

微型计算机 体系结构及程序设计

〔美〕 斯坦福大学 JONH F. WAKERLY

清华大学微处理机设计组译

清华大学出版社

微型计算机 体系结构及程序设计

[美] 斯坦福大学JOHN F.WAKERLY著

清华大学微处理机设计组 译

清华大学出版社

内 容 简 介

本书分三部分：一、对计算机硬件、系统软件、算法和数据类型作了一般介绍，扼要地介绍了Pascal语言和数制；二、计算机的基本结构、汇编语言程序设计、寻址方式、操作、子程序和参数、输入输出、中断和过程以及程序开发；三、描述了八种广泛应用的16位新型微处理器——有PDP—11和LSI—11、68000、Z8000、9900、6809、8086以及MCS—48。

本书的主要特点：一是详细阐明了计算机构造的总原则，并对不同计算机的共同特性进行了比较。其中对位置无关码，存储器变换和管理、中断过程和并行处理的叙述是其它书中少见的；二是讨论了结构型汇编语言，把Pascal语言贯穿到有关算法和内部操作的讨论中。并采用16位微处理器作为例子来说明现代软件工程中如何进行程序设计的。有程序设计实例200多个，三是集中描述了七种最新型的16位微处理器，对基本结构、汇编语言、寻址方式和操作作了详细的讨论和比较。每章末尾有指导性参考书目和习题。附录中有ASCII码、扩展Pascal和串行通讯等资料。

本书适用于科技人员和程序设计人员学习微型计算机，也可作为计算系学生和研究生的参考书或教材。

微型计算机体系结构及程序设计

[美]斯坦福大学JOHN F. WAKERLY著

清华大学微处理机设计组 译

★

清华大学出版社出版

北京 清华园

北京丰华印刷厂印刷

新华书店北京发行所发行·各地新华书店经售

★

开本：787×1092 1/16 印张：32 1/4 字数：800千字

1985年1月第一版 1985年1月第一次印刷

印数：000001~50,000

统一书号：15235·132 定价：6.40元

译者的话

本书是 John F. Wakerly 为美国斯坦福大学研究生讲授微计算机课程的教材，全书共分三部分。第一部分为计算机初步，对计算机及程序设计、PASCAL 语言及其数据结构以及数制进行了介绍；第二部分为一般原理，介绍了微型计算机的基本组成、汇编语言程序设计、寻址方式、指令操作、子程序和参数、输入/输出、中断、直接存储器存取及进程以及程序开发等微计算机的一般原理；第三部分为特定微计算机的体系结构，介绍了 DEC PDP-11 和 LSI-11、MOTOROLA 68000、ZILOG Z 8000、TEXAS INSTRUMENTS 9900、MOTOROLA 6809、INTEL 8086 及 MCS-48 系列等的基本组成、汇编语言、寻址方式、指令操作、程序实例、I/O、中断及陷阱。

本书硬件、软件兼备，既系统介绍了微型计算机的一般原理，又对几种先进的16位微处理器进行了全面而又简要的阐述。在讨论每个机种时，都附有程序实例，并以汇编语言及 PASCAL 语言贯穿全书，因此本书具有很强的系统性、先进性及实用性。

本书可以作为科技人员学习微型计算机原理及16位微型机的自学教材或参考书，也可以作为高等院校计算机科学与工程系、电子工程系或自动化系研究生或本科生教科书。

本书由清华大学微电子学研究所微处理机设计组集体翻译。其中岳震伍翻译第一章 计算机和程序设计一般介绍、第九章 子程序与参数、第十五章 ZILOG Z8000；朱家维翻译第五章 基本的计算机组成、第七章 寻址方式；李晔翻译第二章 程序设计语言 PASCAL、第三章 PASCAL 程序的数据结构、第十四章 MOTOROLA 68000；石秉学翻译第六章 汇编语言程序设计、第八章 操作、附录，齐家月翻译第十章 输入/输出、第十一章 中断、直接存储器存取和进程、第十八章 INTEL 8086；羊性滋翻译第四章 数制及其算术运算、第十三章 DEC PDP-11 和 LSI-11；王嵩梅翻译第十二章 程序开发；许丽珠翻译第十六章 TEXAS INSTRUMENTS 9900；郭懋新翻译第十七章 MOTOROLA 6809；郭尚才翻译第十九章 INTEL MCS-48 系列；杨之廉翻译前言、致讲授教师的话。

参加本书全部或大部分章节校对的有朱家维、岳震伍、齐家月、杨之廉。参加个别章节校对的有郭尚才、孙义和。

本书承蒙李志坚和高联佩先生的关心与支持，并对我们的翻译工作提出了宝贵意见，在此一并表示感谢。

还应说明的是，徐葭生和李瑞伟同志大力支持本书的翻译工作，并提供很多帮助，使翻译工作得以及早进行，特向他们表示衷心的感谢。

由于译者水平有限，一定有很多错误及不妥之处，敬请读者批评指正。

清华大学微电子学研究所
微处理机设计组

1983年9月

前 言

由于认识到一台微型计算机就是一台价格低廉的计算机，“微型计算机革命”这话才得到最恰当的理解。这一革命带来的在计算机结构和程序设计方面的变化并不大。微型计算机的主要影响是对懂得计算机原理并知道如何应用这些原理的程序设计人员的需求大大增加了。

本书的目的是用微型计算机为例阐明计算机的基本结构和程序设计。我们着重于基本原理而不是某个微型计算机指令系统的细节。但我们必须用某种特定的计算机结构作例子。因而在阐明基本原理时采用了两个“假想”处理器，它们的指令和特征是两个实际处理器 Motorola 6809 和 Zilog Z 8000 的子集。在讨论了基本原理以后，我们介绍七种当代微处理器的详细情况。

本书分三部分。第一部分包括四章，这些介绍性的内容读者以前可能学过也可能未曾学过。书的核心是第二部分共八章，通过两个“假想”处理器详细描述计算机的结构和汇编语言程序设计的基本原理。第三部分中利用这些基本原理，适当详尽而扼要地描述了七种当代重要的微处理器结构：DEC PDP-11 和 LSI-11、Motorola 68000、Zilog Z8000、Texas 9900、Motorola 6809、Intel 8086、Intel MCS-48。

本书的用途

本书适用于（微）计算机结构和汇编语言程序设计的导论性课程，一般是计算机科学或计算机工程课程中的第三门程序设计课。（头两门课通常包括高级语言程序设计原理，例如 Pascal 或 FORTRAN）。这类导论性课程会用到本书第一部分和第二部分的基本原理，也会涉及第三部分一种微处理器的详细论述。本书讨论了 CS 3 课程所要求的全部题目，并着重于78年 ACM 课程表要求的几种程序设计风格的技巧方面。

第一部分和第二部分是概念的形成，再综合地应用这些概念描述第三部分中的各种微处理器，使本书也适用于“计算机结构”或“微型计算机结构”课程。这种课程是介绍和比较几种不同计算机的结构和特性。

本书的第三种用途是作为学生和计算机专业人员的一本参考书。本书的题目范围是经过很好的编排的，因而用于参考时很方便，另外还有一个综合的索引。

各章的内容

第一到第四章构成第一部分，是书中其它各章要用到的介绍性和补习性的内容。

- 第一章定义术语和讨论基本概念。以使一个认真的学生即使以前或最近没有上过程序设计课也能懂得本书的其它各章。
- 第二章叙述 Pascal 语言，这是微型计算机中最普通的结构型高级语言。书的各章利用 Pascal 编制算法以及定义计算机内部操作，因此需要对简单的 Pascal 程序有一种阅读能力。本章既可作为对 Pascal 的一种复习，也可看作是 Pascal 的一种快速入门。

- 第三章介绍数据结构，包括数组、堆栈和队列；这些数据结构在描述计算机结构和程序设计时将会用到。
- 第四章介绍数制的基本概念和用于典型计算机中的算术操作。

第二部分利用两个“假想”处理器描述适用于各种计算机的基本原理。这两个“假想”处理器的指令和特征是两个实际处理器的子集：“假想”处理器 H 6809 是 Motorola 6809 的子集合，H 8000 是 Zilog Z8000 的子集合。

• 第五章叙述这两种假想处理器的基本指令、结构和汇编语言。为了完整起见还包括一节假想堆栈机。

- 第六章更详细地叙述汇编程序和汇编语言程序设计的基本概念，包括重定位、链接、位置无关码、宏汇编和结构化汇编语言。
- 第七章是各种寻址方式的全面总结，它描述了不同处理器中所有通用的寻址方式。还包括存贮器地址变换和存贮器管理。
- 第八章阐述最通用的计算机操作类型的格式和作用。
- 第九章开始讨论 Pascal 过程和函数，然后谈到汇编语言程序中的一些高级概念如子程序调用及参量传送方法。还包括递归和协同的概念。
- 第十章讨论输入/输出格式和有关的软件结构。
- 第十一章包括中断、陷阱和直接存储器存取(DMA)，还介绍处理进程，共享数据结构和重入的高级概念。
- 第十二章包括经常被忽略的关于软件工程和程序开发过程的基本概念。

第三部分由第十三章到第十九章构成。每一章描述一种具体微型计算机的体系结构，这里利用了第二部分中讨论的基本原理。PDP-11 和 LSI-11 作为第一个体系结构例子是很合适的，因为所有当代的微型计算机都从 PDP-11 中吸取了很多好的思想。其它详细谈到的处理器有 Motorola 68000、Zilog Z8000、Texas 9900、Motorola 6809、Intel 8086 和 Intel MCS-480。第三部分的每一章包括六节，其相应的内容在第二部分已经研究过。

- 基本结构
- 汇编语言
- 寻址方式
- 操作类型
- 程序实例
- 输入/输出、中断和陷阱

第三部分中对各种处理器的描述是相当详细的，所以读者可以对它们作出理智的评价。例如对每个处理器所有的寻址方式和指令都作了叙述，但为了简明起见并不包括汇编语言、指令副作用和开发系统的操作等全部细节。也不包括总线结构、时序、接口安排和其它硬件的细节。如果读者想对第三部分的某一处理器进行编写和运行程序，那么可以用从处理器制造商那儿取得的资料来补充本书内容的不足。

书的最后有三个附录。附录 A 描述 ASCII 字符和 CRT 终端的特殊字符。附录 B 叙述用于位处理的扩展 Pascal，虽然扩展 Pascal 在第四、第五和第八章中用到时已作了解释。附录 C 描述异步串行通讯协议和常用的 CRT 终端接口。

本书是如何编著的

本书的原稿是在我自己的微型计算机系统上用字处理程序准备的。存在微型计算机系统

中的最后底稿用一个程序转换成能直接进入照相排版机的形式，再得到排版校样，这就避免了容易出错的重新排版过程。过去这种方法曾被很多计算机科技书籍的作者在大型计算机上用过，包括两位自己编写字处理程序的 Donald E. Knuth[*TEX and Metafont*, Digital Press, 1979]和 Brian W. Kernighan[*Software Tools*, Addison-Wesley, 1976; *The Elements of Programming Style*, Second Edition, McGraw-Hill, 1978; *The C Programming Language*, Prentice-Hall, 1978]。但是我相信本书是利用廉价微型计算机进行排版的第一本书。某些读者可能对书的制作过程有兴趣，现详述如下。

我的微型计算机系统是 Southwest Technical Products 的 6800 系统，它有 48 K 字节存储器，Smoke Broadcasting BFD—68 三个一组的小型磁盘系统以及 DOS 68 操作系统、ADM 3A CRT 终端、Diablo 1620 菊花瓣打印机和一台旧式的纸带输入/穿孔机。这套硬件系统在 1978 年的价格低于一万美元。

在这一任务中我用了三个最重要的程序：文本编辑程序、文本格式化程序和排字转换程序。两年中我花在跟文本编辑程序“打交道”上的时间比跟其它任何程序或人都多，所以我有幸有了一个面向屏幕的满意的文本编辑程序来进行工作。这个编辑程序称为 6800 PIE，是 Tom Crosley 写的。PIE 三个字母代表 Programme International Editor。Tom 还为 Apple I 计算机写了一个版本称“Apple PIE”。

为写这本书我用编辑程序产生未经格式化的文本和格式化命令的文件，例如“段落开始”、“一节标题的开始”、“改变字体”等等。非格式化文本文件经文本格式化程序处理后得到格式化文本。这个格式化程序的基础是贝尔实验室 UNIX 系统的 NROFF 格式化程序，由 Technical Systems Consultants 为 6800 写的，后来由我加以改进使其能处理书本格式和排字参数。

文本格式化程序是宏驱动的，所以每一格式化命令由一个宏文件定义。我建立了两个宏文件，一个产生格式化底稿，它能由 Diablo 打印机打印出来；另一个产生格式化文本，它可以被我们的排字转换程序进行传送和送入照相排版机。

这样，我就能把已由成百个学生用过的打字原稿仿制成最终的排版文本。由于排版文本是从同一个非格式化文本文件即打印原稿中产生的，所以在原稿中发现的所有错误都可以在非格式化文本文件中加以校正，保证在排版处理时（几乎）不引入新的错误。这就得到了一本非常准确的排版书（当然是和作者的原稿一样准确）。

我的计算机系统硬件包括一台纸带输入/穿孔机。其原因是排版工业的变化相当缓慢，唯一的标准输入手段是用称为 TTS 六位字符编码的纸带。因此我必须写一个把 ASCII 码转换成 TTS 码的程序，并穿成一条六孔纸带以便送入照相制版机。纸带在系统中本来是弱连接；纸带出错的几率大约是每 10 至 20 页出现一个。由于学生们对原稿的校对很充分，所以排版的校样除了纸带错误外是接近完善的。要求一个习惯于在每页中找出一对排字错误的校对员在每 10 至 20 页中寻找一个纸带错误是不实际的。所以我必须对书中遗留的错误表示抱歉，不论它们是纸带的错误或其它错误。

错误

可以说没有一个完全排除了错误的程序，而只有错误未被发现的程序。对于书籍也是同样。我很想知道本书中还有的错误以便在再版和重印时纠正。因此我愿意给找出每个未被发现的错误的第一个人以三美元，这包括技术上的、印刷上的或其它错误。请把你的批评寄给我，地址是 Computer Systems Laboratory, Stanford University, Stanford CA94305。

任一读者都可按照上述地址写信给我索取一份勘误表，但应附上一个写明自己的地址并贴好邮票的信封。我想只需一盎司第一级邮件的邮费就够了。

致谢（略）

John F. Wakerly
Mountain View, California
1980.11.13

致讲授教师的话

本序言的内容是想帮助教师更好地利用书中的材料来安排他们的工作。对于把本书作为参考书的读者只需要查阅适当的章节或索引就可找到他们期望的内容。

从本书目录表中可以看出题目的范围是经过合理地编排的，力求每个题目都有充分的内容。结果本书就成为一本内容全面的参考书，它比起曾作为计算机结构和汇编语言程序设计导论课的其它书的内容要广泛得多。典型的课程内容只有书中第一部分和第二部分的1/2至1/3，以及第三部分中的一个微处理器。

在我开始写这本书时，很多书评家建议我写成记叙式，即按照需要每一题目叙述一段，这样目录表将大致上与一个典型的讲课提纲的次序相同。在我看来这种方法的问题在于要由我来决定应取舍哪些内容和按什么次序编写。但为什么要委托我来作出决定呢？甚至在我自己的课程中我也不是每学期都严格地按照同一提纲进行讲授的。

所以我选择了这样一种编排方法，它让你作为一个教员可以很方便地把题目汇集成一个对于你的学生和时间表来讲是最合适的课程大纲。这要求你比起简单地按照书的次序讲授要事先多做些事。但这有两个好处。第一，你可以在本书中找到你所需要的几乎所有材料而不需要许多其它的补充阅读材料。第二，当课程结束时学生有一本内容广泛的参考书，他们可从书中找到你未曾讲授的内容，包括几种不同的微处理器，他们很可能在将来会遇到这些微处理器。

一个合理的课程大纲

我想告诉你我在Stanford大学的课程计划，希望这会使你的任务稍为容易一点。课程的目的是讲授汇编语言程序设计和基本的计算机结构，着眼点是把学生看作一个程序设计人员，他们在这以前曾学习过一门或二门高级语言程序设计课。大学的一学期是10周，有三十次50分钟的讲课时间。

表1给出我的讲授题目、有关的阅读材料和用在每个题目上的大致课时。我每次讲授时

表 1 “计算机组织和汇编语言程序设计”课大纲

基 本 题 目	阅 读	讲 课
一般介绍及课程概述	笔 记	0.5
Pascal的复习	2.1—2.9	1.0
计算机数据类型	1.6, 4.12	0.5
进位数制	4.1—4.4	1.0
补码数制	4.5—4.6	1.0
计算机基本组织	1.3, 5.1—5.2.5, n.1	1.5
基本汇编语言	5.2.6, n.2	0.5
简单的寻址方式	5.2.9, 7.1—7.2.4, n.3	0.8

基 本 题 目	阅 读	讲 课
简单操作	8.1—8.5,n,4	0.8
程序设计课题 1	笔 记	0.4
汇编程序和装入程序	6.1—6.3	1.0
条件码、分支	8.6	0.5
子程序	5.2,10,8.6	0.5
怎样写程序及如何 使用编程实验室	笔 记, 12.3,12.5	1.0
高级的寻址方式	7.2.4—7.3.4	1.0
寻址方式的应用	3.1—3.4	2.0
杂类操作	8.7—8.8	0.5
程序设计课题 2	笔 记	0.5
子程序和参数	9.1—9.3.6	2.0
输入/输出	10.1—10.3,n,6	1.5
终端	A,C,3 或笔记	0.5
队列	3.4,9.3.7	1.0
中断	11.1—11.3	1.5
程序设计课题 3	笔 记	0.5
中断处理过程	11.5—11.6	2.0

任 选 题 目	阅 读	讲 课
程序开发技术	12.1—12.6	1.0
递归	9.4	1.0
协同	9.5	1.0
重定位及连接	6.4	0.5
宏命令	6.5	0.5
链接表	3.5—3.7	1.0
乘法和除法	4.8—4.9,8.10	1.0
多精度算术运算	8.9	0.5
十进制运算	4.10,8.11	0.5
浮点运算	4.11	1.0
堆栈机器	5.4	1.0
计算机分类	5.5	1.0
不同 CPU 的体系结构	13—19(挑一个)	3.0
DMA, 数据块 I/O 设备	11.4, 笔记	1.0
串行 I/O 协议和设备	C.1—C.3	1.0

都包括表中的第一部分，而在进行到学期结束时按照时间和兴趣选择第二部分中的题目。

PASCAL 语言的使用

用一堂课左右的时间，我处理一些管理上的事务和介绍PASCAL的基本宗旨和一些突出的特征，然后要求不熟悉 PASCAL 的学生浏览一下第二章。虽然我用 PASCAL 阐明计算机的操作并用它作为汇编程序设计举例时的文件编制语言，但学生不必为使用这本书而成为一个 PASCAL 程序员。需要有阅读基本的 PASCAL 程序的能力完全是为了看懂书中 PASCAL 程序的例子。

程序设计课外作业

我的课程对所有的程序举例都使用一个实际的计算机，我要求学生为这个机器写几个程序并在我们的程序设计实验室中运行和调试这些程序。课程可以不管使用什么计算机都使用同一题目来举例和进行程序设计作业。例如，我在不同的场合曾利用 Motorola 6809 和 HP 21

MX 来讲授同样的课程。表 1 中的阅读任务例如“n.1”是对应第三部分一个特定实际计算机的某章（例如，n=17 时指 6809）。

课程大纲要求细致地安排各个题目的讲授，但仍留有足够的时间去作相应的课外程序设计作业（特别在 10 周的学期中）。在 Stanford 大学的课程中一开始有一个熟悉实验室的练习。以后有三个重要的程序设计作业。

第一个作业只要求有基本操作（送入、存储、加、分支）和寻址方式（寄存器、立即、绝对）的知识。我给学生一个二进制——ASCII 码转换问题（练习 5.13）或素数因子问题（练习 8.12）的 PASCAL 算法。利用计算机 ROM 监控程序中的实用程序来完成输入/输出（如果有的话）。学生学习如何使用汇编程序和其它软件工具，学习如何使用处理器的基本特性和第一次尝试如何调试。

第二个作业引入进一步的寻址方式，重要的数据结构和简单的输入/输出程序设计。对于半学年的课程这些题目至少应包括在二个独立的作业中。我发现练习 7.5 和 10.16 对于达到这些目的是很理想的，因为得到的程序很有趣和好玩。除了课本知识外，我给学生一个程序设计，要求用已定义的接口把作业分成几个模块，再让他们对每个模块的内部进行设计和编码。

第三个程序设计课题是在第二个课题基础上，采用中断使 I/O 更为有效。练习 11.17 十分适用，我再次只给学生模块说明，让他们自己设计模块内部并实现编程。

程序设计环境

一般的微处理器开发系统不可能支持一大群学生进行文本编辑、汇编和别的程序开发工作。对于大的班级有两种解决问题的办法，一是提供大批微处理器开发系统。

另一种办法在 Stanford 和其它很多学校中采用，它是利用学校已有的大型计算机来进行文本编辑和交叉汇编。由学校大型计算机的交叉汇编程序得到的目标文件向下回送到开发系统执行和调试。为了简化回送，我们对目标文件用一个可打印的 ASCII 格式，在大型计算机和开发系统之间用一个串行链路。但是也可以用软磁盘或盒式磁带。对于串行链路，既可以采用直接连接也可以用拨号连接。我们在开发系统的 ROM 中有一个小程序，它使大型计算机和开发系统操作台之间建立有效的连接，这样，学生能够向大型计算机申请联机，把他们的目标文件装上。

在大班中利用学校的大型计算机装置有很多优点。每个开发系统不需要大容量存储器而只需要一个 CPU、一个存储器、少量输入/输出板，一个系统终端和另一用于作业的终端或其它简单的输入/输出设备。每个这样的系统只要几千美元。由于大多数硬件都在学校的装置中，因而硬件维护问题在那儿出现和处理。

讲师和助教可以利用学校大型计算机的集中式文件存储和投递装置。例如在 Stanford 我们有一些“库存”源码模块，学生可以把它们编入自己的程序。当应交作业的时候，学生递交一个自己程序的汇编列表用于评分，但他们还把目标文件的一份拷贝投递给助教，这些目标文件可以向下回送（到开发系统）并检查它们的正确性。

最终，利用大型的中央计算机进行程序开发是典型的方法，因为大的微处理器程序开发任务是在工业界进行处理的。通用的微处理器开发系统如 HP6400 和 TEK 8001 设计成支持这种方法。

补充材料

本书并不描述不同微处理器、不同汇编语言和开发系统的所有特性。而在编写工作程序

时这是必须掌握的。因此教师和学生必须根据自己的需要参考相应的技术文献。至少需要四种系统专用的文件：

(1) 处理器参考手册（由处理器制造商出版）。该手册完整地描述处理器的指令和副作用。

(2) 汇编语言参考手册（由处理器制造商或交叉汇编程序提供者出版）。该手册描述汇编语言的语法和如何运行汇编语言程序。

(3) 开发系统参考手册（由开发系统制造商出版）。该手册描述开发系统的操作（例如如何启动）和提供的程序工具（例如内部输入/输出端口、ROM 中实用程序、调试过程和命令）。

(4) 实验室规则和配置指南（由教师准备）。它给学生以管理知识例如签名的规定、实验室开放时间等，还描述实验室设备及其配置（例如 I/O 端口的存储器变换图）。

在 Stanford 我们给学生一本实验室手册，它包括上述四个方面：(1) 一张 6809 袖珍参考卡片 (2) 4 页交叉汇编程序和如何使用的说明书 (3) 8 页一览表包括有用的调试命令和系统监控 ROM 中已有的子程序 (4) 5 页有关实验室设备的叙述。另外，实验室手册包括一个熟悉实验室的练习和一个简短的具体系统中有关编写、运行和调试程序的指南。该指南是我的助教 Peter Van Sickel 在观察到一些经常发生的问题的基础上写成的，在本书第 12.5 节已对它的含义作了很多解释。

对我们如何利用 Motorola 6809 安排“计算机结构和汇编语言”课的详细资料有兴趣的讲师可写信给我(用专用信纸)地址 Computer Systems Laboratory, Stanford University, Stanford, CA 94305。这些资料同样可以修改成适宜于非 6809 的课程。你会收到一张表和课程大纲，程序设计作业的例子，考题和解；第二和第四章全部习题的解，其他各章部分习题的解，实验室手册摘录，一组玩笑话，在你的课堂上可在适当的时候使用它。

目 录

前言

致讲授教师的话

第一部分 计算机初步

第一章 计算机和程序设计的一般介绍

- 1.1 什么是计算机? 1
- 1.2 数字计算机硬件 2
- 1.3 计算机的基本组成 4
- 1.4 计算机系统软件 6
- 1.5 算法 7
- 1.6 计算机的数据类型 8
- 1.7 计算机分类 10

参考文献

第二章 程序设计语言PASCAL

- 2.1 综述 12
- 2.2 词汇集 14
- 2.3 程序结构 16
- 2.4 说明 17
- 2.5 标准数据类型和表达式 19
 - 2.5.1 整数类型 19
 - 2.5.2 实数类型 20
 - 2.5.3 字符类型 20
 - 2.5.4 布尔类型 20
- 2.6 由用户定义的类型 21
 - 2.6.1 枚举型 21
 - 2.6.2 子界类型 22
- 2.7 简单语句 23
 - 2.7.1 赋值语句 23
 - 2.7.2 GOTO语句 23
 - 2.7.3 过程语句 24
- 2.8 结构语句 I 25
 - 2.8.1 复合语句 25
 - 2.8.2 IF语句 26
 - 2.8.3 FOR语句 28
- 2.9 结构语句 II 28
 - 2.9.1 WHILE语句 28
 - 2.9.2 REPEAT语句 30
 - 2.9.3 CASE语句 31

参考文献

习题

第三章 PASCAL程序的数据结构

- 3.1 一维数组 36
- *3.2 多维数组 39
- 3.3 堆栈 40
- 3.4 队列 44
- *3.5 单向链接表 47
- *3.6 双向链接表 50
- *3.7 表的存贮管理 52

参考文献

习题

第四章 数制及其算术运算

- 4.1 进位计数制 36
- 4.2 八进制和十六进制数 57
- 4.3 进位计数制的一般转换 58
- 4.4 非十进制数的加法和减法 59
- 4.5 负数的表示法 60
 - 4.5.1 带符号数值的表示法 60
 - 4.5.2 补码数制 61
 - 4.5.3 基数-补码表示法 61
 - 4.5.4 2的补码表示法 62
 - *4.5.5 基数减1补码表示法 63
 - *4.5.6.1 的补码表示法 63
 - *4.5.7 余 2^{n-1} 码表示法 64
- 4.6 2的补码的加法和减法 64
 - 4.6.1 加法规则 64
 - 4.6.2 图示法 65
 - 4.6.3 溢出 65
 - 4.6.4 减法规则 65
 - 4.6.5 2的补码和无符号二 进制数 67
- *4.7 1的补码的加法和减法 67
- *4.8 二进制乘法 69

*4.8.1 不带符号的乘法	69	*4.11.2基本的浮点表示法	77
*4.8.2 带符号的乘法	70	*4.11.3典型计算机中的浮点	
*4.9 二进制除法	71	表示法	78
*4.9.1 不带符号的除法	71	*4.11.4浮点运算	79
*4.9.2 带符号的除法	75	4.12字符码	80
*4.10二进制编码的十进制 表示法	75	参考文献	
*4.11定点和浮点表示法	76	习题	
*4.11.1定点表示法	76		

第二部分 一般原理

第五章 基本的计算机组成

5.1存贮器	83	*5.5 其它处理器的组成及分类	110
5.1.1 存贮器组成	83	参考文献	111
5.1.2处理器与存贮器的Pascal模拟	84	习题	112
5.1.3 存贮器的种类	85	第六章 汇编语言程序设计	
5.2 累加器型处理器	85	6.1 汇编语言	116
5.2.1 单累加器型处理器的组成	85	6.1.1 汇编语言格式	117
5.2.2 基本指令周期	86	6.1.2 伪操作 和PLC	118
5.2.3 机器指令	87	6.1.3 表达式	121
5.2.4 指令组	87	6.1.4 汇编时间、装入时间和运行时间的操作	121
5.2.5 机器语言程序	89	6.2 两次扫描的汇编程序	122
5.2.6 汇编语言	90	6.3 目标模块和装入程序	126
5.2.7 一个简单程序的运行	90	*6.4 浮动汇编程序及装载程序	127
5.2.8 执行指令的Pascal模拟	92	*6.4.1 浮动	127
5.2.9 间接寻址	93	*6.4.2 连接	129
5.2.1 子程序	95	*6.5 宏命令	130
5.3 通用寄存器型处理器	98	*6.6 结构化汇编语言	133
5.3.1 通用寄存器型处理器的组成	98	参考文献	135
5.3.2 指令格式	99	习题	136
5.3.3 寻址方式	160	第七章 寻址方式	
5.3.4 指令系统	100	7.1 一般概念	138
5.3.5 程序实例	101	7.2 单参量寻址方式	140
*5.4 堆栈型机器	103	7.2.1 寄存器寻址	140
*5.4.1 堆栈型机器的组成	104	7.2.2 绝对寻址	140
*5.4.2 指令系统	105	7.2.3 立即寻址	141
*5.4.3 程序实例	107	7.2.4 寄存器间接寻址	142
*5.4.4 堆栈型机器的强处和弱点		7.2.5 自动增量和自动减量	142

注：*号表示高级的课题，初次阅读可以略去。

7.3 多参量寻址方式	145
*7.3.1 页面寻址	145
7.3.2 变址寻址	147
7.3.3 基址寻址	151
*7.3.4 基址变址寻址	153
*7.3.5 相对寻址	154
*7.4 位置无关码	155
*7.5 存贮器的地址变换和管理	156
*7.5.1 存贮器地址变换	156
*7.5.2 存贮器管理	158
参考文献	160
习题	160

第八章 操作

8.1 指令格式	162
8.2 条件位	163
8.3 数据传送	165
8.4 加法和减法	166
8.5 某些单操作数指令	167
8.6 程序控制	168
8.7 逻辑操作	172
8.8 旋转和移位	173
*8.9 多精度加法和减法	177
*8.10 乘法和除法	178
*8.11 十进制运算	179
参考文献	182
习题	182

第九章 子程序和参数

9.1 Pascal中的过程和函数	186
9.1.1 过程	186
9.1.2 函数	189
9.2 高级语言程序中的参数	190
9.2.1 Pascal参数	190
9.2.2 数值参数	191
9.2.3 变量参数	192
*9.2.4 过程和函数的参数	193
*9.2.5 其它高级语言中的参数	193
9.3 汇编语言子程序和参数	194
9.3.1 子程序调用方法	194
9.3.2 子程序参数	195
9.3.3 在寄存器和存贮器单元中传	

递参数	196
9.3.4 参数区域	197
9.3.5 静态和动态分配	198
9.3.6 面向堆栈的参数传递	
方法	198
9.3.7 另一例子: 队列管理子	
程序	201
*9.4 递归	204
*9.4.1 递归过程和函数	204
*9.4.2 递归子程序	205
*9.5 共用程序	207
*9.5.1 一般结构	207
*9.5.2 扩展Pascal共用程序	208
*9.5.3 汇编语言共用程序	209
*9.5.4 共用程序的应用	213
参考文献	213
习题	214

第十章 输入/输出

10.1 I/O的组成	217
10.1.1 总线	217
10.1.2 设备与接口	218
10.1.3 端口	218
10.2 I/O编程	219
10.2.1 独立编址的I/O	21
10.2.2 存贮器编址的I/O	221
10.3 I/O规约	223
10.3.1 输入操作	223
10.3.2 重迭的I/O	225
10.3.3 输出操作	226
10.4 I/O驱动程序	227
10.4.1 终端I/O	227
10.4.2 共用的I/O驱动程序	232
参考文献	233
习题	233

第十一章 中断、直接存贮器存取和进程

11.1 基本的中断系统结构和编程	236
11.1.1 一般考虑	236
11.1.2 一个简单的中断系统	237
11.1.3 一个使用中断的I/O	

程序	238
11.1.4 其他中断线	240
11.1.5 一个中断系统的Pascal 模拟	242
11.2 中断系统和编程的种类	243
11.2.1 中断级别和允许	243
11.2.2 中断优先权	244
11.2.3 中断的鉴别和查询	244
11.2.4 向量中断	244
*11.3 陷阱和软件中断	246
*11.3.1 陷阱	246
*11.3.2 软件中断	246
*11.4 直接存储器存取	247
*11.4.1 动机的形成	247
*11.4.2 DMA通道的编程	248
*11.4.3 存储器映像屏幕	250
11.5 中断进程	250
11.5.1 进程	250
11.5.2 简化的限制	251
11.5.3 并行操作和多重中断进程	252
*11.5.4 实现的细节	252
*11.5.5 醒着和睡着	254
11.5.6 与时序有关的错误和临界区	255
*11.5.7 检测临界区	255
*11.5.8 锁位及信号	256
11.6 共用的数据和指令	257
11.6.1 共用的变量和数据结构	257

11.6.2 共用的指令(可重入程序)	259
11.6.3 汇编语言程序举例	261
参考文献	266
习题	266

第十二章 程序开发

12.1 程序开发的步骤	270
12.2 说明和设计	270
12.2.1 程序结构	271
12.2.2 详细的模块设计	271
12.2.3 设计的逐步完善	272
12.2.4 数据结构	273
12.3 编制文档	274
12.3.1 前言	274
12.3.2 程序模块前言	274
*12.3.3 数据模块前言	275
*12.3.4 全程数据结构的所 有权	276
12.3.5 文档的编制	277
12.4 编码	277
12.4.1 编码规则	277
*12.4.2 编码技巧	278
12.5 测试和调试	281
12.5.1 开发的方法	281
12.5.2 测试	282
12.5.3 调试	282
12.6 维护	284
参考文献	285
习题	285

第三部分 特定微型计算机的体系结构

第十三章 DEC PDP-11和LSI-11

13.1 基本组成	287
13.1.1 计算机结构	287
13.1.2 处理器编程模型	288
13.1.3 指令格式	288
13.2 汇编语言	289
13.3 寻址方式	290
13.3.1 基本寻址方式	290
13.3.2 PC寻址方式	291

13.3.3 寻址方式摘要	291
13.3.4 寻址方式的Pascal 模拟	291
13.3.5 存储器地址变换和管理	293
13.4 操作类型	294
13.4.1 双操作数指令	294
13.4.2 单操作数指令	294
13.4.3 程序控制指令	294
13.4.4 杂类指令	296
13.4.5 扩充指令	297

13.4.6 PDP-11作为一个堆栈型机器	298
13.5 程序实例	299
13.6 输入/输出、中断和陷阱	308
13.6.1 输入/输出	308
13.6.2 中断	310
13.6.3 陷阱	312
参考文献	314
习题	314
第十四章 MOTOROLA 68000	
14.1 基本组成	316
14.1.1 计算机结构	316
14.1.2 处理器编程模型	317
14.1.3 指令格式	318
14.2 汇编语言	318
14.3 寻址方式	320
14.3.1 寄存器直接寻址	320
14.3.2 立即寻址	321
14.3.3 绝对寻址	322
14.3.4 地址寄存器间接寻址	322
14.3.5 自动增量和自动减量寻址	322
14.3.6 基址寻址	322
14.3.7 变址寻址	323
14.3.8 基址变址寻址	323
14.3.9 相对寻址	323
14.3.10 相对变址寻址	324
14.4 操作	324
14.4.1 存贮器访问指令	324
14.4.2 特殊的存贮器访问指令	324
14.4.3 移位和循环指令	327
14.4.4 程序控制指令	327
14.4.5 规则性和一致性	330
14.5 程序实例	332
14.6 输入/输出、中断和陷阱	342
14.6.1 输入/输出	342
14.6.2 中断	344
14.6.3 陷阱	346
参考文献	347
习题	347

第十五章 ZILOG Z8000

15.1 基本组成	349
15.1.1 计算机结构	349
15.1.2 处理器编程模型	350
15.1.3 指令格式	351
15.2 汇编语言	351
15.3 寻址方式	351
15.3.1 Z8001 同 Z8002 的寻址比较	352
15.3.2 寄存器寻址	354
15.3.3 立即寻址	354
15.3.4 绝对寻址	354
15.3.5 寄存器间接寻址	355
15.3.6 变址寻址	355
15.3.7 基址寻址	355
15.3.8 基址变址寻址	356
15.3.9 相对寻址	356
15.4 操作	356
15.4.1 存贮器访问指令	356
15.4.2 特殊的存贮器访问指令	358
15.4.3 寄存器访问指令	358
15.4.4 程序控制指令	359
15.4.5 串操作指令	361
15.5 程序实例	361
15.6 输入/输出、中断和陷阱	372
15.6.1 输入/输出	372
15.6.2 中断	375
15.6.3 陷阱	379
参考文献	379
习题	380

第十六章 TEXAS INSTRUMENTS 9900

16.1 基本组成	381
16.1.1 计算机结构	381
16.1.2 处理器编程模型	381
16.1.3 指令格式	383
16.2 汇编语言	383
16.3 寻址方式	383
16.3.1 寄存器直接寻址	384
16.3.2 立即寻址	384
16.3.3 绝对寻址	384