



普通高等教育“十一五”国家级规划教材

高等学校计算机科学与技术系列教材

汇编语言程序设计及应用(第2版)

王保恒 等 编著



高等教育出版社
Higher Education Press

普通高等教育“十一五”国家级规划教材
高等学校计算机科学与技术系列教材

汇编语言程序设计及应用

Huibian Yuyan Chengxu Sheji ji Yingyong

(第2版)

王保恒 柳 靖 沈 立 王学慧 编著



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书是普通高等教育“十一五”国家级规划教材。全书由基础篇和应用篇组成,全面、系统地阐述汇编语言程序设计及应用。基础篇简单介绍 80x86 系列微机的硬件和四种工作模式,详细介绍 80x86 通用指令集实模式下可用的指令,并以 MASM 6. x 为蓝本,详细讲述汇编语言源程序的结构和伪指令,深入讨论分支、循环和子程序等基本汇编语言程序设计技术和宏指令、条件汇编、重复汇编等高级汇编语言程序技术。应用篇介绍保护模式的有关概念、80x86 微处理器与保护模式有关的硬件及专用的指令,系统说明设计保护模式下的程序所需的数据结构和如何实现保护模式下的程序设计,介绍 80x86 微处理器 FPU 的硬件、浮点指令集及程序设计方法,阐明汇编语言和高级语言混合编程的各种约定规则与实现方法,介绍三个多媒体指令集 MMX、SSE 和 SSE2 及其编程技术,简单介绍了 Win32 环境下的汇编语言程序设计方法。

本书内容丰富,系统全面,涵盖了汇编语言程序设计的各个方面。可作为普通高校本科计算机及相关专业汇编语言程序设计课程教材,也适用于信息类各专业学生、自考学生、计算机应用开发人员和希望深入学习微机应用技术的读者。

图书在版编目(CIP)数据

汇编语言程序设计及应用/王保恒等编著. —2 版
北京:高等教育出版社,2010.4
ISBN 978-7-04-028837-7

I. ①汇… II. ①王… III. ①汇编语言-程序设计-高等学校-教材 IV. ①TP313

中国版本图书馆 CIP 数据核字(2010)第 011568 号

策划编辑	倪文慧	责任编辑	张海波	封面设计	于文燕	责任绘图	尹 莉
版式设计	马敬如	责任校对	胡晓琪	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 100120
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 高等教育出版社印刷厂

购书热线 010-58581118
咨询电话 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landraco.com>
<http://www.landraco.com.cn>
畅想教育 <http://www.widedu.com>

开 本 787×1092 1/16
印 张 33.25
字 数 820 000

版 次 1992 年 8 月第 1 版
2010 年 4 月第 2 版
印 次 2010 年 4 月第 1 次印刷
定 价 42.00 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

物料号 28837-00

前言

汇编语言是面向机器的语言,用汇编语言编写的程序在时间和空间两个方面的效率都最高,也是唯一能够利用计算机所有硬件特性并能对其进行直接控制的语言。对于需要软硬件结合开发的计算机应用系统,如实时控制程序;计算机底层软件的设计,如操作系统内核、高级语言的编译程序、磁盘驱动和磁盘读写程序;计算机中的特殊应用场合,如病毒分析、加解密以及破译软件等程序,汇编语言更有着高级语言无法替代的作用。同时汇编语言又是计算机各专业学生掌握计算机基本知识必备的语言。

本书是普通高等教育“十一五”国家级规划教材,参照教育部计算机科学与技术教学指导委员会制定的《计算机专业规范》对本课程的要求,在国防科技大学计算机系统结构系列教材编委会的指导下,结合多年的教学实践经验和体会,为本科教学的需要编写而成。

汇编语言因计算机不同而异。本书以 80x86 系列微机作为模型机讲授汇编语言程序设计的基础知识、程序设计方法与技巧及其应用技术,因此必然带有该系列微机的浓重色彩。但实践和经验证明,这是无关大局的,因为用汇编语言进行程序设计的基本概念、基本技巧和基本方法是普遍适用的,掌握一种系列或型号计算机的汇编语言程序设计,其他的便可触类旁通。

本书由基础篇(第一章至第五章)、应用篇(第六章至第十章)和三个附录组成。第一章简单介绍 80x86 系列微机的硬件,它是掌握 80x86 系列微机指令系统中的通用指令集和实模式下编程的硬件基础。第二章简要介绍 80x86 四种工作模式的特征,详细介绍 80x86 与数据有关的寻址方式以及数据传送、算术运算、位操作和处理器控制四类简单指令。第三章以微软的 MASM 6.x 为蓝本,详细讲述汇编源程序的结构,常用伪指令的格式、功能、使用方法与使用约定,以及各种操作符的使用方法。第四章讨论顺序、分支、循环、子程序和系统功能调用等程序设计技术与技巧,并介绍与这些设计技术密切相关的程序控制类、串操作类复杂指令性语句和过程定义、模块通信伪指令语句。第五章阐述宏指令、条件汇编、重复汇编,介绍 MASM 6.0 及其后续版本增加的高级语言结构流程控制伪指令的功能及使用方法。第六章介绍保护模式的有关概念、80386 及其后继微处理器与保护模式有关的硬件及专用的指令,系统说明了保护模式程序设计所需的数据结构、程序的结构及程序设计方法。第七章介绍 FPU 的硬件、可直接处理的数据类型,阐述浮点指令集及程序设计方法。第八章说明汇编语言和高级语言混合编程的各种约定规则与实现方法。第九章重点介绍三个多媒体指令集 MMX、SSE 和 SSE2 及其编程技术。第十章简单介绍 Win32 环境下的汇编语言程序设计方法。

本书基础与应用并重,理论与实践结合。前五章是汇编语言程序设计的基础部分,后五章是汇编语言程序设计的提高、应用与拓展。本书具有如下特色。

1. 内容丰富,系统全面,涵盖了汇编语言程序设计的各个方面

系统地介绍 80x86 系列微机指令系统的通用指令集、浮点指令集和多媒体指令集,具体而又详细地阐明了实模式、保护模式和虚拟 86 模式程序结构及程序设计方法,讨论了浮点程序设计、

汇编语言与高级语言混合编程以及多媒体程序设计的约定与方法。

2. 语句说明全面,顺序安排合理

对所有语句,均说明其格式、功能、对操作数的要求、使用方法与使用约定。对绝大多数语句给出了应用示例。鉴于MASM 6.0及其后续版本中语句功能强大而繁多,为使读者学习更有针对性,通用指令集分散到第一章到第六章讲授,常用伪指令在第三章讲授,高级汇编使用的伪指令在第五章、第八章讲授,浮点指令集在第七章讲授,多媒体指令集在第九章讲授。附录A按字母顺序列出并说明所有指令性语句。

3. 内容编排有利于教师教学和读者学习实践

本书力图做到通俗易懂,概念清晰。在章节的安排上,从简到难,循序渐进,基本避免了对未介绍内容的提前引用。全书例题两百多个,思考题和练习题一百六十多个,所有给出完整代码的例题都是精心设计并经过上机验证的,读者通过阅读它们,可以加深对概念的理解以及对基本程序设计技术与常用技巧的掌握。

4. 可剪裁使用,适应面宽

从第五章开始,各章之间既自成体系又互相配合,适应高等学校信息类专业按其要求适当剪裁后作为教材使用。前四章可单独作为专科生汇编语言程序设计的教学内容,前五章和第七章组合可作为本科生汇编语言程序设计的教学内容,前五章、第六章和第八章组合可作为从事系统开发读者的学习资料,前四章和后四章组合可作为从事应用开发读者的学习资料。

本书带星号(*)的章节和习题有一定的难度,带*的语句在汇编语言程序设计的基础部分很少使用。基础篇不带*的章节可作为一般院校计算机专业的教学内容,讲授学时44,上机实习学时10。对于重点院校,可依教学的要求酌定。

本书第一章和第五章由柳靖编写,第二章至第四章和第六章由王保恒编写,第七章至第九章由王学慧编写,第十章和附录B由沈立编写,其他附录由王学慧编写。全书由王保恒负责统编和定稿。中山大学吴向军教授、北京理工大学陈朔鹰教授审阅了全书,并提出了许多具体的修改意见,编者据此尽力进行了修改,特在此表示诚挚的谢意。本书得到国防科技大学计算机系统结构系列教材编委会的教授和专家们的指导和帮助,计算机科学与技术系的领导也给予极大关心和帮助,在此一并致谢。

编者在成书前对书稿进行了多次校正,但由于水平和经验有限,疏忽之处敬请专家、同行及广大读者批评指正。

编 者

2009年10月

目 录

上篇 基 础 篇

第一章 80x86 系列微机结构组成	3
1.1 80x86 系列微处理器结构	3
1.1.1 8086 CPU	7
1.1.2 80386 CPU	8
1.1.3 80486 CPU	9
1.1.4 Pentium 微处理器	9
1.2 80x86 系列微处理器的寄存器	11
1.2.1 8086 寄存器	11
1.2.2 80386 寄存器	16
1.2.3 80486 寄存器	18
1.2.4 Pentium 寄存器	18
1.3 80x86 系列微处理机的主存储器组织	18
1.3.1 主存概述	19
1.3.2 堆栈	20
1.4 80x86 系列微处理器能直接处理的数据及其在主存储器中存放形式	20
小结	22
习题一	22
第二章 80x86 系列微处理器寻址方式和指令系统	23
2.1 80x86 的工作模式	23
2.1.1 实模式	23
2.1.2 保护模式	25
2.1.3 虚拟 86 模式	25
2.1.4 系统管理模式	25
2.2 80x86 指令格式	26
2.2.1 80x86 机器指令格式	26
2.2.2 80x86 机器指令在汇编语言中的表示方法	29
2.3 80x86 指令寻址方式及其符号表示	30
2.3.1 寻址方式的有关问题	30
2.3.2 与数据有关的寻址方式	31
2.4 80x86 通用指令系统	39
2.4.1 传送类指令	41
2.4.2 算术运算和辅助操作指令	51
2.4.3 位操作指令	62
2.4.4 处理器控制指令	74
小结	75
习题二	75
第三章 80x86 的宏汇编语言	79
3.1 概述	79
3.1.1 汇编语言与汇编程序	79
3.1.2 80x86 宏汇编语言概述	82
3.2 汇编语言源程序语句中的域	84
3.2.1 标号名字域	84
3.2.2 操作助记符域	85
3.2.3 操作数域	85
3.3 伪指令	90
3.3.1 指令集选择伪指令	91
3.3.2 数据定义伪指令	92
3.3.3 符号定义伪指令	104
3.3.4 模块定义伪指令	106
3.3.5 完整段定义伪指令	108
3.3.6 简化段定义伪指令	114
3.4 COM 文件的源程序结构	120
小结	122

习题三	122	5.1.1 宏定义	191
第四章 程序设计的基本技术	127	5.1.2 宏调用和宏扩展	193
4.1 顺序程序设计	127	5.1.3 宏体中的标号和变量	195
4.2 分支程序设计	128	5.1.4 宏指令的嵌套	197
4.2.1 与程序控制类指令有关的 寻址方式	129	5.1.5 特殊的宏操作符	199
4.2.2 转移指令	133	5.1.6 宏指令与子程序的区别	202
4.2.3 分支程序设计	134	5.2 条件伪指令	202
4.3 循环程序设计	139	5.2.1 条件伪指令概述	203
4.3.1 概述	139	5.2.2 条件伪指令用法举例	204
4.3.2 循环程序设计举例	147	5.3 重复伪指令	205
4.4 子程序设计	158	5.3.1 数值重复伪指令 REPT	206
4.4.1 子程序概述	158	5.3.2 参数值重复伪指令 IRP	207
4.4.2 过程定义与模块通信及调用 返回语句	159	5.3.3 字符重复伪指令 IRPC	208
4.4.3 子程序的设计方法	164	5.3.4 中止宏的伪指令 EXITM	210
4.4.4 系统功能调用	174	5.4 高级语言结构流程控制伪指令 及其应用	210
4.4.5 子程序设计举例	178	5.4.1 分支结构流程控制伪指令及其 应用	210
*4.4.6 递归子程序	182	5.4.2 循环结构流程控制伪指令及其 应用	216
小结	186	小结	222
习题四	186	习题五	222
第五章 高级汇编语言技术	191		
5.1 宏指令的有关问题	191		

下 篇 应 用 篇

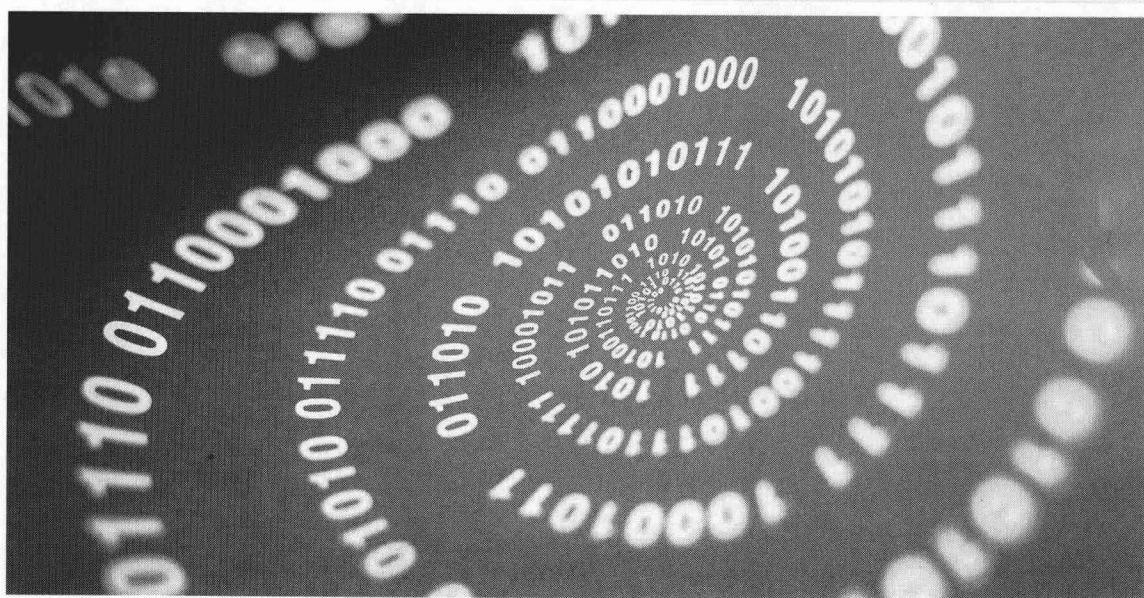
*第六章 保护模式及其程序设计	227	6.3.1 任务间段的保护与共享	250
6.1 保护模式的预备知识	227	6.3.2 虚拟地址到线性地址的 转换	250
6.1.1 保护模式的三种地址空间	227	6.4 保护模式下的控制转移	253
6.1.2 保护模式的段和页	228	6.4.1 概述	253
6.1.3 保护模式涉及的数据结构及 相关硬件	230	6.4.2 任务内特权级不变的段间 转移	255
6.2 操作系统类指令	245	6.4.3 任务内特权级变换的段间 转移	255
6.2.1 实模式和任何特权级下都可 执行的指令	245	6.4.4 任务切换	257
6.2.2 实模式或保护模式 0 特权级下 可执行的指令	246	6.5 保护模式程序设计举例	259
6.2.3 只能在保护模式下执行的 指令	247	6.5.1 预备知识	259
6.3 分段管理机制	250	6.5.2 任务内特权级不变的段间转移 实例	266

6.5.3 任务内特权级变换的段间转移实例	272	小结	377
6.5.4 任务切换实例	279	习题七	378
6.6 分页管理机制	286	第八章 汇编语言与高级语言的混合编程	379
6.6.1 线性地址到物理地址的转换	287	8.1 混合编程的约定规则	379
6.6.2 页级保护和对虚拟存储器的支持	290	8.1.1 命名规则	380
6.6.3 分页管理机制举例	291	8.1.2 调用规则	380
6.7 中断和异常的处理	298	8.2 MASM 宏汇编语言对混合编程的支持	381
6.7.1 中断和异常的概念	298	8.3 C/C++与汇编语言混合编程	383
6.7.2 异常类型及各种异常产生的原因	299	8.3.1 模块链接法	383
6.7.3 中断和异常处理的控制转移	303	8.3.2 嵌入式汇编	395
6.7.4 演示中断处理举例	306	8.4 Pascal 与汇编语言混合编程	400
6.8 输入输出保护	313	小结	404
6.8.1 输入输出保护的方法	314	习题八	405
6.8.2 重要标志位的保护	316	第九章 多媒体指令及其编程	406
6.8.3 演示输入输出保护举例	316	9.1 MMX 指令集	406
6.9 虚拟 86 模式	327	9.1.1 MMX 技术基础	406
6.9.1 进入和离开 V86 模式	327	9.1.2 MMX 指令集	409
6.9.2 进入和离开 V86 模式举例	330	9.1.3 MMX 指令程序设计	417
6.9.3 V86 模式的受限指令	341	9.2 SSE 指令集	419
小结	342	9.2.1 SSE 技术基础	419
习题六	342	9.2.2 SSE 指令集	421
第七章 浮点程序设计	344	9.2.3 SSE 指令程序设计	430
7.1 预备知识	344	9.3 SSE2 指令集	431
7.1.1 FPU 能直接处理的数据及其有关问题	344	9.3.1 SSE2 编程环境	431
7.1.2 与编程相关的 FPU 的硬件	348	9.3.2 SSE2 指令集	432
7.2 FPU 的指令语句	351	9.3.3 SSE2 指令程序设计	442
7.2.1 传送类指令语句	352	小结	443
7.2.2 算术运算类指令语句	358	习题九	443
7.2.3 超越函数类指令语句	360	第十章 Win32 环境下汇编语言程序设计	445
7.2.4 比较类指令语句	362	10.1 一个例子:Hello world	445
7.2.5 FPU 控制类指令语句	367	10.1.1 指令系统选择	446
7.3 FPU 汇编语言程序设计	369	10.1.2 工作模式选择	446
7.3.1 FPU 程序设计的两个问题	369	10.1.3 选项定义	447
7.3.2 FPU 汇编语言程序设计举例	370	10.1.4 链接头文件和库文件	447
		10.1.5 段定义	448
		10.2 Win32 应用程序的特点	448
		10.2.1 API 函数	449

10.2.2 动态链接库	450	10.3.4 Windows 程序设计方法	463
10.2.3 句柄	451	10.3.5 Windows 汇编程序设计 流程	464
10.2.4 用户界面对象	451	10.4 应用程序实例	464
10.2.5 资源	457	10.4.1 控制台程序	464
10.2.6 控件	458	10.4.2 动态链接库	466
10.3 Win32 汇编程序设计方法	462	小结	468
10.3.1 Windows 系统的消息机制 ...	462	习题十	468
10.3.2 Windows 消息的格式	462		
10.3.3 Windows 程序的运行模式 ...	463		
附录 A 80x86 指令系统一览表	471		
附录 B 汇编语言程序的上机实习指导	499		
附录 C DOS 系统功能调用 (INT 21H)	515		
参考文献	523		

上篇

基 础 篇



第一章 80x86 系列微机结构组成

汇编语言是面向机器的语言,它是介于计算机能直接识别的机器语言与人们容易理解的高级语言之间的一种语言。它与计算机硬件密切相关,因不同的计算机而异。它除设置与机器语言代码指令对应的指令性语句外,还增加了专用于定义变量、常量、过程及存储空间等的说明性语句。它能对计算机的任一存储单元和某些寄存器直接操作,甚至是对其中的某一位进行操作。因此,要进行汇编语言程序设计,就必须掌握计算机硬件系统相关知识。本章将讲述汇编语言程序设计使用的 80x86 系列微型计算机(简称微机)系统的有关硬件知识,主要包括微处理器结构、寄存器、主存储器(简称主存)组织及能直接处理的数据及其在主存中的存放形式。

1.1 80x86 系列微处理器结构

作为超大规模集成电路(VLSI)的微型计算机是第四代计算机的一个重要分支。微型计算机的发展是与微处理器的发展相互促进的。30 多年来,微处理器集成度几乎每两年增加一倍,1~4 年更新换代一次,各代的划分通常以速度、字长为主要依据。世界上有多家公司加入微处理器的开发与研制,经过激烈的市场竞争和优胜劣汰,目前最有成就的是 Intel 和 Motorola 两大公司。Motorola 公司生产了 MC6800、68000、68010、68020、68030、68040、68050、68060 等,最近又推出了采用 RISC 技术的 PowerPC。Intel 公司生产的系列微处理器,以 Intel 80x86 系列产品独领风骚,在各种通用、专用微机和工作站中应用最为广泛。本书仅介绍 80x86 系列微处理器。

80x86 系列计算机硬件系统主要由中央处理机(Central Processing Unit, CPU)、主存储器(Memory)和输入输出子系统(Input/Output, I/O)三部分组成,各部分之间通过系统总线(System Bus)相连,如图 1.1 所示。

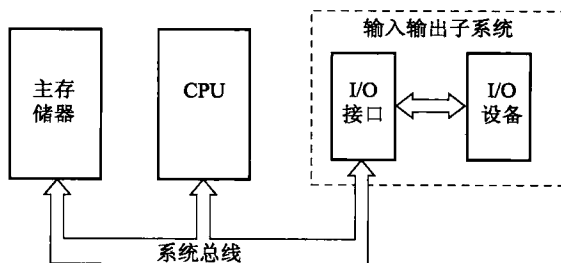


图 1.1 80x86 系列计算机的基本结构

主存储器是计算机内部存储指令和数据的部件。输入输出子系统用于实现人机间通信和信

息交流,包括磁盘、光盘等外存储设备、输入输出设备及它们的控制部件和 I/O(输入输出)接口。CPU 是计算机内部对数据进行处理并对处理过程进行控制的部件,它是计算机的核心部件,由运算器(ALU)和控制器(CU)构成。随着 VLSI 技术的发展,已将 CPU 集成在一个半导体芯片上,这种芯片称为微处理器。现代高档微机的 CPU 还集成了高速缓冲存储器(Cache)。

从 1971 年 Intel 公司推出第一片微处理器 4004 至今,计算机技术取得了长足的发展,表 1.1 列出 Intel 公司微处理器芯片的发展年表及其主要性能、特点。其中有的生存期很短,有的甚至仅在实验室中使用过,如 80186。

表 1.1 Intel 公司系列微处理器

产品型号	推出年份	主频(Hz)	片内晶体管数(万个)	最大可配置主存空间	通用寄存器位数/外部数据总线宽度	地址总线宽度	其他性能与特点
4004	1971	740 K	0.23	2048 B	4/4	12	数据类型 4 位,45 种指令,速度 50 ~ 60 KIPS(每秒千条指令)
8008	1972	800 K	0.33	16 KB	8/8	14	数据类型 8 位,7 个 8 位寄存器,48 种指令,速度 80 KIPS
8080	1974	2 ~ 3 M	0.60	64 KB	8/8	16	数据类型 8 位,70 多种指令,速度 600 KIPS
8085A	1977	3 ~ 6 M	0.9	64 KB	8/8	16	数据类型 8 位,696 种指令,速度 770 KIPS
8086	1978	4.77 ~ 10 M	2.9	1 MB	16/16	20	数据类型 8/16 位,2 万多种指令,速度 2.5 MIPS(每秒百万条指令)
8088	1978	4.77 ~ 10 M	2.9	1 MB	16/8	20	除外部数据总线宽度为 8 外同 8086
80186	1982	8 ~ 16 M		1 MB	16/16	20	除增加两级 DMA、三级中断控制器外同 8086
80286	1982	8 ~ 12.5 M	13.4	16 MB	16/16	24	数据类型 8/16 位;对 8086 原指令功能扩充并新增 23 种指令;速度 4 MIPS;可使用实地址和保护工作模式和虚拟存储器,虚存容量 1 GB

续表

产品型号	推出年份	主频(Hz)	片内晶体管数(万个)	最大可配置主存空间	通用寄存器位数/外部数据总线宽度	地址总线宽度	其他性能与特点
80386DX	1985	16 ~ 33 M	27.5	4 GB	32/32	32	数据类型 8/16/32 位;对 286 指令功能扩充并新增多种指令,确立了 32 位指令系统;速度 6 ~ 12 MIPS;增加虚拟 86 模式,虚存容量 64 TB
80386SL	1988	16 ~ 20 M	27.5	32 MB	16/16	25	为使用 16 位指令系统编写的程序,低功耗,面向便携式而推出
80486DX	1989	25 ~ 50 M	120	4 GB	32/32	32	将 8 KB Cache、浮点单元 FPU 即 80387 和 80386DX 集成在一起,速度最高可达 50 MIPS。此后所有处理器的虚存容量均为 64 TB
80486DX2	1992	50 ~ 66 M	120	4 GB	32/32	32	双倍频的 486DX,速度最高可达 100 MIPS
80486SL	1992	20 ~ 25 M	120	4 GB	32/32	32	低功耗,面向便携式的 486DX
80486DX4	1994	75 ~ 100 M	146	4 GB	32/32	32	片内增加 8 KB Cache 即 L_1 ,3 倍频的 486DX,速度最高可达 150 MIPS
Pentium	1993	60 ~ 233 M	320	4 GB	32/64	32	16 KB 的 L_1 分为 8 KB 指令 Cache (I-Cache) 和 8 KB 数据 Cache (D-Cache),速度最高可达 200 MIPS。此后所有处理器的 L_1 均分为指令和数据两种 Cache
Pentium Pro	1995	133 ~ 200 M	550	64 GB	32/64	36	除 16 KB 的 L_1 外,增加二级 Cache 即 L_2 ,容量为 256 KB ~ 1 MB;首次采用双总线(DIB)结构;速度最高可达 200 MIPS
Pentium P55C	1997	166 ~ 233 M	450	64 GB	32/64	36	L_1 为 32 KB, L_2 为 0 ~ 256 KB;增加多媒体扩展(MMX)指令集。此后所有处理器均支持 MMX 指令集

续表

产品型号	推出年份	主频 (Hz)	片内晶体管数 (万个)	最大可配置主存空间	通用寄存器位数/外部数据总线宽度	地址总线宽度	其他性能与特点
Pentium II	1997	233 ~ 450 M	750	64 GB	32/64	36	L ₁ 为 32 KB, L ₂ 为 512 KB; 增加 8 个专用于 MMX 的 64 位寄存器和可重命名寄存器。此后所有处理器均采用双总线 (DIB) 结构
Pentium III	1999	400 M ~ 1.4 G	950	64 GB	32/64	36	增加实现 SSE 新指令集硬件和 8 个专用于 SSE 的 128 位 XMM 寄存器, 其他同 Pentium II
Pentium 4	2000	1.5 ~ 3.66 G	4 200	64 GB	32/64	36	增加实现 SSE2 新指令集硬件, 采用全新的 NetBurst 微结构, Cache 子系统由 12 KB 执行跟踪 Cache、8 KB 数据 Cache L ₁ 和 256 KB 的 L ₂ 组成

对表 1.1 的几点说明如下。

① 80x86 系列微处理器: 8086 及其以后的微处理器称为 80x86 系列微处理器, 排列在前面的称为低档 80x86 微处理器, 后面的称为高档 80x86 微处理器, 它们的指令系统是“向上兼容”的。所谓“向上兼容”指的是, 使用低档的微处理器指令系统编写的程序, 可不必修改地在高档微处理器上运行; 而使用高档微处理器指令系统编写的程序, 在低档微处理器上一般不能运行。

② 80x86 系列微处理器主频: 其中微处理器的主频都给出一定的范围, 表示该型号有多种频率的微处理器产品。同一型号产品, 主频越高, 处理速度越快。同一主频高档微处理器产品处理速度高于低档微处理器产品。

③ 浮点单元 (FPU): FPU 即 Floating Point Unit 的缩写。80486DX 及其后继微处理器内部都集成了浮点单元 FPU, 它是实现浮点指令集的硬件。较 80486DX 低档的微处理器内部无 FPU, 编程时若使用 FPU 提供的浮点运算指令, 需配置相应的称之为协处理器的芯片。8086/8088 微处理器需配置 8087 协处理器芯片, 80286 微处理器需配置 80287 协处理器芯片, 80386 微处理器需配置 80387 协处理器芯片。有关 FPU 的结构、FPU 浮点指令集和编程方法将在第七章介绍。

④ MMX (Multi-Media eXtension) 指令集: 多媒体扩展指令集, 包含 57 种指令。它的主要特点是采用单指令多数据 (Single Instruction Multiple Data, SIMD) 技术, 允许一条指令处理多个数据, 提高了并行性, 极大地提高了文字识别、图形图像、语音视频、通信和虚拟现实等应用程序的运行速度。MMX 指令集及其编程方法将在第九章讲授。

⑤ SSE (Streaming SIMD Extension) 指令集: 流式单指令多数据扩展指令集, 共包含 70 种指

令。它主要解决 MMX 指令集不能提高浮点运算速度的问题。使用 SSE 指令集,大大提高了三维几何运算、动画处理、游戏等程序的运行速度。

⑥ SSE2 指令集:SSE 指令集的扩展。在 SSE 指令集的基础上新增 76 种指令,目的是进一步提高文字识别、图形图像、语音识别与压缩、视频编辑与压缩解压和虚拟现实等应用程序的运行速度。SSE 和 SSE2 指令集及其编程方法将在第九章讲授。

Pentium 4 微处理器的指令系统包含通用指令集、浮点指令集、MMX 指令集、SSE 和 SSE2 指令集。下面,从程序设计的角度重点介绍 8086、80386、80486 微处理器结构,简要介绍 Pentium 及其后继微处理器结构特点。

1.1.1 8086CPU

8086 CPU 是 16 位微处理器。其内部数据总线和外部数据总线均为 16 位,地址总线为 20 位,具有 6 字节的指令流队列。图 1.2 为 8086 CPU 结构示意图,图中虚线表示控制线。

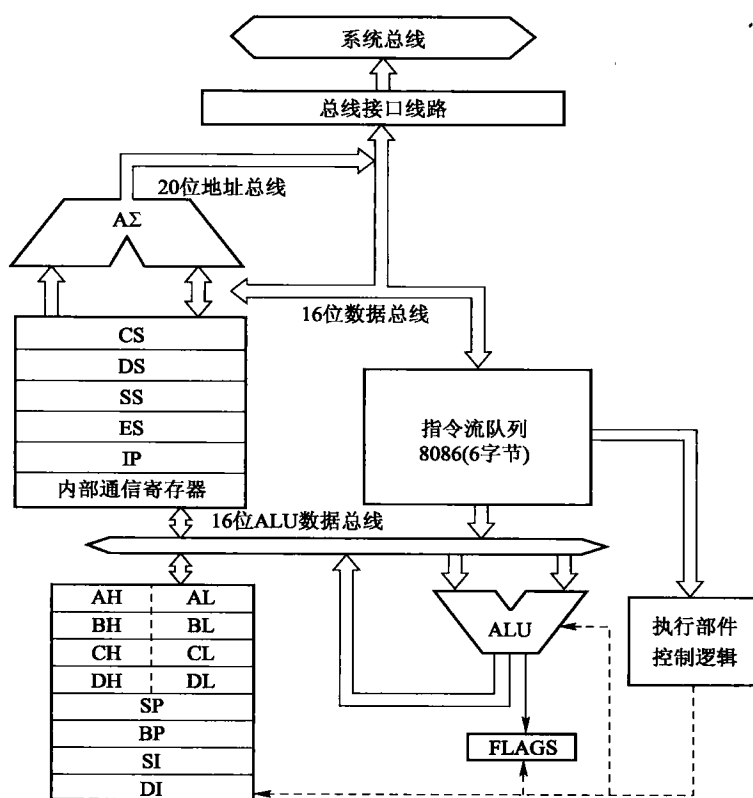


图 1.2 8086 CPU 结构示意图

8086 CPU 可分为两大组成部分。以 16 位 ALU 数据总线为界,下面的部分为执行部件(Execution Unit, EU),上面的部分为总线接口部件(Bus Interface Unit, BIU)。

执行部件(EU)由 8 个 16 位通用寄存器、标志寄存器 FLAGS、算术逻辑运算部件(ALU)以

及执行部件控制逻辑构成。ALU 可实现 16 位的算术逻辑运算,FLAGS 寄存执行结果的状态信息,8 个 16 位通用寄存器的功能将在 1.2 节详细介绍。

EU 的功能:一是根据指令进行算术和逻辑运算;二是如果操作数在主存储器中,则根据指令中操作数的寻址方式,计算出操作数的偏移地址,再通过 BIU 形成一个 20 位的物理地址(详见 2.1.1 节),据此从主存储器中存取所需的操作数。

总线接口部件(BIU)由地址加法器 $A\Sigma$ 、段寄存器 CS、DS、SS 和 ES,16 位指令指示器 IP 以及 6 字节的指令流队列组成。BIU 的功能是完成 CPU 与主存储器和 I/O 设备之间的信息传送。具体过程是,BIU 总是预先从主存储器中取出后续指令送指令流队列,排队等待执行。指令执行时,当指令要求从主存储器或 I/O 设备读取数据,或者将结果存放到主存储器或 I/O 设备时,EU 就向 BIU 发出请求,BIU 根据 EU 的请求完成这些操作。

由于 EU 和 BIU 的操作是独立进行的,它们在很大程度上可以并行工作。所以在 EU 执行一条指令的过程中,BIU 就可取出下一条或多条指令,并存放到指令流队列中。当 EU 执行完一条指令后,可立即到指令流队列取下一条指令执行,大大减少了 CPU 等待取指令所需的时间,提高了 CPU 的效率,加快了系统的运行速度。

8086 处理器仅支持基本的通用指令集,如果希望提高浮点运算的速度则需安装支持浮点指令集的 8087 芯片。

1.1.2 80386 CPU

80386DX CPU 采用 32 位的数据总线和地址总线,最多可寻址 4 GB 内存和 64 TB 虚拟内存,其内部寄存器也从 16 位扩展到 32 位。80386DX CPU 由六个部件组成,它们分别是执行部件、指令预取部件、指令译码部件、总线接口部件、分段部件和分页部件。这些部件按流水线结构设计,彼此功能独立,能并行工作。它扩展了 8086 的通用指令集,结构示意图如图 1.3 所示。

总线接口部件(BIU)的功能与 8086 的 BIU 类似,负责 CPU 与外部总线的连接。不同的是,指令的预取功能从 BIU 中分离出来,成为一个独立的功能部件。CPU 内部的其他部件都能与 BIU 直接通信,并将它们的信息传输请求传送给 BIU。BIU 对这些请求按优先级来选择执行。在总线空闲时,指令预取部件(Instruction Prefetch Unit,IPU)指示 BIU 从存储器中预取指令放入 16 字节的预取队列中,等待指令译码部件的处理。指令译码部件(Instruction Predecode Unit,IDU)从指令预取队列中取出指令并对它们进行译码,将其以先进先出方式存入译码指令队列中,等待执行。只要译码指令队列有闲置空间,指令译码部件就会从指令预取部件中取出下一条指令进行译码。执行部件 EU 由控制部件、数据处理部件和保护测试部件组成。数据处理部件包括一个 ALU、一个乘法器、八个用于地址计算和数据处理的 32 位通用寄存器以及一个 64 位桶形移位器,用于加速移位、循环和乘除法操作。EU 功能是执行指令,并负责与执行该指令有关部件进行通信。分段部件和分页部件共同构成了存储器管理部件(Memory Management Unit,MMU),用于实现对存储器的段页式管理。

80386 处理器仅支持通用指令集,如果希望提高浮点运算的速度则需安装支持浮点指令集的 80387 芯片。