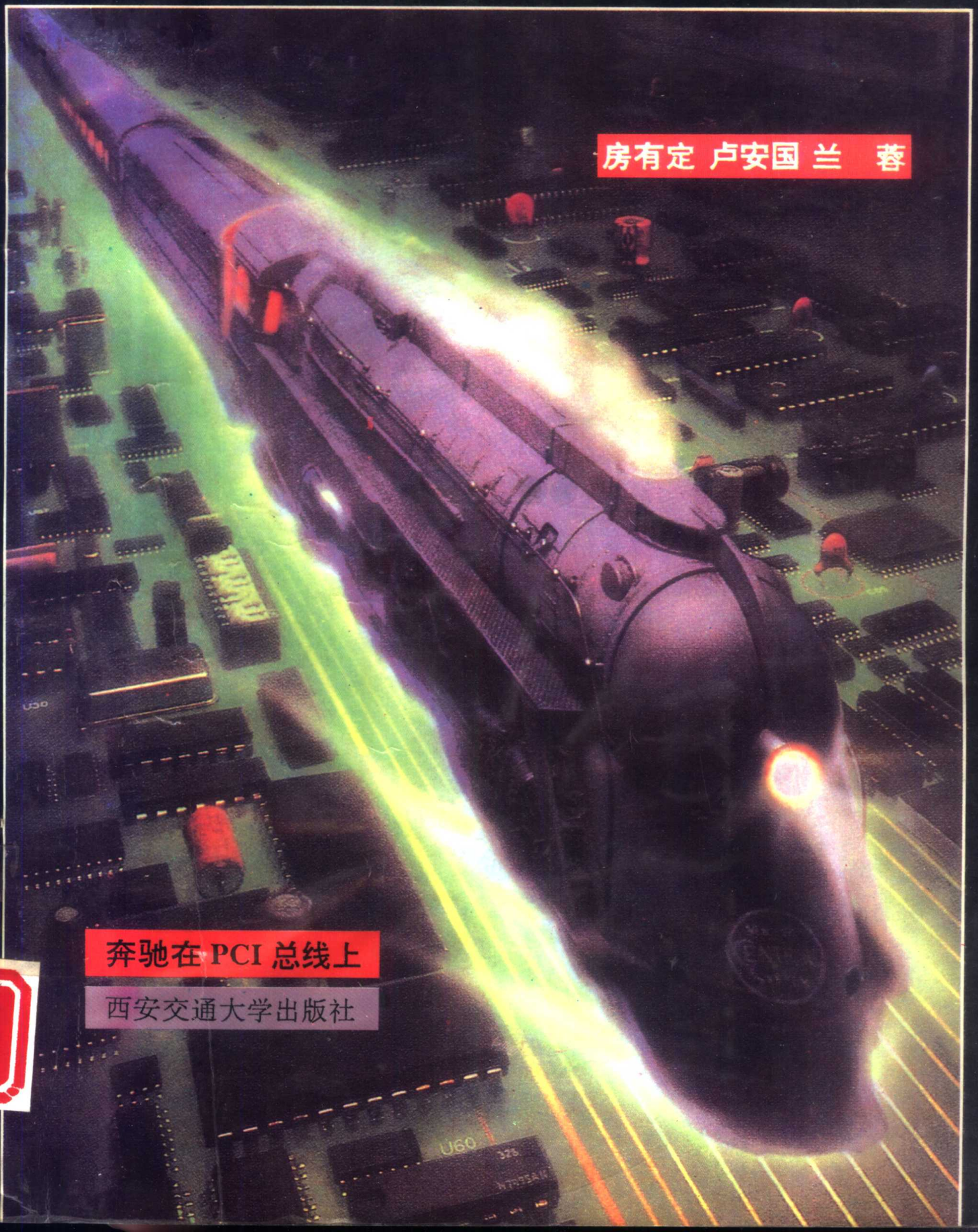
80x86 系列

保护模式高级程序设计



房有定 卢安国 兰 蓉

奔驰在 PCI 总线上

西安交通大学出版社

80x86 系列

保护模式高级程序设计

房有定　卢安国　兰　蓉

西安交通大学出版社

内 容 提 要

本书是 80x86 保护模式(x 为 286,386,486)机构分析和程序设计技术的指导性著作。全书分为上下两篇:上篇主要分析了 286 保护模式程序设计技术;下篇主要分析了 386 和 486 保护模式程序设计技术。

本书内容丰富且全面,其重点不在于系统的硬件体系结构,而在于提高系统性能的软件开发。不仅对保护模式的几个重要机构,即模式切换、存储器保护、控制转移以及高速任务切换等进行了详细的分析,而且,从软件的角度,以大量详实的程序实例,对保护模式的编程技术进行了深入的分析,这些程序实例,多属于国内首次公开。

本书作为 80286,80386 和 80486 等机器软件开发的应用技术,特别适用于从事保护模式程序开发和研究的工程技术人员,对于高等院校教学人员来说,也具有很高的实用参考价值。

(陕)新登字 007 号

80x86 系列

保护模式高级程序设计

房有定　卢安国　兰　荟

责任编辑　白居宪

西安交通大学出版社出版发行

(西安市咸宁西路 28 号　邮政编码　710049)

西安电子科技大学印刷厂印装

陕西省新华书店经销

开本:787×1092　1/16　印张:16.875　字数:409 千字

1996 年 3 月第 1 版　1996 年 3 月第 1 次印刷

印数:1 — 5000

ISBN7-5605-0810-3/TP・118　定价:16.00 元

前　言

国内外专家预言，90 年代将是 32 位微型计算机的时代，而 80386 和 80486 是当今 32 位微处理机的代表机型，采用 80386，80486 甚至 80586 的微型机已成为目前的潮流。

1978 年前后，INTEL 公司推出了 8086CPU 后，8086 就成了微型机 IBM PC/AT 的主宰。8086 是一个 16 位的 CPU，配备了 16 位的数据总线和 20 位地址总线，所以，8086 每次传送和接收 16 位数据。虽然 8088CPU 的寻址范围也为 1M 字节，但它每次只能传送和接收 8 位数据。所以，8088 仅是 8086CPU 系列中的低档机。也就是说，8086 和 8088 内部工作都采用了 16 位，但与外部联系时却分别采用了 16 位和 8 位。8086 和 8088 除数据线宽度不同外，其余大致相同。

在 8086 和 8088 问世不久，INTEL 公司就着手对其进行改进，于 1982 年推出了增强型 8086，也就是所说的 80186。它的数据总线还是 16 位，寻址能力也是 1M 字节，可运行全部的 8086 软件。不同之处在于：它增加了 10 余条扩展指令，并且将一些外围电路芯片如时钟、DMA 控制器等这样的专用部件功能集成到了 80186CPU 中。

虽然 80186 是 8086 的一步重大扩展，但在性能上，它比原来的处理器并没有显著的改进。于是，INTEL 公司又开发成功了第二代高性能 CPU80286。

80286 与 8086 和 80186 相比，它增加了一种特殊的工作模式。在这个工作模式下，80286 可以使用高达 16M 字节的内存空间；支持了虚拟存储管理功能，尽管 80286 只有 16M 字节的实际内存寻址能力，但它可以利用外存储器来模拟高达 1G 字节的虚拟存储器；增加了多任务管理功能，这就使得各个任务之间可以通过硬件机构，实现快速切换；增加了存储保护功能，加强了程序的可靠性。这个特殊的工作就是人们经常讨论的保护模式。

不幸的是，虽然 80286 有了上述的几种特殊功能，但人们仍然只能将其作为高速 8086 使用。这是因为大部分人都一直使用的是 DOS 操作系统，而 DOS 操作系统是按照当初 8086 的硬件结构设计的，所以，一旦 CPU 转换到 80286 特有的保护模式后，DOS 操作系统都将失效。但 80286 却与 8086 不同，它有两种独立的模式：实地址模式和保护模式。DOS 程序只能运行于实模式之下，无法在保护模式下运行，这是 80286 保护模式未能得到充分使用的主要原因之一。

在 80286 保护模式下，80286 提供了虚拟存储、多任务高速切换等功能，各个任务可独立运行，而不会造成任务间的相互影响。除 DOS 操作系统外，其它一些操作系统，如 OS/2，UNIX，WINDOWS 等操作系统，都利用了 80286 的保护模式和其它特性，提高系统的处理能力。

继 80286 之后，INTEL 公司又推出了功能更强大的第三代 CPU 即 80386。它是一个真正的 32 位 CPU，数据总线为 32 位，地址总线也为 32 位，寄存器得到了扩充。而且，它又进一步扩充了 80286 的保护模式功能；增加了页式存储管理功能；增加了一个特殊的保护模式，即虚拟 86 模式。在虚拟 86 模式下，多个 DOS 程序可同时运行，每个程序就像一台实际的 8086 机器，地址空间仍为 1M 字节。

80386 在内部一次能处理 16 位，而在外部一次用 32 位进行通信。另外，80386CPU 中有一个 16 位的指令预取缓冲寄存器，实行指令的预取，使得 80386 的指令执行可以以流水线方式执行，加快了执行速度。

80486 和 80386 相比，虽然在内在功能上未发生太大的变化，但在速度上却提高了许多。它将两个重要的部件：数学协处理器和管理高速内存的高速缓冲存储器加到了 80386 的电路中，使得速度大大高于 80386，但与 80386 又全兼容。所以，原来的程序不必修改，就可运行于 80486 之上。

总而言之，80286，80386 和 80486 扩展了 INTEL 系列(80186，8086，8088)的体系结构，并增加了指令条数、寻址方式和空间，对存储器进行了系统管理，可分段管理内存，还具有存储器安全保护机构。况且，80386 扩展了 80286 的分段模型，支持了高达 4G 字节的物理存储器的分段，同时又支持了以页为单位的存储管理，以满足对存储器管理的特殊要求。目前流行的众多优秀操作系统，如 OS/2 和 WINDOWS NT 操作系统，都是用了这些保护特征和能力，使系统的能力的得到了极大的提高。

然而遗憾的是，无论是在 80286，80386 或 80486 机器中，还是在使用了保护模式特性的操作系统中，如何使用保护模式特性，对于大多数使用者来说，仍然是一个不解之谜。而且，近几年来，虽然有不少有关 80x86(286，386 和 486)保护模式方面的资料或著作发表，这些资料或专著不同程度地对保护模式作了介绍，对了解保护模式起到了很大的指导作用。但到目前为止，对保护模式编程技术进行详细介绍的资料或著作仍然十分贫乏，使得人们难于很好地使用保护模式来进行程序设计。

为了揭开保护模式的神秘之处，我们特编写了这本书，奉献给需要了解和掌握保护模式编程技术的广大用户和科技工作者。

本书分上下两篇：上篇主要介绍了保护模式的始祖，即 80286 的保护模式编程技术。其中兰蓉同志编写了第 1 章～第 5 章的内容。下篇介绍的是 80386/80486 保护模式编程技术。由于 80386 和 80486 的保护模式在本质上是相同的，所以，在下篇内容中，主要围绕着 80386 的保护模式进行了讨论，但所讨论的内容同样可用于 80486，甚至 80586 机器中。

本书内容丰富且全面，其重点不在于系统的硬件体系结构，而在于提高系统性能的软件开发。不仅对保护模式的几个重要机构，即模式切换、存储器保护、控制转移以及高速任务切换等进行了详细的分析，而且，从软件的角度，以大量详实的程序实例，对保护模式的编程技术进行了深入的分析。通过这本书的学习，您可以彻底掌握保护模式程序设计的奥秘，进入到保护模式程序设计的新阶段。

在这本书的编写和修改过程中，得到了西北大学计算机系教授罗景仁老师的热情指导，同时，航天工业部骊山电子公司段茂贤研究员、靳芸生工程师对此书的编写提出了许多宝贵的建议，在此深表感谢。

由于编者水平之所限，书中内容难免有误和不详之处，望计算机方面的同仁和广大读者给予指正为感。

编　者

1996 年 1 月

目　　录

上　篇

下　篇

上　篇

第1章　80286概要及特征

1.1　80286概述

80286是INTEL公司8086系列微处理机的一种，它是继80186微处理机之后的第三代高性能16位微处理机CPU。从8086到80186的扩展主要集中于CPU内部功能的高度集成，即CPU内部增加了DMA控制器、中断控制器、16位计时器/定时器等外围功能，以及增加了对应于高级语言的指令这些方面。

与此相反，从8086到80286的扩展，主要体现在吸收并采用了多用户多任务系统中需求的虚拟存储器管理，异常保护和多任务管理等功能。也就是说，80186主要是以提高CPU内部的外围控制器的功能和降低处理器成本为目标，而80286则以提高CPU自身的功能，以适应于工作站这样的高级应用为目标。

80286处理器充分考虑了与8086，80186以及80386等系列CPU之间的兼容性。例如，对于8086，80186等低档CPU来说，80286的指令系统中，包括了8086CPU的所有指令，以及80186CPU增加的扩展指令。而且，在实模式下，通过模拟8086的功能，可以将80286作为高速8086来使用。

自从IBM公司推出了CPU为80286的PC/AT机以来，采用了80286为CPU的PC/AT兼容机如雨后春笋般地相继开发出来，并迅速得到了广泛的应用。然而，这些机器都是把80286置于实模式下作为高速8086来使用的，并没用真正地使80286本身的内在功能得到最大的发挥。为了克服这种状况，美国IBM公司开发了第一个充分利用了80286自身功能的操作系统OS/2。由于OS/2是一个充分发挥了80286内部功能的操作系统，所以，到目前为止，不适合在单任务操作系统DOS下开发的大型应用软件，都可以在OS/2下来进行开发。

1.2　80286的动作模式

80286具有两种动作模式：一种是模拟8086动作的实地址模式；另一种为完全使用了80286扩展功能的保护模式。

在实地址模式下，80286被作为高速8086来使用。也就是说，80286具有1M字节的物理地址空间和64K字节的I/O访问空间，使用段寄存器生成20位的物理地址，分段功能与8086

完全相同。而且，指令系统中也包括了 80186 的扩展指令，并且保证了二进制指令的兼容性。然而，由于 CPU 内部及总线访问采用了流水线工作方式，使得 CPU 内部处理时间大大缩短。因此，与 8086 相比，即使在相同的 CPU 时钟频率下，80286 的命令处理更为高速。

为了发挥 80286 的高级内在功能，必须把 CPU 置于保护模式下运行。在这个模式下，80286 可访问高达 16M 字节的物理地址空间，每个任务也可以访问 1G 字节的逻辑地址空间。但 I/O 访问的空间仍为 64K 字节。然而，通过使用描述符表，就可以使用虚拟地址变换和存储管理功能；通过 4 个层次的特权级别，就可以完成任务间的保护功能；通过高速任务切换和内容交换，就可以实现实时多任务管理。

在保护模式中，虽然也可以运行实地址模式程序，但必须注意下面这样的情况。在使用了两个物理地址段的程序中，当从一个段访问另一个段中的程序时，这在实地址模式下是可行的，但在保护模式下就会产生保护异常。

另外，在实地址模式下，中断向量表是固定于从 0～3FF 这段地址空间的，而且也可以直接访问到它。但在保护模式下，由于中断向量表的地址是不固定的，所以直接对中断向量表操作的程序是不可执行的。在系统复位之后，80286 置于实地址模式状态之下，当要从这个状态切换到保护模式状态，必须使能内部寄存器 MSW(机器状态字)的 PE 位(保护模式使能位)为 ON 状态。相反，当要从保护模式切换到实地址模式状态时，只能通过硬件复位来实现。

1.3 80286 寄存器构成

80286 的寄存器是在 8086 的寄存器的基础上追加了在保护模式下使用的寄存器而形成的。这些寄存器分为通用寄存器、状态/控制寄存器、段寄存器和系统表寄存器四类。

1. 通用寄存器

这些寄存器与 8086 的寄存器完全相同。也就是说，它们是由 AX，BX，CX，DX，BP，SI，DI，SP 这 8 个 16 位寄存器构成的。如图 1-1 所示，这些寄存器的作用和使用与 8086 的寄存器使用完全相同。

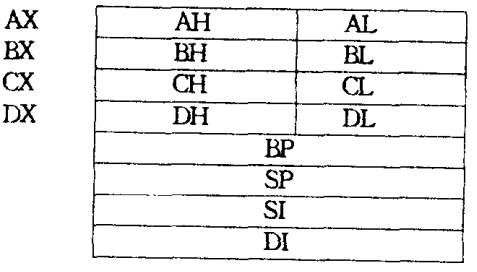


图 1-1 通用寄存器

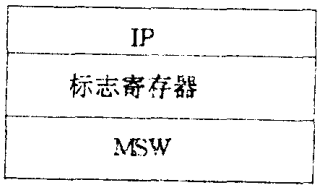


图 1-2 状态/控制寄存器

2. 状态/控制寄存器

它是由 IP(指令计数器)、标志寄存器、MSW 这三个 16 位寄存器构成的，如图 1-2 所示。标志寄存器各位的含义如图 1-3 所示。其中，2 位 IOPL(I/O 特权级)域和 1 位 NT 位(嵌套任务)是 80286 新增标志位。这些新增加位只能在保护模式下使用，在实地址模式下是没有意义

的，标志寄存器各位的功能如表 1-1 所示。

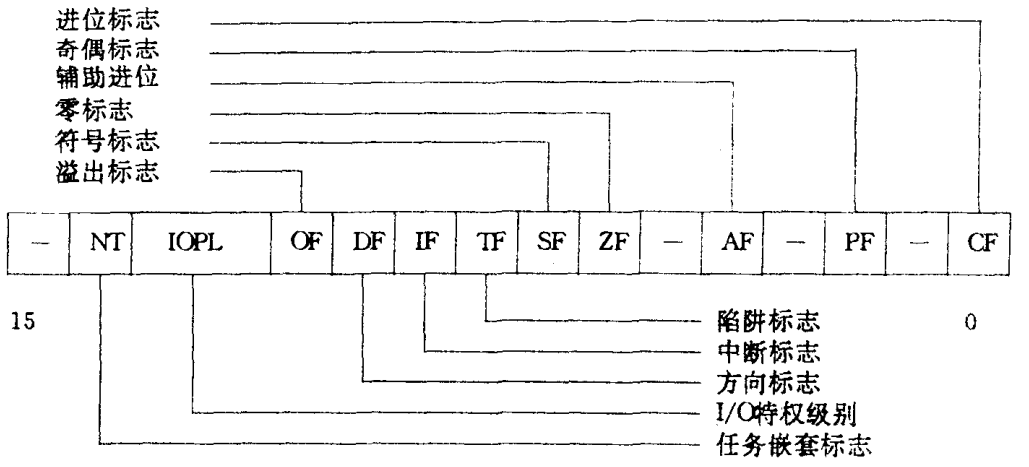


图 1-3　标志寄存器构成

表 1-1　标志寄存器各标志位的功能和含义

标志位	含　义	功能描述
0	CF：进位标志	当运算结果最高位产生进位或借位时为1，否则为0
2	PF：奇偶标志	当运算结果低8位中1的个数为偶数时为1，为奇数时为0
4	AF：辅助进位标志	当AL寄存器的低4位产生进位或借位时为1，用于BCD码运算时进行校正
6	ZF：零标志	当运算结果为0时为1，否则为0
7	SF：符号标志	当运算结果的最高位为1时为1，否则为0
8	TF：陷阱标志	此位为1时，每执行一条指令，就产生一次中断
9	IF：中断标志	此位为1时，禁止中断发生。在保护模式下，此位为被保护对象
10	DF：方向标志	当操作指令的操作方向位为0时自动递减，为1时自动递增
11	OF：溢出标志	运算结果产生溢出时为1，否则为0
12~13	IOPL：I/O特权级	IO命令的特权级别。当低于此级别的程序进行IO操作时，就产生异常
14	NT：任务嵌套标志	当CALL命令或中断引起任务切换时，此位被置为1。当此任务完成后，根据此位来判定是否应返回到切换前的任务

MSW 是 80286 新设的 16 位寄存器。如图 1-4 所示，MSW 只使用了最低的 4 位。第 0 位 PE 是在从实地址模式切换到保护模式时使用的。MP 位(协处理器监视)、EM 位(协处理器模拟)以及 TS 位(任务切换)这三个标志位在与协处理器 80287 接口时被使用，这些位的使用方

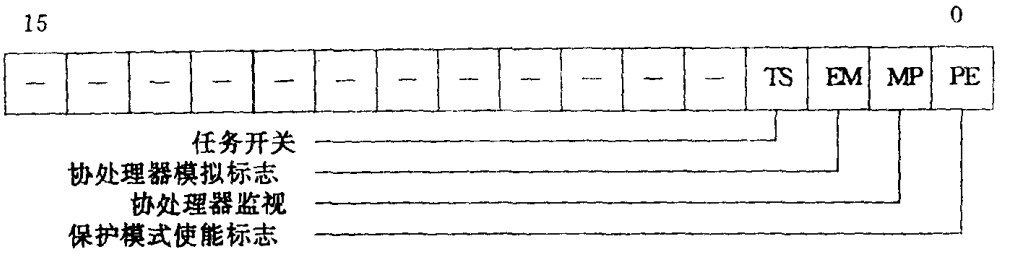


图 1-4　机器状态字

法如表 1-2 所示。MP 位表示协处理器处理器模拟)以及 TS 位(任务切 换)这三个标志位在与协处理器 80287 接口时被使用,这些位的使用方法如表 1-2 所示。M P 位表示协处理器是否存在,EM 位表示是否用软件 来模拟实现协处理器的功能,TS 位表示在多任务环境下是使用协处理器还 是用软件来模拟协处理器的功能。这对多个任务共享一个协处理器时,任务 进行判别是很有效的。这是由于协处理器 的寄存器在任务切换动作期间,并不被自动地加以保存。但 80286 在任 务切换时,自动地设置了 TS 位,表示发生了任务切换。如果在新的任务中 ,TS 位仍然为 1,这时执行协处理器命令时,就会产生软件中断。当发生 了这种软件中断后,中断程序就应将 80287 中属于前一任务的寄存器状 态保存起来,重新装入现在运行任务的 80287 状态,并清除 TS 位,然 后返回到被中断的协处理器指令,继续执行程序。

表 1-2　MSW 协处理器接口位的使用方法

TS	MP	EM	使用方法
0	0	0	复位后,与8086作用相同
0	0	1	没有协处理器存在,利用软件来模拟协处理器
0	1	0	协处理器存在
1	0	1	没有协处理器存在。在多任务环境下,使用软件模拟协处理器。这种组合可用来判别软件模拟的内容是否属于正在运行的任务
1	1	0	协处理器存在时,在多任务环境下,用于判别协处理器的内容是否属于正在运行的任务

3. 段寄存器

段寄存器与 8086 同样,有 CS,SS,DS,SS 四个段寄存器。与实地址模式下 8086 段寄存器具有同样功能的 16 位部分被称为段选择 寄存器。在保护模式下使用这个段选择寄存器,系统从描述符表中选择出所 需的描述符,如图 1-5 所示。

在 80286 的段寄存器中,还有几个特殊域:8 位访问权限域、24 位基地址域和 16 位段大小域。这 48 位组成了段寄存器的高速缓冲寄存器 ,用户程序是不能访问这些特殊域的。这些特殊域是由系统内部使用的,系 统自动完成对这些域的存取。

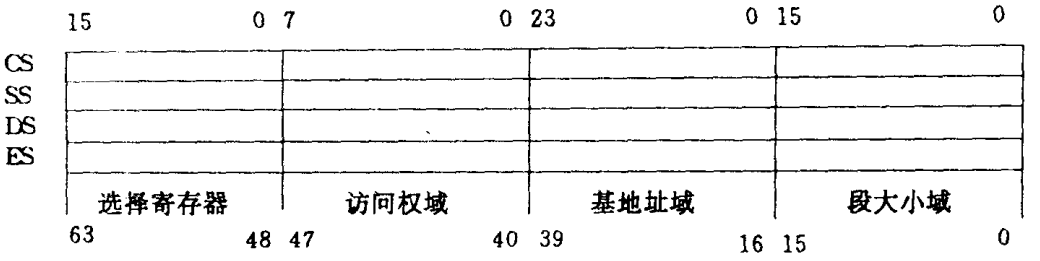


图 1-5　段寄存器的结构

4. 系统表寄存器

系统表寄存器用来管理在保护模式下使用的系统表。系统表有 GDT(全局描述符表)、LDT(局部描述符表)、IDT(中断描述符表)以及 用于存储任务内容的 TSS(任务状态段)。而且这些表分别要用 GDTR (GDT 寄存器)、IDTR(IDT 寄存器)、LDTR(LDT 寄存器) 和 TR (任务寄存器)四个不同的寄存器来指定,如图 1-6 所示。

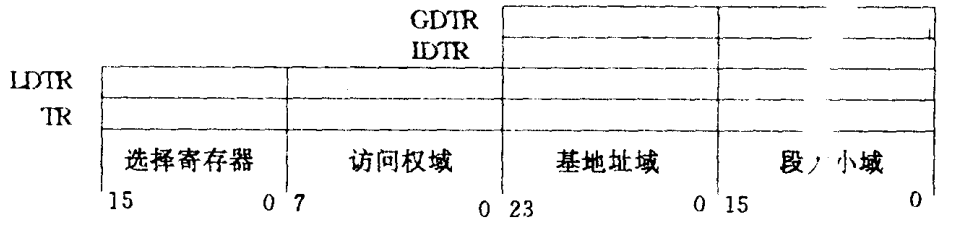


图 1-6　系统表寄存器

其中,GDTR 和 IDTR 是由 24 位基地址域和 16 位段大小域所组 成的 40 位字长的寄存器。相反,LDTR 和 TR 是由包含 8 位访问权限域 和 16 位段选择寄存器这两个特殊域所组成的 64 位字长的寄存器,它们的 构成与段寄存器的构成相同。而且,GDTR 和 IDTR 必须在转入保护模 是式之前进行初始值设定,这两个寄存器在实地址模式下可以访问。然而, LDTR 和 TR 仅可以在保护模式下使用,程序仅可以访问的是段选择寄存 器,其它域是在任务切换时由 LDT 描述符和 TSS 描述符中自动装入的。

1.4　实地址模式 80286 与 8086 的差别

正如前面提到的那样,80286 的实地址模式是高速模拟 8086 动作的一种模式。因此,在实地址模式下,80286 的物理地址空间为从 00000H～0FFFFFH 的 1M 字节。然而这个物理地址正如图 1-7 所示的那样,是把 16 位的段寄存器的段选择寄存器的内容左移 4 位以后(扩大 16 倍)而得到的值,加上 16 位的地址偏移量产生的 20 位物理地址。

指令系统分为基本 8086 指令,80186 扩展指令,80286 保护模式专用指令三种。其中,在实地址模式下可以使用的指令包扩基本 8086 指令,80186 扩展指令,以及与 GDTR,IDTR 和 MSW 有关的初始化时的装入/存取指令。80186 的扩展指令主要包括如下那些指令。

(1)16 位寄存器/存储器与 16 位立即数带符号乘法指令 IMUL；
(2)用 8 位立即数指定其移位/循环次数的移位指令；
(3)把立即数压入堆栈的 PUSH 指令；
(4)把 8 个通用寄存器内容一次压入/弹出堆栈的 PUSHA/POPA 指令；
(5)在 I/O 与存储器间进行串传送的 INS/OUTS 指令；
(6)堆栈空间分配/释放指令 ENTER/LEAVE；
(7)检查数组下标范围的 BOUND 指令。

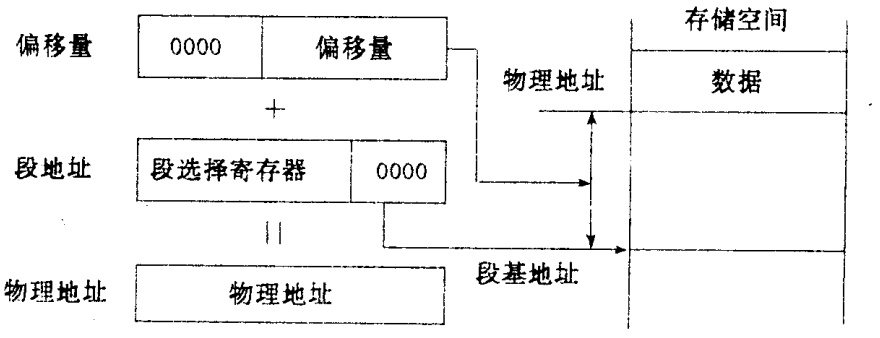


图 1-7　实地址模式物理地址的形成过程

其中，ENTER/LEAVE 指令与 BOUND 指令是与高级语言相关的指令，ENTER/LEAVE 指令在分配函数或过程所需的动态空间时特别有效。其它的 16/32 位微处理器也具备与之相似的指令，如 68000 的指令 LINK/UNLNK。但 80286 的 ENTER 指令需要指出过程或函数的嵌套层次。

标志寄存器也追加了 IOPL 和 NT 两项内容，但这两项内容仅可在保护模式下是使用。另外，可在实地址模式下使用的 MSW 寄存器也包括了与协处理器有关的标志位 TS，EM，MP 以及切换到保护模式时必须使用的 PE 标志位。

与 8086 相同，用 0～255 个向量来访问中断，执行给定的中断。向量表为从 0～3FF 的 1K 字节的内存区域，其中每个中断入口占据 4 个字节。实地址模式的中断向量编号如表 1-3 所示。中断向量编号为 5～9，13 和 16 这些中断都是 80286 新增加的中断。

下面简述一下各中断的含义：

①BOUND 检查异常中断(中断 5)。BOUND 命令执行时，如果数组的下标超出了其数组的范围时，就会产生该异常。

②无效操作码异常中断(中断 6)。无定义的指令以及仅可在保护模式下执行的指令执行时，就会引发该异常。

③协处理器不存在异常(中断 7)。MSW 的 EM 位为 1 时，发出 ESC 指令后，就会引发该异常。除此而外，当 TS 和 MP 位为 1 时，WAIT 指令也可引发该异常。

④中断表大小异常中断(中断 8)。保护模式初始化时，由 LIDT 指令指定的中断向量表的大小小于 3FF 时，就会产生该异常。

⑤协处理器段溢出异常中断(中断 9)。协处理器指令中的操作数超出了段的范围时，就会引发该异常。

⑥段溢出异常中断(中断 13)。在指令执行或存储器访问中,当段的偏移量大于 0FFFFH 时,就会产生该异常。

⑦协处理器错误异常中断(中断 16)。当协处理器产生异常状态时,80286 便可检测到与之相连的 ERROR 端的输入,此时就会产生异常。

表 1-3　实地址模式下中断向量编号

向量号	中 断 原 因	相 关 指 令
0	除法错误	DIV,IDIV
1	单步中断	所有指令
2	NMI中断	所有指令
3	断点中断	INT
4	溢出中断	INTO
5	BOUND检查异常	BOUND
6	无效操作码异常	所有未定义指令
7	协处理器不存在异常	ESC,WAIT
8	中断表大小异常	LIDT
9	协处理器段溢出异常	ESC
13	段溢出异常	所有存储器访问指令
16	协处理器出错异常	ESC,WAIT
32－255	用户自定义异常	

注:中断 10～12,14,15,17～31 都作为系统的预留中断。

第 2 章　80286 存储器管理机制

2.1　存储器管理与虚拟存储

这里所讲的存储器管理，实际上就是把所使用的程序的逻辑地址如何转换为实际存储器的物理地址而施行的机制或策略。在实地址模式的 8086 之下，逻辑地址空间的大小和物理地址空间的大小都为 1M 字节。而且，地址转换如同第 1 章中所说的那样，是由表示基地址的段寄存器和相对于该基址的地址偏移量来决定的。

与此相反，在保护模式下，如图 2-1 所示的那样，平均每个任务的逻辑地址空间为 1G 字节，如果使用了描述符这样的地址转换机制来实现地址映射之后，就可以把逻辑地址空间映射到 16M 字节的物理地址空间中去。这样，与 8086 相比，程序可以使用的地址空间要大得多，这就解决了存储器空间不足这一实际问题。

与逻辑地址空间相比，实际物理存储器的大小是很有限的。为此，把暂时不能装入内存储器中的命令和数据存储到硬盘等高速外部存储装置上，当实际需要访问这些命令或数据时，再把它们从外部存储装置上装入到内存储器中，这种技术称之为虚拟存储。以这种方法使用的逻辑地址被称之为虚拟地址。内存储器和外部存储装置之间以段为单位来进行数据交换。

在具有虚拟存储功能的系统中，对于应用程序来讲，并不需要考虑到虚拟存储功能的存在，而只需用虚拟地址来访问存储器就可以了，但不能直接访问物理地址。在 80286 中使用的描述符地址转换机构和虚拟存储功能具有类似之处。描述符地址转换机构管理的单位为可变长内存段，段的最大长度为 64K 字节。也就是说，虚拟存储是 80286 所具有的固有功能。

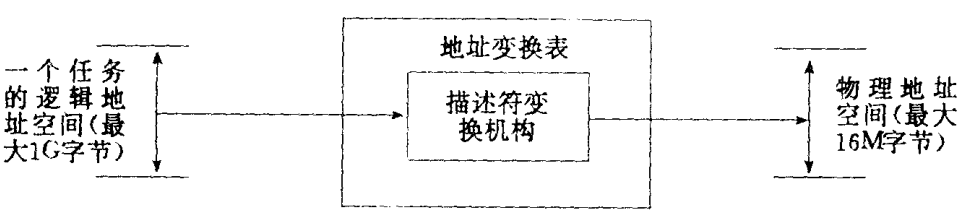


图 2-1　保护模式下物理地址的形成

2.2　描述符表与地址变换

保护模式中使用的虚拟地址是一个由 16 位地址偏移量和 16 位段选择寄存器值组成的 32 位指针，图 2-2 给出了变换到物理地址的过程。

首先，利用段选择寄存器从描述符表中选择到包含了目的数据段的段描述符。在这个过程中，必须根据描述符表的基地址，检查虚拟地址中的段选择寄存器的值是否超出了描述符表规