UNIX环境与80x86丛书

UNIX系统下的80386

周明德 张淑玲 编著

UNIX 80X86

清华大学出版社

UNIX 系统下的 80386

周明德　张淑玲　编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书是《UNIX 环境与 80x86》丛书中的一本。本书全面、深入地分析了当前微机主流 CPU 芯片 80386 的任务及任务切换机制与 UNIX 的进程、进程切换和调度;80386 的片上存储管理单元与 UNIX 的虚存管理;80386 的中断、异常与 UNIX 的中断、陷阱、信号与系统调用。最后还有几个是在保护方式下编程的、十分实用的例子。本书是学习和掌握 80386、80486 的各种微机的必读课本,是大学本科生、研究生学习和应用高档微机的重要参考书。

图书在版编目(CIP)数据

UNIX 系统下的 80386/周明德,张淑玲编著. —北京:清华大学出版社,1994
(UNIX 环境与 80X86 丛书/周明德等主编)
ISBN 7-302-01488-4

Ⅰ.U… Ⅱ.①周… ②张… Ⅲ.UNIX 操作系统-手册 Ⅳ.TP316

中国版本图书馆 CIP 数据核字(94)第 01909 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)
印刷者:通县人民文学印刷厂
发行者:新华书店总店北京科技发行所
开 本:787×1092 1/16 印张:18 字数:424 千字
版 次:1994 年 8 月第 1 版 1994 年 8 月第 1 次印刷
本社分类号:TP·593
印 数:0001—8000
定 价:14.60 元

《UNIX 环境与 80x86》丛书

出 版 说 明

微电子技术、计算机技术是当今世界发展最为迅速的科学技术。1976 年 Intel 公司推出它的第一个 16 位微处理器 8086 时，芯片上集成的晶体管数大约 2 万个；1985 年 Intel 公司第一次宣布 32 位的微处理器 80386 时，芯片上集成的晶体管数已达 20 万个；目前正在大量推广的 80486，集成的晶体管数达 100 万个；Intel 公司计划于 1993 年初推出的 P_5(也即 80586)，估计集成的晶体管数可达 400 万个。芯片的工作速度则是成 10 倍地提高：8086 的主振频率为 5MHz，80486 的主振频率为 50MHz，不久，微处理器的主振频率将可达 100MHz 或 200MHz。相应地，微处理器的功能和性能也有了极大的提高。

8086 有 20 条地址线，寻址能力达到 1M 字节，比之 8 位微处理器的 64K 字节的寻址能力，提高了 16 位，这在当时已是一种可观的发展。但是，应用证明，1MB 的内存是远远不够用的。80386 等 32 位微处理器的地址线扩展为 32 条，直接寻址能力达到了 4000MB (4GB)，这显然是一个十分庞大的内存空间。随着半导体存储器容量的迅速提高和价格的急剧下降，386、486 微机的常用内存容量已达到 4MB、8MB 或 16MB。

总之，自 1981 年 IBM PC 问世以来的短短 10 余年，微型计算机有了极大的飞跃，系统的配置有了很大的扩展，性能有了很大的提高，年产量已突破了 2500 万台。目前，80386、80486 已成为微型计算机的主流 CPU。

微型计算机性能的极大的提高，使得以单用户、单任务为主要特征的、得到了广泛应用的操作系统 MS-DOS(PC-DOS)已经远远跟不上硬件系统的发展，也不能满足广大用户应用的需要。

多用户、多任务分时式操作系统——UNIX，因其具有适用面宽、可移植性好等开放系统的基本特征，在 80 年代得到了广泛的应用，已成为高档微型机、工程工作站和超级小型计算机的主流操作系统。在 90 年代必将得到极大的发展。

为发展我国的软件产业，1989 年机电部决定集中资金、集中技术力量，以操作系统为突破口，大力进行国产系统软件的开发。此项目并被国家列为“85”科技攻关的重点软件开发项目。

国产操作系统要求遵循国际标准“可移植操作系统界面(POSIX)”，与国际上的主流 UNIX 系统相兼容，它的第一目标机是以 386、486 为 CPU 的高档微机。

为适应计算机的发展，为满足广大读者的要求，我们决定组织编写一套《UNIX 环境与 80x86》丛书。这套丛书详细、深入地分析在 UNIX 环境下工作的 80x86 的原理、特点和使用。Intel 公司在设计 80386 时考虑了多用户多任务操作系统的需要，在芯片上提供了 4 种特权极，提供了任务和任务切换的机制，提供了片内的存储管理单元等硬件支持。

UNIX操作系统则充分利用了这些机制，充分挖掘了 80386 的潜力。只有透彻地了解 80386 的全部机制和功能，才能设计出在 80386 上运行的多用户、多任务操作系统。

这套丛书还全面地、分层次地介绍在 80x86 上运行的 UNIX 操作系统。

这套丛书的作者都是国产操作系统项目的主要技术骨干和技术带头人，是国产操作系统的主要设计者。他们对各种 UNIX 系统和在 UNIX 环境下的 80386 都作了深入的分析。

这套丛书具有理论性、先进性、系统性和实用性，适合各种不同层次的计算机工作者使用。

前　言

半导体技术的飞跃发展，使得功能强大的 Intel 80386、80486 芯片成为当前微型计算机的主流 CPU 芯片。目前的这些 80386、80486 芯片就综合性能来说，已超过了 10 年前 IBM 公司推出 IBM-PC 机时的 CPU 芯片的 100 倍。以单用户、单任务、简单、易学，使用方便为特征的微型机的主流操作系统——MS-DOS(PC-DOS)，已经不能充分发挥这些 CPU 的性能，也不能满足广大用户的应用需求。

功能强大、齐全而又短小精悍，适用面宽便于剪裁，易于移植的多用户、多任务操作系统 UNIX，适应了这样的需要，迅速地由工作站、服务器、超级小型机领域进入了高档微机的领域。

UNIX 操作系统充分发挥了 80386、80486 的能寻址 4G 字节的物理地址空间，虚拟存储器功能，任务与任务切换机制，4 级特权等强大的功能，使在微型机上实现多用户、多任务管理。而 80386、80486 为多用户、多任务操作系统的实现提供了硬件上的直接支持，使操作系统软件与硬件得到完满的结合。

本书正是从这样的结合点出发，深入地分析了 80386、80486 的任务与任务切换机制及 UNIX 的进程和进程切换、进程调度；80386、80486 的片内存储管理单元与 UNIX 的虚存管理；80386、80486 中的中断、异常与 UNIX 的中断、陷阱、信号及系统调用机制。重点在于对 80386、80486 的深入、全面、透彻的分析，从而能更好地使用 80386、80486。最后一章介绍了几个在 80386 的保护方式下编程的汇编程序实例，全面而又深入地介绍了保护方式下编程的所有重要的问题，有很大的实用价值。

本书是作者参加国家“八五科技攻关计划”的重大软件课题——《国产操作系统开发》研究工作的总结，具有先进性和实用性。

张淑玲同志参加了本书第五章的汇编语言程序举例的设计、调试和总结工作。

本书可作为高校计算机专业的本科生、研究生学习微型计算机的教材和教学参考书，也可以作为广大微型计算机用户的重要参考书及培训教材。

限于时间与水平，书中必然存在不少缺点，敬请广大读者不吝赐教。

周明德

一九九三年十月九日

目　录

第一章

引　论

第一节　80386 基础

一、80386 的寄存器结构

1. 80386 的 CPU 结构

80386 的 CPU 结构如图 1-1 所示。

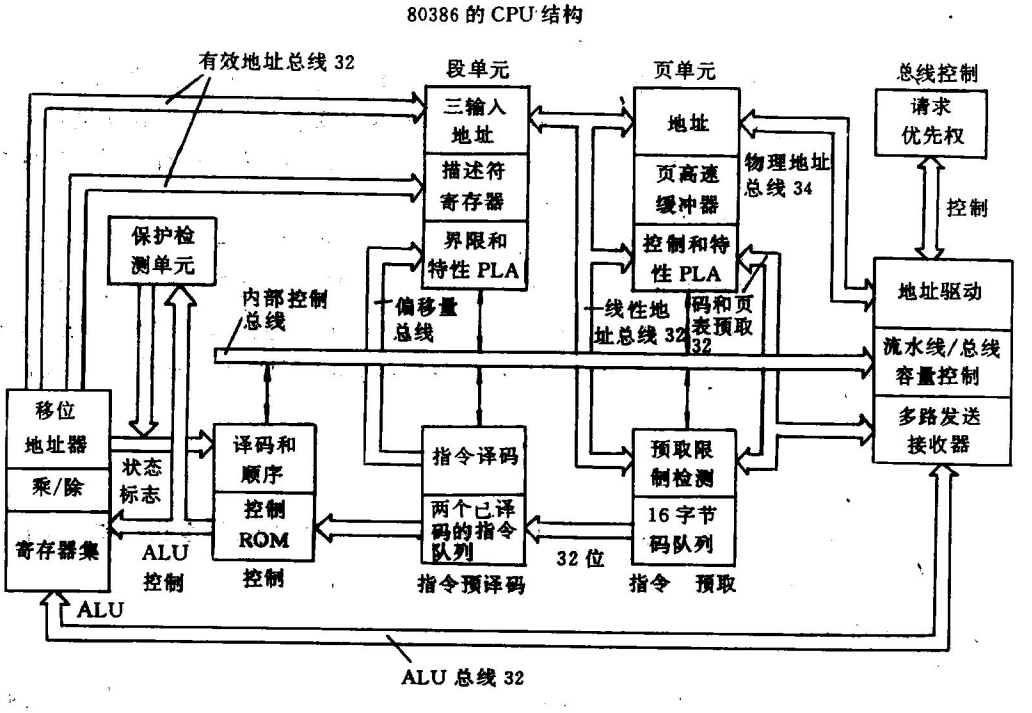


图 1-1　80386 的 CPU 结构

80386 芯片是由中央处理器(CPU)、存储管理部件(MMU)和总线接口部件(BIU)组

成的。

中央处理器又是由指令部件和执行部件所组成。指令部件可以预取指令(最多为16个字节),且对指令操作码进行译码,并把它们存放在已译码的指令队列里供执行部件使用(这样指令在执行时就可以省去取指令和译码的时间)。执行部件包括8个通用寄存器,它们既可用于数据操作,也可用作地址指针。还包括一个64位的桶形移位器(barrel shifter),用于加速移位、循环以及乘除法操作,这使典型的32位乘法可在1μs内执行。

存储管理部件(MMU)由分段部件和分页机构组成。分段部件通过提供一个额外的寻址器件对逻辑地址空间进行管理,可以实现任务之间的隔离也可以实现指令和数据区的重定位。分页机构提供了对物理地址空间的管理,整个空间分为若干页,每一页的大小是固定的,为4K(4096)字节。每一个段可以是一页也可以是若干页,最多可为1M页(4G字节)。为了实现虚拟存储管理,80386对页故障支持完整的再启动功能。

80386中的存储器是按段来组织的,每一段的大小都是可变的,最大可达到4千兆(4G)字节。一个给定范围的线性地址空间或一个段可以有相应的属性。这些属性包括它的起始地址、大小(界限)、类型(码段、数据或堆栈)及其保护特性。80386中的每一个任务都可以有多达16381(16K)个段,每一个段最大都可达4G字节。因此,80386为每个任务都可提供最大为64兆兆字节的虚拟地址空间。

为了使应用程序和操作系统相互隔离和各自得到保护,分段部件提供了4级保护。这种由硬件实施的保护,使各种系统的设计具有高度的完整性。

80386有两种操作方式:实地址方式和受保护的虚拟地址方式。在实地址方式下,80386的操作象一个极快的8086,且需要的话可以把结果扩展为32位。但在实地址方式下,只能寻址1M字节的地址空间,不能启用分页机制,不能使用特权。只有在保护虚地址方式下,才能充分发掘80386的潜力,启用分页机制。

在保护方式下,可通过软件切换进入虚拟8086方式。在这种方式下的每个任务都可用8086的语义运行,从而可以运行8086的各种软件(应用程序或整个操作系统)。通过采用分页和模拟I/O指令,各种虚拟的8086任务可以与80386主操作系统相互隔离并受到保护。虚拟8086方式的存在,使得在多用户操作系统(例如UNIX)中,为运行DOS任务提供了硬件基础。

2. 寄存器结构

80386共有7类32个寄存器。它们是:通用寄存器、段寄存器、指令指针和标志寄存器、控制寄存器、系统地址寄存器、排错寄存器和测试寄存器。

(1) 通用寄存器

在80386中有8个32位的通用寄存器,如图1-2所示。

每一个通用寄存器都可以存放数据或地址量,它们支持1、8、16、32和64位的数据操作数以及16位或32位地址操作数。

对这些通用寄存器的低16位可以分别存取,好象8086中的16位寄存器一样使用,而且它们的命名也与8086中的相同。

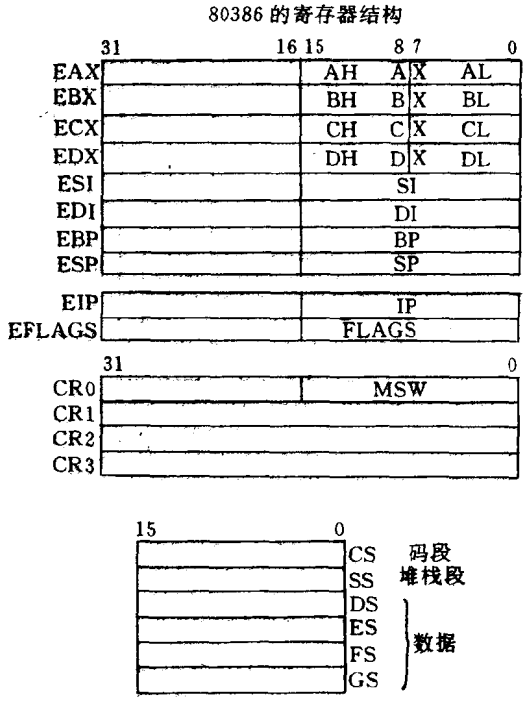


图 1-2　80386 中的编程结构

数据寄存器 EAX、EBX、ECX、EDX 中的低字中的两个 8 位字节(位 0—7 和位 8—15),可以作为 8 位的寄存器单独存取,它们的命名也与 8086 中一样。

(2) 指令指针和标志寄存器

80386 有 32 条地址线,故指令指针也为 32 位,是 IP 的扩展称为 EIP。EIP 中存放的是下一条要取出的指令(一般情况下,不是下一条要执行的指令)相对于码段基地址的偏移量。EIP 的低 16 位(位 0—15)称为 IP,在 16 位码段中使用。

80386 中的标志寄存器扩展为 32 位,称为 EFLAGS,如图 1-3 所示。

它的低 16 位(位 0—15)包含了命名为 FLAGS 的 80286 的标志寄存器,它的低 12 位(位 0—11)即为 8086 的标志寄存器。所以,在执行 8086 和 80286 操作码时,这个低 16 位的标志寄存器是有用的。

80386 中扩展了标志位 VM(位 17)和 RF(位 16)。

VM(Virtual 8086 Mode:虚拟 8086 方式,位 17)

在 80386 的保护虚地址方式中,由 VM 位提供一种虚拟 8086 方式。即当 80386 工作在保护虚拟地址方式时,如果使 VM 位置位,80386 将转为虚拟 8086 方式。VM 位只能在保护方式下由 32 位的 IRET 指令(若当前的特权级=0)或在任何特权级下由任务切换设置。VM 位不受 POPF 指令的影响,PUSHF 指令总是使该位清零。当在作为虚拟的 8086

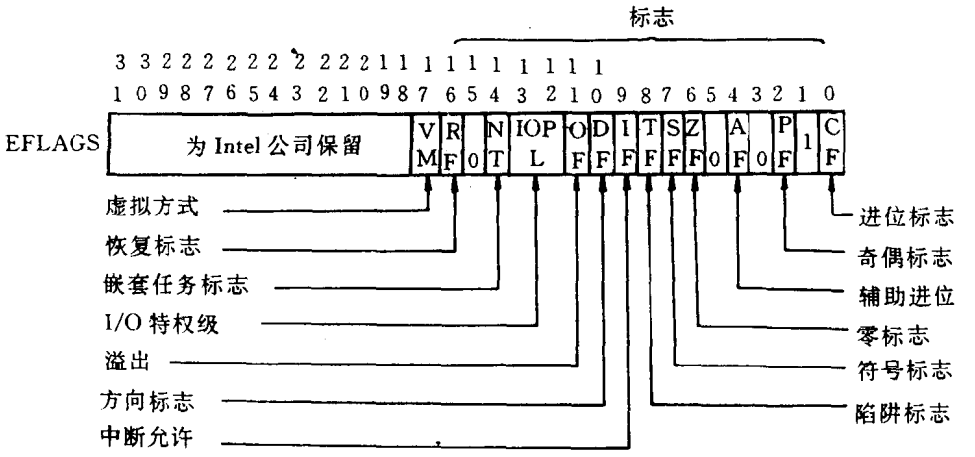


图 1-3　80386 中的标志寄存器

任务运行时，遇到中断，或执行了任务切换，则压入堆栈的或要保存的 EFLAGS 映象中的 VM 位为 1。

RF(Resume Flag：恢复标志，位 16)

RF 标志是与排错寄存器的断点或单步操作一起使用的。在断点处理之前，在两条指令之间对该位进行检查。若 RF 位置位，则在下一条指令执行期间不顾任何排错故障。然而，每当成功地完成一条指令(表明没有故障)时，RF 标志都将自动复位。但是，IRET、POP 指令和引起任务切换的 JMP、CALL、INT 指令例外。这些指令根据存储器映象所确定的值设置 RF。例如，在断点服务子程序的结尾处，IRET 指令可以弹出一个带有 RF 位置位的 EFLAGS 映象，这样在断点地址处恢复程序执行时，就不会在同一位置上产生另一次断点故障。

(3) 段寄存器

在 80386 中存储单元的地址仍是由两部分组成，即段基地址与段内偏移量。但在 80386 中，地址已扩展为 32 位，故段内偏移量和段基地址都是 32 位的。为了与 8086、80286 兼容，80386 中的段寄存器仍是 16 位的，于是 32 位的段基址就不是由段寄存器中的值直接确定，而是通过一个表来确定，段寄存器的值只是此表的索引(详细的在后面介绍)。

在 80386 中有 6 个 16 位段寄存器，即 CS、SS、DS、ES、FS 和 GS。

CS 表示当前的码段，SS 表示当前的堆栈段，而 DS、ES、FS 和 GS 都可以用来表示当前的数据段。

图 1-2 中所表示的段寄存器是编程可见的，在 80386 中，每一个段寄存器都有一个与之相联系的但编程时不可见的段描述符(用以描述一个段，例如此段的基地址，段的大小和段的属性等)寄存器，如图 1-4 所示。

每个段描述符寄存器都保持着一个 32 位的段基地址，一个 32 位的段界限(大小)和其他一些必要的属性。

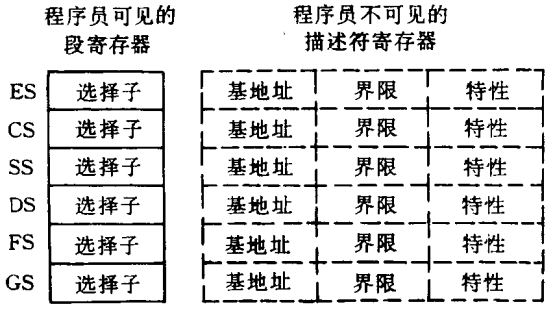


图 1-4　段描述符寄存器

每当用一个值加载段寄存器时，80386 的硬件会自动以段寄存器中的值为索引，从表（描述符表）中取出一个 8 个字节的描述符，经过适当的转换，装入到对应的段描述符寄存器中。每当出现存储器访问时，就可从指定的段寄存器，直接用相应的段描述符寄存器中的段基地址，而不必再去查表，取出段基地址。这就是 80386 对它的寻址方式的硬件支持，从而加快了存储器访问。

（4）系统地址寄存器

在 80286 和 80386、80486 中，地址是 24 位或 32 位的，段基地址就不能由段寄存器中的值左移若干位来形成，而是采用了选择子和描述符这样的数据结构来确定存储单元的段基地址的。这样做不仅可以用只有 16 位的段寄存器（段选择子）来确定 24 位（80286）或 32 位（80386、80486）的段基地址，更重要的是可以确定段的一些属性例如特权，从而为基于 80286、80386 的操作系统和保护机制，提供了广阔的活动舞台。

80286、80386 中的段基地址是由一个 8 个字节的描述符（如图 1-5 所示）确定的。

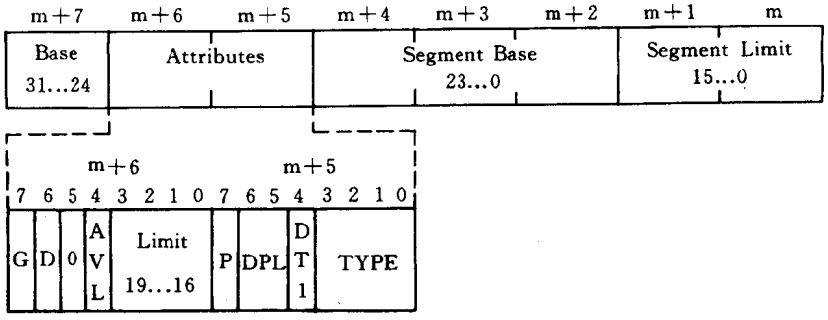


图 1-5　80386 中的段描述符

由相关的描述符组成一个表。这些表是：

GDT（Global Descriptor Table：全局描述符表）；

LDT（Local Descriptor Table：局部描述符表）；

IDT（Interrupt Descriptor Table：中断描述符表）。

在 80386 上运行的软件是基于任务的，为描述一个任务就需要 TSS(Task State Segment：任务状态段)，为支持任务切换就需要任务寄存器 TR。

上述的一些表，需要用寄存器来指定它们的起始地址和大小。这些系统地址寄存器如图 1-6 所示。

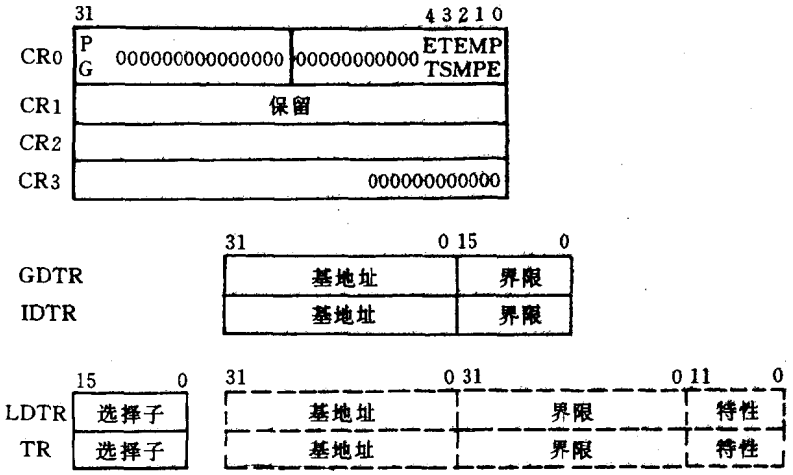


图 1-6　系统地址寄存器

GDTR 和 IDTR

这两个寄存器分别保存着 GDT 和 IDT 的 32 位的线性基地址和 16 位的界限值。这两个表的界限都是 16 位的，即每一个表最大为 64K 字节。每个描述符为 8 个字节，故每个表最大可以有 8K 个描述符。一般来说，这是足够的。对于 80386 来说，最多可以处理 256 个中断和异常，所以 IDT 的最大界限为 256×8＝2K 字节。

由于 GDT 和 IDT 对系统中的所有任务都是全局性的，因此，GDT 和 IDT 所在的段由 32 位的线性地址和 16 位的界限值确定。

LDTR 和 TR

局部描述符表 LDT 和任务状态段 TSS 是面向任务的，故它们所在的段不是由这些表本身决定的，而是由任务确定的，即由任务的系统段寄存器中的选择子值决定的。故局部描述符表寄存器 LDTR 和任务状态段寄存器 TR 中的值，只是一个 16 位的选择子，如图 1-6 中所示。它们的作用与段寄存器相似，故 80386 中为它们提供了相应的编程不可见的描述符寄存器，作为对存储单元访问的硬件支持，以加快存储单元的访问。

在多任务、多用户操作系统中，可以并发运行许多个任务，每一个任务都有自己的 LDT 和 TSS。但 80386 的硬件上，只有一个 LDTR 和 TR，所以其上装载的是当前任务的 LDT 和 TSS 的选择子，在对应的描述符寄存器上装载相应的描述符。

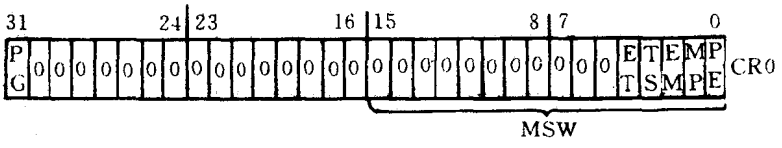
(5) 控制寄存器

自 80286 开始，就有了实地址和保护虚地址两种工作方式，从而有了支持这两个方式

转换的硬件——机器状态字 MSW。80386 中把它扩展为 3 个 32 位控制寄存器 CR0、CR2 和 CR3。这些寄存器与上面介绍的系统地址寄存器一起，保存着影响系统中所有任务的机器状态。

① CR0——机器控制寄存器(包括 80286 中的机器状态字)

CR0 中定义了 6 个控制和状态位，如图 1-7 所示。



注：0表示 Intel 公司已保留、不要定义。

图 1-7 控制寄存器 CR0

为了与 80286 的保护方式兼容，CR0 的低 16 位可以作为 MSW。

CR0 中各位的定义如下：

PG(Paging Enable：分页启用，第 31 位)

若 PG 位置位，启用 80386 的片内分页机制；否则，禁止分页部件工作。

ET(Processor Extension Type：处理器扩展类型，第 4 位)

ET 表示所扩展的协处理器的类型，是 80287 还是 80387。这可由 80386 复位后，由 ERROR #引线的电平确定。如果需要，在程序控制之下，通过对 CR0 送数，可以使 ET 位置位或复位。如果 ET 位置位，就使用与 80387 兼容的 32 位规程；如果 ET 位复位，则采用与 80287 兼容的 16 位规程。

注意：为严格保持与 80286 的兼容性，ET 位并不受 LMSW 指令的影响。但当存储 MSW 或 CR0 时，位 4 准确地反映 ET 位的当前状态。

TS(Task Switched：任务切换，第 3 位)

不论什么时候完成任务切换操作，TS 位都自动置位。一条协处理器操作码，将引起协处理器无效异常(异常 7)。此异常处理程序，典型地保存属于上一个任务的 80287/80387 内容，装载属于当前任务的 80287/80387 状态，且在返回失败的协处理器操作码之前，清除 TS 位。

EM(Emulate Coprocessor：仿真协处理器，第 2 位)

EM 位置位，将使所有的协处理器操作码都产生协处理器无效异常(异常 7)，从而由仿真程序执行相应的指令。若 EM 位复位，将允许协处理器操作码在实际的 80287 或 80387 上执行(在复位后的默认状态为 80387)。

注意：WAIT 指令不受 EM 位置位的影响。

MP(Monitor Coprocessor：协处理器监控位，第 1 位)

MP 位与 TS 位一起使用以确定在 TS=1 时，WAIT 操作码是否产生协处理器无效异常(异常 7)。即当 MP=1 且 TS=1 时，WAIT 操作码产生异常；否则，WAIT 操作码不

产生异常。

注意：当任务切换操作完成时，TS 位是自动置位的。

PE(Protection Enable：保护启用，第 0 位)

要使CPU 进入保护虚拟地址方式，必须使 PE 位置位。若 PE 位复位，则 CPU 再次处于实地址方式。PE 位可由加载 MSW 或 CR0 指令置位；但为了严格与 80286 兼容，PE 位不能由 LMSW 指令复位。

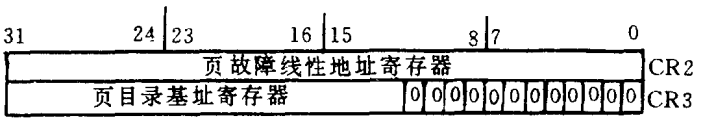
CR1：保留。

② CR2——页故障线性地址

当产生页故障时，在 CR2 中保存了产生页故障时的线性地址，这在故障处理程序中是十分重要的。

③ CR3——页目录表基地址

CR3 中包含的是页目录表的起始物理地址，如图 1-8 所示。



注：0 表示 Intel 公司保留，不要定义

图 1-8 CR2 和 CR3

80386 中的页目录表和页表都是页对齐的(4K 字节对齐)，因此，当写 CR3 时，它的低 12 位是忽略的(只加载高 20 位)，当存 CR3 时，这低 12 位也是未定义的。

若用明显的值加载 CR3 时，或在任务切换时两个任务的页目录表基地址不同，则将使分页机制的高速查找缓冲器中的所有缓存的页表项无效。

对控制寄存器操作，要用指令

```
MOV   CR0,reg
MOV   CR2,reg
MOV   CR3,reg
MOV   reg,CR0
MOV   reg,CR2
MOV   reg,CR3
```

这些都是特权指令。为了与 80286 兼容，也有只对 CR0 的低 16 位操作的 LMSW 和 SMSW指令。

(6) 排错(Debug)寄存器

80386 中有 6 个程序员可访问的排错寄存器，如图 1-9 所示。

这 6 个寄存器用于排错，寄存器 DR0—DR3 可指定 4 个断点的线性地址，寄存器 DR6 用于设置断点，而寄存器 DR7 用于显示断点的当前状态。

排错寄存器

31	0
线性断点地址 0	DR0
线性断点地址 1	DR1
线性断点地址 2	DR2
线性断点地址 3	DR3
Intel公司保留，不要定义	DR4
Intel公司保留，不要定义	DR5
断点状态	DR6
断点控制	DR7

测试寄存器

31	0
测试控制	TR6
测试状态	TR7

图 1-9　排错和测试寄存器

三、实地址方式

为了与 8086 兼容，在 80386 中设置了实地址工作方式。当 80386 复位或接通电源时，初始处于实地址方式。实地址方式与 8086 有相同的基本结构，但允许访问 80386 的 32 位寄存器集。80386 实地址方式的寻址方式、存储器大小、中断处理与 80286 的实地址方式相同。

80386 的指令系统中，除了专用于保护虚地址方式的指令外，其余的包括 lgdt、lidt 等指令在实地址方式下都是有效的。在实地址方式下操作数的默认长度是 16 位，就象 8086 的一样。在要使用 32 位寄存器和寻址方式时，必须使用超越前缀。80386 在实地址方式下的段最大是 64K 字节。设置实地址方式是为了保持 80386 和 8086 的兼容。又可以从实地址方式进入保护方式。

1. 存储器寻址

在实地址方式下，80386 的低 20 条地址线有效，故只能寻址 1M 字节的地址空间。

在实地址方式下不允许分页，所以线性地址与物理地址是一样的。

在实地址方式下，物理地址的形成与 8086 相同，某个段寄存器的内容包含着段的基地址，它的内容左移 4 位即为段基地址；与有效地址(由某种寻址方式确定)相加形成存储单元的物理地址，如图 1-10 所示。

由于在形成物理地址时，段寄存器内容左移 4 位，故各段总是起始于 16 字节的边界。

在实地址方式下，所有的段最大为 64K 字节，所有的段都可以读、写或执行。如果在取一条指令或读一个操作数时，地址超出了段的结尾(即一个操作数的地址偏移量大于 FFFFH，例如有一个字操作数，低字节在地址 FFFFH，而高字节在 0000H)，则 80386 产