x86 系统架构概览-读书笔记

1.1 系统级体系结构概览

系统级架构由一组寄存器、数据结构和指令组成，旨在支持基本的系统级操作，例如内存管理、中断和异常处理、任务管理以及多处理器控制。

下图提供了适用于 32 位模式的系统寄存器和数据结构的摘要：

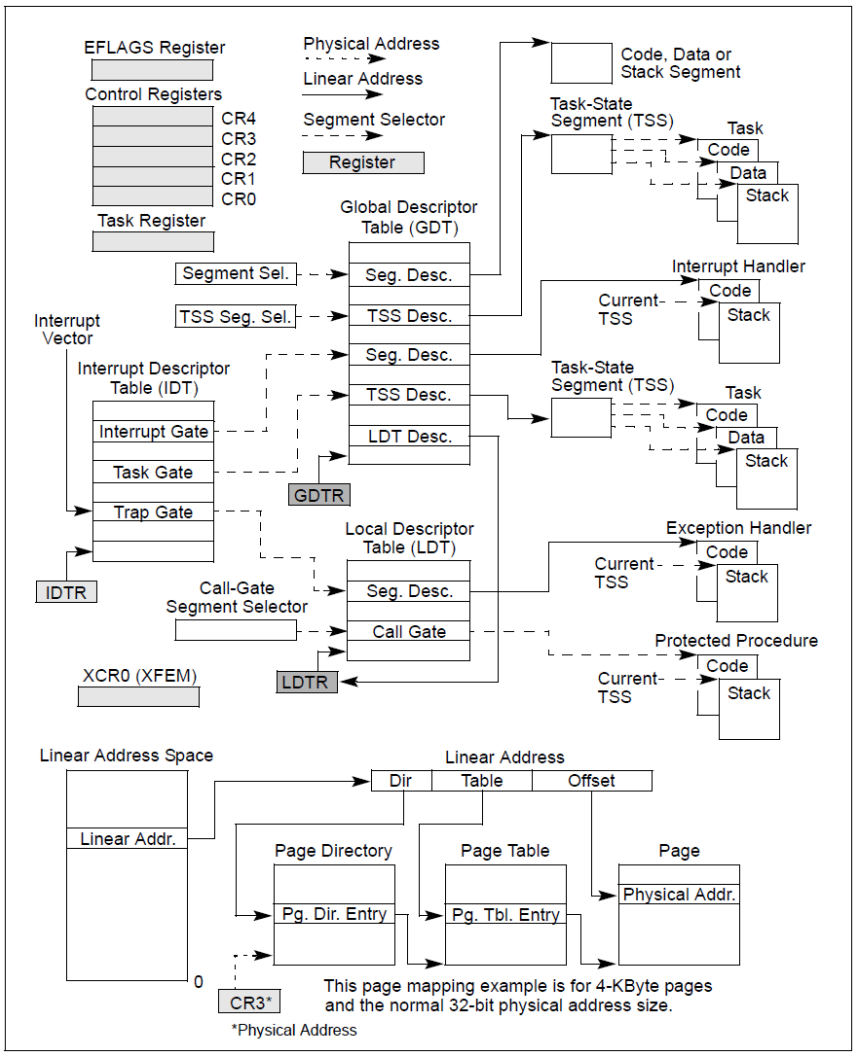

Figure 2-1. IA-32 System-Level Registers and Data Structures

1.1.1 IA-32e 模式下的全局和本地描述符表

在 IA-32e 子模式（64 位模式和兼容模式）中，GDTR 和 LDTR 寄存器都扩展为 64 位宽。有关更多信息，请参见第 3.5.2 节“IA-32e 模式下的段描述符表”。全局和本地描述符表在 64 位模式下扩展以支持 64 位基址（16 字节 LDT 描述符保存 64 位基址和各种属性）。在兼容模式下，描述符不会扩展。

1.1.2 系统段、段描述符和门

除了构成程序或过程执行环境的代码、数据和堆栈段之外，体系结构还定义了两个系统段：任务状态段（TSS）和 LDT。GDT 不被视为段，因为它不是通过段选择器和段描述符访问的。TSS 和 LDT 具有为其定义的段描述符。

体系结构还定义了一组称为门（调用门、中断门、陷阱门和任务门）的特殊描述符。它们为系统过程和处理程序提供了受保护的网关，这些过程和处理程序可能在与应用程序和大多数过程不同的特权级别上运行。例如，对调用门的 CALL 可以访问与当前代码段处于相同或数值较低特权级别（特权级别更高）的代码段中的过程。要通过调用门访问过程，调用过程 1 需要提供调用门的选择器。然后，处理器对调用门执行访问权限检查，将 CPL 与调用门的特权级别和调用门指向的目标代码段进行比较。

如果允许访问目标代码段，处理器将从调用门获取目标代码段的段选择器和该代码段的偏移量。如果调用需要更改特权级别，处理器还会切换到目标特权级别的堆栈。新堆栈的段选择器是从当前正在运行的任务的 TSS 中获得的。门还促进了 16 位和 32 位代码段之间的转换，反之亦然。

1.1.3 任务状态段和任务门

TSS（见图 2-1）定义任务执行环境的状态。它包括通用寄存器、段寄存器、EFLAGS 寄存器、EIP 寄存器和段选择器的状态，以及三个堆栈段（每个特权级别一个堆栈）的堆栈指针。TSS 还包括与任务关联的 LDT 的段选择器和分页结构层次结构的基址。

受保护模式下的所有程序执行都发生在任务（称为当前任务）的上下文中。当前任务的 TSS 的段选择器存储在任务寄存器中。切换到任务的最简单方法是调用或跳转到新任务。这里，新任务的 TSS 的段选择器在 CALL 或 JMP 指令中给出。在切换任务时，处理器执行以下操作：

1. 将当前任务的状态存储在当前 TSS 中。
2. 使用新任务的段选择器加载任务寄存器。
3. 通过 GDT 中的段描述符访问新的 TSS。
4. 将新任务的状态从新的 TSS 加载到通用寄存器、段寄存器、LDTR、控制寄存器 CR3（分页结构层次的基址）、EFLAGS 寄存器和 EIP 寄存器中。
5. 开始执行新任务。

还可以通过任务门访问任务。任务门类似于调用门，不同之处在于它提供对 TSS 而不是代码段的访问（通过段选择器）。

1.1.4 中断和异常处理

外部中断、软件中断和异常通过中断描述符表 (IDT) 进行处理。IDT 存储一组门描述符，用于访问中断和异常处理程序。与 GDT 一样，IDT 也不是一个段。IDT 基址的线性地址包含在 IDT 寄存器(IDTR)中。

IDT 中的门描述符可以是中断、陷阱或任务门描述符。要访问中断或异常处理程序，处理器首先通过 INT、INTO、INT 3 或 BOUND 指令从内部硬件、外部中断控制器或软件接收中断向量（中断号）。中断向量提供 IDT 的索引。如果选定的门描述符是中断门或陷阱门，则以类似于通过调用门调用过程的方式访问相关的处理程序过程。如果描述符是任务门，则通过任务切换访问处理程序。

1.1.5 内存管理

系统架构支持内存的直接物理寻址或虚拟内存（通过分页）。使用物理寻址时，线性地址被视为物理地址。使用分页时：所有代码、数据、堆栈和系统段（包括 GDT 和 IDT）都可以分页，只有最近访问的页面才会保存在物理内存中。

页面（有时称为页框）在物理内存中的位置包含在分页结构中。这些结构驻留在物理内存中。

分页结构层次结构的基物理地址包含在控制寄存器 CR3 中。分页结构中的条目确定页框基的物理地址、访问权限和内存管理信息。

要使用此分页机制，线性地址被分成几部分。这些部分提供分页结构和页框的单独偏移量。系统可以有一个或多个分页结构层次结构。例如，每个任务都可以有自己的层次结构。

1.1.6 系统寄存器

为了帮助初始化处理器和控制系统操作，系统架构在 EFLAGS 寄存器和几个系统寄存器中提供了系统标志：

• EFLAGS 寄存器中的系统标志和 IOPL 字段控制任务和模式切换、中断处理、指令跟踪和访问权限。

• 控制寄存器（CR0、CR2、CR3 和 CR4）包含用于控制系统级操作的各种标志和数据字段。这些寄存器中的其他标志用于指示对操作系统或执行程序中特定处理器功能的支持。

• 调试寄存器允许设置断点，以用于调试程序和系统软件。

• GDTR、LDTR 和 IDTR 寄存器包含其各自表的线性地址和大小（限制）。

• 任务寄存器包含当前任务的 TSS 的线性地址和大小。

• 特定于模型的寄存器。

特定于模型的寄存器(MSR)是一组主要可供操作系统或执行程序（即在特权级别 0 上运行的代码）使用的寄存器。这些寄存器控制诸如调试扩展、性能监控计数器、机器检查架构和内存类型范围(MTRR)等项目。这些寄存器的数量和功能在 Intel 64 和 IA-32 处理器系列的不同成员之间有所不同。

大多数系统限制应用程序访问系统寄存器（EFLAGS 寄存器除外）。但是，系统可以设计为所有程序和过程都以最高特权级别（特权级别 0）运行。在这种情况下，应用程序将被允许修改系统寄存器。

1.2 实模式和保护模式转换

上电或复位后，处理器将处于实地址模式。然后，控制寄存器 CR0 中的 PE 标志控制处理器是在实地址模式下还是在保护模式下运行。EFLAGS 寄存器中的 VM 标志决定处理器是在保护模式还是虚拟 8086 模式下运行。保护模式和虚拟 8086 模式之间的转换通常作为任务切换或中断或异常处理程序返回的一部分进行。LMA 位（IA32_EFER.LMA[bit 10]）决定处理器是否在 IA-32e 模式下运行。在 IA-32e 模式下运行时，64 位或兼容子模式操作由代码段的 CS.L 位决定。

处理器通过启用分页并设置 LME 位（IA32_EFER.LME[bit 8]）从保护模式进入 IA-32e 模式。当处理器处于实地址、受保护、虚拟 8086 或 IA-32e 模式时，只要收到 SMI，处理器就会切换到 SMM。执行 RSM 指令后，处理器始终会返回到发生 SMI 时所处的模式。

1.3 80x86 系统指令寄存器

1.3.1 标志寄存器（EFLAGS）

EFLAGS 寄存器的系统标志和 IOPL 字段控制 I/O、可屏蔽硬件中断、调试、任务切换和虚拟 8086 模式（见图 2-5）。只有特权代码（通常是操作系统或执行代码）才允许修改这些位。

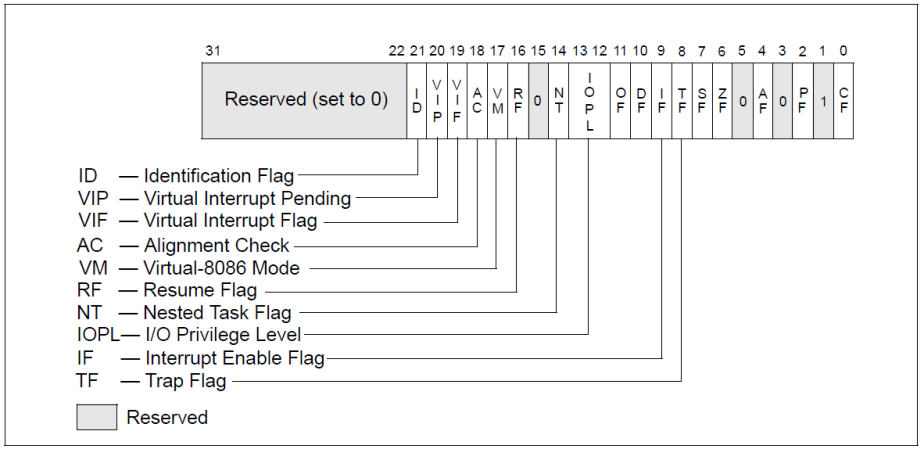

Figure 2-5. System Flags in the EFLAGS Register

1.3.2 内存管理寄存器（GDTR、LDTR、IDTR、TR）

处理器提供了四个内存管理寄存器（GDTR、LDTR、IDTR 和 TR），用于指定控制分段内存管理的数据结构的位置（见图 2-6）；提供了用于加载和存储这些寄存器的特殊指令。

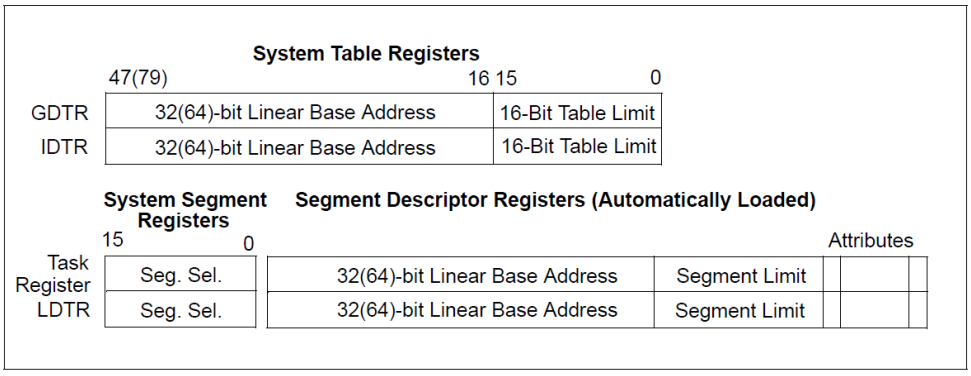

Figure 2-6. Memory Management Registers

（1）GDTR：

GDTR 寄存器保存 GDT 的基址（保护模式下为 32 位；IA-32e 模式下为 64 位）和 16 位表限制。基址指定 GDT 字节 0 的线性地址；表限制指定表中的字节数。

LGDT 和 SGDT 指令分别加载和存储 GDTR 寄存器。处理器上电或复位时，基址设置为默认值 0，限制设置为 0FFFFH。作为保护模式操作的处理器初始化过程的一部分，必须将新的基址加载到 GDTR 中。

（2）LDTR：

LDTR 寄存器保存 16 位段选择器、基址（保护模式下为 32 位；IA-32e 模式下为 64 位）、段限制和 LDT 的描述符属性。基址指定 LDT 段字节 0 的线性地址；段限制指定段中的字节数。LLDT 和 SLDT 指令分别加载和存储 LDTR 寄存器的段选择器部分。包含 LDT 的段必须在 GDT 中具有段描述符。当 LLDT 指令在 LDTR 中加载段选择器时：LDT 描述符中的基址、限制和描述符属性会自动加载到 LDTR 中。

发生任务切换时，LDTR 会自动加载新任务的 LDT 的段选择器和描述符。在将新的 LDT 信息写入寄存器之前，不会自动保存 LDTR 的内容。

在处理器上电或复位时，段选择器和基址设置为默认值 0，限制设置为 0FFFFH。

（3）IDTR：

IDTR 寄存器保存 IDT 的基地址（保护模式下为 32 位；IA-32e 模式下为 64 位）和 16 位表限制。基地址指定 IDT 字节 0 的线性地址；表限制指定表中的字节数。LIDT 和 SIDT 指令分别加载和存储 IDTR 寄存器。在处理器上电或重置时，基地址设置为默认值 0，限制设置为 OFFFFH。然后可以在处理器初始化过程中更改寄存器中的基地址和限制。

（4）TR：

任务寄存器保存当前任务的 TSS 的 16 位段选择器、基址（保护模式下为 32 位；IA-32e 模式下为 64 位）、段限制和描述符属性。选择器引用 GDT 中的 TSS 描述符。基址指定 TSS 字节 0 的线性地址；段限制指定 TSS 中的字节数。LTR 和 STR 指令分别加载和存储任务寄存器的段选择器部分。当 LTR 指令在任务寄存器中加载段选择器时，TSS 描述符中的基址、限制和描述符属性将自动加载到任务寄存器中。在处理器上电或复位时，基址设置为默认值 0，限制设置为 OFFFFH。

当发生任务切换时，任务寄存器会自动加载新任务的 TSS 的段选择器和描述符。在将新的 TSS 信息写入寄存器之前，任务寄存器的内容不会自动保存。

1.3.3 控制寄存器（CR0-CR3）

控制寄存器（CR0、CR1、CR2、CR3 和 CR4；参见图 2-7）决定处理器的操作模式和当前执行任务的特性。这些寄存器在所有 32 位模式和兼容模式下均为 32 位。

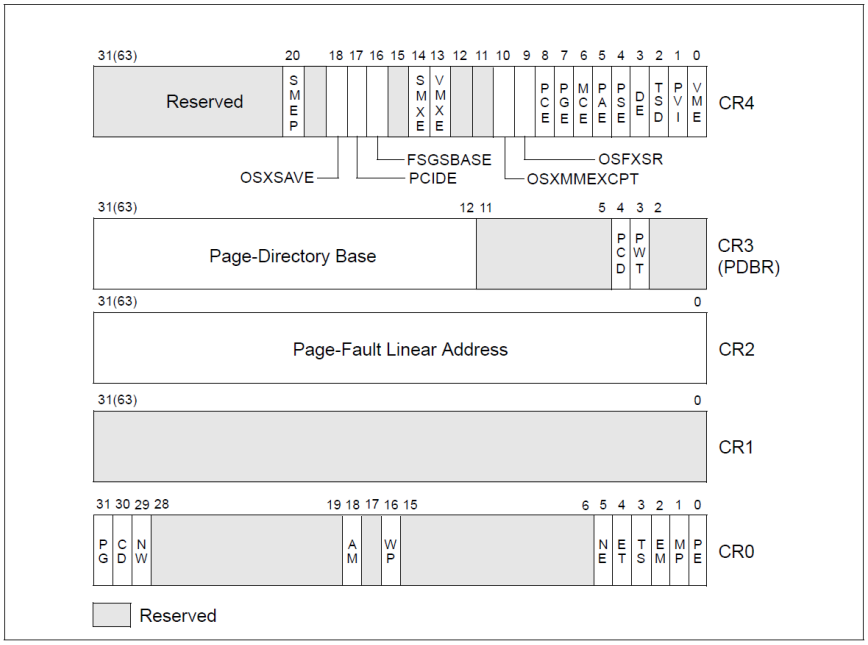

Figure 2-7. Control Registers

在 64 位模式下，控制寄存器扩展为 64 位。MOV CRn 指令用于操作寄存器位。这些指令的操作数大小前缀将被忽略。以下情况也是如此：

- CR0 和 CR4 的 63:32 位是保留的，必须用零写入。将非零值写入任何高 32 位都会导致一般保护异常 #GP(0)。
- CR2 的所有 64 位均可由软件写入。
- CR3 的位 51:40 是保留的，必须为 0。
- MOV CRn 指令不检查写入 CR2 和 CR3 的地址是否在实现的线性地址或物理地址限制内。
- 寄存器 CR8 仅在 64 位模式下可用。

控制寄存器总结如下，这些控制寄存器中每个架构定义的控制字段均单独描述。在图 2-7 中，64 位模式下寄存器的宽度在括号中表示（CR0 除外）。

- CR0 — 包含控制处理器操作模式和状态的系统控制标志。
- CR1 — 保留。
- CR2 — 包含页面错误线性地址（导致页面错误的线性地址）。
- CR3 — 包含分页结构层次结构基址的物理地址和两个标志（PCD 和 PWT）。仅指定基地址的最高有效位（低于低 12 位）；地址的低 12 位假定为 0。因此，第一个分页结构必须与页面（4 KB）边界对齐。PCD 和 PWT 标志控制该分页结构在处理器内部数据缓存中的缓存（它们不控制页目录信息的 TLB 缓存）。

使用物理地址扩展时，CR3 寄存器包含页目录指针表的基地址。在 IA-32e 模式下，CR3 寄存器包含 PML4 表的基地址

1.4 系统指令

系统指令处理系统级功能，例如加载系统寄存器、管理缓存、管理中断或设置调试寄存器。其中许多指令只能由操作系统或执行程序（即在特权级别 0 上运行的程序）执行。其他指令可以在任何特权级别执行，因此可供应用程序使用。

表 2-3 列出了系统指令，并指出它们是否可用于应用程序。

Table 2-3. Summary of System Instructions

Instruction	Description	Useful to Application?	Protected from Application?
LLDT	Load LDT Register	No	Yes
SLDT	Store LDT Register	No	No
LGDT	Load GDT Register	No	Yes
SGDT	Store GDT Register	No	No
LTR	Load Task Register	No	Yes
STR	Store Task Register	No	No
LIDT	Load IDT Register	No	Yes
SIDT	Store IDT Register	No	No
MOV CR*n*	Load and store control registers	No	Yes
SMSW	Store MSW	Yes	No
LMSW	Load MSW	No	Yes
CLTS	Clear TS flag in CR0	No	Yes
ARPL	Adjust RPL	Yes[1,5]	No
LAR	Load Access Rights	Yes	No
LSL	Load Segment Limit	Yes	No
VERR	Verify for Reading	Yes	No
VERW	Verify for Writing	Yes	No
MOV DR*n*	Load and store debug registers	No	Yes
INVD	Invalidate cache, no writeback	No	Yes
WBINVD	Invalidate cache, with writeback	No	Yes
INVLPG	Invalidate TLB entry	No	Yes
HLT	Halt Processor	No	Yes
LOCK (Prefix)	Bus Lock	Yes	No
RSM	Return from system management mode	No	Yes
RDMSR[3]	Read Model-Specific Registers	No	Yes
WRMSR[3]	Write Model-Specific Registers	No	Yes
RDPMC[4]	Read Performance-Monitoring Counter	Yes	Yes[2]
RDTSC[3]	Read Time-Stamp Counter	Yes	Yes[2]
RDTSCP[7]	Read Serialized Time-Stamp Counter	Yes	Yes[2]
XGETBV	Return the state of XCR0	Yes	No
XSETBV	Enable one or more processor extended states	No[6]	Yes

- LGDT（加载 GDTR 寄存器）——将内存中的 GDT 基地址和限制加载到 GDTR 寄存器中。
- SGDT（存储 GDTR 寄存器）——将 GDTR 寄存器中的 GDT 基地址和限制存储到内存中。

• LIDT（加载 IDTR 寄存器）——将内存中的 IDT 基地址和限制加载到 IDTR 寄存器中。
• SIDT（加载 IDTR 寄存器）——将 IDTR 寄存器中的 IDT 基址和限制存储到内存中。
• LLDT（加载 LDT 寄存器）——将 LDT 段选择器和段描述符从内存加载到 LDTR 中。（段选择器操作数也可以位于通用寄存器中。）
• SLDT（存储 LDT 寄存器）——将 LDTR 寄存器中的 LDT 段选择器存储到内存或通用寄存器中。
• LTR（加载任务寄存器）——将 TSS 的段选择器和段描述符从内存加载到任务寄存器中。（段选择器操作数也可以位于通用寄存器中。）
• STR（存储任务寄存器）— 将当前任务 TSS 的段选择器从任务寄存器存储到内存或通用寄存器中。