

 高等学校教材



新一代汇编语言 程序设计

曹加恒 苏光奎 许先斌 吕 慧



高等教育出版社

<http://www.hep.com.cn>

<http://www.hep.edu.cn>

778
高等学校教材

711313-43
C2/b

新一代汇编语言程序设计

曹加恒 苏光奎 许先斌 吕 慧

高等教育出版社

内容提要

本书以新一代 Pentium 系列微处理器和 MASM 6.1X 为背景,全面系统地叙述了 16/32 位 PC 汇编语言程序设计方法和最新知识与技术。全书共 12 章,分为三个层次:基本原理方法篇(第一~第六章)介绍了 Pentium 工作方式和扩展技术,16/32 位寄存器,存储寻址机制,指令寻址与指令系统,汇编语法与伪指令,顺序分支和循环程序设计,子程序嵌套与递归及 COM 文件的编制。程序设计应用篇(第七~十章)介绍了表处理的排序检索,加密、解密,磁盘文件管理,时钟与驻留,BIOS 中断调用技术,机器人图形动画,输入/输出接口程序,异步通信,汇编语言与高级语言的混合编程。知识拓展技术篇(第十一~十二章)介绍了分段、分页管理与多任务保护机制,实方式与保护方式切换示例,虚拟 86 方式的使用程序,多媒体汇编 MMX 和 SSE 指令及其编程设计与优化处理等。通过程序示例与软件工具用法示范以及附录列表,说明了如何进行软件开发。各章之后均附有适量的习题。该书体系结构逐层推进,层内循序渐进;融会多年教学研究成果,内容新颖丰富,重点突出学用结合;教学内容易于教师和学生根据实际层次水平按需组合。

本书可作为高等学校计算机或电子信息、自动控制等相关专业的教材;同样也可供相关专业继续教育、微机培训的学生使用;还可供科研和软件开发人员及一切想掌握微机汇编知识的人员学习或作为自学参考书。

图书在版编目(CIP)数据

新一代汇编语言程序设计/曹加恒等编. —北京:高等教育出版社, 2003.2

ISBN 7-04-011557-3

I. 新… II. 曹… III. 汇编语言—程序设计
IV. TP313

中国版本图书馆 CIP 数据核字(2002)第 100150 号

出版发行	高等教育出版社	购书热线	010-64054588
社 址	北京市东城区沙滩后街 55 号	免费咨询	800-810-0598
邮政编码	100009	网 址	http://www.hep.edu.cn
传 真	010-64014048		http://www.hep.com.cn
经 销	新华书店北京发行所		
排 版	高等教育出版社照排中心		
印 刷	北京人卫印刷厂		
开 本	787×1092 1/16	版 次	2003 年 2 月第 1 版
印 张	24	印 次	2003 年 2 月第 1 次印刷
字 数	500 000	定 价	26.00 元

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

前 言

与时俱进,开拓创新,编著出版《新一代汇编语言程序设计》是 21 世纪汇编课程与教学改革的需要,也是读者渴望学习新知识和新技术的迫切需要。

“汇编语言程序设计”是我国高等学校计算机各专业必修的一门主干课程,也是电子信息、自动控制、信息管理等专业及其相关学科应用的一门重要基础课程。对于训练学生掌握程序设计方法与技术,从事自主版权计算机芯片和系统软件的开发,在信息安全、过程控制等多方面的应用,均具有十分重要的作用。

本书编写根据国家教育部所属高等院校计算机软、硬件专业四年制教学大纲的要求,结合目前学科发展的实际情况,全面系统地叙述了 32 位 PC 汇编语言程序设计的基本原理和 Pentium 系列微型计算机汇编的最新知识与技术。全书可分为 3 个层次,共 12 章。主要内容有:

- 基本原理方法篇:第一~六章

第一章介绍了与新一代汇编语言有关的 Pentium 系列处理器工作方式和扩展技术,数据表示与类型,汇编语言概念与特征,源程序示例;第二章介绍了 32 位 PC 汇编可编程寄存器体系,实方式存储器寻址机制,堆栈存储技术,汇编连接与调试;第三章介绍了实方式指令寻址与指令系统,字符设备 I/O 功能调用;第四章介绍了高级汇编语言的语法,包括表达式、伪指令、宏功能、模块连接及记录结构;第五章介绍顺序程序和循环程序、算术运算与代码转换的程序设计;第六章介绍子程序概念、设计方法,嵌套与递归子程序以及 COM 文件编制。

- 程序设计应用篇:第七~十章

第七章介绍实用程序设计,包括:线性表的顺序、二分查找程序,排序方法与冒泡排序,串操作指令与加密、解密程序;第八章介绍 MS-DOS 高级调用设计,包括:扩展磁盘文件管理程序,时钟程序,驻留程序;第九章介绍 BIOS 中断调用技术,包括:键盘、显示器和打印机 I/O 的 BIOS 中断调用,写彩色字符和图形的显示控制程序;第十章介绍输入/输出程序设计,包括:I/O 指令寻址与数据传送,I/O 接口程序,发声与异步通信程序设计,汇编语言与高级语言的混合编程。

- 知识拓展技术篇:第十一~十二章

第十一章介绍保护虚拟机制及其程序设计,包括:保护方式,分段、分页管理与多任务保护机制,实方式与保护方式可执行的指令以及特权指令的特殊指令集,实方式与保护方式的切换,32 位与 16 位代码切换的示例,虚拟 86 方式、敏感指令及 V86 方式的使用;第十二章介绍多媒体汇编指令及其编程,包括:MMX 指令,MMX 程序设计与优化,SIMD 浮点指令寄

寄存器与浮点指令、整数指令、高速缓存优化处理指令。附录中还给出了指令与中断调用列表。

通过许多上机程序实例,说明了如何利用汇编语言从事软件开发。各章之后均附有适量的习题,希望读者掌握其中的内容并结合实际上机实习。

我们早在 1984 年就编写了《PC 汇编语言程序设计》教材,1988 年和 1996 年先后编著出版了《IBM PC 汇编语言程序设计》与《80386/80486 高级汇编设计与技术》两本书;主持了教育部《汇编语言网络课程建设》等项目;研制出版了 PC 单机版的《汇编语言多媒体课件系统》,正在完善《新一代汇编语言课程》的网络版;获得了“面向 21 世纪汇编课程教学改革”的创新研究与实践”的省部级优秀教学成果奖。这些成果是我们编写此书的坚实基础。

该书主要的特色:

1. 具有三层体系结构的独特编写方式

以 Pentium 系列微机为背景,以 MASM6.1X 为工具,按 16 位汇编和 32 位汇编有分有合的方式,逐层推进,层内循序渐进,正如上面三个层次所述。

2. 内容新颖丰富,学用结合

融合多年教学经验、科研和教学法研究的成果,将新一代 Pentium 系列丰富的指令逐层分解,有序分布在各个章节,重点突出,学用结合,实例精选,简练实用。

3. 教学内容按需组合

教学内容模块结构化的编写方法,易于教师和学生根据实际层次的水平,进行有选择的讲授与学习。同时为教师的拓展教学、学生扩大知识面的自主学习留有相当大的发展空间,因而适应面广。内容按需组合的成教/本科/计算机本科/提高涉及的 4 个层次,分别在章节目录中用无*/**/**/**来标出,当然后面的递进层次包含了前面相应层次应该学习的内容。

本书大纲和统稿审定由博士生导师曹加恒承担。全书共 12 章和一个附录:由博士许先斌副教授编写第十到第十二章,吕慧博士编写第七到第九章和附录,苏光奎教授编写第五和第六章,曹加恒教授编写第一到第四章。在编写过程中,我们参考了有关单位的资料和教材,在此对他们表示感谢!

本书适宜于本科、高职院校计算机或相关专业的教材,也可供科研和软件开发人员及一切想掌握微机汇编知识的人员学习或自学参考。

由于时间有限,书中难免有遗漏或不妥之处,诚请读者不吝批评指正!

编者

2002.9.18

目 录

基本原理方法篇

第一章 新一代汇编语言程序设计

概述 1

1.1 新一代微机系统简介 1

1.1.1 计算机系统概述 1

1.1.2* Pentium 处理器特性 3

1.1.3* Pentium 工作方式 5

1.1.4* Pentium 扩展技术 8

1.2 数据表示与类型 9

1.2.1 数值数据表示 9

1.2.2 BCD 码数据表示 11

1.2.3 非数值数据表示 11

1.2.4 基本数据类型 13

1.3 汇编语言基本概念 14

1.3.1 汇编语言相关特征 14

1.3.2 汇编语句行基本知识 17

1.3.3 汇编语言源程序示例 18

习题一 21

第二章 32 位 PC 汇编程序设计环境 22

2.1 32 位可编程寄存器体系 22

2.1.1 通用寄存器 22

2.1.2 基本控制寄存器 24

2.1.3* 32 位的相关寄存器 26

2.2 实方式存储器寻址机制 27

2.2.1 存储单元的地址和内容 27

2.2.2 存储器分段寻址 29

2.2.3 段的分配方式 30

2.3 堆栈存储技术 32

2.3.1 堆栈概念 32

2.3.2 堆栈操作原则 32

2.3.3 堆栈操作示例 33

2.4 程序汇编连接与调试方法 34

2.4.1 程序编辑 34

2.4.2 汇编与连接执行 36

2.4.3 程序跟踪调试方法 41

习题二 45

第三章 实方式指令寻址与指令系统 47

3.1 指令的基本寻址方式 47

3.1.1 指令的基本格式 47

3.1.2 数据寻址方式 48

3.1.3 数据寻址与数据结构的关系 53

3.1.4 程序转移寻址 54

3.2* 实方式 32 位指令寻址 57

3.2.1 数据与地址类型 57

3.2.2 32 位的指令寻址方式 57

3.2.3 实地址 32 位指令寻址 58

3.3 实方式指令系统 60

3.3.1 常用指令类型集 60

3.3.2 数据传送类指令 61

3.3.3 逻辑和移位指令 65

3.4 字符设备 I/O 功能调用 69

3.4.1 DOS 子程序调用方式 69

3.4.2 单字符输入/输出调用 70

3.4.3 多字符输入显示输出 71

习题三 72

第四章 高级汇编语言的语法 74

4.1 汇编语句的表达式 74

4.1.1 汇编语言语句格式 74

4.1.2 数值表达式	75	5.3.2 分支程序的结构形式	129
4.1.3 标号变量与地址表达式	77	5.3.3 分支程序设计方法	130
4.1.4 特殊运算符	78	5.4 循环程序设计	139
4.2 汇编伪指令语句	82	5.4.1 循环程序的结构	139
4.2.1 方式伪指令	82	5.4.2 循环控制方法	139
4.2.2 数据与符号定义伪指令	83	5.4.3 单重循环程序设计	142
4.2.3 段定义与段管理伪指令	87	5.4.4 多重循环程序设计	150
4.3 宏功能语句	90	5.5 算术运算与代码转换程序 设计	152
4.3.1 宏指令和宏库	90	5.5.1 算术运算程序设计	152
4.3.2* 重复块语句	96	5.5.2 代码转换程序设计	155
4.3.3* 条件汇编与列表控制	98	习题五	160
4.4 模块连接及记录结构	100	第六章 子程序设计技术	162
4.4.1* 程序模块间的符号通信	100	6.1 子程序的概念	162
4.4.2** 多模块程序文件的连接 操作	102	6.2 子程序的一般设计方法	163
4.4.3** 子程序库的利用及管理	103	6.2.1 子程序的调用和返回指令	163
4.4.4*** 记录与结构	104	6.2.2 过程定义	166
习题四	107	6.2.3 现场保护和恢复	168
第五章 程序设计的基本方法	110	6.2.4 子程序参数传递方法	169
5.1 程序设计的基本步骤	110	6.3 嵌套与递归子程序	183
5.2 顺序程序设计	111	6.3.1* 子程序的嵌套	183
5.2.1 算术运算指令	111	6.3.2** 递归子程序	187
5.2.2 处理机控制指令	120	6.4** COM 文件的编制	191
5.2.3 顺序程序设计方法	122	习题六	193
5.3 分支程序设计	126		
5.3.1 转移指令	126		

程序设计应用篇

第七章 实用程序设计	196	7.3 串操作指令与加密解密程序 ...	207
7.1* 线性表的检索程序	196	7.3.1 串操作指令的共性	207
7.1.1 线性表检索方法	196	7.3.2 串操作指令与查表转换指令	207
7.1.2 顺序检索程序	197	7.3.3 加密解密程序	210
7.1.3 二分查找程序	200	习题七	214
7.2* 排序程序	203	第八章 DOS 高级调用程序设计	215
7.2.1 排序方法	203	8.1* 磁盘文件管理程序	215
7.2.2 冒泡排序	204	8.1.1 扩展磁盘文件管理有关概念	215

8.1.2 常用扩展磁盘文件管理系统 功能调用	216	第十章 输入/输出程序设计	245
8.1.3 扩展文件管理程序示例	219	10.1 I/O 指令寻址与数据传送	245
8.2** 时钟程序	222	10.1.1 I/O 指令端口寻址	246
8.2.1 取/置日期与时间功能调用	222	10.1.2 直接程序控制输入/输出 方式	247
8.2.2 时钟程序	223	10.1.3* 程序查询输入/输出方式	249
8.3 驻留程序	224	10.1.4** 中断输入/输出方式	253
8.3.1 程序驻留的功能调用	224	10.1.5*** 数据通道输入/输出 方式	259
8.3.2 驻留程序的编制	225	10.2* I/O 接口程序设计	262
习题八	227	10.2.1 键盘接口程序设计	262
第九章 BIOS 中断调用程序设计	229	10.2.2 打印机接口程序设计	264
9.1 中断	229	10.3*** 发声与异步通信程序 设计	266
9.1.1 中断和中断源	229	10.3.1 发声系统程序设计	266
9.1.2 中断类型码和中断向量表	229	10.3.2 异步通信程序设计	269
9.2 BIOS 中断调用	230	10.4** 汇编语言与高级语言的 混合编程	279
9.2.1** 显示器 I/O 中断调用	230	10.4.1 嵌入汇编的方法	280
9.2.2* 键盘 I/O 中断调用	234	10.4.2 调用汇编的方法	281
9.2.3* 打印机 I/O 中断调用	234	习题十	284
9.3** 显示控制程序	235		
9.3.1 写彩色字符程序	235		
9.3.2 写彩色图形程序	242		
习题九	244		

知识拓展技术篇

第十一章*** 保护虚拟机制及其 程序设计	285	示例	301
11.1 保护程序环境	285	11.3.1 实方式与保护方式切换示例	301
11.1.1 保护方式	285	11.3.2 32 位与 16 位代码切换示例	306
11.1.2 分段管理机制	287	11.4 虚拟 86 程序	312
11.1.3 分页管理机制	290	11.4.1 V86 方式与敏感指令	312
11.1.4 多任务保护机制	292	11.4.2 V86 方式的使用	314
11.2 特殊指令集	294	习题十一	318
11.2.1 实方式可执行的指令	294	第十二章*** 多媒体汇编指令及其 编程	319
11.2.2 保护方式可执行的指令	297	12.1 MMX 指令	319
11.2.3 特权指令	300	12.1.1 MMX 指令需求与数据结构	319
11.3 实方式与保护方式切换		12.1.2 MMX 指令格式与分类传送	321

12.1.3 算术运算、比较与类型转算 指令	324	12.3.4 高速缓存优化处理指令	343
12.1.4 逻辑运算、移位与状态清除 指令	327	习题十二	345
12.2 MMX 程序设计	328	附录	346
12.2.1 使用 MMX 指令的注意事项 ...	328	附录一 指令表	346
12.2.2 MMX 程序示例	329	附录二 伪指令表	351
12.3 SSE 指令介绍	331	附录三 MSDOS 与 BIOS 调用表	354
12.3.1 SIMD 浮点指令寄存器	332	附录四 DEBUG 命令表	365
12.3.2 SIMD 浮点指令	333	附录五 汇编期间错误信息表	366
12.3.3 SIMD 整数指令	341	附录六 中断类型码分配表	370
		参考文献	372

基本原理方法篇

第一章 新一代汇编语言程序设计概述

目前以 Pentium 系列为主体的计算机广泛用于各个行业。为了充分挖掘这种 32 位 PC 机潜在的汇编效率,应用、开发自主产权的软件和硬件,本书从新一代汇编语言角度叙述程序设计的原理、方法与技术。本章介绍计算机系统概念,CPU 工作方式,数据处理类型,汇编语言的有关特性和模块程序示例。

1.1 新一代微机系统简介

新一代汇编语言程序设计是指在新一代微处理器 Pentium 系列微机上进行的汇编语言程序设计。为此,应熟悉和理解计算机系统软件和硬件的一些基本知识;了解 Pentium 体系的逻辑结构、四种操作方式和 Pentium 的扩展技术。

1.1.1 计算机系统概述

计算机系统包括硬件和软件两大部分。硬件(Hardware)包括各种功能部件及其相关的外部设备。软件(Software)则是为运行、管理、维护与应用计算机而编制的各种程序和文档。

1. 计算机组成结构

计算机的基本组成结构一般由中央处理器 CPU、存储器和输入/输出(I/O)子系统组成,它们由系统总线连接在一起,如图 1.1 所示。

CPU(Central Processing Unit)由运算器和控制器组成。运算器包括具有高速存储性能的寄存器和算术逻辑单元 ALU,执行所有的算术和逻辑运算指令;控制器负责从存储器中取出指令,经译码分析后向全机发出有关的控制命令,保证程序功能的正确完成。

存储器(Memory)是计算机的记忆部件,它可以存放程序、数据、信息及中间结果。这种在机器内部的半导体存储器称为内存或主存,且具有随机读出或写入的特点,因此又称为随机存取存储器(RAM)。而另一种只能读出信息的存储器,称为只读存储器(ROM)。存储器所具有的基本单位是字节的存储单元的个数,称为存储容量。一个字节 B(Byte)由 8 个二进制位 b(bit)组成,因此,存储容量常用的单位是:B 表示字节,KB 表示千字节,MB 表示兆字节,GB 表示吉字节,TB 表示太字节。 $1\text{ KB} = 2^{10}\text{ B} = 1\,024\text{ B}$, $1\text{ MB} = 2^{20}\text{ B}$, $1\text{ GB} = 2^{30}\text{ B}$,

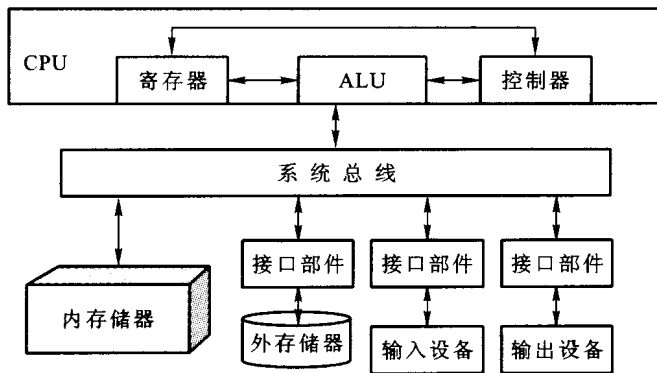


图 1.1 PC 系列机的基本组成结构

1 TB = 2^{40} B。存储单元的存取是通过存储单元的地址(类似房间号)来进行的。

输入/输出(I/O)子系统通常包括大容量存储器和 I/O 设备。大容量存储器是指如磁盘、磁带、CD-ROM 光盘等能存储大量信息的外部存储器(外存)。除了必要的系统程序,如 DOS 引导程序及直接与 I/O 设备进行数据交换的程序是存放在内存中,一般的软件是存放在外存中,只有运行时,它们才调入内存,由 CPU 控制执行。

I/O 设备是指计算机与外部进行通信的输入/输出设备,如显示器、键盘、打印机等。

系统总线连接 CPU、存储器和 I/O 设备等相关部件,在 CPU 总线控制逻辑的作用下,实现各部件之间的信息传送。

在 Pentium 系列微机中,包含的 PC 16 位机性能是 32 位汇编语言程序设计的基础。

2. 外部设备接口

对外部设备(Peripheral)的管理是汇编语言的重要用途之一。外部设备(外设)与主机(包括 CPU 和存储器)的通信是通过外设的接口电路来实现的。在每个接口电路中,都有一组寄存器。一般说来,这些寄存器有三种不同的类型。

(1) 数据寄存器:存放外设与主机之间传送的数据,起着缓冲器的作用。

(2) 状态寄存器:保存外设或接口电路的状态信息,以便 CPU 测试并了解它们的当前工作状态。例如,每个设备都有忙闲位,用来表示当前是否在工作,是否有空闲接受 CPU 安排的新任务等。

(3) 控制寄存器:暂存 CPU 给外设或接口电路的控制命令,并在适当时候向相关接口或外设发出各种命令。例如 CPU 要启动磁盘工作,必须发出启动命令等。

为了方便主机访问外部设备,外设中的每个寄存器被赋予一个端口(Port)地址,又称端口号,从而组成一个独立于内存空间的 I/O 地址空间。通常,I/O 空间可达 64 K 个端口(外设寄存器)。存取端口地址的内容是通过输入/输出指令(IN/OUT)来完成的。

PC 系列微机提供了 BIOS(基本 I/O 系统)和 MS-DOS(磁盘操作系统)两种类型的系统程序,供用户调用外部设备。这种系统程序是系统提供的功能子程序,通过中断方式(INT)调用需要的子程序,当子程序执行完后,返回到原来的程序继续执行。

3. 计算机软件

软件可分为系统软件和应用软件两大类。

系统软件是用户使用机器时为产生、准备和执行用户程序所必需的一组程序。系统软件的组成一般有操作系统、I/O 驱动程序、文件管理程序、调试程序、文件编辑程序、翻译程序、连接程序、装入程序、系统程序库等。

系统软件的核心为操作系统(OS)。操作系统的主要部分是常驻监督程序,只要一开机它就开始运行,能接受用户命令,并使操作系统执行相应的操作。I/O 驱动程序则对 I/O 设备进行控制和管理,完成 CPU 和 I/O 设备之间的信息传递。

翻译程序 Translator 可以将一种语言书写的源程序翻译成机器语言程序(称为目标程序)。汇编程序、解释程序和编译程序均是翻译程序的几种形式。

连接程序(Linker)用来把要执行的程序与库文件或其他已翻译过的子程序连接在一起,形成机器能执行的程序。装入程序能将这种可执行程序从外存储器直接加载到内存,以便机器正确执行。

调试程序(Debug)是系统提供给用户的能监督和控制用户程序的一种工具程序。它可以装入、修改、显示、逐条或连续执行一个程序,特别适宜于跟踪程序的执行,方便查找程序的逻辑错误。

系统程序库和用户程序库是各种标准程序、子程序及一些文件的集合,它可以被系统程序或用户程序调用。操作系统还允许用户建立程序库,以提高不同类型用户的工作效率。

1.1.2* Pentium 处理器特性

Pentium(奔腾)系列微处理器,是随着人们对图形图像处理、实时视频处理、语音识别、CAD/CAE/CAM、大规模财务分析和大流量客户机/服务器应用等与时俱进的显著需求而不断诞生的。奔腾系列微处理器包括 Pentium、MMX Pentium、Pentium Pro、Pentium II、Pentium III、Pentium 4 等。微型计算机的飞速发展和更新换代,使人们可以进一步拓宽微机应用领域和深化微机应用知识的能力,从而在这个年轻的行业里有所发展和作为。

奔腾系列 32 位 PC,增加了一系列新的功能,采用并融入了许多新的设计方法与技术。但是 Pentium 系列并没有改变 IA-32 结构(Intel Architecture),仍与过去 32 位 Intel 80386 和 80486 系列 CPU 兼容,同时也兼容先前的 8086/8088 和 80286,因而具有良好的软、硬件兼容性。了解 Pentium 的特性是进行新一代汇编程序设计的基础。现从程序设计的角度简单介绍 Pentium 新增、扩充或改进的主要内容,其逻辑结构如图 1.2 所示。

Pentium 支持的数据总线位数达到 64 位;支持的物理地址位数是 32 位;内部寄存器仍

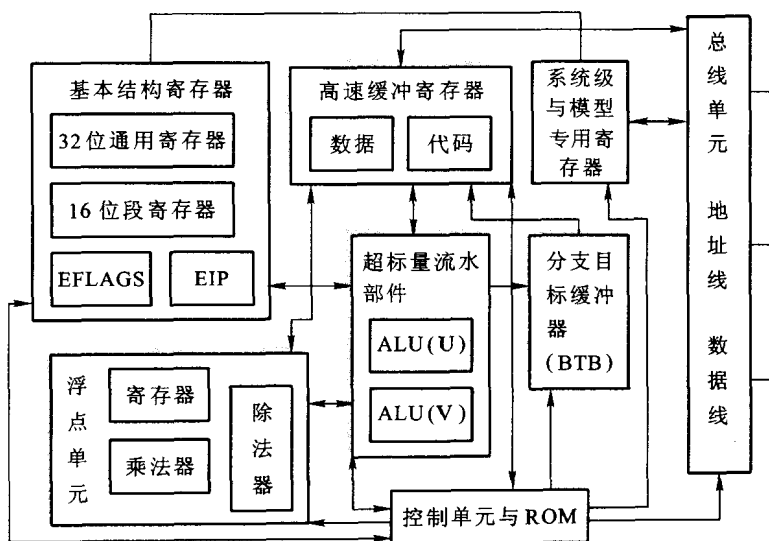


图 1.2 Pentium 逻辑结构示意图

是 32 位。Pentium 的基本结构寄存器有:8 个 32 位的通用寄存器,6 个 16 位的段寄存器,一个 32 位的指令指针 EIP,这些均与 80386、80486 相同;32 位的标志寄存器 EFLAGS 除新增了若干个标志位外,其他标志位的位置及意义保持与 80486 相同。

系统级与模型专用寄存器: Pentium 的系统地址寄存器包括全局描述符表寄存器 GDTR、局部描述符表寄存器 LDTR、中断描述符表寄存器 IDTR 和任务状态段寄存器 TR,它们均与 80386 相同;4 个控制寄存器 CR0、CR2、CR3 和 CR4 与 80486 相比,一是新增了 CR4,二是其中有的位新定义了如高速数据缓存工作方式等相关的内容。新增模型专用寄存器主要用于系统测试和运行性能检测。而控制单元与 ROM 则控制和协调 Pentium 处理器的工作。

Pentium 与 Intel 486 相比有重要改进,其新型体系结构的特点可归纳如下:

1. 超标量流水线

超标量流水线设计是 Pentium 处理器技术的核心。超标量体系结构拥有两条“流水线”,称为“U”流水线和“V”流水线,每条流水线都拥有自己的 ALU、地址生成电路和 Cache 的接口。“U”流水线和“V”流水线都能执行整数指令,“U”流水线还能执行浮点指令,这种流水线结构允许在单个时钟周期内执行两条整数指令,极大地提高了指令并行的执行速度。

2. 片上高速缓冲存储器 Cache

Pentium 有两个独立的高速缓冲存储器 Cache,每个容量是 8 KB,可在片上存储常用的数据和指令。指令 Cache 和数据 Cache 采用 32×8 线宽,是对 Pentium 64 位总线的有力支

持。数据 Cache 有两个接口,分别通向“U”和“V”两条流水线,并采用 Cache 回写技术,因此大大节省了处理时间。

3. 快速运算的浮点单元

Pentium 的浮点单元 FPU(Floating-Point Unit)是对集成在片内 80486 数值协处理器的基础上,重新设计并进行了彻底改进的浮点部件。其执行过程分为 8 级流水,使每个时钟周期能完成一个浮点操作(某些情况下可完成两个)。对于一些常用指令如 ADD(加)、MUL(乘)和 LOAD(取)等采用了新的算法,同时,用电路进行了固化,即用硬件来实现。从而能够更加有效快速地进行浮点数值运算。

4. 动态分支预测

分支预测是指对一个分支是否真正改变程序流而进行的猜测。循环操作在软件设计中使用十分普遍,而每次在循环当中对循环条件的判断占用了大量的 CPU 时间。为此,Pentium 提供了分支目标缓冲器 BTB(Branch Target Buffer)来动态地预测程序分支。当一条指令导致程序分支时,BTB 记忆下这条指令和分支目标的地址,并用这些信息预测这条指令再次产生分支时的路径,预先从此处预取,保证流水线的指令预取步骤不会空置。从而能有效地提高分支处理执行速度。

Pentium 新增了一条 8 字节比较交换指令和一条处理器识别指令以及 4 条系统专用指令。所有这些都大大提高了 Pentium 的整体性能。

为进一步加快 32 位代码的运行,Pentium Pro(高能奔腾)地址总线达到 36 位,可寻址主存 64 GB 容量。除了两个分别为 8 KB 的 L1 级 Cache 外,还集成了一个 256 KB 或 512 KB 的 L2 级高速缓存,L2 级高速缓存能以处理器的全速运行。扩展了超标量技术,具有 3 个整数处理单元,能同时执行 3 条指令。既支持乱序执行,又能支持寄存器重命名和推测执行,并对 32 位指令进行了优化处理。Pentium Pro 又新增了 3 条指令。这些均有利于提高流水线的效率。

1.1.3* Pentium 工作方式

Pentium 系列处理器虽有多种型号,但是它们包含 80X86 CPU 的基本体系结构和基本指令,因而系列机的软件向上兼容。即在低档机型上用汇编语言设计编写的程序,一般不加修改或稍加修改后就可以在高档机器上运行。Pentium 系列的四种工作方式如图 1.3 所示。

32 位 80X86 微处理器全面支持 32 位数据、32 位指令和 32 位寻址。它不仅兼容 8086 的工作方式,而且增强了 80286 的保护方式,新增的虚拟 8086 方式还可以更好地运行多个实地址方式的程序。

1. 实地址方式

实地址方式又称 R 方式(Real Mode),它是在上电或复位后,32 位 CPU 首先初始化为实地址的工作方式。在 R 方式下可以使用 32 位寄存器和 32 位操作数,也可采用 32 位的寻

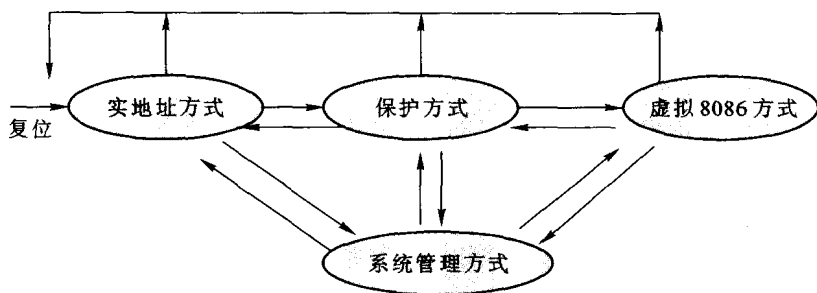


图 1.3 Pentium 系列的工作方式

址方式,执行 32 位非保护虚拟的指令。但是 32 位 CPU 与 16 位 PC 机一样,只能寻址 1 MB 物理存储器空间,程序分段时段的大小最大为 64 KB,段基址和偏移量都用 16 位表示,所以称这样的段为“16 位段”。

事实上对程序员而言,实地址方式的 32 位 CPU 相当于可以进行 32 位处理的高速 8086。原来为 80X86 设计的非保护虚拟程序几乎都可以在实方式下运行。

2. 保护方式

保护方式又称 P 方式(Protected Mode),它是 32 位 CPU 的固有工作方式。由实地址方式可以进入 32 位保护方式,用于支持多任务的需要,程序员可以使用多达 64 TB 的虚拟(逻辑)存储空间。

P 方式不仅具有段式存储管理功能,而且提供了页式存储管理功能,可以更好地支持虚拟存储器。在保护方式下,32 位 CPU 才能发挥其全部功能,可以充分利用其强大的存储管理和保护能力。P 方式下的 32 位 CPU 可使用全部 32 条地址线,使微处理器寻址的物理存储器空间达到 4 GB。它们的段基址和段内偏移量都是 32 位的,这样的段称为“32 位段”。Pentium Pro 以上的 32 位 CPU 还可支持 64 GB 物理存储器。

多任务是通过 32 位 CPU 的多任务硬件机构,使其在各种任务之间来回快速而方便地切换所完成的,即为每个任务提供一台虚拟处理器来仿真多台处理器。也就是说,在任一时刻,操作系统把真实的处理器分配给任一个虚拟处理器,接着由它运行该虚拟处理器的任务。这种工作方式之所以称为保护方式,是因为处理器在处理多任务时,每个程序各自分开,在自己的空间运行。实地址方式下,一个瞎跑的程序会使整个系统崩溃,而在保护方式下,一个程序最多也就是损坏自己。

为此,保护方式提供了一些保护设施,如:① 任务地址空间的分离;② 4 个特权级的建立;③ 特权指令的使用;④ 段和页的访问权限(如:只读、只执行);⑤ 段的上限检查。

建立的 4 个特权级,使系统程序 and 用户程序分离,如图 1.4 所示。由此可以实现多任务及高速的任务切换以及对各个执行任务环境的地址保护能力。在虚地址保护方式下,能将

每个任务的 2^{46} B 的虚拟地址空间映射到 2^{32} B 的物理地址空间,具有 4 GB 的寻址能力。在特权级中,0 级为最高级,3 级为最低级。

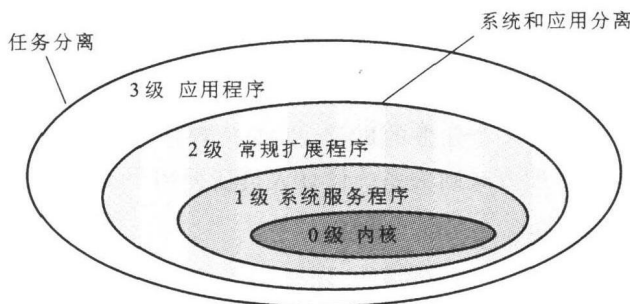


图 1.4 特权保护结构示意图

3. 虚拟 8086 方式

虚拟 8086 方式又称 V86 方式 (Virtual-8086 Mode), 它在保护方式下, 通过设置控制标志, 使 32 位 CPU 可以转入虚拟 8086 工作环境。利用 P 方式提供的硬件支持多任务的优点, 不但可以有多个 V86 任务, 每个任务执行一个 8086 程序, 而且 V86 任务可以与其他 32 位 CPU 任务一起执行多道程序。

V86 工作方式使得多个 DOS 程序能同时运行, 每个程序就像在自己的 8086 机器中运行一样。事实上, 正是由于这种方式, 才使得硬件能够支持 Microsoft Windows 同时运行相互隔开的多个 DOS 程序。

虚拟 8086 方式是一种在保护方式下运行的类似实方式的工作环境。V86 方式下, 段寄存器的使用与 R 方式一样, 左移 4 位加 16 位偏移量得到 20 位地址; 如果采用分页存储管理, 则任务的 1 MB 地址空间可以映射到 4 GB 物理地址的任何位置。这样, 多个 8086 程序就可以利用分页机制将各自的逻辑 1 MB 空间映射到不同的物理地址, 从而实现共存于主存并行运行。

虚拟 8086 方式的程序都是在最低特权级 3 下运行的, 而实方式程序却在最高特权级 0 下运行。因此 V86 程序都要经过 P 方式确定的所有保护性检查。

由此可见, 虚拟 8086 方式是一种更好的运行实方式程序的工作环境, 但是它并不能替代实地址方式。

4. 系统管理方式

系统管理方式又称 SM 方式 (System Manager Mode), 它与保护方式一样, 也是 Pentium 的一种操作方式。当处理器接收到系统管理方式中断时, 就会进入系统管理方式 (如节能状态); 处理器利用系统管理方式返回指令 RSM, 可以恢复到被中断之前的程序中。

SM 方式不是为应用软件访问而设计的, 该方式的特征是由植入机器内部的固件 (装有

系统级程序代码的 ROM) 来控制的。

SM 方式使系统设计人员可以实现十分高级的功能,其中包括电源管理以及对操作系统和正在运行的程序提供透明的安全性等。

综上所述,程序设计者对于 32 位的 CPU,在实地址方式下执行 16 位段的程序;保护方式下可以对 32 位和 16 位的程序都能单独或混合操作;虚拟 8086 方式则虚拟 8086 CPU 进行工作,在保护方式下可执行多个任务的 8086 的 16 位程序,这是传统执行单个任务的 8086 CPU 所不能代替的;系统管理方式则为系统设计人员所使用。

1.1.4** Pentium 扩展技术

为了适应和增强对多媒体数据的处理能力,出现了多媒体扩展技术 MMX (MultiMedia eXtension);为了针对国际互联网和三维多媒体程序的应用要求,产生了数据流 SIMD 扩展技术 SSE(Streaming SIMD Extensions),SIMD 是指单指令多数据技术。

1. MMX 技术简介

多能奔腾 MMX Pentium 是 Intel 将 MMX 技术融入了内部 Cache 为 32 KB 的 Pentium。后来,Intel 又在 Pentium Pro 中继承了 MMX 技术,并结合 Pentium Pro 的动态执行技术,推出了 Pentium II。Pentium II 内部一级 Cache 增至 64 KB,二级 Cache 为 512 KB。

MMX 的重点技术包括 SIMD 技术和新增的 57 条多媒体指令,8 个 64 位宽的 MMX 寄存器和四种新的数据类型。其核心是针对多媒体信息处理中的数据特点,可用新增的 57 条指令对图像、图形、动画、音频、视频和通信方面的程序进行优化。MMX 技术使程序员可设计出更多、更丰富、更令人惊奇的应用程序。

MMX 的数据结构包括的 64 位紧缩数据,可以表示 8 个字节、4 个字(1 字为 2 个字节)、2 个双字或 1 个四字,对于大量使用 8 位、16 位或 32 位数据的多媒体软件,这样一条 MMX 指令就能同时处理 8 个、4 个或 2 个数据单元,这就是所谓的单指令多数据结构 SIMD (Single Instruction Multiple Data)。此种结构是 MMX 技术把机器性能提高的最根本因素。

2. SSE 技术说明

采用 MMX 指令的 Pentium 和 Pentium II 处理器取得了很大的成功之后,Intel 又采用数据流 SIMD 扩展的 SSE 技术(原称为 MMX-2)开发了 Pentium III。Pentium III 在 Pentium II 的基础上又新增了 70 条 SSE 指令,极大地提高了浮点 3D 数据的处理能力。SSE 指令集类似于 AMD 公司发布的 3D Now! 指令集。

SSE 指令集可分为三组:50 条 SIMD 浮点指令、12 条 SIMD 整数指令和 8 条高速缓存优化处理指令。其中 SIMD 浮点指令是 SSE 指令集的主要指令,也是 Pentium III 处理器性能提高的一个关键。SSE 指令集可以在实地址方式、保护方式和虚拟 8086 方式下运行。

由于采用紧缩浮点数据,所以多数 SIMD 浮点指令一次可处理 4 对 32 位单精度浮点数。Pentium III 处理器中增加的 8 个 128 位浮点寄存器,用于与 SSE 浮点指令相配合。一