

摇摇摇摇	愿愿愿愿	摇摇摇摇中断与中断源	· · · · · ·	(员缘)
摇摇摇摇	愿愿愿愿	摇摇摇摇中断分类	· · · · · ·	(员缘)
摇摇摇摇	愿愿愿愿	摇摇摇摇中断向量表	· · · · · ·	(员苑)
摇摇摇摇	愿愿愿愿	摇摇摇摇中断过程	· · · · · ·	(员苑)
摇摇摇摇	愿愿愿愿	摇摇摇摇中断优先级	· · · · · ·	(员愿)
摇摇摇摇	愿愿愿愿	摇摇摇摇中断指令	· · · · · ·	(员愿)
摇摇愿愿愿	愿愿愿愿	摇摇摇摇中断处理程序设计	· · · · · ·	(员怨)
摇摇摇摇	愿愿愿愿	摇摇摇摇中断处理程序的编写	· · · · · ·	(员怨)
摇摇摇摇	愿愿愿愿	摇摇摇摇设置和获取中断向量	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇中断程序设计举例	· · · · · ·	(员园)
摇摇愿愿愿	愿愿愿愿	摇摇摇摇月影杂中断调用	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇月影杂概述	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇月影杂中断调用方法	· · · · · ·	(员园)
摇摇愿愿愿	愿愿愿愿	摇摇摇摇阂字功能调用	· · · · · ·	(员缘)
摇摇摇摇	愿愿愿愿	摇摇摇摇阂字功能调用概述	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇基本阂字功能调用	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇应用举例	· · · · · ·	(员愿)
摇摇愿愿愿	愿愿愿愿	摇摇摇摇磁盘文件管理	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇传统文件管理方式	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇扩充文件管理方式	· · · · · ·	(员源)
摇摇愿愿愿	愿愿愿愿	摇摇摇摇输入输出程序设计	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇程序直接控制方式	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇程序中中断方式	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇直接存储器访问(阂云)方式	· · · · · ·	(员源)
摇摇摇摇	愿愿愿愿	摇摇摇摇通道传输方式	· · · · · ·	(员缘)
摇摇本章小结			· · · · · ·	(员缘)
摇摇习题八			· · · · · ·	(员园)
第 怨章	愿愿愿愿	摇摇摇摇汇编语言与高级语言的混合编程	· · · · · ·	(员苑)
摇摇愿愿愿	愿愿愿愿	摇摇摇摇调用协议	· · · · · ·	(员苑)
摇摇摇摇	愿愿愿愿	摇摇摇摇入口参数传递规则	· · · · · ·	(员苑)
摇摇摇摇	愿愿愿愿	摇摇摇摇返回值传递规则	· · · · · ·	(员愿)
摇摇摇摇	愿愿愿愿	摇摇摇摇寄存器保护规则	· · · · · ·	(员愿)
摇摇愿愿愿	愿愿愿愿	摇摇摇摇与 悦语言的接口	· · · · · ·	(员愿)
摇摇摇摇	愿愿愿愿	摇摇摇摇模块连接法	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇伪变量法	· · · · · ·	(员园)
摇摇摇摇	愿愿愿愿	摇摇摇摇行内汇编法	· · · · · ·	(员园)
摇摇愿愿愿	愿愿愿愿	摇摇摇摇与 孕葬粤语言的接口	· · · · · ·	(员源)
摇摇本章小结			· · · · · ·	(员园)
摇摇习题九			· · · · · ·	(员园)

第 4 章 保护模式程序设计 .....	( 4-1)
4.1 保护模式存储器管理 .....	( 4-2)
4.2 段寄存器与存储器分段管理 .....	( 4-3)
4.3 控制寄存器与存储器分页管理 .....	( 4-5)
4.4 调试寄存器与测试寄存器 .....	( 4-6)
4.5 描述符的类型 .....	( 4-7)
4.5.1 存储段描述符 .....	( 4-7)
4.5.2 系统段描述符 .....	( 4-8)
4.5.3 控制门描述符 .....	( 4-9)
4.6 保护模式下的系统类指令 .....	( 4-10)
4.6.1 实模式和保护模式任何特权级下都可执行的指令 .....	( 4-10)
4.6.2 实模式和保护模式 0 特权级下可以执行的指令 .....	( 4-11)
4.6.3 仅在保护模式下执行的指令 .....	( 4-11)
4.6.4 特权指令 .....	( 4-11)
4.7 控制转移与特权级变换 .....	( 4-12)
4.7.1 任务状态段 .....	( 4-12)
4.7.2 同一任务内特权级不变的段间转移 .....	( 4-13)
4.7.3 同一任务内特权级变换的段间转移 .....	( 4-13)
4.7.4 不同任务间的控制转移 .....	( 4-14)
4.8 中断异常处理 .....	( 4-15)
4.8.1 中断 .....	( 4-15)
4.8.2 异常 .....	( 4-15)
4.8.3 中断异常处理的控制转移 .....	( 4-16)
4.9 虚拟 8086 工作方式 .....	( 4-17)
4.9.1 虚拟 8086 方式 .....	( 4-17)
4.9.2 离开虚拟 8086 方式 .....	( 4-18)
4.9.3 进入虚拟 8086 方式 .....	( 4-18)
4.10 实模式下保护模式编程 .....	( 4-19)
4.11 本章小结 .....	( 4-20)
4.12 习题十 .....	( 4-20)
附录 A 动态调试程序 阅读 .....	( 4-21)
附录 B 普通 80386 指令系统 .....	( 4-22)
附录 C 常用 80386 杂项功能调用 .....	( 4-23)
参考文献 .....	( 4-24)

# 第1章 基础知识

## 1.1 数据表示方法

### 1.1.1 数与数制

数可以用不同的计数制来表示，习惯上常用十进制计数法。在计算机中，为了便于信息存储和计算，采用二进制计数法。一般地，基数为  $r$  的  $r$  进制数的值可以表示为：

$$a_n r^n + a_{n-1} r^{n-1} + \dots + a_0 r^0 + a_{-1} r^{-1} + a_{-2} r^{-2} + \dots + a_{-m} r^{-m}$$

其中  $a_i$  可以是  $0, 1, \dots, r-1$  中的任一数码， $r^i$  则是各位的权。

#### 1. 二进制数(Binary)

二进制数由 0、1 两个数码构成，基数为 2，第  $i$  位的权为  $2^i$ ，运算逢二进一。为了和十进制数区别，书写时在尾部加注字母 B。

二进制数  $a_n a_{n-1} \dots b_{-1} b_{-2} \dots b_{-m}$  的值为：

$$a_n \times 2^n + a_{n-1} \times 2^{n-1} + \dots + a_0 2^0 + b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \dots + b_{-m} \times 2^{-m}$$

其中  $a_i$  和  $b_i$  为 0, 1 两个数码中一个。

例如： $101011B = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 43_{10}$

$n$  位二进制数可以表示  $2^n$  种组合。3 位能够表示 8 种组合，0 至 7 的整数；4 位能够表示 16 种组合，0 至 15 的整数；8 位能够表示 256 种组合，0 至 255 的整数，16 位能表示  $64 \times 1024(64K)$  种组合。

#### 2. 八进制数(Octal)和十六进制数(Hexadecimal)

二进制数在计算机中容易实现，易于存储，抗干扰性强；但二进制数的一个很大缺点是表示一个数所需位数多，人们阅读、书写、记忆等不太方便。例如十进制数  $1000_{10}$ ，用二进制数表示则需要 10 位二进制数字  $1111101000_2$ 。为了便于人们阅读与书写，经常用八进制数或十六进制数来代替二进制数。

八进制数由 0, 1, 2, 3, 4, 5, 6, 7 八个数码构成，基数为 8，第  $i$  位的权为  $8^i$ ，运算逢八进一。

八进制数  $a_n a_{n-1} \dots a_0 b_{-1} b_{-2} \dots b_{-m}$  的位是：

$$a_n \times 8^n + a_{n-1} \times 8^{n-1} + \dots + a_0 8^0 + b_{-1} \times 8^{-1} + b_{-2} \times 8^{-2} + \dots + b_{-m} \times 8^{-m}$$

其中  $a_i$  和  $b_i$  为 0, 1, ..., 7 八个数码中一个。书写时在尾部加注字母 O，由于字母 O 与数字 0 容易混淆，八进制数亦常用尾标 Q 标识。

十六进制数由 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F 十六个数码构成，其中数码 A, B, C, D, E, F 对应于十进制数 10, 11, 12, 13, 14, 15，基数为 16，第  $i$  位的权为  $16^i$ ，运算逢十六进一。

书写时在尾部加注字母H或下标16描述以字母开始的十六进制数的前面应加一数码0，以区别于字符串。十进制数可以省略尾注D。

十六进制数  $a_n a_{n-1} \dots a_0 b_{-1} b_{-2} \dots b_{-m}$  的值是：

$$a_n \times 16^n + a_{n-1} \times 16^{n-1} + \dots + a_0 16^0 + b_{-1} \times 16^{-1} + b_{-2} \times 16^{-2} + \dots + b_{-m} \times 16^{-m}$$

其中  $a_i$  和  $b_i$  为 0, 1, ..., F 十六个数码中一个。

例如：14AFH =  $1 \times 16^3 + 4 \times 16^2 + 10 \times 16^1 + 15 = 5295_{10}$

十、二、八和十六进制数之间的对应关系见表 1-1。

表 1-1 十、二、八、十六进制对应关系表

十进制	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
二进制	000	001	010	011	100	101	110	111								
八进制	0	1	2	3	4	5	6	7	8							
二进制	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
十六进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

### 3. 数制转换

#### (1) 非十进制数转换成十进制数

非十进制数转换成十进制数的方法很简单，只需按权展开后相加即可。

【例 1-1】

$$101101.01B = 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-2} = 45.25_{10}$$

$$345Q = 3 \times 8^2 + 4 \times 8^1 + 5 \times 8^0 = 229_{10}$$

$$0F2DH = 15 \times 16^2 + 2 \times 16^1 + 13 \times 16^0 = 3885_{10}$$

#### (2) 十进制数转换成非十进制数

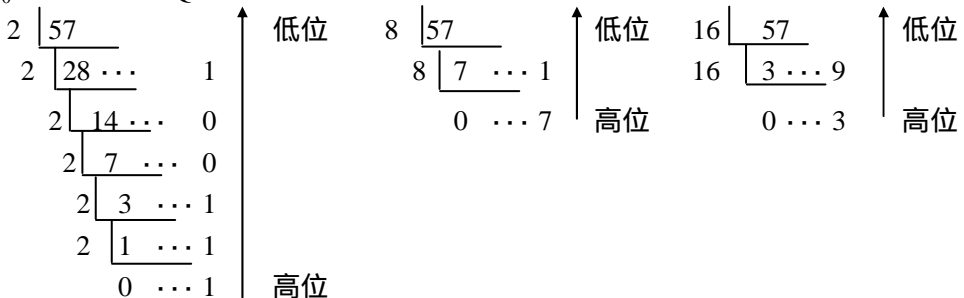
转换时将整数部分与小数部分分别转换。

整数部分采用除基数取余法，直至商为 0。先得到的余数为低位，后得到的余数为高位。

小数部分采用乘基数取整法，直至乘积为整数或达到控制精度。

【例 1-2】

$$57_{10} = 111001B = 71Q = 39H$$



$$0.625_{10} = 0.101B = 0.5Q = 0.AH$$

$$0.625 \times 2 = 1.25 \Lambda 1$$

$$0.625 \times 8 = 5 \Lambda 5$$

$$0.625 \times 16 = 10 \Lambda A$$

$$0.25 \times 2 = 0.5 \Lambda 0$$

$$0.5 \times 2 = 1 \Lambda \Lambda 1$$

#### (3) 二、八、十六进制数之间的转换

将二进制数转换成八进制数可按三位一组进行，转换成十六进制可按四位一组进行，每一组对应八进制或十六进制相应数码。分组时如果位数不够，整数部分在最左边补 0，小数部分在最右边补 0。

将八进制数转换成二进制数，只需将八进制数一位对应换成二进制数三位即可；同样对十六进制数，只需将其中一位对应换成二进制数四位即可。

【例 1-3】

$$\begin{aligned} 1100100.11010B &= \underbrace{001}_1 \underbrace{100}_4 \underbrace{100}_4 \underbrace{110}_6 \underbrace{100}_4 = 144.64Q \\ &= \underbrace{0110}_6 \underbrace{0100}_4 \underbrace{1101}_D \underbrace{0000}_0 = 64.DH \end{aligned}$$

### 1.1.2 计算机中的数据表示

数的原值称为真值，它用常规的数符正负号及常规的数码表示。

在计算机中，数的符号也用二进制来表示。连数符一起数字化了的数，称为机器数。一般用最高有效位表示数的符号，正数用 0 表示，负数用 1 表示。机器数可用不同码制表示，常用的有原码、补码、反码三种表示法，广泛应用的是原码和补码两种表示法。

#### 1. 数的原码表示

原码表示是一种比较直观的机器数表示方法，原码与真值的区别仅仅是符号位数字化。 $+1011$  采用原码表示为 01011， $-1011$  采用原码表示 11011。数据 0 的原码有两种，即正 0 和负 0。

$$[+0] = 000 \dots 000$$

$$[-0] = 100 \dots 000$$

采用原码做乘除运算比较方便，可取绝对值直接运算，而符号单独处理。对于应用最多的加减运算，原码表示不太方便，像  $(-2)+3$  表面做加法操作，而实际需做  $(3-2)$  减法操作；在计算机中实现时，比较繁琐。

#### 2. 数的补码表示

为了简化运算，人们总结出数的另一种表示方法——补码表示法。

正数的补码就是其本身，形式与原码相同。负数的补码符号与原码相同，其余各位取反，然后末位加 1。

$+1011$  采用补码表示为 01011， $-1011$  采用补码表示为 10101。

$[x-y]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}}$ ，符号位同时参加运算，所以补码表示可以将减法变成加法。

从补码求原码或真值，可按求补码同样的方法进行。

$n$  位补码表示的数的范围为  $[-2^{n-1}, 2^{n-1}-1]$ 。

【例 1-4】求  $-59+61$  的值。

$$(-59)_{10} + (61)_{10} = 1100\ 0101 + 0011\ 1101 = \underbrace{1\ 0000\ 0010}_{\text{溢出}} = (2)_{10}$$

不仅二进制数有补码的概念，任意进制数都可以引入补码的概念。R 进制数的补码定义为：

$$[x]_{\text{补}} = r^n + [x]$$

【例 1-5】求十进制数  $-41$  和  $59$  的补码。

$$[-41]_{补} = 10^2 + (-41) = 59$$

$$[59]_{补} = 10^2 + 59 = 159 = 59$$

└ 溢出

### 3. 数的反码表示

正数的原码与原码相同，负数的反码最高位为 1，其余各位依次求反。

由反码的定义可以得出补码的定义：正数的补码是它本身，负数的补码是它的反码加 1。

在反码表示中，“0”的表示法也不惟一。

$$[+0]_{反} = 0000\ 0000 \quad [-0]_{反} = 1111\ 1111$$

### 4. BCD 码

十进制数的二进制编码称为 BCD 码。它的引入是为解决日常习惯的十进制与机器内的二进制之间的矛盾，方便进行十进制数与二进制数之间的转换，最常用的是 8421 码。它对每一位十进制数码用 4 位二进制编码表示。表示规则见表 1-2。

表 1-2 BCD 码编码规则

十进制数	0	1	2	3	4	5	6	7	8	9
BCD 码	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

BCD 码在 0 至 9 的范围与纯二进制无区别，不同之处仅在于逢十进位，而 4 位纯二进制数逢十六进位，因此对大于等于 10 的数需做加 6 修正。

十进制数 1329 用 BCD 码表示为：0001 0011 0010 1001

BCD 码有压缩 BCD 码和非压缩 BCD 码两种形式。压缩 BCD 码的每个十进制数字按四个二进制位一组存放，一个字节可以存放两位压缩 BCD 码。非压缩 BCD 码的每个十进制数字占每个字节的低 4 位，高 4 位内容是什么不重要，每个字节存放一位 BCD 码。

【例 1-6】写出十进制数 31 的压缩 BCD 码和非压缩 BCD 码。

31 的压缩 BCD 码为：0011 0001

31 的非压缩 BCD 码为：0000 0011 0000 0001

### 5. ASCII 码

ASCII 码是国际上比较通用的字符二进制编码，它的全称为美国信息交换标准码 (American Standard Code for Information Interchange)。ASCII 码是 7 位二进制编码，表 1-3 列出了 ASCII 码。

从表 1-3 可看到，它对 94 个常用的一般符号进行了编码，其中包括 26 个英文字母的大小写符号、10 个数字符号和 32 个其他符号。空格也作为一个符号，其编码是 20H，它介于一般符号和 32 个控制符之间。所有这些一般符号和控制符统称为字符。从表 1.3 还可看到，数字符号的编码、大写字母符号的编码和小写字母符号的编码分别是连续的，所以只在记住数字符号的编码从 30H 开始、大写字母符号的编码从 41H 开始和小写字母的编码从 61H 开始，那么就可推出其他数字符号和字母符号的编码。

由于 ASCII 码只使用了 7 位二进制进行编码，故最多表示 128 个字符。这往往不能满足使用要求。为此，在 IBM PC 系列及其兼容机上使用扩展的 ASCII 码。扩展的 ASCII 码使用

8 位二进制进行编码，故可表示 256 个字符。另外，在扩展的 ASCII 码中，控制符所对应的编码同时也表示其他图形符号。

表 1-3 ASCII 表

高位 低位		000	001	010	011	100	101	110	111
		0	1	2	3	4	5	6	7
0000	0	NUL	DEL	SP	0	@	P		p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	“	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	\$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0001	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[	k	{
1100	C	FF	FS		<	L	\	l	
1101	D	CR	GS	-	=	M	]	m	}
1110	E	SO	RS	.	>	N	^	n	~
1111	F	SI	US	/	?	O	-	o	DEL

## 6. 变形国标码

有了 ASCII 码，计算机就能处理数字、英文字母等字符，但不能处理汉字。为使计算机能够处理汉字信息，就必须对汉字进行编码。我国在 1981 年 5 月对六千多个常用汉字制定了交换码的国家标准，即 GB2312-80《信息交换用汉字编码字符集——基本集》。该标准规定了汉字信息交换用的基本汉字符和一般图形字符，共计 7445 个。其中汉字分成两级，共计 6763 个。该标准同时也给定了它们的二进制编码，即国标码。

国标码是 16 位编码，高 8 位表示汉字的区号，低 8 位表示汉字的位号。实际上，为了给汉字符编码，该标准把代码表分成 94 个区，每个区有 94 个位。区号和位号都从 21H 开始。一级汉字安排在 30H 区至 57H 区，二级汉字安排在 58H 区至 77H 区。例如，“啊”字的国标码是 3021H。国标码为汉字的输入提供一种标准输入方式。

目前，计算机中文平台中普遍采用汉字编码是变形国标码。变形国标码是 16 位二进制编码，顾名思义它是国标码的变形。用得最多的变形方法是把国标码的第 15 位和第 7 位均置成 1，以区别 ASCII 码。由于国标码中第 15 位和第 7 位都是 0，所以这种变形方法实际上就是在国标码上加 8080H。

尽管 16 位的变形国标码与两个扩展的 ASCII 码的组合有冲突，但它在相关系统模块的支持下，有效地实现了汉字在计算机内的表示。

### 1.1.3 基本数据类型

计算机存取的以二进制位表示的信息位数一般是 8 的倍数，它们有专门的名称。

#### 1. 字节

一个字节由 8 个二进制位组成。字节的最低位一般称为第 0 位，最高位称为第 7 位。

通常，硬件存储器的每个存储单元由 8 个连续的位组成，可用于存储一个字节的

信息。若用一个字节来表示一个无符号数，那么表示范围是 0~255；如果表示有符号数，则表示范围是-128~+127。一个字节足以表示一个 ASCII 字符，也可以表示一个扩展的 ASCII 字符。

一个字节可分成 2 个 4 位的位组，称为半字节。高 4 位称高半字节，低 4 位称低半字节。

#### 2. 字

两个字节（即 16 个二进制位）组成一个字。字的最低位称为第 0 位，最高位称为第 15 位。字的低 8 位称为低字节，高 8 位称为高字节。由于一个字由 16 个二进制位组成，所以用一个字来表示无符号数，则表示范围是 0~65535；如果表示有符号数，则表示范围是-32768~+32767 间的有符号数。一个字还可表示一个变形国标码。

注意，有时候字是涉及处理器一次能够处理的信息量的一个术语，字长是衡量处理器品质的一个重要指标。

#### 3. 双字

就和听起来一样，双字由两个字组成，也即包含 32 个二进制位。低 16 位称为低字，高 16 位称为高字。双字表示的数的范围更大。

#### 4. 四字

四字就是由四个字组成的，包含 64 个二进制位。如果双字不能表达所需的数值精度，那么四字也许就能解决问题了。

#### 5. 十字节

十字节就和它的名称一样，由 10 个字节组成，含 80 个二进制位。可用于存储非常大的数或表示较多的信息。

#### 6. 字符串

字符串是指由字符构成的一个线性数组。通常每个字符用一个字节表示，但有时每个字符也可用一个字或一个双字来表示。

## 1.2 汇编语言程序设计

### 1.2.1 程序设计语言

#### 1. 指令与程序

指令是指出计算机所要进行的操作和操作对象的一组代码，实际上是计算机能够识别的