



普通高等教育“十一五”国家级规划教材

微机系统汇编语言 与接口技术

冯萍 吴晓 编

第2版



普通高等教育“十一五”国家级规划教材

微机系统汇编语言与接口技术

第 2 版

冯 萍 吴 晓 编
冯博琴 吴 宁 主审



机械工业出版社

本书为普通高等教育“十一五”国家级规划教材。

本书以 Intel 80X86 为背景机介绍微处理器、汇编语言与接口技术的基础知识、原理和使用方法。全书分为三部分，第一部分是 Intel 微处理器系列基本工作原理介绍；第二部分是汇编语言部分，以 MASM6.11 的 Programmer's Work Bench 为平台，介绍 80X86 指令系统及汇编语言程序设计技术基础，并通过典型应用帮助读者深入学习和掌握汇编语言程序设计的方法；第三部分是接口技术部分，首先引入微机基本接口技术，系统和详细地介绍了中断、串行和并行通信、时钟以及总线等技术的基本原理和应用方法，进一步讲述了 Pentium PC 发展的部分接口技术。

本书可以作为本科计算机专业、自动控制类专业“微机原理与接口技术”或“汇编语言与接口技术”课程的教材，亦可供从事系统开发的工程技术人员学习使用。

图书在版编目 (CIP) 数据

微机系统汇编语言与接口技术/冯萍, 吴晓编. —2 版. —北京: 机械工业出版社, 2011. 3

普通高等教育“十一五”国家级规划教材

ISBN 978-7-111-34862-7

I. ①微… II. ①冯…②吴… III. ①汇编语言—程序设计—高等学校—教材②微型计算机—接口—高等学校—教材 IV. ①TP3

中国版本图书馆 CIP 数据核字 (2011) 第 098177 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 王小东 责任编辑: 王小东 任正一

版式设计: 霍永明 责任校对: 李秋荣

封面设计: 姚毅 责任印制: 乔宇

北京机工印刷厂印刷 (三河市南杨庄国丰装订厂装订)

2011 年 9 月第 2 版第 1 次印刷

184mm × 260mm · 31 印张 · 771 千字

标准书号: ISBN 978-7-111-34862-7

定价: 58.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

社服务中心: (010)88361066

门户网: <http://www.cmpbook.com>

销售一部: (010)68326294

教材网: <http://www.cmpedu.com>

销售二部: (010)88379649

读者购书热线: (010)88379203

封面防伪标均为盗版

第 2 版前言

目前,国内最广泛使用的微机系统,主要使用 Intel 80X86 系列微处理器或者兼容的微处理器。在 Intel 的 80X86 家族中,16 位的 8086/8088 微处理器是基础,实现了以分段方式管理存储器;32 位的 80X86 微处理器,实现了支持多任务的保护工作方式;其后推出的 Pentium 系列微处理器,其微架构不断更新,支持三级智能缓存、硬件防病毒技术 EDB、节能省电技术 EIST、超线程技术及多核技术等,代表着微处理器的发展方向。因此,本书以 80X86 作为教材背景机。鉴于篇幅有限,本教材概要介绍了 Intel 微处理器基本结构及工作原理,可以作为讲授微机原理的参考教材。

汇编语言面向机器,能够为程序员提供最直接操纵机器硬件系统的途径,利用它可以编写出在“时间”和“空间”两个方面最具效率的程序,同时也是作为对微处理器工作原理深入了解的必备知识。随着微型计算机应用的日益广泛,微机接口技术的重要性也日益明显地表现出来。汇编语言与接口技术部分内容属于计算机各专业的技术基础课程,是必修的核心课程之一,该技术基础课程对于训练学生掌握汇编语言程序编程和进行微机接口设计都有重要作用。此外,该课程也是其他相关专业的必修或选修课。

为了适应教学需要,我们编写了《微机系统汇编语言与接口技术》一书,该书在微机原理部分,重点介绍了 Intel 的 80X86 系统中最基本的、且具有代表性的几种微处理器的结构和特点,其目的是通过本部分内容的介绍,使学生能够对微处理器的基本原理和基本结构有一个概要的了解,为后续汇编和接口技术的学习奠定基础。

在汇编语言程序设计部分,讲解了汇编语言程序设计的一般概念、基本技术和常用技巧,模块化程序设计及保护方式程序设计的方法,应用编程技术及实现细节。在微机接口技术部分介绍了接口技术的基本概念、总线连接技术、中断技术、并行通信和串行通信接口技术等。先讲述基本原理,再介绍实现这些原理的典型芯片,并且给出这些原理或芯片的应用实例,力求理论联系实际,做到原理、技术和应用并重,硬件和软件结合。

全书共分 10 章。第 1 章绪论,简要介绍了 Intel 微处理器的发展、汇编语言与微机接口技术的特点及应用、计算机中数制及运算基础;第 2 章 Intel 80X86 微处理器,主要介绍了微处理器的基本结构、寄存器结构、工作方式等;第 3 章 80X86 寻址方式和指令系统,对 32 位机在 16 位模式或 32 位模式中工作的寻址方式及操作都做了较详细的阐述,同时列举了一些程序,帮助读者深入理解其指令的功能;第 4 章汇编语言及程序设计,论述了汇编语言源程序的设计方法,常用的伪指令使用技巧,宏和模块化程序设计的方法以及汇编语言程序和高级语言程序的相互调用;第 5 章微机接口技术,讲述了 I/O 接口技术、DMA 技术、传统中断及高级中断控制技术等;第 6 章存储器接口技术,介绍了微型计算机中的内存储器及地址译码原理,重点介绍了微处理器与存储器的扩展连

接技术；第7章总线技术，主要介绍了总线的有关概念、功能、总线标准及连接技术，常用总线的有关规范等；第8章可编程接口芯片，重点介绍了常用并行和串行接口芯片、定时器/计数器芯片的结构及其与微处理器接口方式和编程，为了与当前各大、专院校普遍使用的16位微型计算机实验设备相衔接，主要以8位接口芯片为例；第9章数/模与模/数转换，主要讲述了数/模转换器和模/数转换器的一般工作原理，重点介绍与微处理器的接口技术及其编程；第10章应用程序设计，介绍了DOS应用程序设计方法，实模式和保护模式切换等接口技术。

本书每章都有思考题与习题，以便帮助读者理解和掌握有关内容。

由于水平有限，书中不妥和谬误之处在所难免，恳请读者批评指正。

编 者

目 录

第 2 版前言

第 1 章 绪论 1

1.1 Intel 微处理器发展综述 1

1.1.1 微处理器内核不断翻新 1

1.1.2 微处理器芯片制造工艺不断提高 3

1.1.3 并行化技术不断发展 4

1.1.4 X86 指令集不断扩展 6

1.2 汇编语言与接口技术的特点和使用场合 8

1.2.1 汇编语言的特点和使用场合 8

1.2.2 接口技术的特点 10

1.3 运算基础 11

1.3.1 进位计数制及其相互转换 11

1.3.2 二进制数的运算规则 14

1.3.3 计算机中的四则运算 15

1.3.4 计算机中带符号数的表示方法 16

1.3.5 计算机中数的小数点表示方法 19

1.3.6 二进制编码 21

1.3.7 逻辑运算与基本逻辑电路 23

思考题与习题 25

第 2 章 Intel 80X86 微处理器 27

2.1 微处理器的基本结构 27

2.1.1 微处理器的内部基本结构 27

2.1.2 微处理器外部基本引脚 31

2.1.3 80X86 微处理器的主要逻辑结构 35

2.2 80X86 内部寄存器结构 40

2.3 微处理器的工作方式 45

2.3.1 实地址方式 45

2.3.2 虚拟 8086 方式 45

2.3.3 保护方式 46

2.3.4 3 种工作方式的状态转换 46

2.3.5 系统管理方式 (SMM) 47

2.4 实模式存储器寻址 47

2.5 保护模式存储器分段管理 48

2.6 保护模式存储器分页管理 54

思考题与习题 57

第 3 章 80X86 寻址方式和指令系统 58

3.1 80X86 寻址方式 58

3.1.1 数据寻址方式 58

3.1.2 程序地址寻址方式 64

3.1.3 堆栈地址寻址 66

3.2 80X86 指令格式 67

3.2.1 80X86 指令编码格式 67

3.2.2 80X86 指令格式 68

3.3 80X86 指令系统 69

3.3.1 数据传送指令 69

3.3.2 算术运算指令 73

3.3.3 逻辑运算指令 79

3.3.4 控制转移类指令 85

3.3.5 串操作指令 90

3.3.6 输入/输出指令 93

3.3.7 处理器控制 95

3.3.8 中断指令与 DOS 功能调用 95

思考题与习题 98

第 4 章 汇编语言及程序设计 101

4.1 汇编语言 101

4.1.1 汇编语言格式 101

4.1.2 简化的段定义伪指令 102

4.1.3 完整段定义伪指令 103

4.1.4 常用伪指令 105

4.1.5 汇编程序两种基本格式 109

4.2 分支程序设计 111

4.2.1 简单分支程序 112

4.2.2 复杂分支程序 116

4.3 循环程序设计 118

4.3.1 单重循环程序 118

4.3.2 多重循环程序 123

4.4 子程序设计 125

4.4.1 过程定义 125

4.4.2 调用指令 125

4.4.3 堆栈帧指令 127

4.4.4 子程序设计 128

4.4.5 子程序的递归和重入	135	应用	189
4.5 控制汇编语言程序语句	139	5.3.7 8259A 编程应用	190
4.5.1 IF 语句	139	5.3.8 中断控制方式的设计与应用	191
4.5.2 DO—WHILE 语句	140	5.4 APIC 技术	192
4.5.3 REPEAT—UNTIL 语句	141	5.4.1 82093AA I/O 高级可编程中断 控制器 (IOAPIC)	193
4.6 结构和记录	142	5.4.2 APIC 系统的组成	198
4.6.1 结构和联合	142	5.4.3 APIC 系统工作模式	199
4.6.2 记录	145	5.4.4 APIC 总线周期	203
4.7 宏	147	5.4.5 APIC 系统工作过程	205
4.7.1 宏指令的定义和使用	147	5.4.6 APIC 总线仲裁	206
4.7.2 宏指令中参数的使用	148	5.4.7 APIC 系统的中断类型	206
4.7.3 宏与子程序的区别	149	5.5 直接存储器存取控制	210
4.7.4 与宏有关的局部变量说明伪 指令 LOCAL	149	5.5.1 DMAC 的基本功能和基本操 作	210
4.8 源程序的结合	150	5.5.2 DMAC 占用总线的方式	211
4.9 PUBLIC 和 EXTERN 伪指令	152	5.5.3 操作类型	212
4.10 库文件	154	5.5.4 操作方式	212
4.11 汇编语言与高级语言的接口	155	5.5.5 可编程 DMA 控制器 8237A	213
4.11.1 C 语言调用协议	155	思考题与习题	227
4.11.2 内存模式和段的约定	156	第 6 章 存储器接口技术	229
4.11.3 数据类型与结果返回	156	6.1 存储器与微处理器的接口	229
4.11.4 PASCAL 语言调用协议	157	6.1.1 微型计算机中的内存储器及地址 译码原理	229
4.11.5 MASM 调用高级语言	157	6.1.2 32 位存储器的组成与多字节 访问	230
4.11.6 调用举例	157	6.2 微处理器与存储器的扩展连接技术	231
4.12 条件汇编和条件错误汇编	159	6.2.1 存储器芯片引脚的连接	231
4.13 汇编和连接程序	160	6.2.2 存储容量的字扩展	232
4.14 汇编语言程序上机过程	161	6.2.3 存储容量的位扩展	236
思考题与习题	162	6.2.4 存储容量的体扩展	237
第 5 章 微机接口技术	166	6.2.5 主存储器与微处理器的连接	237
5.1 I/O 控制	166	思考题与习题	238
5.1.1 I/O 控制方式	166	第 7 章 总线技术	239
5.1.2 程序查询	167	7.1 概述	239
5.2 中断控制	168	7.1.1 总线规范的基本内容	239
5.2.1 基本中断控制方式	168	7.1.2 总线分类	240
5.2.2 中断判决与控制	169	7.1.3 采用标准总线的优点	241
5.2.3 80X86 实方式中断	170	7.2 总线数据传输	241
5.3 8259A 可编程中断控制器	175	7.2.1 总线传输过程	241
5.3.1 基本功能	176	7.2.2 总线传输信号特性	242
5.3.2 结构及引脚	176	7.2.3 总线传输控制	244
5.3.3 8259A 管理中断过程	178	7.2.4 总线仲裁	247
5.3.4 8259A 的命令	179		
5.3.5 8259A 工作方式	184		
5.3.6 8259A 在 PC/XT 系统中的			

7.2.5 奔腾系列微处理器总线技术	250	9.2.1 数/模转换基本原理	359
7.2.6 现代总线发展	252	9.2.2 D/A 的性能参数和术语	362
7.2.7 总线的性能指标	253	9.3 数/模转换芯片及接口技术	364
7.3 常用总线介绍	254	9.3.1 DAC0832	364
7.3.1 PCI 总线	254	9.3.2 DAC1210/1209/1208 系列 D/A 转换器及接口	367
7.3.2 PCI Express	276	9.3.3 DAC8562	371
7.3.3 AGP 总线	281	9.3.4 数/模转换器芯片和微处理器的 接口需要注意的问题	374
7.3.4 SCSI 总线	289	9.4 模/数转换原理	375
7.3.5 IDE 总线	291	9.4.1 模/数转换过程	375
7.3.6 Centronics 总线	291	9.4.2 A/D 转换器基本工作原理	378
7.3.7 RS-232C 总线	292	9.4.3 A/D 的性能参数和术语	383
7.3.8 通用串行总线 USB	296	9.5 模数转换芯片及接口技术	384
思考题与习题	300	9.5.1 ADC0808/0809	384
第 8 章 可编程接口芯片	302	9.5.2 AD1674	391
8.1 可编程并行输入输出接口 8255A	302	9.5.3 模/数转换器芯片和微处理器的接 口需要注意的问题	396
8.1.1 8255A 内部结构与引脚信号	302	9.6 D/A 和 A/D 器件的选择	401
8.1.2 8255A 控制字	304	思考题与习题	402
8.1.3 8255A 工作方式	307	第 10 章 应用程序设计	404
8.2 可编程定时器/计数器 8254	316	10.1 保护方式编程	404
8.2.1 主要功能	316	10.1.1 由实方式进入保护方式	404
8.2.2 8254 的内部结构及引脚	316	10.1.2 保护模式操作	415
8.2.3 8254 编程命令	319	10.2 中断程序和中断拦截	418
8.2.4 8254 工作方式	322	10.2.1 中断处理程序	418
8.2.5 8254 在 PC 系列机中的应用	331	10.2.2 中断拦截	419
8.2.6 8254 编程应用	332	10.3 EXEC 功能和程序段前缀	427
8.3 可编程串行输入/输出接口芯片 8251A	335	10.3.1 EXEC 功能	427
8.3.1 串行通信概述	335	10.3.2 程序段前缀 (PSP)	430
8.3.2 8251A 的内部结构	339	10.4 Window 环境下汇编语言程序设计	435
8.3.3 8251A 的引脚分配	340	10.4.1 前言	435
8.3.4 8251A 的编程	342	10.4.2 编写一个简单的 Win32 汇编 语言程序	435
8.3.5 8251A 的接口技术与应用 举例	345	10.4.3 编写显示窗口的 Win32 汇编 语言程序	437
8.4 可编程 I/O 控制模块芯片 82801EB ICH5/82801ER ICH5R	348	10.4.4 菜单	443
8.4.1 内部结构及引脚分配	349	10.4.5 动态链接库	449
8.4.2 功能描述	351	10.5 设备驱动程序	452
思考题与习题	353	10.6 图形显示	462
第 9 章 数/模转换与模/数转换	355	10.6.1 文本方式	462
9.1 信号转换技术	356	10.6.2 图形显示	466
9.1.1 概述	356	10.7 鼠标器	471
9.1.2 几种传感器	356		
9.2 数/模转换原理	359		

思考题与习题	474	附表 B INT 2FH DPMI 功能调用	476
附录	476	附表 C INT 31H DPMI 功能调用	477
附表 A ASCII 码表	476	参考文献	488

第 1 章 绪 论

1.1 Intel 微处理器发展综述

不同系列或同一系列的微处理器都会有不同的核心类型，甚至同一种核心都会有不同版本的类型，核心版本的变更是为了修正上一版存在的缺陷，并增强性能。每一种核心类型都有其相应的制造工艺、面积（成本与核心面积基本上成正比）、电压和电流大小、晶体管数量、各级缓存的大小、主频范围、流水线架构和支持的指令集、功耗和发热量的大小、封装方式、接口类型、前端总线频率（FSB）等。核心类型在某种程度上决定了微处理器的工作性能。

微处理器核心的发展方向是更低的电压、更低的功耗、更先进的制造工艺、集成更多的晶体管、更小的核心面积、更先进的流水线架构和更丰富的指令集、更高的前端总线频率、集成更多的功能、双核心或多核心等。

1.1.1 微处理器内核不断翻新

1. 80X86 系列微处理器

1979 年英特尔推出了 8088/8086 芯片，它属于 16 位微处理器，时钟频率为 4.77MHz，地址总线为 20 位，可使用 1MB 内存。内部数据总线是 16 位，8088 和 8086 的主要区别是：8088 外部数据总线为 8 位，而 8086 外部数据总线为 16 位。

1985 年英特尔推出了 80386 芯片，80386 的内部和外部数据总线都是 32 位，地址总线也是 32 位，可寻址高达 4GB 内存。它除具有实模式和保护模式外，还增加了一种叫虚拟 8086 的工作方式，可以通过同时模拟多个 8086 处理器来提供多任务能力。

1989 年英特尔推出 80486 芯片，80486 是将 80386 和数学协处理器 80387 以及一个 8KB 的高速缓存集成在一个芯片内，并且在 80X86 系列中首次采用了 RISC（精简指令集）技术，可以在一个时钟周期内执行一条指令。它还采用了突发总线方式，大大提高了与内存的数据交换速度。由于这些改进，80486 的性能比带有 80387 数学协处理器的 80386 DX 提高了 4 倍。

2. Pentium 系列微处理器

在 1993 年英特尔推出了新一代的高性能处理器 Pentium。Pentium 系列微处理器的典型核心类型列举如下：

(1) MMX 1997 年英特尔推出了 Pentium MMX，MMX 多媒体扩展指令集技术的诞生使得当时微处理器处理多媒体信息的能力提升幅度达到 60% 以上。MMX 技术开创了微处理器开发的新纪元，之后的 SSE 等多媒体指令仍可看到其身影。

1997 年英特尔发布了 Pentium II 处理器。集合 Pentium MMX 和 Pentium PRO 的特点，将高速缓存与处理器整合在一块 PCB 上。Pentium II 的多媒体能力，也达到了当时的最高水平。

(2) Katmai、Coppermine 1999 年初，英特尔发布了第三代的奔腾处理器——Pentium

Ⅲ，采用了 Katmai 内核，这个内核最大的特点是更新了 SSE 多媒体指令集，以增强三维和浮点应用，并且可以兼容以前的所有 MMX 程序。随后英特尔在 1999 年底发布 0.18 μm 的 Coppermine（铜矿）核心的 Pentium Ⅲ，最大的改进是将二级缓存削减到 256KB，但以全速运行，因此性能大幅提升。

(3) NetBurst 2000 年 11 月，英特尔发布了 Pentium 4，采用了全新的 NetBurst 架构设计，拥有频率为 400MHz 的前端总线、SSE2 指令集、256KB 的二级缓存，起步频率达到 1.5GHz。

(4) Northwood 2002 年初，英特尔推出了基于 Northwood 核心的 Pentium 4，微处理器内部集成了 512KB 的全速二级缓存，使得新一代 Northwood 性能大幅提高，并且拥有极佳的超频潜力，这一年，英特尔发布了一个重要的技术——超线程技术（Hyper-Threading, HT），超线程技术就是利用特殊的硬件指令，把两个逻辑内核模拟成两个物理芯片，让单个处理器都能使用线程级进行并行计算，从而提高多任务、多线程性能，主频可达 3.4GHz。

(5) Prescott 2004 年英特尔正式推出了 P4E 处理器，采用 Prescott 核心，拥有 1MB 的二级缓存、支持 800MHz FSB、HT 二代技术和 SSE3 指令集。新核心 Prescott P4 处理器采用了 90nm 7 层铜互连制造工艺，内部集成的晶体管数达到了 1.25 亿个，这个数目比 Northwood 核心 P4 处理器多了一倍。采用了 31 级超长流水管线的结构来进一步提高主频。

它与 Northwood 最大的区别是采用了更多的流水线结构，核心电压 1.25 ~ 1.525V，前端总线频率可达 800MHz（支持超线程技术），主频可达 3.4GHz，与 Northwood 相比，其二级缓存从 512KB 增加到 1MB。

虽然 Prescott 相比 Northwood 的改进不少，但性能却几乎没有提升，在 3.0GHz 以下的频率，效能甚至比不上 Northwood。这与 Prescott 的 31 级超长流水管线有一定关系。而另一方面，采用 90nm 制造工艺没有为这款 P4 降低发热量，甚至比 Northwood 更热。更换新核心没得到性能提升，反而带来高发热、高功耗的诟病，因此 Prescott 也成为历史上最有争议的核心。

(6) Smithfield 2005 年英特尔发布了第一款双核心处理器的核心类型，基本上可以认为 Smithfield 核心是简单的将两个 Prescott 核心松散地耦合在一起的产物，两个核心通过同一条前端总线与芯片组相连，缺乏必要的协同和资源共享能力，而且还必须频繁地对二级缓存作同步化刷新动作，以避免两个核心的工作步调出问题。这种独立缓存的松散型耦合方案，其优点是技术简单，缺点是性能不够理想。基于该核心的产品是 Pentium D 和 Pentium EE，核心电压 1.3V 左右，都支持硬件防病毒技术 EDB 和 64 位技术 EM64T，支持节能省电技术 EIST。前端总线频率为 800MHz，Pentium EE 主频达 3.2GHz，且支持超线程技术。

(7) Conroe 2006 年 7 月 27 日英特尔正式发布 Conroe 核心，这是全新的 Core（酷睿）微架构（Core Micro-Architecture）应用在桌面平台上的第一种微处理器核心。采用此核心的有 Core 2 Duo E6X00 系列和 Core 2 Extreme X6X00 系列。与上代相比，Conroe 核心具有流水线级数少、执行效率高、性能强大以及功耗低等优点。Conroe 核心采用 65nm 制造工艺，核心电压为 1.3V 左右。Core 2 Extreme 的前端总线频率可达到 1333MHz；在一级缓存方面，每个核心都具有 32KB 的数据缓存和 32KB 的指令缓存，并且两个核心的一级数据缓存之间可以直接交换数据；在二级缓存方面，Conroe 核心都是两个内核共享 4MB，并通过改良了的 Intel Advanced Smart Cache（英特尔高级智能高速缓存）共享缓存技术来实现缓存数据的同步。Conroe 核心都支持硬件防病毒技术 EDB、节能省电技术 EIST 和 64 位技术 EM64T 以及

虚拟化技术 Intel VT。Conroe 核心是先进的桌面平台处理器核心，其在高性能和低功耗上找到了一个很好的平衡点。

(8) Nehalem 2008 年 Intel Atom 节能平台的推出，成功推动上网本的概念。Intel 新一代四核处理器 Core i7 接替高端 Core 2 四核成为新旗舰。Core i7 是一款基于全新 Nehalem 架构的微处理器，集众多先进技术于一身，如集成内存控制器、三通道技术支持、全新 QPI 总线、超线程技术的回归、Turbo Mode 内核加速等。虽然 Core i7 采用新架构，但还建立在 Core 微架构的基础上，对 Core 微架构作了较大幅度的改进，并增添了超线程（HT）、三级 Cache、TLB 和分支预测的等级化、集成内存控制器（IMC）、QPI 总线和支持 DDR3 等技术。Core i7 这款四核微处理器，采用 HT 后，可达到八个线程，性能大幅领先上一代的 Core 2 Quad 四核。

(9) Sandy Bridge 2011 年英特尔计划推出代号为 Sandy Bridge 的处理器，Sandy Bridge 将有八核心版本，二级缓存仍为 512KB，但三级缓存将扩容至 16MB。而 Sandy Bridge 最主要特点则是加入了 game instruction AVX（Advanced Vectors Extensions）技术。

该处理器采用英特尔第二代 32nm HKMG 工艺制造，原始主频 2.50GHz，外频 100MHz（Nehalem 是 133MHz），倍频 25 ×，一级缓存 8 路关联 256KB，二级缓存 8 路关联 4 × 256KB，三级缓存 12 路关联 6MB（小于 Nehalem），整合英特尔第六代图形核心，支持超线程技术和 AES-NI、AVX 指令集，其中 AES-NI 新增 7 条指令，可加速数据加密和解密，AVX（高级矢量扩展）则针对密集型浮点运算，并协助加速一般用途和工程应用的计算。

1.1.2 微处理器芯片制造工艺不断提高

英特尔的创始人戈登摩尔（Gordon Moore）通过长期研究后发现：微处理器芯片中的部件（指晶体管）在不断增加，其价格也在不断下降。“随着单位成本的降低以及单个集成电路集成的晶体管数量的增加，到 1975 年，从经济学来分析，单个集成电路应该集成 65000 个晶体管”。英特尔此后几年的发展都符合“摩尔定律”。摩尔及其同事总结出一句极为精练的公式“集成电路所包含的晶体管每 18 个月就会翻一番”。之后芯片内集成的晶体管数量也证实了他的这句话，并且发展速度还在加快。

从芯片制造工艺来看，在 1965 年推出的 10 μm （微米）处理器后，经历了不断的技术改进，而 32nm（纳米）的制造工艺是目前微处理器的最高工艺。

在 1971 年，英特尔发布了第一个微处理器 4004，采用 10 μm 工艺生产，仅包含 2300 多个晶体管，时钟频率为 108kHz。在 1978 年英特尔推出了 8086 处理器，这时工艺已经缩减为 3 μm 工艺，含 2.9 万个晶体管。1985 年，推出了 386 处理器，含 27.5 万个晶体管。1989 年，英特尔发布了 486 处理器，处理器首次突破了 100 万个晶体管大关，处理器工艺已经全面采用了 1 μm 工艺，并且在芯片内集成了 125 万个晶体管。1993 年采用 800nm 的奔腾（Pentium）的出世，让微处理器全面从微米时代跨入了纳米时代。奔腾含有 310 万个晶体管。1995 年后，半导体行业已普遍采用 0.35 μm （350nm）工艺进行主流芯片的生产。Pentium MMX（多能奔腾，P55C）是最典型的产品，集成了 450 万个晶体管，功耗 17W。在 0.35 μm 工艺的帮助下，工作频率突破了 200MHz。此后 0.25 μm 工艺便应运而生。与 0.35 μm 工艺相比，使用 0.25 μm 制程可使处理器的运算速度提升一倍以上，且工作电压更低，功耗更少。同时，芯片的封装面积更小、成本更低、功能也更强。采用 0.25 μm 工艺的

Intel 处理器最具代表性的产品当数 Pentium III，采用 0.25 μm 制造工艺，集成了 900 万个晶体管。

正如摩尔定律所说，制造工艺的进步无可阻挡，微处理器在经历了 180nm 工艺后，在 2001 年直接进入了 130nm 时代。与 180nm 工艺相比，新的 130nm 工艺的氧化层可减少 30% 以上，工作电压可达到更低，芯片面积更小。采用 130nm 工艺的处理器有 Tualatin 系列（Pentium III-S 及 Celeron III）、Northwood 系列（Pentium 4 A/B/C、Celeron 4）等产品。

在 2004 年推出核心为 Prescott 的 Pentium 4E 处理器，一个显著的特点就是工艺再次改进为 90nm，集成了 1 亿个晶体管。但工艺的提升，没有使得功耗降低，主频的提升，使得 Prescott 功耗开始走高。

但随着芯片中晶体管数量增加，原本仅数个原子层厚的二氧化硅绝缘层会变得更薄进而导致泄漏更多电流，随后泄漏的电流又增加了芯片额外的功耗。此时，由于受“泄漏电流”的影响，导致后续产品频率无法提升，功耗高居不下。为此英特尔迅速部署 65nm 产品计划，在 2005 年推出了 Pentium Extreme Edition 955。

2008 年推出了首款 45nm Penryn 处理器。Penryn 双核心版本内建 4.1 亿个晶体管，四核心则有 8.2 亿个晶体管，微架构经强化后，在相同频率下较上代 Core 产品拥有更高性能，同时 L2 Cache 容量亦提升 50%，明显提高数据读取执行的命中率。此外，亦加入 47 条全新 Intel SSE4 指令，提高多媒体性能和实现高性能运算应用。

目前 Core i7 980X 是全球第一款桌面六核微处理器，采用 32nm 工艺，功耗 130W。

未来 16nm 制造工艺，无疑将成为先进技术的代表。相比而言，它体积更小，性能更高。但同时，要实现 16nm 制造工艺必须解决各个方面的诸多技术难题。未来 16nm 制造工艺面临的挑战依然严峻。

1.1.3 并行化技术不断发展

微处理器的基本工作大致分为指令的获取、解码、运算和结果的写入四个步骤，采用流水线设计之后，指令就可以连续不断地进行处理。在同一个较长的时间段内，显然拥有流水线设计的微处理器能够处理更多的指令。

例如 80486 和 Pentium 均使用了 6 步流水线结构，流水线的 6 步为：

(1) 取指令 微处理器从高速缓存或内存中取一条指令。

(2) 指令译码 分析指令性质。

(3) 地址生成 很多指令要访问存储器中的操作数，操作数的地址也许在指令字中，也许要经过某些运算得到。

(4) 取操作数 当指令需要操作数时，就需再访问存储器，对操作数寻址并读出。

(5) 执行指令 由 ALU 执行指令规定的操作。

(6) 存储或“写回”结果 最后运算结果存放至某一内存单元或写回累加器 A。

在理想情况下，每步需要一个时钟周期。当流水线完全装满时，每个时钟周期平均有一条指令从流水线上执行完毕，输出结果。Pentium、Pentium Pro 和 Pentium II 处理器的超标量设计更是分别结合了两条和三条独立的指令流水线，每条流水线平均在一个时钟周期内执行一条指令，所以它们平均一个时钟周期分别可执行 2 条和 3 条指令。

流水线技术是通过增加计算机硬件来实现的。例如要能预取指令，就需要增加取指令的

硬件电路，并把取来的指令存放到指令队列缓存器中，使微处理器能同时进行取指令和分析、执行指令的操作。因此，在 16 位/32 位微处理器中一般含有两个算术逻辑单元 ALU，一个主 ALU 用于执行指令，另一个 ALU 专用于地址生成，这样才可使地址计算与其他操作并行进行。

随着微处理器频率不断地攀升，英特尔总是在自己某个核心的处理器到达极限之时采用新的、更长流水线的核心来消除频率的瓶颈。但是由于现有芯片制造工艺的限制，频率的提升带来高功耗、高发热量的问题。尽管流水线增长，频率提升的空间相应增大，但是处理器频率提升的其他瓶颈却无法解决。而且过长的流水线意味着更加复杂的内部结构，生产的良品率也难以保证。同时在微处理器的工作中，指令往往不是孤立的，许多指令按一定的顺序执行才能完成一个任务。而一旦某个指令在运算过程中发生了错误，或者执行了没有用的指令，那么其后与之相关的指令就都没有用了。这些指令必须清除掉，然后再执行其他的指令，微处理器相当于做了许多无用功！流水线越长，一旦出错影响也就越大，比如一个指令在最后一级出错，那么可能在后续流水线中的所有指令都要被清除，Northwood 核心处理器要浪费 20 级工序的时间，而 Prescott 核心处理器就要浪费 31 级工序的时间！流水线越长、级数越多就会导致延迟次数越多，总延时就越大，微处理器完成单个任务的时间就会越长。

随着主频的不断攀升，NetBurst 架构的弊端越来越明显。第三代 Prescott 奔腾 4 流水线达到 31 级，以至于它每个时钟周期比 Northwood 多产生大约 60% 的热量，同时功率消耗也增加大约 10%！3.2GHz 的 Prescott TDP 达到了 103W！奔腾 4 最终止步于 3.8GHz，2006 年英特尔宣布新一代处理器将采用 Core 微架构。英特尔指出未来处理器的技术发展重点将是“每瓦特性能”（Performance per Watt）。

Conroe 处理器的数据流水线长度从 Prescott 的 31 级大幅度缩短至 14 级。其算术逻辑运算单元（ALU）数量由上代 NetBurst 微架构的 2 组提升至 3 组，整体运算性能大大增加。

Core 微架构的目标就是构建一个高效的双核心架构，在 Core 架构中，其指令流水线深度达到 14 级，Core 架构是兼顾执行效率和降低功耗的折中设计。采取共享式二级缓存设计，2 个核心共享二级缓存。内核采用高效的 14 级有效流水线设计，每个核心内建 4 组指令解码单元，支持微指令融合与宏指令融合技术，每个时钟周期最多可以解码 5 条 X86 指令，并拥有改进的分支预测功能。每个核心内建 5 个执行单元，执行资源庞大。采用新的内存相关性预测技术。支持增强的电源管理功能。支持硬件虚拟化技术和硬件防病毒功能。内建数字温度传感器，配合系统实现动态的功耗控制和散热控制。

流水线的“条数”与“级数”是完全不同的概念。能够完整执行各种指令的一系列功能单元组成“一条”流水线。而关于流水线级数，可以这样简单理解：一条流水线所包含的功能单元一般可以被划分为多个部分，它可以被划分成几个部分，就称这条流水线是“几级”的。

Core 微架构的 14 级有效流水线与 Prescott 核心的 31 级有效流水线的对比，只有参考意义。不能够由此判断 Core 微架构只能达到很低的频率。

此外，Core 架构的每个核心都拥有 3 个算术逻辑单元（ALU），而之前的微处理器架构只有 1 到 2 个 ALU，这样的设计使得 Core 架构拥有比较高的处理能力。英特尔随后推出的系列微处理器架构都是在 Core 架构之上扩展多核的改进。

1.1.4 X86 指令集不断扩展

1. Intel X86 指令集

微处理器最初只能用加法器来完成整数以及固定小数点位置（整点）的算术运算，最早的微处理器并没有能力进行浮点运算（8088/8086，80286，80386SX），需要浮点运算时，由微处理器通过软件模拟来实现，所以，进行浮点运算时就会慢很多。

8086/8088 是微处理器的鼻祖，所谓 X86 架构也就是指 8086/8088 处理器所开创的指令集体系。为了弥补 8086/8088 在进行浮点运算时的不足，英特尔 1980 年设计了 8087 数学协处理器，专门处理浮点运算，8087 提供两个基本的 32/64bit 浮点形态和额外的扩展 80bit 内部支援来改进复杂运算之精度。8087 后被 80287、80387DX/SX 和 487SX 所取代。

随着时代的发展，越来越多的程序要求使用更高精度的浮点运算，X87 协处理器几乎成为必备品。于是在制造工艺日趋成熟之后，英特尔在 486 一代将 X86 和 X87 整合在了一起，浮点运算成为了微处理器的一项基本功能，而且重要性越来越大。Intel 486DX、Pentium 之后的微处理器都内含了协处理器，所以此后就很少会提及协处理器的概念了。

所谓 X86 架构的处理器就是采用了 Intel X86 指令集的处理器，为了增强计算机的浮点运算能力，增加了 X87 数学协处理器并引入了 X87 指令集，于是就将采用了 X86 指令集和 X87 指令集的处理器统称为 X86 架构的处理器。

2. MMX 指令集：增强多媒体性能

MMX（Multi Media eXtension，多媒体扩展指令）指令集是英特尔在 1996 年为旗下的 Pentium 系列处理器所开发的一项多媒体指令增强技术。MMX 指令集中包括了 57 条多媒体指令，通过这些指令可以一次性处理多个数据，在处理结果超过实际处理能力的时候仍能够进行正常处理，如果在软件的配合下，可以得到更强的处理性能。MMX 指令集非常成功，在之后生产的各型微处理器都包括这些指令集。

但是，MMX 指令集不能与 X86 的浮点运算指令同时执行，必须做密集式的交错切换才可以正常执行，这样一来，就会造成整个系统运行速度的下降。

3. SSE 指令集：加强浮点和 3D 性能

SSE 是 Streaming SIMD Extension（SIMD 扩展指令集）的缩写，而其中 SIMD 为 Single Instruction Multiple Data（单指令多数据）的缩写，所以 SSE 指令集也叫单指令多数据流扩展。

SSE 指令集是为提高处理器浮点性能而开发的扩展指令集，它共有 70 条指令，其中包含提高 3D 图形运算效率的 50 条 SIMD 浮点运算指令、12 条 MMX 整数运算增强指令、8 条优化内存中的连续数据块传输指令。理论上这些指令对当时流行的图像处理、浮点运算、3D 运算、多媒体处理等众多多媒体的应用能力起到全面提升的作用。SSE 指令与 AMD 公司的 3DNow! 指令彼此互不兼容，但 SSE 包含了 3DNow! 中的绝大部分功能，只是实现的方法不同而已。SSE 也向下兼容 MMX 指令，它可以通过 SIMD 和单时钟周期并行处理多个浮点数据来有效地提高浮点运算速度。

该指令集最先运用于英特尔的 Pentium III 系列处理器。

4. SSE2 指令集：进一步优化浮点运算

由于使用 SSE 指令之后，程序执行性能得到极大的提升，于是英特尔又在 SSE 的基础

上推出了更先进的 SSE2 指令集。

SSE2 包含了 144 条指令，由两个部分组成：SSE 部分和 MMX 部分。SSE 部分主要负责处理浮点数，而 MMX 部分则专门计算整数。SSE2 的寄存器容量是 MMX 寄存器的两倍，寄存器存储数据也增加了两倍。在指令处理速度保持不变的情况下，通过 SSE2 优化后的程序和软件运行速度也能够提高两倍。由于 SSE2 指令集与 MMX 指令集相兼容，因此被 MMX 优化过的程序很容易被 SSE2 再进行更深层次的优化，达到更好的运行效果。

5. SSE3 指令集：加强并行数据处理能力

SSE3 指令是目前规模最小的指令集，它只有 13 条指令。它共划分为五个应用层，分别为数据传输命令、数据处理命令、特殊处理命令、优化命令、超线程性能增强 5 个部分，其中超线程性能增强是一种全新的指令集，它可以提升处理器的超线程的处理能力，大大简化了超线程的数据处理过程，使处理器能够更加快速的进行并行数据处理。

SSE3 中 13 个新指令的主要目的是改进线程同步和特定应用程序领域，例如多媒体和游戏。这些新增指令强化了处理器在浮点转换至整数、复杂算法、视频编码、SIMD 浮点寄存器操作以及线程同步等五个方面的表现，最终达到提升多媒体和游戏性能的目的。

英特尔是从 Prescott 核心的 Pentium4 开始支持 SSE3 指令集的。

6. SSSE3 (SSE3S) 指令集：加强多媒体图形图像处理

SSSE3 (Supplemental Streaming SIMD Extensions 3) 是 Intel 命名的 SSE3 指令集的扩充，在 65nm Core 2 Duo 中引入了 SSSE3 指令集。

SSSE3 包含了 16 条新的不同于 SSE3 的指令。均可运作于 64 位的 MMX 寄存器或是 128 位 XMM 寄存器之中。因此，有些 Intel 的文件表示有 32 条新指令。SSSE3 指令集增强了微处理器的多媒体、图形图像处理、多媒体编码、整数运算和 Internet 等方面的处理能力。

7. SSE4.1 指令集：大幅提升浮点运算，优化 MPU 和 GPU 数据共享

2001 年以来英特尔最重要的指令集扩展是 SSE4.1，SSE4.1 指令集包含 54 条指令。英特尔在 Penryn 处理器中加入了对 SSE4.1 的支持，共增加了 47 条新指令，令处理器的多媒体处理能力得到最大 70% 的提升。SSE4 加入了 6 条浮点型点积运算指令，支持单精度、双精度浮点运算及浮点产生操作，且 IEEE 754 指令 (Nearest, - Inf, + Inf, Truncate) 可立即转换其路径模式，大大减少延误，这些改变将对游戏及 3D 内容制作应用有重要意义。

此外，SSE4 加入串流式负载指令，可以提高图形帧缓冲区的读取数据频宽，理论上可获取完整的快取缓存行，即每次读取 64bit 而非 8bit，并可保持在临时缓冲区内，让指令最多可带来 8 倍的读取频宽效能提升，对于视频处理、成像以及 GPU 与 MPU 之间的共享数据应用，有着明显的效能提升。在 45nm Core 2 Duo 中引入了 SSE4.1 指令集。

SSE4 指令集让 45nm Penryn 处理器增加了 2 个不同的 32bit 向量整数乘法运算单元，并加入 8bit 无符号 (Unsigned) 最小值及最大值运算，以及 16bit 及 32bit 有符号 (Signed) 运算。在面对支持 SSE4 指令集的软件时，可以有效地改善编译器效率及提高向量化整数及单精度代码的运算能力。同时，SSE4 改良了插入、提取、寻找、离散、跨步负载及存储等动作，令向量运算进一步专门化。

8. SSE4.2 指令集：优化 XML 和交互式应用性能

在 Nehalem 架构的 Core i7 处理器中，引入了 SSE4.2 指令集，加入了 STTNI (字符串文本新指令) 和 ATA (面向应用的加速器) 两大优化指令。STTNI 包含了 4 条具体的指令。

STTNI 指令可以对两个 16bit 的数据进行匹配操作，以加速在 XML 分析方面的性能。英特尔表示，新指令可以在 XML 分析方面取得 3.8 倍的性能提升。

ATA 包括冗余校验的 CRC32 指令、计算源操作数中非 0 位个数的 POPCNT 指令，以及对于打包的 64bit 算术运算的 SIMD 指令。CRC32 指令可以取代上层数据协议中经常用到的循环冗余校验，英特尔表示其加速比可以达到 6.5 ~ 18.6 倍；POPCNT 用于提高在 DNA 基因配对、声音识别等包含大数据集中进行模式识别和搜索等操作的应用程序性能。

9. AVX 高级矢量扩展指令集

在 Sandy Bridge 架构的 Core i3、Core i5 及 Core i7 中最大的亮点是引入“高级矢量扩展”指令集，game instruction AVX (Advanced Vectors Extensions) 技术，简称“AVX”（之前称做 VSSE），其重要性堪比 Pentium III 引入 SSE。其主要特性有：更宽的矢量运算：从 128bit 增至 256bit，并保持向下兼容性；增强的数据重排；单个操作可同时处理 8 个 32bit 数据；支持三操作数和四操作数，非破坏性句法；支持弹性的访存地址不对齐；可扩展的新操作码 (VEX)。使用 AVX 技术进行矩阵计算的时候将比 SSE 技术快 90%。

1.2 汇编语言与接口技术的特点和使用场合

本节介绍汇编语言与接口技术的特点和使用场合。

1.2.1 汇编语言的特点和使用场合

汇编格式指令是机器指令符号化的指令。机器语言是用二进制编码的机器指令的集合及一组使用机器指令的规则。它是 MPU 能直接识别的唯一语言。只有用机器语言描述的程序，MPU 才能直接执行。用机器语言描述的程序称为目的程序或目标程序。机器指令在形式上表现为二进制编码。机器指令一般由操作码和操作数两部分构成，操作码在前，操作数在后。操作码指出要进行的操作或运算，如加、减、传送等；操作数指出参与操作或运算的对象，也指出操作或运算结果存放的位置，如 MPU 的寄存器、存储单元和数据等。机器指令与 MPU 有着密切的关系。通常，MPU 的种类不同，对应的机器指令也就不同。同一个系列 MPU 的指令集常常具有良好的向下兼容性，例如，Pentium 指令集包含了 8086 指令集。

机器语言用于描述数据在机器中的流动，要求使用者熟悉机器内部结构，由于采用人们所不熟悉的形式来描述计算机要执行的任务，所以用机器语言编写程序十分繁杂，而且需采用编码的方式表示机器要执行的任务，因此用机器语言编制出的程序不易为人们理解、记忆和交流，而且极易出错，一旦有错，难于检查。为了克服机器语言的上述缺点，人们采用便于记忆、并能描述指令功能的符号来表示指令的操作码，这些符号被称为指令助记符。指令中一般包括指令功能和操作数，说明指令功能的助记符采用英文缩写，指令执行的操作数用符号地址来表示，例如 MPU 的寄存器、存储单元地址等。汇编语言是指令和伪指令的集合。伪指令主要用于解释和说明指令中操作数的存放形式、指令和数据的分段和指令段之间的关系等。用汇编语言书写的程序称为汇编语言程序，或称为汇编语言源程序。用汇编语言编写的程序要比用机器语言编写的程序容易理解、调试和维护。

由于 MPU 能直接识别的唯一语言是机器语言，所以用汇编语言编写的源程序必须被翻