



教育科学“十五”国家规划课题研究成果

微型计算机结构与编程

葛幼秋 韩宇龙 武梦龙



高等教育出版社

教育科学“十五”国家规划课题研究成果

微型计算机结构与编程

葛纫秋 韩宇龙 武梦龙

高等教育出版社

内容提要

本书系统介绍了计算机的组织结构与汇编语言程序设计的基本方法。

本书内容主要包括:计算机的基本结构与操作、8086CPU、指令系统、汇编语言程序设计、存储器、I/O 系统、现代计算机系统以及实验。

本书选材与内容组织充分考虑了本科教学的特点与实际需求,力求突出基础性,同时也充分注意适应当前技术的发展。本书可作为高等院校相关专业的本科教材,也可供相关领域工程技术人员学习与参考。

图书在版编目(CIP)数据

微型计算机结构与编程/葛纫秋,韩宇龙,武梦龙. —北京:高等教育出版社,2005.6
ISBN 7-04-017194-5

I. 微... II. ①葛... ②韩... ③武... III. ①微型计算机-结构②微型计算机-程序设计 IV. TP36

中国版本图书馆 CIP 数据核字(2005)第 029811 号

策划编辑 吴陈滨 责任编辑 李 刚 封面设计 于文燕
责任绘图 朱 静 版式设计 王 莹 责任校对 王 超
责任印制 韩 刚

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

经 销 北京蓝色畅想图书发行有限公司
印 刷 北京原创阳光印业有限公司

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>

开 本 787×960 1/16
印 张 30.25
字 数 570 000

版 次 2005年6月第1版
印 次 2005年6月第1次印刷
定 价 38.00 元(含光盘)

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 17194-00

总 序

为了更好地适应当前我国高等教育跨越式发展需要,满足我国高校从精英教育向大众化教育的重大转移阶段中社会对高校应用型人才培养的各类要求,探索和建立我国高等学校应用型人才培养体系,全国高等学校教学研究中心(以下简称“教研中心”)在承担全国教育科学“十五”国家规划课题——“21世纪中国高等教育人才培养体系的创新与实践”研究工作的基础上,组织全国100余所培养应用型人才为主的高等院校,进行其子项目课题——“21世纪中国高等学校应用型人才培养体系的创新与实践”的研究与探索,在高等院校应用型人才培养的教学内容、课程体系研究等方面取得了标志性成果,并在高等教育出版社的支持和配合下,推出了一批适应应用型人才培养需要的立体化教材,冠以“教育科学‘十五’国家规划课题研究成果”。

2002年11月,教研中心在南京工程学院组织召开了“21世纪中国高等学校应用型人才培养体系的创新与实践”课题立项研讨会。会议确定由教研中心组织国家级课题立项,为参加立项研究的高等院校搭建高起点的研究平台,整体设计立项研究计划,明确目标。课题立项采用整体规划、分步实施、滚动立项的方式,分期分批启动立项研究计划。为了确保课题立项目标的实现,组建了“21世纪中国高等学校应用型人才培养体系的创新与实践”课题领导小组(亦为高校应用型人才立体化教材建设领导小组)。会后,教研中心组织了首批课题立项申报,有63所高校申报了近450项课题。2003年1月,在黑龙江工程学院进行了项目评审,经过课题领导小组严格的把关,确定了首批9项子课题的牵头学校、主持学校和参加学校。2003年3月至4月,各子课题相继召开了工作会议,交流了各校教学改革的情况和面临的具体问题,确定了项目分工,并全面开始研究工作。计划先集中力量,用两年时间形成一批有关人才培养模式、培养目标、教学内容和课程体系等理论研究成果报告和在研究报告基础上同步组织建设的反映应用型人才培养特色的立体化系列教材。

与过去立项研究不同的是,“21世纪中国高等学校应用型人才培养体系的创新与实践”课题研究在审视、选择、消化与吸收多年来已有应用型人才培养探索与实践成果基础上,紧密结合经济全球化时代高校应用型人才培养工作的实际需要,努力实践,大胆创新,采取边研究、边探索、边实践的方式,推进高校应用型人才培养工作,突出重点目标,并不断取得标志性的阶段成果。

教材建设作为保证和提高教学质量的重要支柱和基础,作为体现教学内容

和教学方法的知识载体,在当前培养应用型人才中的作用是显而易见的。探索、建设适应新世纪我国高校应用型人才培养体系需要的教材体系已成为当前我国高校教学改革和教材建设工作面临的十分重要的任务。因此,在课题研究过程中,各课题组充分吸收已有的优秀教学改革成果,并和教学实际结合起来,认真讨论和研究教学内容和课程体系的改革,组织一批学术水平较高、教学经验较丰富、实践能力较强的教师,编写出一批以公共基础课和专业、技术基础课为主的有特色、适用性强的教材及相应的教学辅导书、电子教案,以满足高等学校应用型人才培养的需要。

我们相信,随着我国高等教育的发展和高校教学改革的不断深入,特别是随着教育部“高等学校教学质量和教学改革工程”的启动和实施,具有示范性和适应应用型人才培养的精品课程教材必将进一步促进我国高校教学质量的提高。

全国高等学校教学研究中心

2003 年 4 月

前 言

本书是为工科电类专业编写的本科教材,是在编者长期从事电类专业计算机硬件技术课程教学与相关领域科研实践的基础上编写的。在内容选择与组织时,充分考虑了本科教学的特点与实际需求,力求突出基础性、概念清楚、难度适当。为学生以后学习和掌握本领域的新技术,不断更新知识提供较为充实的基础知识。

计算机硬件技术是电类专业的一门重要的专业基础课程。作为电类专业的主要专业基础课程,随着计算机技术的发展,它的内容一直在变化。过去我们习惯于介绍一个典型的 CPU 芯片(8086 或 8051)的结构、指令集、汇编语言程序设计以及外部扩展技术(接口技术)。其指导思想是介绍如何以一个 CPU 为核心,建立一个计算机硬件系统,使其在程序控制下完成指令的功能。课程采用的这种结构模式已越来越不适应目前计算机技术的发展状况。

在电子设备中嵌入计算机,或者以 CPU 为核心构建电子系统已经成为电子技术发展的必然趋势。在电子系统中嵌入的计算机具有灵活、低价、体积小、功耗低等优点,适应这种需求的嵌入式系统设计技术正在成为电子系统设计的一项主要技术。因此,在今天电类专业学习计算机硬件技术更侧重于对其基本结构与工作时序的理解。本书正是以此为基本出发点,更加突出了对硬件系统工作原理的介绍。

本书可以作为硬件技术的入门教材,在编写时更加注意强调基础、结合应用,使学生在掌握这些知识后有能力通过以后的工作与学习进一步扩大知识面和更新知识。

本书共分为 8 章:

第 1 章是基础部分,介绍计算机中数的表示方法、运算规则及运算电路。在此基础上,通过一个最简单的模型机来说明计算机的硬件结构、指令编码、电路实现及工作过程。通过简单而具体的实例说明了计算机的基本工作原理,是深入理解以后各章的基础。

第 2 章以 8086 为典型实例,说明一个实用计算机的体系结构。介绍了 8086 的结构、时序、存储器组织、I/O 组织及系统结构。

第 3 章、第 4 章介绍 8086 指令系统与汇编语言程序设计。了解指令系统是为了进一步理解计算机的工作原理。随着技术的发展,汇编语言程序设计技巧已不再是本课程的重点,但它仍是各种程序设计方法的基础。在这里力求突出

重点,结合实例说明设计思想,尽量避免繁琐的定义与解释。

第5章更加突出存储器的系统配置与存储管理技术。第6章则是对I/O技术的说明。

第7章介绍了现代计算机技术在通用计算机及微控制器两个分支上的发展。

本书由葛纫秋教授主编,负责全书统稿与定稿,并编写了本书的第1章、第2章、第4章和第5章。韩宇龙老师编写了第3章与实验,并对全书内容进行了协调、校对与修订。武梦龙老师编写了第6章、第7章。北京工业大学贾克斌教授仔细审阅书稿并提出宝贵意见,在此深表谢意。

由于编者水平有限,书中难免有疏漏与不当之处,敬请批评指正。

编 者

2004年12月

目 录

第 1 章 计算机的基本结构与操作	1
1.1 数的表示方法	1
1.1.1 数的位置表示法及各种进位制的数	1
1.1.2 各种进位制数的表示	2
1.2 无符号数的机器数表示与机器数运算	4
1.2.1 无符号数的机器数表示	4
1.2.2 无符号数相加与全加器	4
1.2.3 无符号数相减	6
1.2.4 状态标志与条件判断	7
1.3 有符号数的机器数表示与机器数运算	8
1.3.1 原码表示法	8
1.3.2 有符号数的补码	9
1.3.3 补码的运算	10
1.4 计算机中的运算电路	11
1.4.1 运算电路	11
1.4.2 总线结构	12
1.4.3 状态特征标志	13
1.5 计算机中字符与数的表示方法	16
1.5.1 数的定点表示与浮点表示	16
1.5.2 字符的编码	17
1.6 计算机硬件的基本组成	20
1.6.1 总线	21
1.6.2 CPU	21
1.6.3 接口通道与 I/O 设备	21
1.6.4 存储器	22
1.7 最简单的 CPU	24
1.7.1 运算器	24
1.7.2 控制器	25
1.7.3 整机的工作	33
习题一	35
第 2 章 8086 CPU	38
2.1 CPU 的技术发展	38

2.1.1	更多的内部寄存器和高速缓存	38
2.1.2	CPU 内部的多总线	39
2.1.3	指令流水线式处理	39
2.1.4	更大的指令集	39
2.1.5	子程序和中断	40
2.2	8086 的编程结构	40
2.2.1	总线接口单元(BIU)	40
2.2.2	执行单元(EU)	42
2.3	8086 系统结构	44
2.3.1	两种工作方式	44
2.3.2	两种工作方式下相同的引脚定义	44
2.3.3	最小方式引脚定义和系统结构	47
2.3.4	最大方式引脚定义和系统结构	52
2.4	总线时序	56
2.4.1	系统的复位和启动操作	57
2.4.2	总线读/写操作周期时序	58
2.4.3	中断响应周期时序	62
2.4.4	总线请求和总线响应时序	62
2.5	存储器结构和 I/O 端口编址	65
2.5.1	数据存储格式	65
2.5.2	存储器的分段和物理地址的形成	65
2.5.3	I/O 端口编址	67
习题二		68
第 3 章 指令系统		70
3.1	指令编码格式与寻址方式	70
3.1.1	指令编码格式	71
3.1.2	寻址方式	76
3.2	8086 的指令系统	79
3.2.1	数据传送类指令	79
3.2.2	寻址与传送类指令举例	83
3.2.3	算术运算类指令	89
3.2.4	逻辑运算类指令	96
3.2.5	控制转移类指令	99
3.2.6	串操作指令	107
3.2.7	调用与返回指令以及堆栈操作	112
3.2.8	中断调用指令与中断返回指令以及 DOS 系统功能调用	115
3.2.9	控制类指令	116
习题三		118

第 4 章 汇编语言程序设计	120
4.1 概述	120
4.1.1 机器语言、汇编语言与高级语言	120
4.1.2 汇编语言程序的开发	121
4.2 汇编语言程序格式	123
4.2.1 汇编语言程序的实例	124
4.2.2 宏汇编所产生的列表文件	127
4.2.3 连接后所产生的可执行文件	128
4.3 宏汇编的基本语法	130
4.3.1 语句	130
4.3.2 常数	133
4.3.3 段定义	134
4.3.4 ASSUME 语句	135
4.3.5 过程定义与结束语句	136
4.3.6 变量定义语句	137
4.3.7 属性操作符	139
4.3.8 宏指令	144
4.4 汇编语言程序设计的基本方法	145
4.4.1 汇编语言程序设计的基本步骤	145
4.4.2 程序结构	147
4.4.3 DOS 系统功能调用	148
4.5 顺序程序	153
4.6 分支程序	156
4.7 循环程序	160
4.7.1 计数控制的循环程序	161
4.7.2 条件控制的循环程序	163
4.7.3 用逻辑尺方法控制的循环程序	165
4.7.4 多重循环	166
4.8 子程序	169
4.8.1 子程序的使用	169
4.8.2 参数传递	170
4.8.3 子程序嵌套与递归	174
4.9 模块化程序结构	177
4.9.1 模块的定义	178
4.9.2 模块间的交叉访问	178
4.9.3 多模块间的段连接	181
习题四	183
第 5 章 存储器	186

5.1 概述	186
5.1.1 存储器的分类及性能指标	186
5.1.2 主存储器	188
5.1.3 内存条	189
5.2 存储器芯片	191
5.2.1 数据与地址	192
5.2.2 RAM 和 ROM	193
5.2.3 RAM 的基本存储电路	195
5.2.4 ROM 的基本存储电路	197
5.2.5 芯片的内部组成	200
5.3 存储器子系统配置	202
5.3.1 位扩展	202
5.3.2 地址扩展	203
5.3.3 控制信号的连接	204
5.3.4 多字节数据	204
5.3.5 基本功能的扩展	206
5.3.6 冯·诺依曼结构和哈佛结构	206
5.4 8086 系统中存储器的组织	207
5.4.1 存储器的分体结构	207
5.4.2 存储系统连接实例	208
5.4.3 8086 的存储器管理	211
5.4.4 PC 内存空间的分配	214
5.5 高速缓冲存储器	217
5.5.1 存储器的层次结构	217
5.5.2 程序访问的局部性原理与 cache 的作用	219
5.5.3 cache 的种类	220
5.6 虚拟存储器	222
5.6.1 虚拟存储技术	222
5.6.2 虚拟存储器的实现	223
5.6.3 Pentium/Windows 个人计算机的内存管理	231
5.7 辅助存储器	232
5.7.1 磁盘	232
5.7.2 软盘	235
5.7.3 硬盘	235
5.7.4 光盘	236
习题五	237
第 6 章 I/O 系统	238
6.1 I/O 系统的组织	238

6.1.1 I/O 信息的组成	238
6.1.2 I/O 接口的作用	239
6.1.3 I/O 端口寻址方式	240
6.2 I/O 传送方式	241
6.2.1 程序控制的 I/O 方式	241
6.2.2 中断控制的 I/O 方式	245
6.2.3 直接存储器存取传送方式	246
6.2.4 可编程并行接口芯片 8255A	246
6.2.5 键盘及其接口技术	254
6.2.6 发光二极管显示及其接口	258
6.3 中断技术	263
6.3.1 中断概述	263
6.3.2 中断源	264
6.3.3 中断分类	265
6.3.4 中断处理	265
6.3.5 中断服务子程序	266
6.3.6 中断优先权	267
6.4 80X86/Pentium 中断系统	269
6.4.1 中断结构	269
6.4.2 中断向量表	271
6.4.3 中断响应过程	273
6.4.4 IBM PC/XT 中断分配	275
6.4.5 高档微处理器中断系统	277
6.5 可编程中断控制器	278
6.5.1 8259A 的内部结构和引脚	278
6.5.2 8259A 的中断控制过程	281
6.5.3 8259A 的中断触发方式	282
6.5.4 8259A 的状态设定	282
6.5.5 8259A 应用举例	286
6.6 中断程序设计	288
6.7 DMA 控制器	293
6.7.1 DMA 概述	293
6.7.2 8237A 控制器	295
6.7.3 8237A 的初始化编程	301
习题六	302
第 7 章 现代计算机系统	304
7.1 微处理器的逻辑结构	307
7.1.1 80486 的寄存器组织	307

7.1.2 RISC 介绍	311
7.2 虚拟存储器的实现机制	313
7.2.1 虚拟地址、物理地址和线性地址	314
7.2.2 存储器的分段管理	315
7.2.3 存储器的分页管理	322
7.3 8086 微处理器家族其他成员指令介绍	326
7.3.1 80286 新指令	326
7.3.2 80386 新指令	328
7.4 存储器管理技术	330
7.4.1 扩充内存和扩展内存的概念	331
7.4.2 扩充内存的实现	331
7.4.3 扩展内存的实现	333
7.4.4 Windows 95 的内存管理	335
7.5 MCS-51 单片机	335
7.5.1 内部结构	335
7.5.2 存储器	336
7.5.3 端口	340
7.5.4 专用寄存器	344
7.6 MCS-51 的内部定时器与串行接口	346
7.6.1 定时器/计数器	346
7.6.2 串行接口	351
7.7 MCS-51 的中断系统	355
7.8 凌阳 16 位单片机	360
7.8.1 技术性能与结构	360
7.8.2 主要特点	361
7.8.3 指令集	362
7.9 C8051 SOC 单片机	363
7.9.1 CIP-51 内核	364
7.9.2 存储器	365
7.9.3 JTAG 调试和边界扫描	365
7.9.4 I/O 系统	366
7.9.5 模拟数字转换系统	367
7.10 Nios 嵌入式处理器	368
7.10.1 Nios CPU 结构	369
7.10.2 Avalon 交换结构总线	370
7.10.3 外围设备	372
7.10.4 片内调试模块	373
7.10.5 Nios 开发环境	373

实验	379
实验一 软件基础(1)	379
实验二 指令系统(1)	385
实验三 指令系统(2)	389
实验四 软件基础(2)	394
实验五 汇编语言程序设计(1)	399
实验六 汇编语言程序设计(2)	403
实验七 汇编语言程序设计(3)	405
实验八 LED 接口	409
实验九 键盘接口	413
实验十 中断处理	418
实验十一 定时显示装置设计	420
附录	422
附录 1 80X86 指令系统一览表	422
附录 2 8086/8088 指令系统编码格式	434
附录 3 中断向量地址和 DOS 功能调用	445
附录 4 Microsoft MASM6.11 和 CodeView	451
参考文献	468

第1章 计算机的基本结构与操作

不论一台计算机有多么强大的功能,它终归要通过人们预先设置的指令来工作,而直接由计算机执行的指令其实并不复杂。最基本的指令无非是实现下面的操作:数据传送、两数相加或逻辑运算以及判断(例如判断某数是否为0)。这些操作均可由数字逻辑电路实现。在数字电子技术课程中已经讲述过实现上述操作的基本操作单元电路,例如数字逻辑门、锁存器、存储器等,而像加法器这样的器件也是由一些逻辑门构成的组合逻辑电路。

计算机中的指令实际上是一组代码,它控制电路的工作。计算机对每一组代码做出具体的响应,产生相应的控制信号,控制其各个组成单元按预定的顺序工作,最终实现这条指令应该实现的功能。这些原始指令就成为可供人与计算机交流的语言——机器语言。为了降低电路的成本和简化电路,原始指令应该尽可能简单。

计算机中采用二进制数,并且将符号数表示为补码形式。采用这样的表示方法可以很方便地用数字电路来实现算术运算。

下面介绍数的一般表示方法及计算机中所采用的数的表示方法。

1.1 数的表示方法

1.1.1 数的位置表示法及各种进位制的数

人们习惯用一组约定的符号来表示数。这些符号可以是数符(0~9),也可以是字符(A、B、C、D、E、F等)或其他符号(I、II、III、IV等)。

在表示一个数的一组符号中,每一个符号所代表的量不仅取决于这个符号本身,还取决于它所在的位置,这就是位置表示法。

在位置表示法中,对每一数位的单位量赋以一定的位值,称为位权。每个数位上的数字所表示的量是这个数字和该位位权的乘积。

例如,十进制数33;右边的3代表3,左边的3代表30。也就是说,十进制数的最低位的位权为1,次低位的位权为10。

如果相邻两位中高位的位权与低位的位权之比是常数,则称此常数为基数。例如,十进制数的基数为10。

假定一个数 $A_{n-1}A_{n-2}\cdots A_1A_0A_{-1}\cdots A_{-m}$ 的基数为 X , 则它所代表的数值 N 可表示为

$$N = A_{n-1}X^{n-1} + A_{n-2}X^{n-2} + \cdots + A_0X^0 + A_{-1}X^{-1} + A_{-2}X^{-2} + \cdots + A_{-m}X^{-m}$$

即

$$N = \sum_{i=-m}^{n-1} A_i X^i$$

式中, 从 A_0X^0 起 (包含该项在内), 左边的部分为 N 的整数部分, 而右边的部分则为 N 的小数部分。

整数部分最低位为第 0 位。每向左 1 位, 位数增加 1; 每向右 1 位, 位数减 1 (顺序为 -1 位、-2 位等)。

将整数部分的最低位表示为第 0 位 (而不是第 1 位) 可以使各位的位权与它对应的位数相关, 即第 0 位的位权为 X^0 、第 1 位的位权为 X^1 等。一般而言, 第 i 位的位权为 X^i , 这是计算机中通用的表示方法。

基数 X 可以取不同的值, 对应可得到不同进位制的数。

若 $X=10$, 则可得到十进制的表达式为

$$(N)_{10} = \sum_{i=-m}^{n-1} A_i 10^i$$

式中, 系数 A_i 可为 0~9 的任一个。每个数位的位权则是 10 的对应次幂, 幂的次数为该数位的位数, 即整数最低位的位权为 $10^0=1$, 次低位的位权为 $10^1=10$ 等。例如

$$(1392.67)_{10} = 1 \times 10^3 + 3 \times 10^2 + 9 \times 10^1 + 2 \times 10^0 + 6 \times 10^{-1} + 7 \times 10^{-2}$$

十进制数运算时遵循“逢十进一”、“借一当十”的规则。若最低位为 9, 再加 1 将使该位变为 0, 同时使次低位增加 1, 向上各位间的关系与此相似。

当 $X=2$ 时, 得到二进制的表达式为

$$(N)_2 = \sum_{i=-m}^{n-1} A_i 2^i$$

式中, 系数 A_i 只能取 0 或 1, 每个数位上的位权是 2 的对应次幂。例如

$$(1011.011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$$

二进制数运算时遵循“逢二进一”或“借一当二”的规则。

二进制数的每一位只有 0 和 1 两种可能的取值, 特别适合于用两态器件来存储和运算, 是电子计算机内部采用的计数制。

1.1.2 各种进位制数的表示

不同进位制数可以用下标 2、8、10、16 等区别, 也可以用后缀区别。

表 1.1 所示为四种进位制数的表示法, 其中 B(Binary)、Q(Octal, 用 Q 代替 0 以与数字区别)、H(Hexadecimal) 为后缀, 分别表示其前面的数为二进制数、八进制数或十六进制数。十进制数的后缀 D(Decimal) 通常省略。

十进制数是人们日常生活中习惯使用的计数制。因此,在人机接口中总是使用十进制数。人们直接提供给计算机的数是十进制数,而计算机最终输出给人的数也是十进制数。但是在计算机内部实际使用的数则是二进制数,这是因为二进制数最适合用数字电路进行存储和运算。

由于二进制数书写麻烦,因此通常用十六进制数来代表二进制数。例如,一片内存区的地址是 73FFFH,这个地址用二进制的数则表示为

0111 0011 1111 1111 1111

实际上这个地址是通过 20 根地址线传送的。从高位到低位,地址线上的状态可用上面的 20 位二进制数表示。它的物理含义很清楚:有 3 根地址线上是低电平,其余 17 根地址线上是高电平。但是这种表示过于繁琐,因此通常使用 73FFFH 来表达这个地址。

对于一个熟练的硬件技术人员,当看到 73FFFH,头脑中实际上立刻就会联想到以上的 20 位二进制代码。同样,将 20 位二进制代码转换为等值的十六进制数也不是太困难的事。

表 1.1 十进制、二进制、八进制、十六进制数对照表

十进制数	二进制数	八进制数	十六进制数
0	0000B	00Q	0H
1	0001B	01Q	1H
2	0010B	02Q	2H
3	0011B	03Q	3H
4	0100B	04Q	4H
5	0101B	05Q	5H
6	0110B	06Q	6H
7	0111B	07Q	7H
8	1000B	10Q	8H
9	1001B	11Q	9H
10	1010B	12Q	AH
11	1011B	13Q	BH
12	1100B	14Q	CH
13	1101B	15Q	DH
14	1110B	16Q	EH
15	1111B	17Q	FH
16	10000B	20Q	10H