80386扩展内存编程

王建文　程　军　贺乐天

西安电子科技大学出版社

80386扩展内存编程

王建文　程　军　贺乐天

西安电子科技大学出版社
1994

(陕)新登字 010 号

内 容 简 介

80386 提供了 16M～4G 的可寻址空间，给用户提供了极好的运行环境。本书主要介绍如何利用 80386 所提供的内存空间进行应用程序设计。

本书共分八章。第一章介绍 80386 内存管理机制及其相关技术。第二章介绍 80386 硬件调试功能及其应用。第三章至第六章介绍扩充内存编程技术。第七章通过剖析一个 DOS 扩展器来介绍如何编写扩展内存驱动程序。第八章介绍了几个系统/应用软件使用扩充内存的情形。

本书适用于软件开发人员和广大计算机用户，也可作为大专院校计算机专业的教学参考书。

80386 扩展内存编程

王建文　程　军　贺乐天　编著

责任编辑　汪海洋

西安电子科技大学出版社出版发行

陕西省大荔县印刷厂印刷

新华书店经销

开本 787×1092　1/16　印张 24　字数 572 千字

1994 年 4 月第 1 版　1994 年 4 月第 1 次印刷　印数：1—5 000

ISBN 7-5606-0310-6/TP・0112　　定价：19.50 元

目　录

引　言

一、DOS 内存管理现状

MS－DOS 操作系统是目前市场上 PC 机的主流操作系统，PC/XT 机所使用的 8088/8086 芯片最大寻址空间为 1M 字节；更高档的微机所使用的芯片 80286 最大寻址空间为 16M 字节；80386 芯片的寻址空间更大。然而，基于这些芯片的操作系统 MS－DOS 所能管理的最大 RAM 空间却仅为 640K，也就是说，在 MS－DOS 下执行的程序受限于 640K 这个狭小的空间。这个数字与以上所列 80286/80386 芯片最大寻址空间相比可以看出，80286/80386 芯片的最大寻 址能力还远未得到发挥。

从操作系统角度看，常常需要牺牲时间来换取空间，或者以牺牲空间来换取时间，因此从根本意义上来讲，内存大小直接影响到计算机的运行速度和应用范围大小。内存越大，相应来说，程序的执行速度也越快，程序的应用范围也越大。因而如何尽可能地发挥 80286/80386 芯片的最大寻址能力便是本书所要讲述的主题。

IBM PC/XT 及其兼容机可直接寻址 1M 的内存空间，这一寻址空间又可分成若干 个区域留作专用。例如 0C0000H～0A0000H 用于视频显示 RAM，0F000H～(1M－1)H 用于系统 ROM 等。640K 以上的所有寻址空间都留作专用，但是其中有些区域还未明确规定其用途，如：0D0000H～0E0000H 的区域未被使用。0～640K 的寻址空间为用户程序 RAM。PC、XT、AT 机内存分配情况如图 1 所示。

二、什么是扩展内存/扩页内存?

一般地，我们将 80286/80386 微机的内存分为 3 种类型：基本内存(BASE MEMORY)，扩展内存(EXTENDED MEMORY)和扩页内存(EXPANDED MEMORY)。基本内存是指以 MS－DOS所能使用的自由 RAM 空间，从 0 到 640K 范围的线性地址空间；扩展内存是指 MS－DOS 应用软件通常所不能访问的 1M 字节以上的线性地址空间；扩页内存是指依据 Lotus，Intel 和 Microsoft 公司联合提出的 EMS 标准实现的页式内存，这种内存是非线性空间。

扩展内存与 80286/80386 芯片实际的寻址能力相对应，而扩页内存与芯片的实际寻址能力并无对应关系。实际上，基于 8088/8086 芯片的微机也可以使用扩页内存，使得应用程序可以使用超出 8088/8086 寻址能力(1M 字节)以上的内存空间。相应地，扩充内存的 RAM 卡必须附加额外的硬件机构。符合 EMS 标准的扩充内存卡我们称之为 EMS 卡。

以下说明符合 EMS 标准的扩页内存示例。

EMS 标准是软件和硬件设计标准的混合，它们规定了扩展内存卡如何与软件一起工作，根据硬件的特点怎样编写或修改软件。在 EMS 卡上的 RAM 不是永久的占据某一 个地址空间，而是由软件命令为 16K 逻辑页(EMS 卡上的存贮体)分配一个特定的 16K PC 地址空间。该地址空间也称为一个寻址窗口。通过使用软件开关，使扩展内存卡提供大量的内存逻辑页，使得多个不同的逻辑页对应于同一个寻址窗口，从而可以将一个大的存贮体空

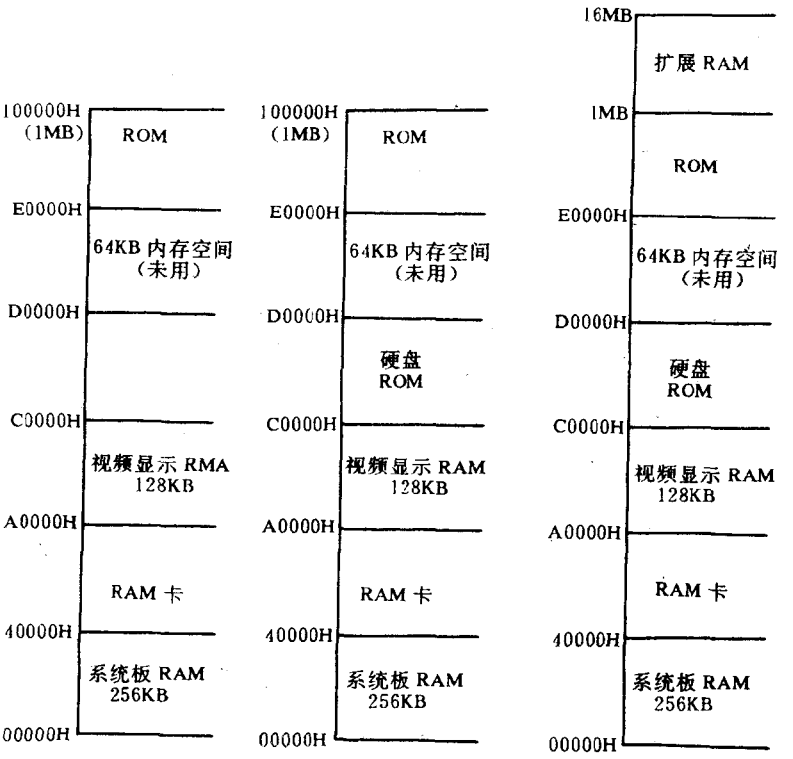


图 1　PC XT 和 AT 机的内存空间分配

间(EMS 卡 RAM) 映射到一个小的 PC 地址空间，从而达到扩充内存的目的。实际上，软体开关是 EMS 卡上的一些内部寄存器。

一旦某个逻辑页由软件命令分配了某个寻址窗口，程序对其存取就像对常规内存的操作一样。图 2 说明了 EMS 扩展卡上多个逻辑页中的一个出现在一个地址窗口上，其它的逻辑页等待的情形，EMS 要求在实际内存 1M 寻址空间上有连续的四个 16K 窗口，从而形成一个 64K 寻址空间主体，该 64K 寻址体的起始地址必须为 64K 整数倍。图 2 示例中，我们选取 0D0000H～0E0000H 未用区域作为 64K 寻址体空间。该体的四个窗口分别编号

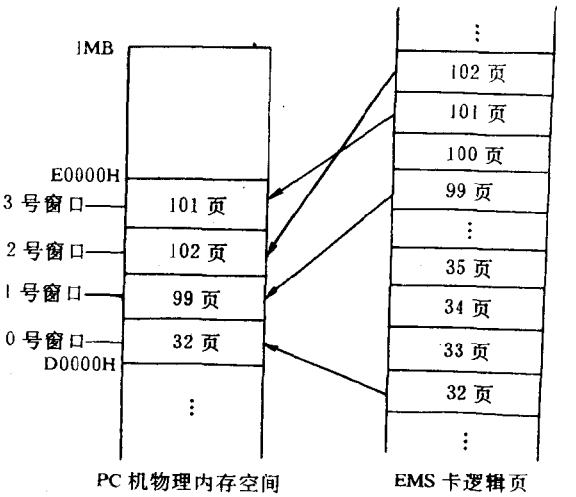


图 2　当前物理窗口与 EMS 逻辑页的映射关系

为 0 至 3 号。在 EMS 扩充卡上 的任一逻辑页都是可以出现在该寻址体的任一个窗口上的。

实现 EMS 标准的管理程序称为 EMM，EMM 实现了以上提到的扩充卡上逻辑页到寻址窗口的控制映射，而且 EMM 将 EMS 标准以 INT 67H 中断调用的方式提供给应用程序。通过加载 EMM 程序，应用程序可借助于软中断 INT 67H 直接与 EMM 建立通信，从而存取高达 16M 字节的扩充内存。

扩展存储器是地址高于 FFFFFH 或 1MB 的任何存储单元，只能由 80286 和 80386 处理器直接访问。80286 处理器有 24 位地址，可寻址到 16MB 内存，80386 处理器有 32 位地址，可寻址到 4GB，见图 3 和图 4。

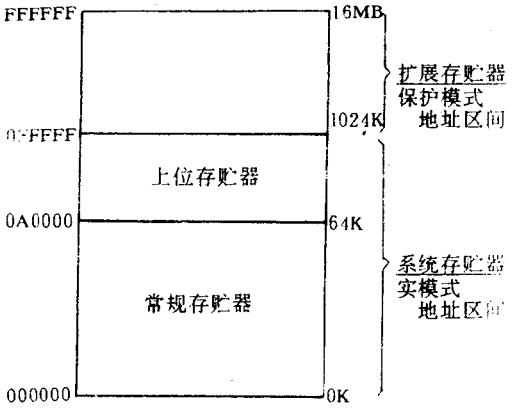


图 3　80286 PC 的扩展存储器

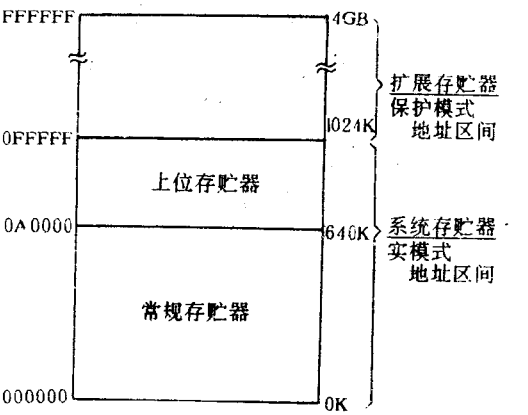


图 4　80286 PC 的扩展存储器

扩展存储器经常也被称为 XMS 存储器。Lotus，Intel ，Microsoft 和 AST 公司共同制定了 Extended Memory Specification (XMS) 2.0 (3.0 已问世)，定义了内存中三个特定区域的分配，即：高端存储区(HMA - High Memory Area)，上端存储块(UMB - Upper Memory Blocks)和扩展存储块(EMB - Extended Memeory Blocks)，见图 5。

HMA 是扩展存储器中紧挨 DOS 1MB 边界的第一个 64KB 区段(0FFFFFH 到 10FFFFH)，只能在 80286 和 80386 机器上使用．通过启动第 21 条地址线(A20)，可以在实模式下访问到。然而，HMA 必须做为一个单独的存储块处理，即不能分割共享，只能调入一个单独的程序。所以，HMA 一般用来存放尽量接近 64KB 的程序。

UMB 是上位存储器中的一些存储地址。虽然上位存储器(384K)技术上是保留的，但仍有一些未用部分可以通过专门的硬件和内存管理程序找到，构成 UMB，并在实模式下访问。而 EMB 是从 10FFFFH 以上的扩展存储器中分配的，只能在保护模式下访问。

在 80286 或 80386 PC 上使用扩展存储器，首先必须增加物理内存。为使性能更好，应把扩展内存与系统内存物理上一起放在 PC 母板上。如果母板上没有额外的存储器位置，可加一块内存适配板，并将其一部或全部配置成扩展存储器。然后，安装一个独立的扩展内存驱动程序，如 HIMEM .SYS 或 QEMM .SYS 等，用来管理系统或应用程序对扩展内存的访问。各种扩展内存驱动程序之间可能互不兼容。

在保护模式下 80286/80386 芯片引入了许多全新的概念：访问权限检查(读写保护)，地址限界检查，多任务切换等。80286/80386 芯片是为支持多任务操作系统而引入这些概

念和管理机制的。目前，在微机上实现运行的多任务操作系统有 OS/2，XEMX，或 UNIX System V（如 Release 4.0 版）等。在这些操作系统下应用程序实际上可以直接运行在扩展内存上。即使是在 MS－DOS 任务操作系统下，我们也可以通过指令来切换到 80286/80386 保护模式下，使用保护模式下的程序设计方法和工具来自由地使用扩展内存。这些程序设计方法和工具正是本书所要讲述的核心。我们将在书中给出全部工具软件的源程序代码。目前，在 DOS 下运行的许多实用程序，如高速磁盘缓冲存储，虚拟磁盘和打印缓冲等实用程序均是使用本书所提到的一些原理和类似方法来实现对扩展内存的访问。

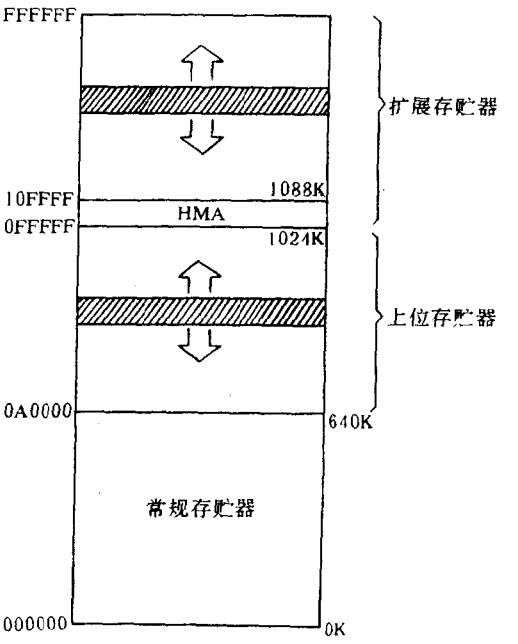


图 5　XMS 分配区域

另外，值得一提的是，如果读者已具备了并发、分时、多任务等操作系统概念，本书所介绍的内容将有助于您在 80286/80386 微机上编制出一个小型的多任务操作系统，如果您还具有某些非凡的天才（如组织管理能力），这个操作系统也许会发展成为商用性质的，从而得到推广。

三、本书内容组织

全书共分八章，第一章主要从原理方面介绍 80386 内存管理机制及其相关技术，比较系统地讲述了 80386 芯片所实现的保护模式及其地址变换机制，任务切换及保护机制，其中详细探讨了保护模式下最重要的数据结构——描述符(descript)，最后简要介绍了 V86 模式。

第二章介绍了 80386 硬件调试及其应用，并分别给出了用汇编语言和 C 语言编写的调试工具程序范例。80386 硬件调试功能可以使用在保护模式下。

第三章全面介绍了扩页内存规范 EMS，讲述了 EMS 原理、EMS 检测方法、常用 EMS 命令等内容，最后给出了一个在 EMS 环境下的程序设计实例——DUP

第四章至第七章逐次深入地讲述了扩展内存的使用方法，从单纯访问扩展内存时使用的 BIOS 调用方法到 DOS 扩展器(extender)源代码举例，系统地为读者介绍使用扩展内存编程的方法和工具，这部分内容也可以称为全书的主干。

最后一章简要介绍了几个扩页/扩展内存实用软件。

附录 I 至 IV 系统，分类列举了全部的 80386 指令集。

第一章　386保护模式及存贮管理机制

1.1　概　　要

第一章是以后各章的理论基础。在这一章里，我们着重介绍80386不同于8086/8088的内存访问方式及386保护模式，为大家以后阅读程序之便，还将80386扩展指令集做为附录列于书后。

386共有三种不同的内存管理模式：实地址模式(real address mode)，保护模式(protected mode)和虚拟8086模式(virtual 86 mode)。本章首先对读者比较熟悉的实地址模式做简单回顾，最后对虚拟8086模式一带而过，重点放在对保护模式及其寻址机制的详细描述上，使读者能够对386保护机制有一个清晰完整的概念。

本章内容如下：

(1) 80386实地址模式；

(2) 保护模式及其地址变换机制；

(3) 描述符；

(4) 任务切换及保护机制；

(5) V86模式。

1.2　80386实地址模式

当80386开机启动时，便自动进入实地址方式．用户可以一直运行在实模式，也可以通过软件指令切换到保护模式。80386实模式和8086/8088的寻址方式是一样的，只不过运行速度更快，内存配置更大些，可以说，在实模式下，80386是增强型的8086/8088。

在实模式下，386可以执行8086/8088基本指令集。这就是说，用户在8086/8088上运行的程序可以不做任何修改直接在386上运行。因此，80386在目标代码一级与8086/8088完全兼容。386另外增加了一些指令，目的在于增强性能，方便操作。例如，386增加了PUSH ALL (PUSHA)，POP ALL (POPA)，堆栈操作指令ENTER和LEAVE，输入/输出串指令INS、OUTS等。有了这些指令，用户可以将常用的大段代码以及大量循环操作用一两条指令代替。

80386的这部分指令称为80386扩展指令集，需强调一点，由于386使用了扩展指令集，使得用户在386上编写的程序无法在8086/8088上运行。因此大家在使用386扩展指令时需注意一点，即单纯为了编写方便是否值得牺牲可移植性？本章后面所说的程序除非特别说明，一般指的是汇编语言程序。

与8086/8088相比，80386还增加(增强)了一些寄存器，如图1.2.1所示，其中，寄存器EAX、EBX、ECX、EDX、EBP、ESP、ESI、EDI、EFLAGS分别由原来的16位扩展为32位，但原来的16位寄存器AX、BX等仍可单独使用，另外386还增加了两个数据段寄存器

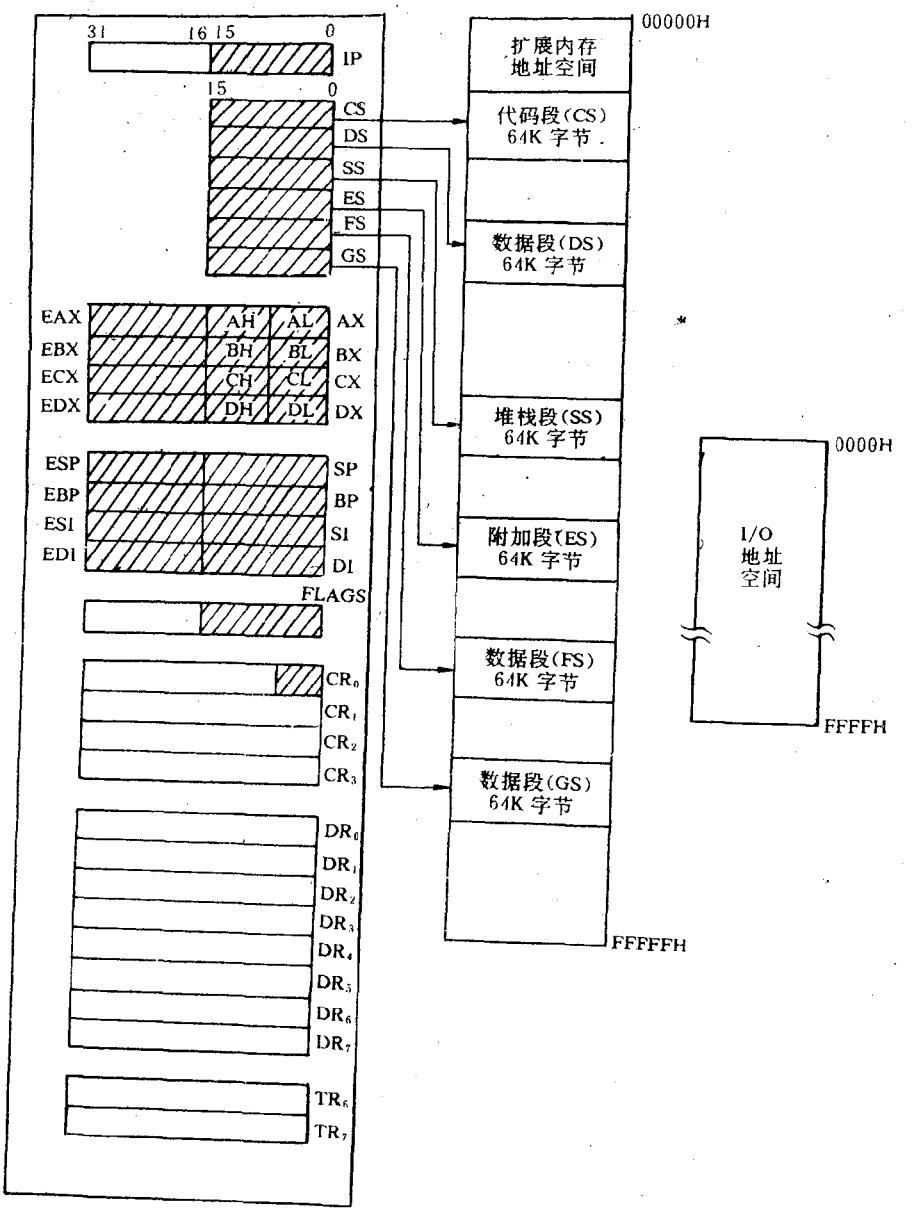


图 1.2.1　实模式下 80386 地址映射

FS 和 GS，这就是说，同时有 6 个段处在活动方式。在实模式下，386 仍使用 CS：IP 指示指令地址。从图 上可以看出，386 每段仍为 64K，寻址空间仍为 1M 字节。这是由于 386 通过段寄存器(内含基地址)和偏移量生成物理地址的方式与 8086/8088 完全相同，如 1.2.2 所示，段寄存器值左移 4 位与段内偏移量相加，形成 20 位物理地址，$2^{20}=1M$，故寻址空间为 00000H—0FFFFFH，即 1 兆字节。

80386 与 8086/8088 有相同的存贮空间和 I/O 空间，386 的 I/O 空间为 64K 字节，0000H—FFFFH，如图 1.2.3 所示。在图 1.2.3 (*a*) 中，我们可以看出，最初的 1K 字节(000H—3FFH)仍然用做中断向量表。在图 1.2.3 (*b*) 中，前 256 字节 I/O 地址空间(00～

FFH)被称做第 0 页。这些端口地址可以直接被 I/O 指令所访问。

最后，我们介绍怎样从实地址模式切换到保护模式。386 增加了四个控制寄存器，分别称为 CR_0、CR_1、CR_2、CR_3，这些控制寄存器均为 32 位，其中 CR_0 的第 0 位称为 PE 位(保护激活位)，在实模式下，PE 位置 0；当 PE 位置 1 时，系统将由实模式进入保护模式。PE 位可

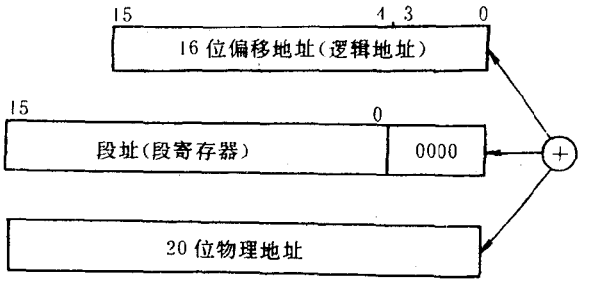


图 1.2.2　实模式下 80386 物理地址生成

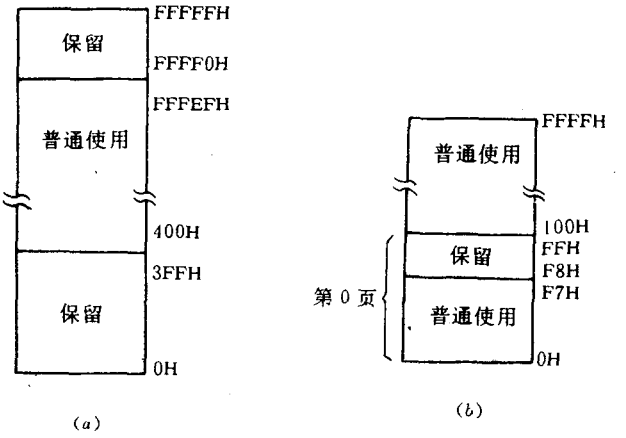


图　1.2.3

(a) 实模式下内存指定使用情况；　(b) I/O 地址空间

以通过 386 指令来改变。

1.3　保护模式及其地址变换机制

通过使控制寄存器 CR_0 的 PE 位置位，我们可以从实地址模式进入保护模式。保护模式是 386 不同于 8086/8088 内存管理模式的高级模式，在保护模式下，386 支持虚地址机制，页面机制，保护机制，以及多任务机制。在这一节里，我们将讲述保护模式下 80386 寄存器及描述符表，虚拟地址空间以及地址变换方式。

1.3.1　保护模式有关概念及 80386 寄存器

在保护模式下出现了许多不同于实地址方式的全新的概念，在此先做简要介绍：

(1) 任务(TASK)：是一个子程序的集合，这些子程序配合起来完成一个特定的功能。

(2) 描述符表(Descriptor Tables)：由多个描述符连续排列所形成的列表，其中每个描述符称为一项，描述符表长度称为限长，描述符表首项所在内存地址称为该描述符表的基地址。

(3) 描述符(Descriptor)：每个描述符在内存中占连续 8 个字节，用来描述一段内存地址及访问权限。

(5) 选择符(Selector)：作用类似于指针，用来在某个描述符表中选取某个描述符。

(6) 描述符表之描述符：一个描述符，该描述符用以指定一个描述符表。

(7) 描述符表寄存器：寄存器中的内容用来指定一个描述符表。

保护模式下所使用的寄存器模式如图 1.3.1 所示。比较图 1.2.1，我们发现保护模式下的寄存器集合是实地址模式下寄存器集合的超集。保护模式下出现 4 个新的寄存器：全局描述符表寄存器(GDTR)，中断描述符表寄存器(IDTR)，局部描述符表寄存器(LDTR) 和任务寄存器(TR)，除了这 4 个新出现的寄存器，另外有几个寄存器进行了扩展，如指令指针寄存器 IP 现在称为 EIP，由原来的 16 位增加为 32 位，标志寄存器 EFLAG 更多的位有了实际定义，控制寄存器 CR_0 到 CR_3 定义了实际的功能。接下去我们讨论这些寄存器扩展的目的及这些寄存器在保护模式下所起的作用。

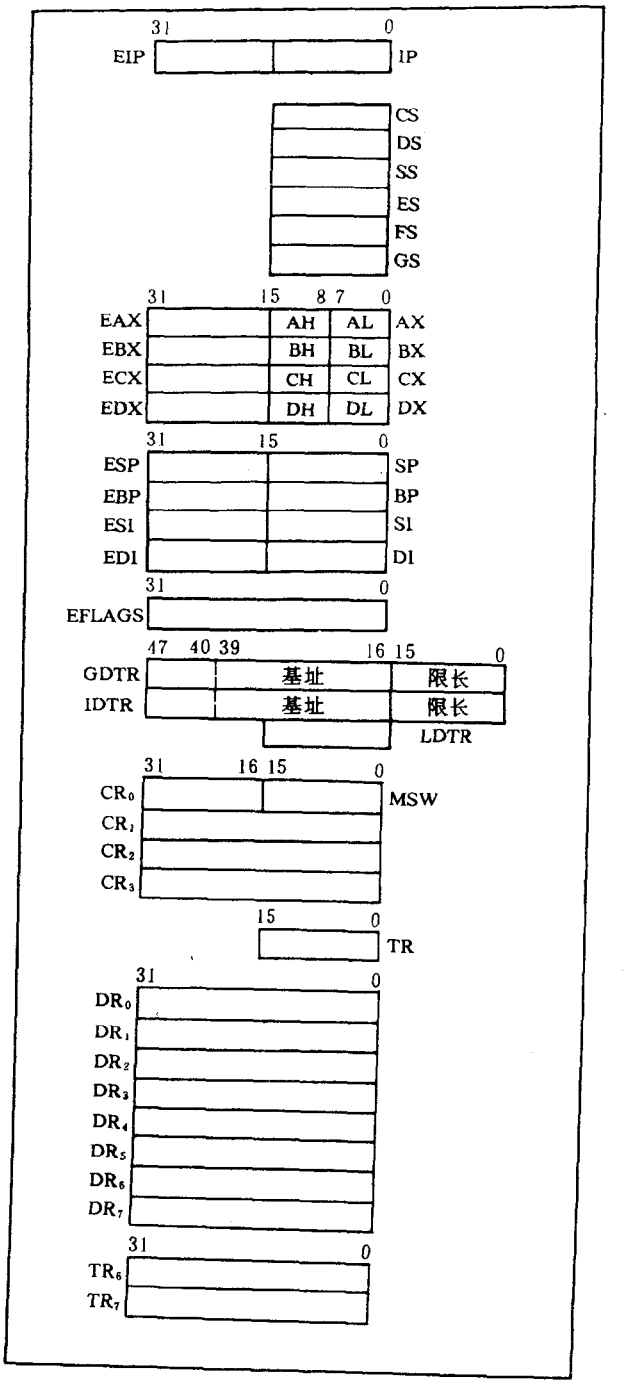


图 1.3.1　保护模式下的寄存器

(1) 全局描述符表寄存器

如图 1.3.2 所示，全局描述符表寄存器由两部分组成：基地址(32 位)及限长(16 位)，因而全局描述符表是一个 48 位的寄存器，通过基地址和限长，全局描述符表寄存器定义了一个全局描述符表(GDT)。

全局描述符表寄存器低 16 位称为限长，由这个限长所决定的全局描述符表 GDT 最大长度为 2^{16} 字

节，由于 GTD 中每个描述符占 8 个字节，故一个 GDT 最大可以包含 $2^{16}/2^3=8\,192$ 个描述符。限长不一定非取最大值，它是一个上限，只要小于等于这个上限且是 8 的整数倍即可，全局描述符表高 32 位称为基地址，它可以取 80386 4 GB 线性地址空间中的任一个地址。例如若 GDTR 值为 0700 0FFFH，则其基地址为 0700H，限长为 0FFFH，因此 GDTR 表长为 0FFFH +1＝1000H 字节，包含 1000H/8 ＝0200H 即 320 个描述符。关于线性地址、逻辑地址等概念详见下一小节。

一般地 80386 保护模式下只需有一个全局描述符表。在由实地址模式切换到保护模式之前，必须将基地址和限长装入全局描述符表寄存器，这可以通过软件指令来完成。指令 LGDT 读出 GDTR 内容，指令 SGDT 写入 GDTR 内容，有关指令详细情况参见附录。一旦装入，DGTR 的值一般不再变动。

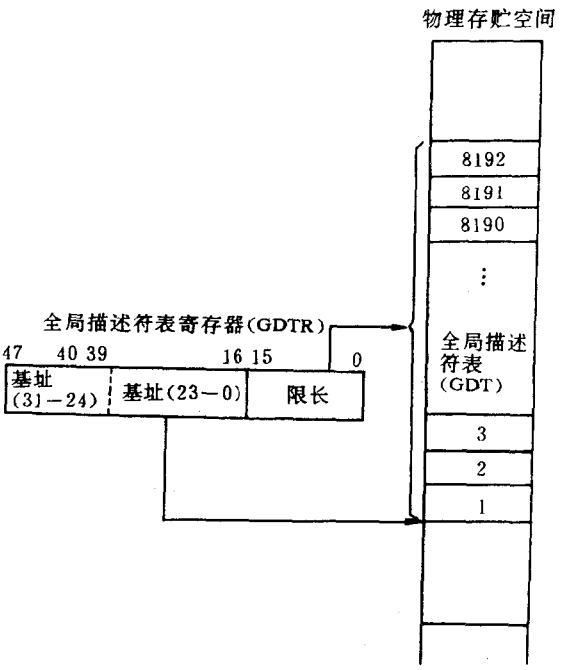


图 1.3.2　全局描述符表

(2) 中断描述符表寄存器

和全局描述符表寄存器类似，中断描述符表寄存器在内存中定义了一个中断描述符表(IDT)，所不同的是，中断描述符表中的描述符称为中断描述符，而不同于通常的段描述符。通过 IDTR 和 IDT，386 提供了一种机制使得微处理器控制权可以转交给中断服务程序或异常服务程序。

中断描述符表寄存器是 48 位寄存器，由两部分组成，即基地址(32 位)和限长(16 位)，如图 1.3.3 所示，基地址可以是 80386 4GB 线性地址空间中的任一个地址，限长

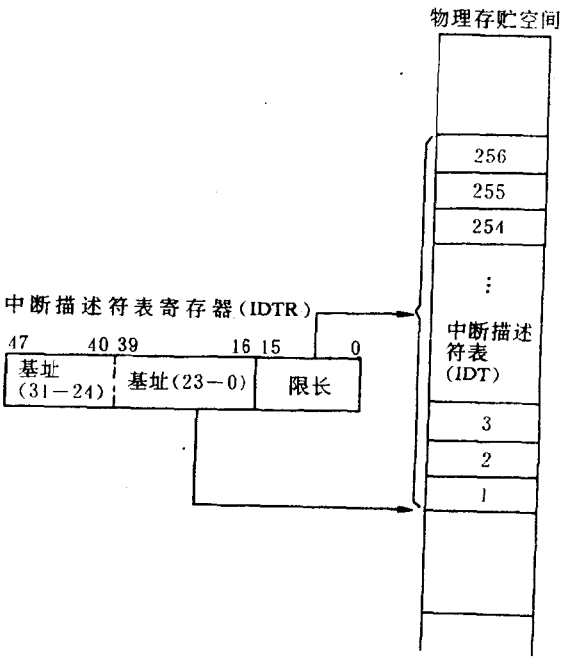


图 1.3.3　中断描述符表

最大可以是 65 535 字节($2^{16}-1$)，但由于 386 只有 256 个中断，所以只需 256 个中断描述符，即 8K 字节。

事实上，IDT 中的中断描述符称为中断门，我们将在下一节详细讨论。通过中断门，我们可调用中断服务程序。每个中断门也是 8 个字节，包含属性和中断服务程序的入口地址。

和全局描述符表一样，在切换到保护模式之前，必须将基地址和限长装入 IDTR 寄存器，一旦装入，IDTR 的值不再改变，指令 LIDT 和 SIDT 分别用来完成读出和写入 IDTR 的功能。

(3) 局部描述符表寄存器

局部描述符表寄存器和其它寄存器一起完成 80386 保护模式下的内存管理机制。局部描述符表寄存器必须和全局描述符表一起使用。如图 1.3.4 (*a*) 所示。每个任务拥有它自己的局部存贮器空间，这些存贮器空间由局部描述符表来定义，每个任务对应一个局部描述符表。在 386 保护模式下，多个任务间可以进行切换，因而内存中保留有多个局部描述表，图 1.3.4 (*a*) 中有几个描述符表。

一个描述符表之描述符包含这个描述符表所在的基地址和限长，因而描述符表之描述符指向一个描述符表。全局描述符表中，包含所有的局部描述符表之描述符，顾名思义，每一个局部描述符表描述符都指示一个局部描述符表，如图 1.3.4 (*b*) 所示。

局部描述符表寄存器是一个 16 位寄存器，里面的内容是一个选择符，通过选择符可以在全局描述符表中找到对应的局部描述符表描述符。每一个任务执行时都将其自身的选择符装入局部描述符表寄存器中，从而指示出自己所使用的局部内存地址空间。80386 微处理器在某一时刻只能执行一个任务，LDTR 内容即是与这个任务对应的选择符。

当一个选择符装入局部描述符表寄存器的同时，386 内部硬件逻辑自动将 GDT 中相应的局部描述符表描述符装入一个高速缓存器(cache)，这个高速缓存器称为局部描述符表高速缓存器，LDTR 与这个高速缓存器相互配合，从而指示出一个局部描述符表。图 1.3.4 (*b*) 虚线示出了这个过程。

(4) 任务寄存器

在 80386 保护模式中，任务间切换是一个重要的概念，而这一切最终都是通过任务寄存器内容的改变来实现的。任务寄存器是一个 16 位的寄存器。与 LDTR 相似，任务寄存器的内容也是一个选择符，它间接地指示出一个任务的任务状态段 TSS。每个任务都有一个任务状态段 TSS，任务状态段中保留有与这个任务相关的外部环境和内部数据，可以说在 386 保护模式中，TSS 是一个任务的唯一标识。

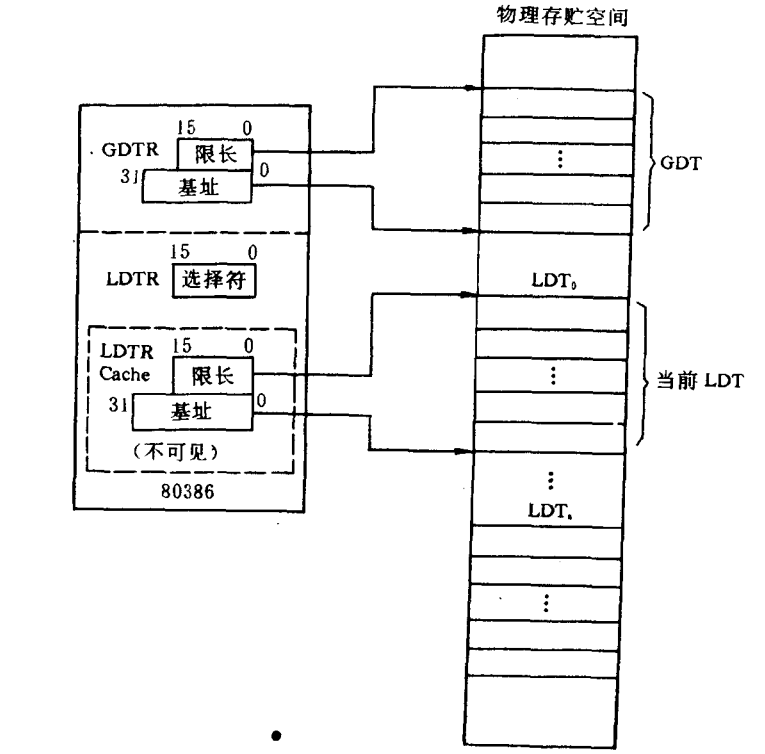
一个 TSS 描述符包含有 TSS 的基址和限长，在全局描述符表中保留有每个任务的 TSS 描述符，通过选择符可以选择某一个 TSS 描述符。

任务寄存器中装有当前任务的选择符。任务切换时，任务寄存器装入新的任务选择符。在任务寄存器内容改变的同时，80386 内部逻辑自动地将这个选择符所指示的 GDT 中的 TSS 描述符装入一个高速缓存器，这个高速缓存器与任务寄存器配合使用，称为任务描述符高速缓存器。

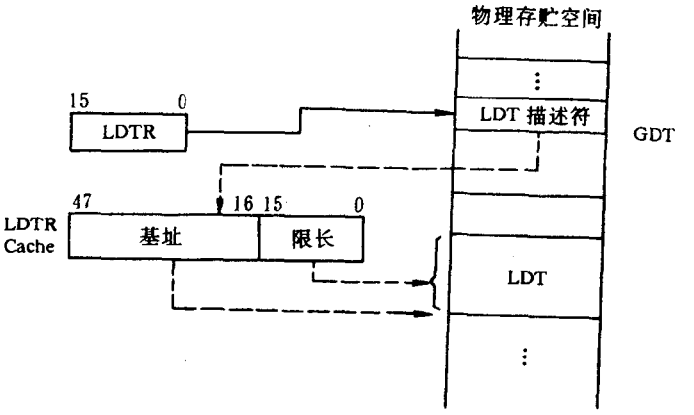
与图 1.3.4 (*b*) 类似，图 1.3.5 描述了任务寄存器间接指示一个任务状态段的全过程。

(5) 控制寄存器

前面我们已经介绍过，386 保护模式有 4 个系统控制寄存器，分别为 CR_0，CR_1，CR_2，



(a)



(b)

图　1.3.4

(a) 全局描述符表与当前任务的局部描述符表；　(b) LDTR 对局部描述符表的寻址机制

CR_3，如图 1.3.6 所示。每个控制寄存器均是 32 位寄存器，CR_0 的低 5 位是系统控制标志，被称为机器状态字(MSW)，CR_0 的最高位 PG 与 CR_2，CR_3 一起用于页面寻址机制。

让我们首先来看看 CR_0。MSW 共有 5 位，从低到高表示为 PE、MP、EM、TS、ET，其中 PE、MP，EM，ET 是保护模式系统设置信号，TS 是任务标志位。所有各位都可以通过语句

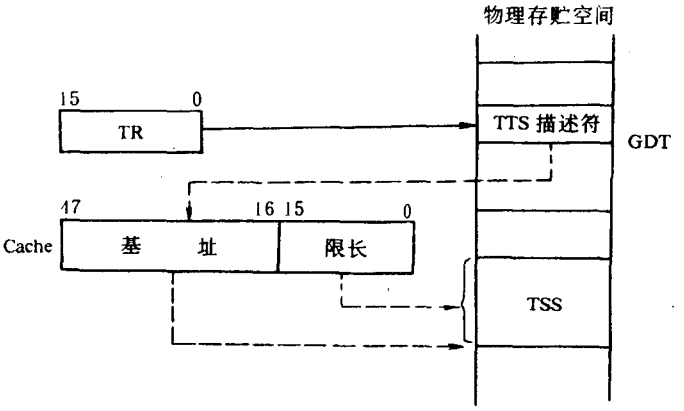


图 1.3.5　任务寄存器与任务切换机制

来改变。

保护模式激活位 PE 定义了 80386 是处于实地址模式还是保护模式。386 启动时，PE 位清零，此时系统处于实地址模式。通过指令设置 PE 位等于 1 时，系统进入保护模式。一旦系统进入保护模式，80386 就不能再通过软件指令切换到实地址方式。此时由保护模式进入到实地址模式的唯一办法便是硬件重新启动。

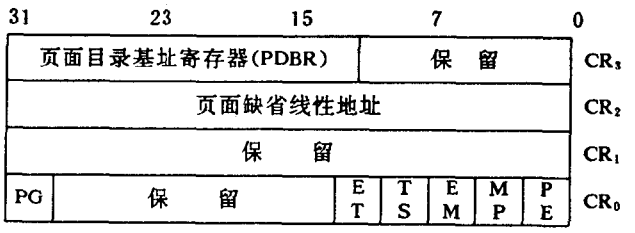


图 1.3.6　控制寄存器

协处理器标志位 MP 置 1 表示数学协处理器存在，另一方面，如果系统通过软件模拟协处理器，则模拟协处理器位 EM 置 1。这两位，一次只能设置 1 位。协处理器扩展类型位 ET 用来指示系统所使用的协处理器是 80387 还是 80287，当 ET 置 1 时，表示系统使用的是 80387 协处理器。

任务切换位 TS，当系统由一个任务切换到另一个任务时，任务切换位自动置位。

通过将 CR_0 的页面激活位 PG 置 1，386 的页面寻址机制开始起作用。与 PG 位配合使用的有 CR_2 和 CR_3 两个控制寄存器。CR_2 的 32 位用来装载页面缺省线性地址，这是由于 386 使用了虚拟地址方式，逻辑地址远远大于实际的物理地址。因而 386 允许操作系统使用换入/换出机制，即可以将暂不使用的内存内容换出到外部大容量磁盘上，在用到时再将其从磁盘换入到内存中来，所以当 80386 寻址机制发现逻辑页面不在内存中时，便将缺省的页面线性地址写入到 CR_2 中以便换入。CR_3 的高 20 位(8－31 位)称为页面目录基址寄存器(PDBR)，用来保存页面目录的基地址。低 12 位全部为 0 以便页面目录基址定位在 4K 字节的整数倍上，这是由于每个目录表是 4K 字节大小的缘故。详细内容参见下一小节页面寻址机制部分。