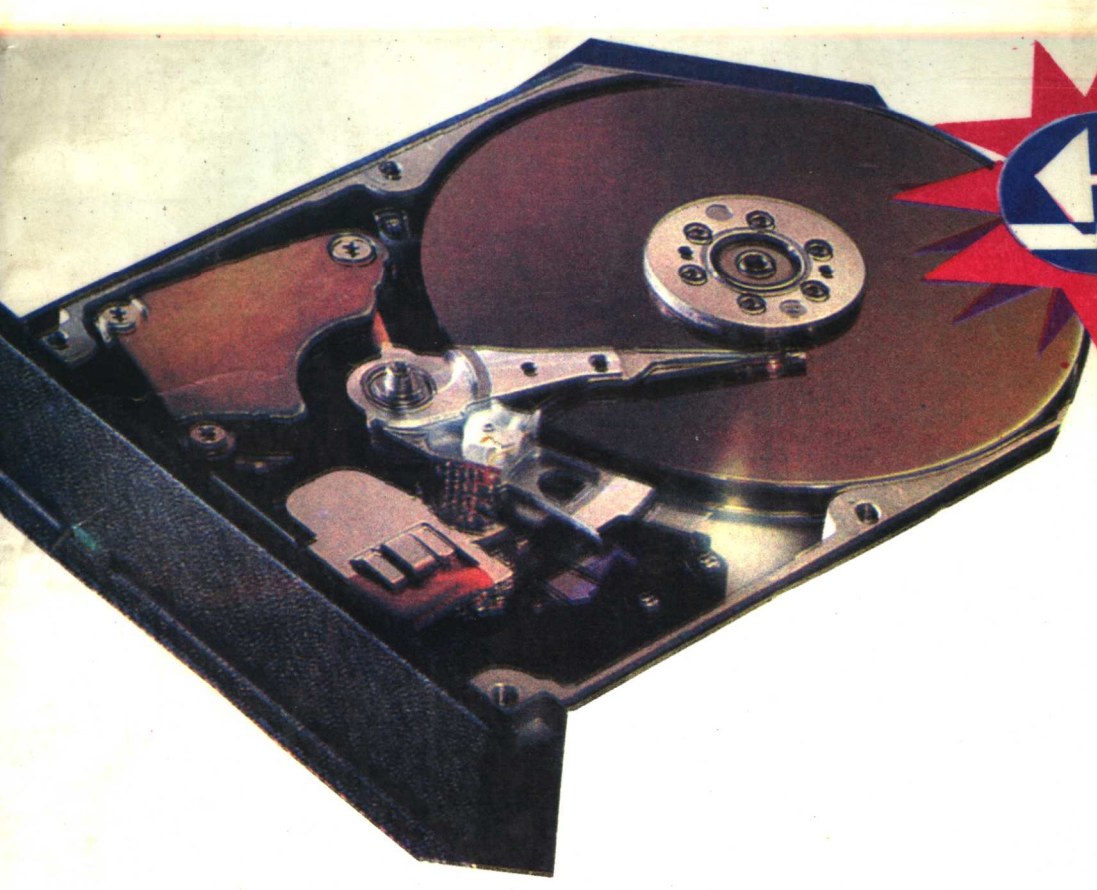
北京希望电脑公司计算机技术丛书



# DOS5.0开发者指南
# DOS高级程序设计指南

刘铁石　余永进
刘占奎　吴　刚　编译
田延斌
王淑芳　　审校

海洋出版社

北京希望电脑公司计算机技术丛书

# DOS 5 开发者指南
# ——DOS高级程序设计指南

刘铁石　余永进　等　编译
刘占奎　吴刚　田延斌

王淑芳　审　校

海洋出版社
1992年·北京

## 内 容 提 要

本书全面介绍了 MS-DOS 程序设计环境，通过实例详细讨论了许多高级编程技巧，其中包括如何编写驻留程序和设备驱动程序，如何在程序中增加鼠标支持和图形功能，以及如何使用扩展存储器以打破 640KB 限制等。

本书包括大量的汇编语言和 C 语言源代码，这些程序代码既是功能强的、实用的软件工具，又能够被用户程序直接引用而不必修改，从而给程序设计带来了极大的方便。

本书能够满足不同层次读者的需要，尤其适合那些想要编写有效的专业程序的高级程序员阅读。

欲要本书的用户请直接与北京 8721 信箱联系，邮编:100080，电话:2562329。

(京)新登字 087

特约编辑：秦人华
责任编辑：阎世尊

# DOS 5：开发者指南
# ——DOS 高级程序设计指南

刘铁石　余永进　等　编译
刘占奎　吴刚　田延斌
王淑芳　审　校

海洋出版社出版(北京市复兴门外大街 1 号)
海洋出版社发行　　北京市双青印刷厂印刷
开本：787×1092 毫米 1/16 印张 39.5 字数：885 千字
1992 年 2 月第一版　1992 年 2 月第一次印刷
印数：1-3000 册
定价：31.00元
ISBN 7-5027-2833-3/TP·114

# 前　言

《DOS 5:开发者指南》是编写所有版本的 MS-DOS 专业程序的完整指南，为了让你对 DOS 程序设计环境有一个整体的了解，书中每一章的内容均建立在前面章节的基础上．通过学习，你将学会如何编写应用程序、驻留程序、设备驱动程序以及任何类型的能够为 DOS 编写的程序．

第一部分包含基本的 PC 硬件以及 BIOS 和 DOS 功能，与许多其它 DOS 书籍不同，这些功能调用按使用功能进行组织，同时，一些未公布的最有用的功能也包括在内．

第二部分涉及到许多更高级的程序设计技巧，你会在这里找到帮助你建立驻留程序和设备驱动程序的固定程序，你也将看到如何把鼠标支持和图形功能增加到你的程序中，以及为了打破 640KB 限制如何使用扩展内存．

最后一部分包含 80286、386 和 486 处理器现有的一些技术，我们将考察保护方式下的调试技术以及扩充内存，然后剖析一个完整的 386 DOS 扩展器．

《DOS 5:开发者指南》包括有超过 500KB 的源代码，这些代码是功能强的、实用的软件工具，你能够把它们作为自己程序的一部分使用．你将观察到:

- 一个高速串行线性分析器
- 一个虚拟存储器磁盘备份程序
- 一个基于鼠标的骨架程序
- 一个用于编写 TSR 程序的工具箱
- 一个完整的 386 保护方式下的 DOS 扩展器

《DOS 5: 开发者指南》中的许多程序范例使用的是 C，其余的程序使用汇编和 C++(所有 C++程序都有相应的 C 程序例子)．无论使用哪一种语言，你都会发现 PC 硬件概述和应用环境两章的内容特别有价值．这两章集中讲述了有关的概念，这样，你能够在需要时处理其它的语言．

《DOS 5:开发者指南》将不限制你只使用某一个特定的编译程序或者汇编程序，你可以使用 Microsoft C 和 Borland 公司的 Turbo C 编译本书中的每一个 C 程序，也能够使用 MIX 公司的廉价的 Power C 编译其中的大多数程序．

如果你是一名工程师，你将从内部了解到 DOS 的工作情况．工具箱使编写低级 TSR 程序和设备驱动程序变得很容易．

如果你是一位软件开发者，你会发现 DOS 5.0 提供的最新特性，也将学会如何在应用程序中使用附加的存储器、图形以及保护方式技术．

如果你是一个学生，你会看到真正的应用程序，而不只是教科书的那些程序例子．

本书在编译出版过程中，得到了许多同志的热情帮助，同时还得到了中国科学院希望高级电脑技术公司资料部的秦人华经理、杨淑新老师的大力支持，特此表示诚挚的谢意!

编译者

1992 年・北京

# 目 录

## 第一部分 MS－DOS 程序设计基础

# 第二部分　MS—DOS 高级程序设计

## 第三部分 保护方式技术

# 第一部分　MS-DOS 程序设计基础

# 第一章　PC 硬件概述

在一本关于软件的书中，第一章就来谈论硬件，这似乎有点不合逻辑。然而，为了有效地对 PC 编程，就必须懂得有关的硬件知识。在下面的几节中，我们将回顾 8086 和有关的处理器的一些基本的结构特点，并详细地考查 PC 硬件。如果用来编程的高级语言隐含了这些细节的话，那么用户就会觉得开始的几节特别有帮助。

## 1.1　基本寻址

8086 型处理器，内存是以 8 位的字节构成。如果涉及的数大于 8 位，8086 就把最低有效字节存放在最低内存地址。尽管这符合逻辑，但是在读列表或内存转存时常会混淆，因为数值似乎是倒序的。例如，计算机存放 B800H 这个字是这样两字节：先存放 00H，然后是 B8H。

Intel 系列处理器采用分段的内存寻址技术。简单地说，一个段就是一个内存区，计算机可以处理多个段。在实址方式(DOS 运行的方式)中，每个段是严格的 64KB 长；一共可以有 65536 个段。段与段之间是可以重叠的，所以一个段可以从上一个段的后 16 字节处开始。这就是为什么 DOS 不能直接访问大于 1MB 内存的原因(65，536*16=1 048 576，即 1MB)。8086 和 8088(用于 XT 级机器上的处理器)无论如何只能寻址 1MB，286、386 和 486 处理器可以支持更大的内存，但 DOS 不能直接访问。

段的编号从 0000H 到 FFFFH。既然，每个段都是 64KB 长，所以我们就用一个称为偏移(offset)的值来确定所需寻址的字节。一个完整的 8086 的地址总是包括一个段值和一个偏移量。如果段值是 0040H，偏移量是 0102H，那么地址就写为 0040:0102。因为相邻段之间间隔 16 个字节(10H)，所以 0000:0010 与 0001:0000 表示同一地址。同样，0040:0000 与 0000:0400 是同一地址，也与 0020:0200 表示同一地址。计算机总是倒序地存放地址。例如，0040:1234 在计算机的内存(以 16 进制表示)中是这样的：

34　12　40　00

程序是通过寄存器和常数的组合来形成地址。基于此我们将在考查了处理器的寄存器之后再详细地讨论寻址问题。

## 1.2　内存类型

用户程序可以使用几种不同类型的内存。就像我们刚看到的，XT 级计算机只能寻址 1MB 内存，这种内存称为常规内存。XT 可以有多达 640KB 的 RAM，I/O 设备、BIOS ROM 和其它系统资源使用其余的 384KB 内存。

当程序需要的内存空间开始超出这个范围时，Lotus,Intel 和 Microsoft 一起致力于建立扩展内存规范(EMS)。可以看到 EMS 的不同版本号，像 EMS 3.2 或 EMS 4.0(最通行的写法)。

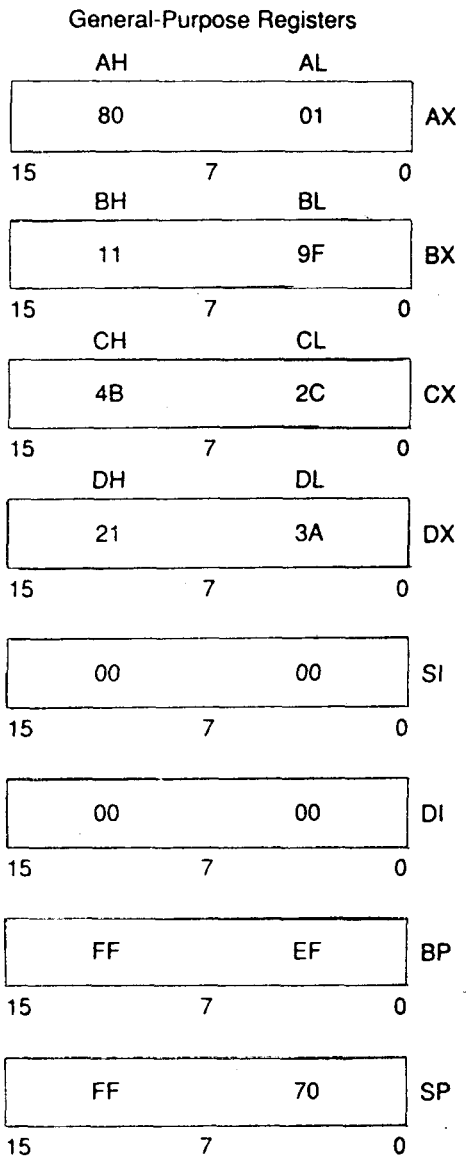
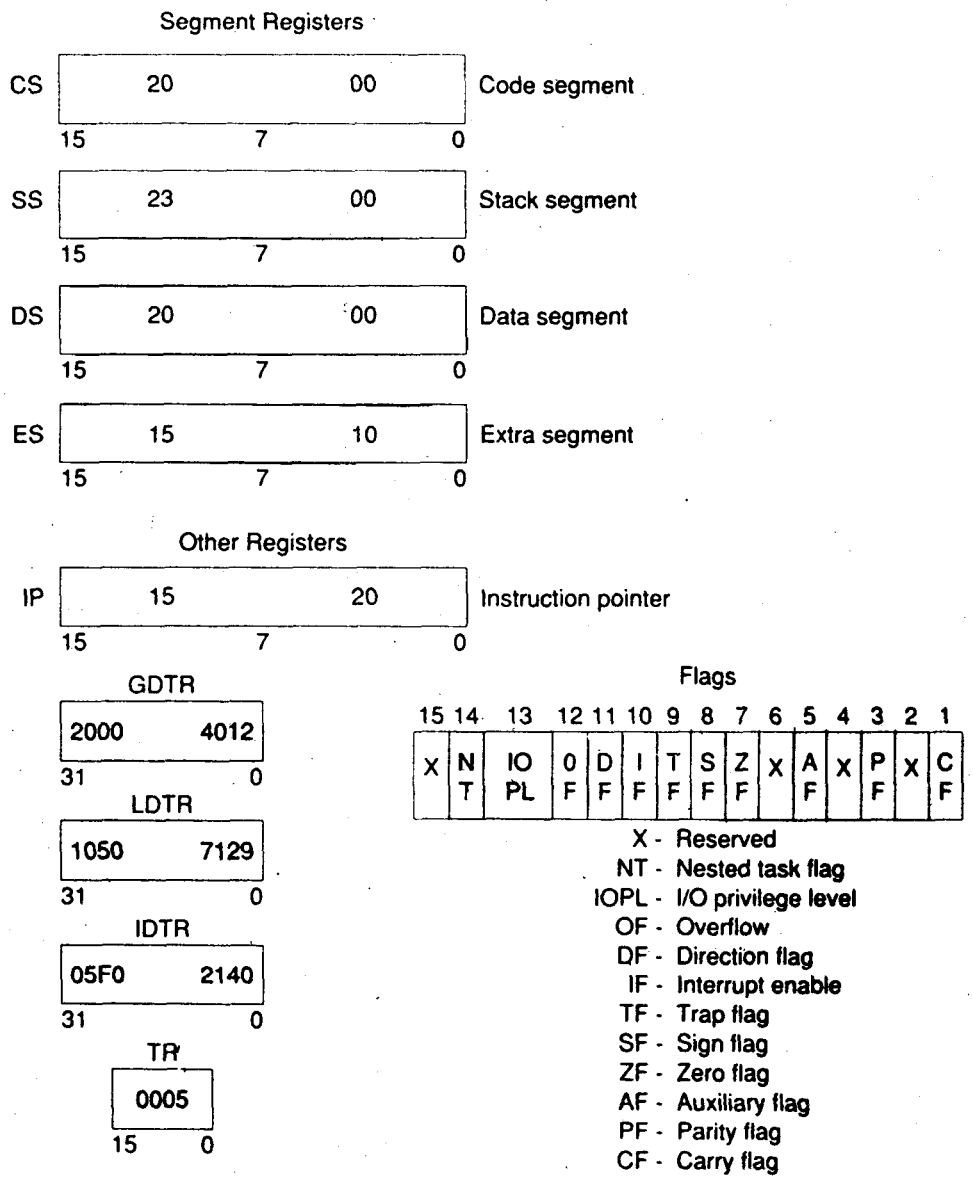



图 1-1 8088/8086/80286 寄存器




Note: Shaded registers and fields are only on the 80286

图 1-1 8088/8086/80286 寄存器(继续)

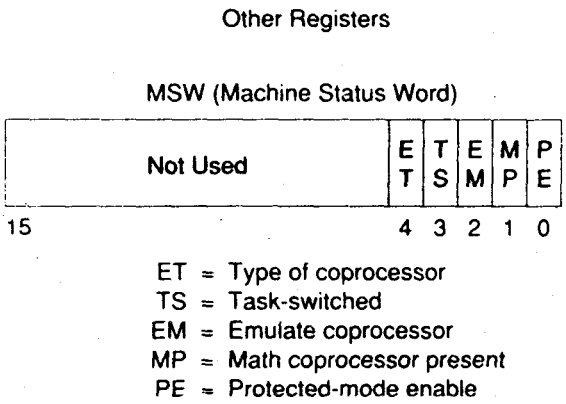



图 1-1 8088/8086/80826 寄存器(继续)

绝大多数的 EMS 系统包括一个或多个 EMS 卡和称之为驱动程序的软件。程序可以通过驱动程序请求使用 EMS 内存，可是这种内存在某一时刻只能访问 64KB。EMS 驱动程序把这 64KB 块放在 1MB 地址空间的高段，通常就在 BIOS 下面。通过调用驱动程序，程序可以把 EMS 内存 16KB 的“页”映射到 64K 字块。

即使 EMS 可以处理的内存多达 32MB，但在某一时刻，程序只能访问有限数量的 16KB 的页。尽管这不太方便，但是扩展内存是 XT 级机器可以使用的唯一的一类附加内存。如果程序使用其它类型的附加内存，那它们是不能在基于 8088/8086 的机器上运行。

EMS 也可以由磁盘来模拟，这不需要额外的硬件，但是与硬件的 EMS 系统相比这速度太慢了。

在 286、386 和 486 上，1MB 的内存并不是上限。286 可以寻址多达 16MB；386 和 486 可以寻址多达 4096MB(4GB)。从技术上讲，这种扩充内存只能在保护方式下访问。但 BIOS 提供了一个功能，数据可以在任意的两个地址之间相互传递，包括在扩充内存中的数据。

扩充内存规范(XMS)驱动程序允许程序用一个巧妙的寻址技术(将在后面讨论)访问扩充内存几乎是 64KB。重要的是要注意在保护方式下，常规内存和扩充内存并没有区别，扩充内存只是简单地从 1MB 地址开始。用软件可将扩充内存转换到扩展内存，在 286 上这一点不是十分有效，并且不是所有的 EMS 特性都能够仿效。可是，386 和 486 强大的内存管理性能，EMS 可以完整有效地用软件来实现。

## 1.3 寄存器

8086 有四个通用寄存器、一个标志寄存器、四个段寄存器、二个索引寄存器、一个堆栈段寄存器和堆栈指针、一个基址寄存器和一个指令指针寄存器。图 1-1 给出了这些寄存器以及 286 中的其它寄存器；图 1-2 给出了在 386 和 486 中的寄存器。注意 286、386 和 486 有多种专用的寄存器。虽然它们在 DOS 下不是十分有用，但当我们讨论对保护方式下的 DOS 扩展

时将会用到它们。

AX、BX、CX 和 DX 是通用寄存器：

- AX 有时也称为累加器，用于算术运算。
- BX 是唯一能在实址方式中参与构成地址的通用寄存器。
- CX 用于循环操作或重复操作计数。
- DX 指定 I/O 地址，特别是大于 0FFH 的 I/O 地址。它还在乘法和除法指令中与 AX 一起使用。

每个通用寄存器都是 16 位的。也可以把其中的任何一个用作二个 8 位寄存器。例如，AH 和 AL 事实上就是 AX 寄存器的两个 8 位寄存器。在图 1-1 中 AX 是 8001H，这就意味着 AH＝80H,AL＝01H。386 和 486 有同样的通用寄存器，只不过那些寄存器是 32 位的。32 位的寄存器都以字母 E 开头。如像图 1-2 当中，EAX 是 72018001H，那么 AX＝8001H,AH＝80H,AL＝01H。不能直接访问 32 位寄存器的高 16 位。

理论上讲，通用寄存器可以用于任何操作。但是每个寄存器都有某些特定功能。有时对于某种特定的操作必须使用某个寄存器，有时使用某个寄存器会比用其它寄存器更有效。

- 段寄存器(CS、DS、SS 和 ES)用来指出当前代码段(CS)、数据段(DS)和堆栈段(SS)，ES 是个附加段，用于段间数据传输。这些段像下面描述的那样参与地址计算。386 和 486 处理器还有二个附加段寄存器：FS 和 GS。
- 二个索引寄存器：SI 和 DI。主要是参与形成地址。与通用寄存器不同，SI 和 DI 是不能分成二半的。在 386 和 486 上，可以使用 ESI 和 EDI 来形成 32 位地址，或用 SI 和 DI 形成 16 位地址。
- 8086 系列提供一个堆栈供 CPU 或程序员使用。堆栈作为数据临时存储区域，在子程序调用和中断功能调用中起重要作用。堆栈段寄存器(SS)和栈指针(SP，或 386/486 中 ESP)给出栈顶地址。用 SS 寄存器作段值，用 SP 寄存器作偏移量，来确定当前堆栈的栈顶。
- 基址寄存器 BP (或 386/486 中 EBP)指向栈中的项。在许多方面，它与 SI、DI 寄存器相似，但我们稍加观察寻址方式便可以看出两者之间明显的差别。
- 指令指针寄存器(IP)给出当前代码段下一条指令的偏移量。尽管跳转指令或子程序调用会修改 IP，但用户却不能直接修改它。当然，在 386 和 486 上，32 位方式时使用 EIP。
- 标志寄存器包含一系列确定处理器状态的标志位。某一位可以表示最后算术运算是否溢出，另一位可以控制处理器对外部事件(或称为中断)的响应。程序通常用特定的指令设置或清除某一位来改动这个寄存器。

## 1.4 再谈寻址

在实址方式中，8086 系列的处理器仅能用某些值的组合来形成地址。为形成一个完整地址，必须计算出偏移量并确定使用的段。

下面的表可以表明偏移量是如何形成的。在每一列中，既可以选其中的一项，也可以不作选择地跳到下一列，把所选择的所有值加起来就是偏移量。

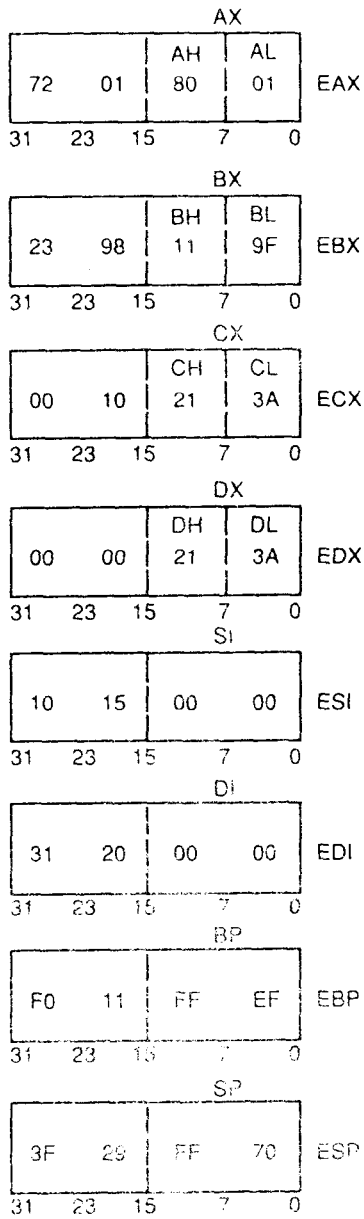
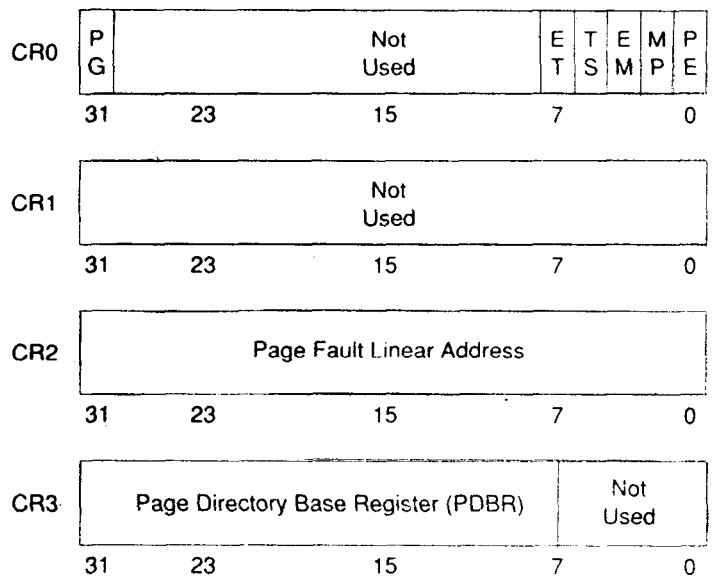



图 1-2 80386/80486 寄存器

Other Registers




PG = Paging enable
ET = Type of coprocessor
TS = Task-switched
EM = Emulate coprocessor
MP = Math coprocessor present
PE = Protected-mode enable

**图 1-2 80386/80486 寄存器 (继续)**