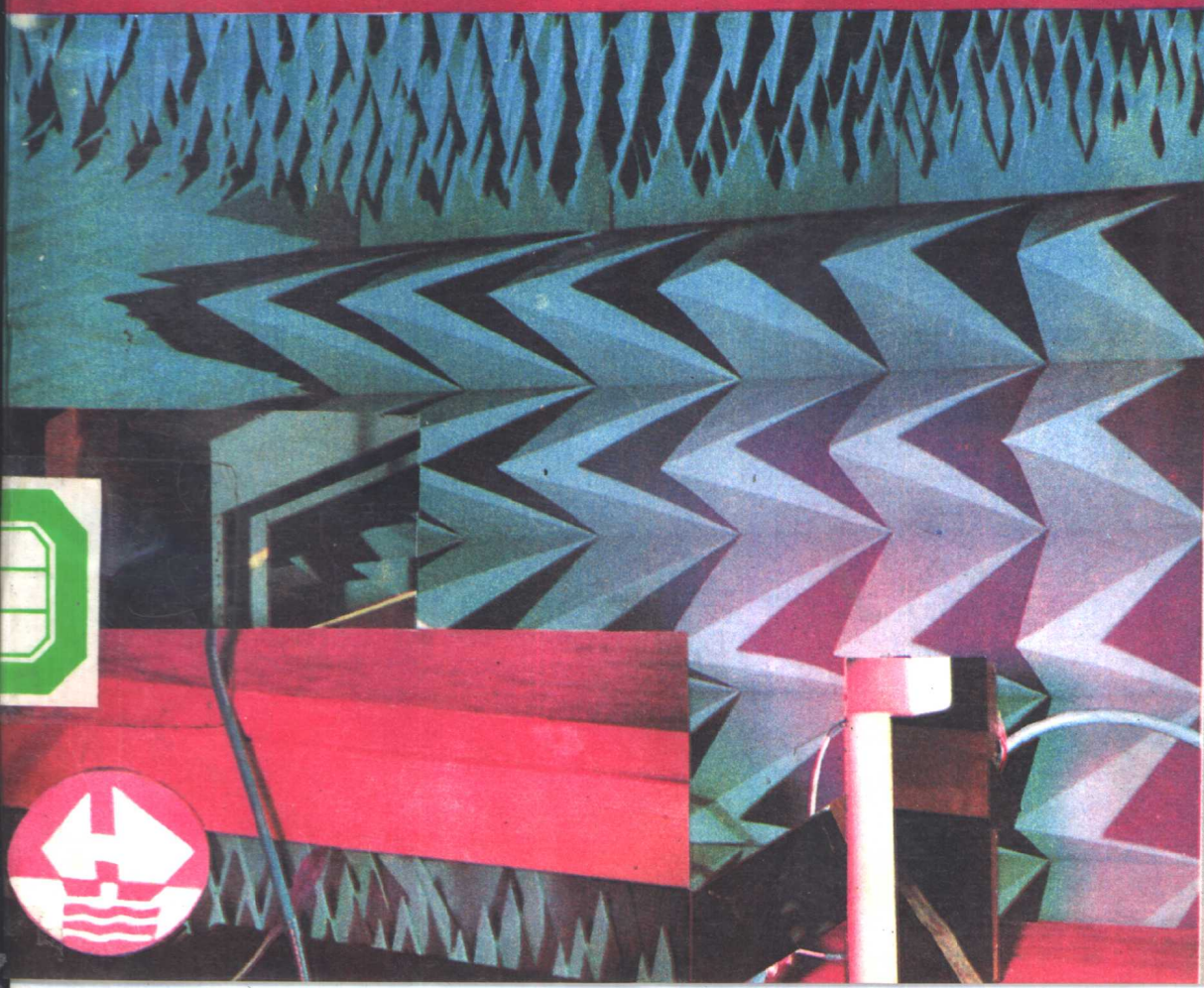


适用于 IBM PC 286、386、486 及其兼容机

80286 / 386 / 387 高级编程实用技术

• 80386 的结构和功能；从 8088 和实方式升级到 286 / 386 保护方式；扩充存储器和虚拟存储器；DOS、环境和 UNIX；os/2；移植到 os/2 和管理程序；286 / 386 程序设计；协处理器；

中国科学院希望高级电脑技术公司



适用于IBMPC 286、386、486及其兼容机

80286／386／387

高级编程实用技术

邹然军

中国科学院希望高级电脑技术公司

一九九一年五月

版权所有
翻印必究

■ 北京市新闻出版局

准印证号：3584—91584

■ 订购单位：北京8721信箱资料部

■ 邮 码：100080

■ 电 话：2562329

■ 传 真：01—2561057

■ 乘 车：320、332、302路
车至海淀黄庄下车

■ 办公地点：希望公司大楼一楼
往里走101房间

1875

目 录

前言	(1)
第一篇 80386 的结构和功能	(3)
第一章 对 386 的剖析	(3)
1.1 观察时钟	(3)
1.2 移动存储器	(5)
1.3 静态手段	(6)
1.4 组织存储器	(8)
1.5 协处理器选择	(8)
1.6 兼容性的考虑	(9)
1.7 大容量存储器	(11)
第二章 升级到80386	(12)
2.1 构造基础	(12)
2.2 80286突破约束	(13)
2.3 一些局限	(15)
2.4 向32位升级	(16)
2.5 指令的改进	(17)
2.6 4 GB 的改进	(18)
2.7 操作方式	(22)
2.8 整个家族的兼容性	(23)
2.9 虚拟特性	(25)
2.10 Intel家族的编程考虑	(26)
2.11 最大限度地利用80386	(27)
第三章 在 80386 上编程	(28)
3.1 操作方式	(28)
3.2 固有结构和指令系统	(30)
3.3 分页	(33)
3.4 性能	(34)
3.5 其它一些特征	(34)
3.6 结束语	(35)
第二篇 从 8088 和实方式升级到286/386和保护方式	(36)
第四章 辨别CPU的类型	(36)
第五章 通用编程	(38)
5.1 通用编程的定义	(39)
5.2 通用PC的定义	(39)

5.3	定义一个特征表	(40)
5.4	特征表的使用	(43)
5.5	判断CPU类型	(46)
5.6	判断视频类型	(54)
第六章	从8088升级到80286	(65)
6.1	8088微处理器	(66)
6.2	8088的寻址	(67)
6.3	8088中断	(67)
6.4	80286中央处理单元	(68)
6.5	80286的执行单元	(69)
6.6	80286中断	(70)
6.7	保护方式	(70)
6.8	不同之处	(73)
6.9	结束语	(73)
第七章	对AT保护方式的实验	(74)
7.1	PC机和AT机的差别	(74)
7.2	段寄存器	(74)
7.3	描述符表如何工作	(75)
7.4	进入保护方式	(76)
7.5	程序实例	(78)
7.6	实验	(80)
7.7	结论	(81)
第八章	PC/AT保护方式下的程序	(86)
8.1	处理器综述	(86)
8.2	保护	(88)
8.3	保护方式下的程序	(89)
8.4	初始化	(91)
8.5	在保护方式下执行	(92)
8.6	故障处理程序	(92)
8.7	扩充建议	(93)
第九章	286/386 保护方式编程	(120)
9.1	操作方式	(120)
9.2	段描述符	(121)
9.3	行为问题	(122)
9.4	敏感指令与特权指令	(123)
9.5	为长远利益而编写良性程序	(126)
第十章	80286和 80386 编程	(126)
10.1	更高的时钟速度	(126)

10.2	每条指令花费更少的时钟周期	(127)
10.3	流水线	(127)
10.4	新指令	(128)
10.5	80386的速度提高	(129)
10.6	80386上为什么需要汇编程序	(130)
10.7	更好的80386程序	(130)
第三篇	使用更大的存贮器：扩充存贮器和虚拟存贮器	(131)
第十一章	虚拟存贮器和虚拟机器	(131)
11.1	虚拟存贮器模式	(132)
11.2	80286 VM 模式	(133)
11.3	80386 VM 模式	(135)
11.4	虚拟存贮器的作用	(139)
11.5	虚拟机器	(140)
11.6	总结	(141)
11.7	为什么没有80386虚拟机器	(141)
第十二章	Intel 80386上的 存储器寻址	(141)
第十三章	LIM EMS 4.0：下一代扩展存贮器的定义	(145)
13.1	新的特征集	(146)
13.2	EMS 4.0 的改进	(146)
13.3	页框	(147)
13.4	新的EMM 参数	(147)
13.5	EMM函数	(147)
13.6	程序函数	(148)
13.7	环境函数	(150)
13.8	程序实例	(151)
13.9	获取EMS 4.0	(162)
第四篇	操作系统第一部分：DOS、环境和UNIX	(163)
第十四章	386 操作环境	(163)
14.1	并发传统	(164)
14.2	模块系统	(168)
14.3	虚拟环境	(172)
14.4	多任务工作台	(174)
14.5	多应用程序现状	(176)
14.6	80386上的虚拟机	(176)
第十五章	Microsoft Windows/386创建一个虚拟计算机环境	(177)
15.1	Intel回顾	(178)
15.2	四种方式	(180)
15.3	设备虚拟化	(180)

15.4	用户接口	(181)
15.5	Windows/386 如何工作	(182)
15.6	Windows/386 多任务	(184)
15.7	许多机器、一个DOS	(184)
15.8	通信	(185)
15.9	未来预兆	(186)
第十六章	操作系统的选择	(187)
16.1	特征范围	(187)
16.2	流行的方法	(191)
16.3	最佳选择	(192)
第十七章	UNIX 与 80386 结合	(193)
17.1	地址转换	(193)
17.2	页面故障	(194)
17.3	内存保护	(195)
17.4	分段与非分段寻址	(195)
17.5	分离或组合用户和内核地址空间	(195)
17.6	谁在上面	(196)
17.7	用户和内核段的布局	(196)
17.8	内核片段 (Section)	(196)
17.9	用户程序片段	(197)
17.10	内存管理数据结构	(198)
17.11	进程和片段	(198)
17.12	页表和页帧管理	(199)
17.13	在内存和磁盘上的页面	(199)
17.14	页面高速缓存	(200)
17.15	工作原理	(200)
17.16	美满的结合	(201)
第十八章	Intel 386 UNIX和DOS软件相结合	(201)
18.1	支持虚拟机器	(201)
18.2	V86加速了DOS执行	(202)
18.3	VM监视器转换8086 调用	(202)
18.4	如何建立一个VM 监视器	(203)
18.5	中断处理程序指挥交通	(203)
18.6	转换程序操作函数	(205)
18.7	仿真程序模拟外设	(205)
18.8	虚拟机器加速了执行	(205)
18.9	80386作为一个 UNIX 工具	(205)
第十九章	在移植到Microsoft System V/386时的编程考虑	(206)

19.1	小模式程序	(208)
19.2	大模式和中模式	(208)
19.3	大代码 (Large Code)	(209)
19.4	大数据 (Large Data)	(209)
19.5	混合模式	(209)
19.6	32位编程	(209)
19.7	新XENIX应用程序	(210)
19.8	应用程序大小	(211)
19.9	可移植性	(211)
19.10	XENIX和UNIX市场	(211)
19.11	性能	(212)
19.12	要求换页和虚拟存贮	(213)
19.13	结论	(215)
第五篇 操作系统第二部分: OS/2 基础		(216)
第二十章 未来的体系结构		(216)
20.1	硬件环境	(217)
20.2	多任务运行的优势	(219)
20.3	存贮器管理	(220)
20.4	DOS 兼容性	(223)
20.5	I/O 服务	(225)
20.6	动态连接	(225)
20.7	灵活的驱动程序	(226)
20.8	定时规范	(229)
20.9	为虚拟存贮器编码	(234)
第二十一章 OS/2 程序员入门		(235)
21.1	OS/2 的关键特性	(235)
21.2	应用程序接口	(236)
21.3	带有优先级的多任务	(237)
21.4	交互过程通信	(238)
21.5	存贮保护和虚拟存贮器	(238)
21.6	动态连接	(239)
21.7	与 MS-DOS 的兼容性	(239)
21.8	一个简单的OS/2 应用程序	(240)
21.9	建立应用程序	(241)
21.10	建立一个家族应用程序	(243)
第二十二章 OS/2 应用程序家族		(244)
22.1	“老的” MS-DOS 应用程序	(244)
22.2	家族应用程序	(245)

22.3	内核应用程序	(246)
22.4	Windows/Presentation Manager应用程序	(246)
22.5	工具	(247)
22.6	一个示例的内核应用程序	(248)
22.7	汇编并连接DUMP.ASM	(249)
22.8	编译和连接DUMP.C	(249)
22.9	移植一个程序	(249)
第二十三章	OS/2 多任务: 利用80286的保护方式	(265)
23.1	进程	(266)
23.2	线索	(267)
23.3	屏幕组	(268)
23.4	OS/2 程序设计	(269)
23.5	管理线索	(270)
23.6	小结	(272)
第二十四章	多任务	(272)
24.1	会话 (Session)	(274)
24.2	进程 (Process)	(274)
24.3	线索 (Thread)	(276)
24.4	调度线索	(277)
24.5	划分应用程序	(278)
24.6	进程间通信	(279)
24.7	信号量	(279)
24.8	管道	(283)
24.9	共享存贮器	(284)
24.10	队列	(284)
24.11	信号	(285)
第二十五章	OS/2 虚拟存贮管理	(287)
25.1	动态连接和共享	(291)
25.2	一些使用忠告	(294)
第六篇	操作系统第三部分: 移植到OS/2 和管理程序的描述	(295)
第二十六章	MS-DOS程序的兼容性和转换	(295)
26.1	兼容方式	(296)
26.2	与设备打交道	(297)
26.3	80286的局限	(298)
26.4	FAPI	(299)
26.5	双态设备驱动器	(300)
26.6	新工具	(300)
26.7	设备监视程序	(301)

26.8	折衷方案	(302)
26.9	OS/2 API 调用	(303)
第二十七章	DOS程序转换成OS/2 保护方式	(306)
27.1	程序员应掌握代码	(306)
27.2	家族方式考虑	(318)
27.3	综述	(318)
第二十八章	移植到 OS/2	(318)
28.1	转换的准备	(319)
28.2	到C 的转移	(320)
28.3	一个人的工作	(321)
第二十九章	管理程序: Paradox 趋近 Windows 2.0	(324)
29.1	优良的设计便于移植	(324)
29.2	为什么是Windows呢?	(324)
29.3	好消息	(325)
29.4	自然配合	(325)
29.5	图形交互	(325)
29.6	实现优势	(326)
29.7	调试困难	(326)
29.8	用户优势	(326)
29.9	性能优点	(327)
29.10	消除顾虑	(327)
29.11	Microsoft OS/2 下的Paradox	(328)
第三十章	OS/2 Windows管理程序, Microsoft Windws的未来	(329)
30.1	保护方式	(330)
30.2	用户接口	(330)
30.3	API	(331)
30.4	未来	(332)
第七篇	286/386程序设计	(335)
第三十一章	80386操作系统示例	(335)
第三十二章	80386软件开发	(361)
32.1	为什么总要被386所干扰	(362)
32.2	80386 DOS运行时的环境	(362)
32.3	几点看法	(363)
32.4	A.I.Architects OS/386	(364)
32.5	IGC X-AM开发环境	(364)
32.6	Phar Lap 386 IDOS扩展程序	(365)
32.7	MetaWare High C 和 High C-386	(366)
32.8	LIP FORTRAN	(368)

32.9	MicroWay NDP FORTRAN-386	(369)
32.10	SAIC FORTRAN-386	(369)
32.11	A.I.Architects 的海明板协处理器	(370)
32.12	外部性能	(371)
第三十三章	开发80386应用程序	(371)
33.1	编译程序	(372)
33.2	编译程序的性能	(373)
33.3	连接程序	(374)
33.4	环境	(374)
33.5	硬件	(376)
33.6	小结	(376)
第八篇	协处理器	(376)
第三十四章	外围芯片增加了80386机的设计选择	(376)
34.1	数学协处理器加速了浮点运算	(377)
34.2	DMA控制器负责数据传送	(379)
34.3	系统硬件实例	(381)
34.4	软件兼容增加了选择机会	(381)
第三十五章	数学协处理器加速程序运行	(382)
35.1	开发工具	(383)
35.2	协作系统	(384)
35.3	数据类型	(384)
35.4	寄存器	(385)
35.5	异常	(388)
35.6	指令	(388)
35.7	实方式与保护方式	(389)
第三十六章	80387及其应用	(394)
36.1	外部体系结构	(395)
36.2	接口协议	(397)
36.3	内部体系结构	(403)
36.4	特性和主要的改进	(403)
36.5	应用程序	(404)
36.6	程序设计实例	(409)
第三十七章	80387协处理器程序设计	(409)
37.1	体系结构概述	(410)
37.2	80387中的新指令	(412)
37.3	扩展操作数范围	(413)
37.4	舍入控制	(414)
37.5	异常操作	(414)

37.6	异常的类型	(415)
37.7	程序设计实例	(416)
37.8	处理器初始化和控制	(418)
37.9	80387性能	(420)
第三十八章	如何得到更好的浮点结果	(420)
38.1	精度及其保证	(421)
第三十九章	解决协处理器的瓶颈问题	(427)
39.1	瓶颈限制了性能	(428)
39.2	等待时间短的数据通路	(428)
39.3	系统的地址总线一览	(429)
39.4	产生指令	(430)
39.5	多寄存器组	(430)
39.6	编译的一个例子	(431)
39.7	设计高质量程序来提高系统性能	(434)
39.8	简化	(434)
第四十章	PC 图形学的新问题	(434)
40.1	两个例子	(435)
40.2	图形优化	(435)
40.3	新技术是需要的	(439)
40.4	通信和同步	(439)
40.5	黑箱	(440)
40.6	处理文本	(441)
40.7	虚拟显示	(441)
40.8	下一代	(442)
40.9	结束语	(443)
第四十一章	高性能图形: Intel 82786	(443)
41.1	BIU控制	(444)
41.2	显示处理器	(446)
41.3	图形处理器	(451)
41.4	局部存贮器	(453)
41.5	从属 (Slaves) 和主控 (Masters)	(453)
41.6	多任务的中心	(454)
第四十二章	82786图形芯片的内部结构	(455)
42.1	图形协处理器的系统结构	(455)
42.2	图形处理器	(456)
42.3	显示处理器	(456)
42.4	内存存取优先级	(456)
42.5	处理位图	(458)

42.6	图形内存	(458)
42.7	程序员模型	(459)
42.8	光栅操作	(459)
42.9	未来影响	(460)
42.10	窗口管理	(460)

附录 IBM PC 系列BIOS比较

前 言

微处理器问世已久，现在已经演变了好几代。回顾到1975年，那时对微处理器芯片进行程序设计就意味着从小小的寄存器、基本指令和狭小的存储器空间中去获得很多的东西。几年以后的Z80，6800和6502时代使这种限制有所拓宽。接着IBM PC采用了Intel 8080家族的微处理器，使得8088/8086编程对工程师们来说成为容易的事。8088和8086与前几代产品相比没有什么不同，但它们确实能使用更大的存储器及16位操作。（被DOS与操作系统限制到640K的1MB的8086存储器空间，曾经被看作很大了，特别是与CP/M系统的64K存储空间相比更是这样。在Apple Macintosh上使用的68000芯片具有类似的改进，尽管其存储空间非常大，达到16 MB。）具有如此大的存储器及功能强的指令，程序设计者可设计出功能强大的程序。然而，几乎所有的微处理器及微型计算机软件都与单任务的直接和简单的领域及物理存储器空间的准确使用紧密相联。

随后80286芯片诞生了，这是Intel 8080家族的下一代产品。最初它被作为一个快速的8086用在IBM AT及兼容机上。不仅如此，80286还包括了从小型机和大型机上移植来的高级的新方式及存储器保护硬件。由于和8088/8086这代产品相兼容的吸引力太大了——这些芯片具有世界上最大的软件库，上述功能并没有马上被使用。使80286运行8086软件及它自身的软件或更高级的软件并不简单。这时Intel公司公布了下一代芯片——80386，它采用了80286的先进功能，增加了更快的速度、更多的小型机技术及8086的兼容性。同时，Microsoft 公司开始公布了实现80286先进特性的新操作系统OS/2的细节和范例。

（DOS操作系统在IBM及其竞争者中实现的8088/8086/80286系统中占有统治地位，对此Microsoft具有很大功绩。由于这一原因，OS/2在市场上比任何参加竞争的操作系统都更有来头。事实上，最普遍的竞争者是UNIX，Microsoft公司也开发了微型机上UNIX最常见的变种——XENIX）。

对80286进行程序设计可用几种不同方式。可以编写8086这代产品的程序，然后在更快的80286上运行。也可使用80286的指令编写程序，其保护方式有更大的存储器空间及进行多道程序设计。本质上，这种选择意味着为DOS（以及与其有关的环境如Windows）和OS/2（以及有关的表示管理环境）编写程序。

为80386编写程序又是另一回事了。你可以再把它当作一个非常快速的8088/8086来使用并为DOS编写程序，或把它在某种程度上假设为一个80286，并在OS/2下编写程序。但80386的功能远不止这些。它具有8086软件的特殊兼容方式，可以同时运行若干个8086程序。（不幸的是对80286没有这样的兼容方式。）DOS的某些环境如Windows/386具有这种先进方式。80386具有足够快的速度及存储器保护硬件去处理巨大的存储空间，甚至更巨大的“虚拟存储器”，这里，物理存储器在逻辑上被放大到其自身容量的成千上万倍。将速度和虚拟存储器相结合，80386是多任务（同时运行一个以上程序或一个以上操作系统，在这些程序和操作系统之间共享处理器）的理想工具。

在286/386时代对编程增加的另一项革新是协处理器。这些专用的处理器芯片曾经在

某种程度上很稀少——只被用于神秘的应用软件，并由一些专门的代码编制人员设计程序。现在，许多计算机依靠算术协处理器、图形协处理器、甚至存储器管理协处理器来分担微处理器曾经担任完成的繁复的特殊任务。Intel为80287及80387设计了一系列的协处理器，包括80287和80387算术芯片，82786图形协处理器，82385高速缓存控制器，及82380 DMA控制器。其它公司，如Weitek，也模仿Intel公司的作法设计它们自己的协处理器。

为80286或80386编制程序有如为8086编制程序那么快速、容易，又如为一台价值上百万元的大型机编写操作系统那样复杂。这两个芯片具有很多共同点，但也有许多显著的不同点。当然，不论你使用哪种芯片，都应有一本入门书——实际上，你应有一本Intel公司自己的286或386的文献，现在286和386书籍充斥着书店的货架。在这里，书中描述了芯片的寄存器、指令系统、存储器寻址及其它基本功能。（Intel还提供了面向系统程序方面的某些书籍。）

然而，我认为你还应该有一本这样的书，对80286/80386执行程序设计时，在环境、操作系统、技巧、线索、易犯的错误方面都给予真实观点的书。我查遍了计算机技术方面的杂志，尽最大可能从Intel公司获取文献，并恳请286/386专家介绍他们为新一代芯片进行程序设计时所希望知道资料。本书的编写人员有的来自象Lotus、Microsoft这样的大公司，也有的来自小公司或咨询部门。在阅读和比较完所有的文件之后，我对它们进行了筛选，剔除了其中太老或重复的文章。其中有些文章与其它文章相重复，但它们从不同观点，不同重点、解释、细节方面提供了不同的信息。

我们并不希望你从头到尾地阅读这本书，虽然当你从入门开始向更专业化材料阅读时，可能会在每个章节中阅读几篇文章。当然，这是本好书，我们希望保留这本书。书中列举了一系列的暗示、对重要的存储器地址、管理或测试芯片方式的代码段、对操作系统概念的解释。所有这些，若无此书将只好到许多昂贵的杂志及书籍中去寻找。

开始的几篇文章描述了80386的历史、概况和结构。接下的一个章节主要叙述了怎样编写在8088和80386环境中的可移植程序。接下的第一章叙述的是存储器——DOS下的虚拟存储器、80386寻址及扩展存储器。

接下的章节是有关操作系统（如OS/2、DOS、XENIX、UNIX、VM/386，及Concurrent DOS）及环境（如Windows、Desqview、PC-MOS/386）的。对OS/2的叙述居多，原因是许多开发人员将会把程序从DOS移植到OS/2中。书中叙述了从DOS向OS/2以及从Windows向Presentation Manager移植程序的经验，同时还解释并用图形表达了386中多任务和特殊操作系统的优点。

下面一系列文章叙述了当前386开发的实用工具——用来将DOS程序转换以达到其存储器最大空间和386软件本机调试方案的工具。

最后对协处理器，尤其是80287及80387算术芯片，进行了剖析和比较。接下来是利用这些协处理器功能的方案和对82786图形协处理器和82385高速缓存控制器的描述。由于大部分的80286及80386程序将在IBM PC及PS/2和有竞争能力的兼容机上运行，因此本书的末尾给出了这些计算机BIOS基础相比较的图。要想设计出功能强大而且既实用又可移植的程序来，懂得BIOS中更改之处及286/386的知识是至关重要的。

第一篇 80386的结构和功能

第一章 对386的剖析

从几个方面来看,基于386的计算机都象好酒。对于没有入门的人来说,他们看来都一样。这种感觉的来源是现在市场上对廉价的而几乎是单一品种的、基于8080的XT系列机及配备80286 AT兼容机的大量需求。当今的复印纸工程的先进技术几乎可以保证能用一种AT机的标签来代替另一种标签(或因缺乏某种标签而复印更多的标签),却几乎不需改变操作或性能。

的确象酒一样,配备了80386的计算机之间显著的差别是瓶子的形状——不同的底座设计,它可能倾向于安放一种磁盘驱动器而不是另一种。但重要的是内在的东西,对计算机来说,设计的微妙差别可以带来实实在在的效果——不仅仅体现在做事的速度上,更体现在能做哪些事情上。

随着Microsoft Windows386的出现,这一点便可以理解了。在其公布的一开始,这个新的操作环境仅仅和基于80386的Compaq计算机系列相兼容(引自Microsoft)。更通用的版本以后将公布。

计算机之间的某些差别是可以感觉到的。比如某些PC机之所以畅销是建立在 其 商标名的基础上的。将IBM PS/2 型号80或Compaq Deskpro 386/20放在你的案头,说明你能买得起最好的计算机。

机器的商标名及其名声还可以带来实实在在的可以看得见的利益。例如,厂商的一部分声望依赖于对支持处理速度和制造质量的许诺以及简单的保证。一个著名的公司不可能倒闭而使你拿着一袋子功能不好的微处理器。

没有一个基于80386的PC机在各方面都名列第一的。比如在同一台机器上,你不可能既获得高性能又获得低价格。所有实用的计算机都不可避免地体现某种折衷。

由于需要将性能提高到极限的价格较高,所以,速度较快的计算机的价格也更加昂贵。(速度与价格的关系不可能是成正比的,也不是说越昂贵的计算机其速度就越快。)其它设备在特定情况下,可能有很独特之处,如大容量快速硬盘在网络上运行很出色,而内嵌式的通讯口对多用户安装很有利。

最后,并不是所有的人都需要一台基于80386的计算机。如果你计划用一台计算机仅仅做一件工作——比如保存书或撰写书,那么你就不需要80386的多任务功能。如果你在寻找最佳的性能价格比,那么邮购的高速AT兼容机就是最佳的选择。

但是如考虑到将来,那么配备了80386微处理器的PC机就是唯一可选择的途径了。他们不仅仅在速度上更快,而且比在IBM以前的环境中在结构上也具有优势。如果今天选择了更差的东西,那么明天将会后悔这一决定的。

1.1 观察时钟

不可否认,基于80386 PC机最大的诱惑就是其性能。这种机器具有两倍基于80286

AT的运算速度，这便意味着做一件工作前者所用时间只是后者的一半。一旦使用32位代码的应用程序设计好以后，这个性能水平将会在同样的设备上翻一番而不需做任何改动。

但就性能而论，不是所有基于80386的PC机都是等同的。尽管最新的重负载80386硬件有指望在吞吐量上达到运行普通DOS软件的普通AT机的3或4倍，但是某些系统甚至还没有达到AT机性能的两倍的水平。

这些令人惊讶的差别是无法从某一方面进行解释的。基于80386计算机的性能因时钟速度、存储器的类型和系统的结构的种类不同而不同。

总的来说，基于80386的计算机和PC机之间最明显的和最公认的差别就是时钟速度了。广告中充满了对MHz的引用。目前，在基于80386的商业PC机中所遇到的低档的机器大约为14MHz，高档的为20MHz。

这里所说的时钟与日期中的时间几乎没有关系，它以毫微秒（ns）而不是小时做单位。它的功能是保证计算机的电子思维顺利进行。它工作起来更象节拍器，打击节拍以便使机器的所有部分都协调工作。

时钟是必要的，因为计算机以非常简捷的电脉冲形式来处理信息。每个脉冲分别以线性顺序一个接一个处理。计算机将数字状态（最小的信息单位：1比特）理解为在一个特定的时刻瞬间脉冲的存在或消失。

虽然电移动很快，但也需要时间。线路中的电阻减慢了脉冲并把它拉长，因为脉冲必须克服计算机电路和元件的电阻、电抗（主要为电容）。所以，要使一个完整的脉冲从计算机的一部分移动到另一部分，就需要一个不连续的周期——在计算机的不同部分之间传递，就需要不同长度的时间。

为了使所有的脉冲同步，系统时钟为移动电荷分配了一个给定的周期，然后触发计算机的各个部件，并在时钟触发的一瞬间来测试各个部件是否接受到了与逻辑“1”或“0”相等价的电信号。一旦系统中所有的脉冲都被寄存之后，系统就可以重新开始传递信号了。

时钟跳变的速率确定了信息在系统中传递的速度。时钟的速度越快，识别和处理比特的速度将越快。不过时钟的速度不可能无限制的增加。在某一点，增加时钟的速度，可能就要求计算机在所有的脉冲到达目的地之前就寄存信息。而且，较快的时钟以较高的频率运行，而较高的频率更有可能在连接处遗漏或寄存错误的结果。这两种情况都可以引起错误。

计算机芯片被分为不同速度级并在特定的时钟速率下运行。它们的速度受耗电量及散热能力的限制，也受内部电阻和电抗的限制。太快的时钟会使芯片混乱，并使它们的工作不可靠。

时钟速率以MHz计量：百万周（或脉冲）/每秒。80386微处理器最慢的时钟速度是12.5MHz。而16或20MHz的速度就是比较快的了。这个速度一般是推荐使用的速度。一般来说，80386芯片可以比包装箱上标出的速度快，但是随着时钟速度的增加，其可靠性下降（一般是大幅度的）。

在大多数个人计算机中的时钟速度是由石英晶体薄片非常精确的振动来决定的。这个晶体或许自身在计算机系统板的金属包装里，又或许和其它电路组合在一起，放在振荡器