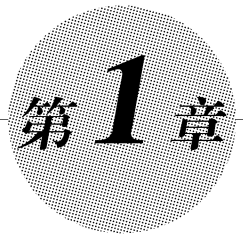

AMD x86-64 系统编程概要

本书是为下面两类软件开发者所写的：

① 编写操作系统、装入程序、连接程序、设备驱动程序或要访问系统资源(这些系统资源通常只对运行在最高特权级(CPL＝0)的软件可用)的实用程序的程序员。

② 需要了解系统的硬件和功能特征以及它们对应用软件的支持的应用程序开发者。

本章介绍软件开发者可用的 x86-64 体系结构的基本特征和能力。这些概念包括：

- 支持的地址格式和它们是如何组织的。
- 内存管理硬件如何使各种地址格式可用于访问内存。
- 处理器操作模式和内存管理硬件如何支持这些模式。
- 用于管理系统资源的系统控制寄存器。
- 中断和异常机制以及它们如何用于中断程序执行和报告差错。
- 附加的、可用于系统软件包括对硬件多任务的支持、报告机器检查异常、调试软件问题和优化软件性能等的各种特性。

许多传统的特性和能力在 x86-64 体系结构下得到了增强以支持 64 位操作系统和应用程序，同时对已存在的软件提供完全的后向兼容特性。

1.1 内存模型

x86-64 体系结构的内存模型允许系统软件用安全的方式管理应用软件和相关的数据，并且与传统的内存模型后向兼容。该模型的硬件转换机制可提供虚拟存储空间和物理内存空间之间的地址映射。其转换机制允许系统软件透明地重定位应用程序和数据，定位在物理内存空间的任何位置或在由操作系统管理的系统硬件驱动器的区域中。

在长模式(有关长模式的介绍，请参见 1.3 节)中，x86-64 体系结构实现平面存储模型。在传统模式中，该体系结构可实现所有的传统内存模型。

1.1.1 内存寻址

x86-64 体系结构支持地址重定位。为此，需要若干种地址类型以完全描述内存组织。特别是由 x86-64 体系结构定义的 4 种地址类型：

- 逻辑地址。
- 有效地址，或是逻辑地址一部分的段偏移量。
- 线性（虚拟）地址。
- 物理地址。

1. 逻辑地址

逻辑地址是至段地址空间的基准，它由段选择子和有效地址组成。逻辑地址表示为：

逻辑地址 ＝ 段选择子：偏移量

段选择子规定在全局或局部描述符表中的一项。规定的描述符表项描述在虚拟地址中的段位置、它的尺寸和其他特性。有效地址用作为在由选择子规定的段中的偏移量。

逻辑地址常常用作为远指针（far pointer）。远指针通常用在软件寻址中的地址在当前段外的情况下。

2. 有效地址

有效地址即存储段内的偏移量。有效地址由基寄存器值、可放大的变址值和位移量值等元素形成。有效地址可用以下等式表示：

有效地址 ＝ 基值 ＋（放大系数 × 变址值）＋ 位移量

用于有效地址计算的各个元素定义如下：

- 基值——存在于任一通用寄存器中的值。
- 放大系数——1、2、4 或 8。
- 变址值——存在于任一通用寄存器中的 2 的补码值。
- 位移量——作为指令一部分编码的一个 8 位、16 位或 32 位 2 的补码值。

有效地址常常作为近指针（near pointer）引用。当段选择子隐含已知时，或当使用平面存储模型时，使用近指针。

长模式定义有效地址的长度是 64 位。若处理器实现不支持全 64 位虚拟地址空间，有效地址必须以规范形式表示。

3. 线性（虚拟）地址

逻辑地址的段选择子部分规定了在全局或局部描述符表中的段描述符项。此规定的段描述符项包含段基地址，它是在线性地址空间中的段的起始地址。线性地址由段基地址加有效地址（段偏移量）形成，它建立了在所支持的线性地址空间的任一字节单元的基准。线性地址常常作为虚拟地址引用，这两个术语在本书中可以交换使用。

线性地址 ＝ 段基地址 ＋ 有效地址

当使用平面存储模型时——像在 64 位模式下——按段基地址为 0 对待。在这样的情况下，线性地址与有效地址相同。在长模式中，线性地址必须用规范的地址形式表示。

4. 物理地址

物理地址是在物理地址空间（典型的为主内存）中的基准。物理地址用页转换机制从虚拟地址转换。当未启用页转换机制时，虚拟（线性）地址作为物理地址使用。

1.1.2 存储器组织

x86-64体系结构将存储器组织为虚拟存储器和物理内存。虚拟存储器和物理内存通常其空间大小能不同，虚拟地址空间比物理内存大得多。系统软件在物理内存和系统硬盘之间重新分配应用程序和数据，以使它比实际存在的有更多的内存可用。然后，系统软件用硬件存储管理机制把更大的虚拟地址空间映射为较小的物理地址空间。

1. 虚拟存储器

软件用虚拟地址访问在虚拟存储空间中的单元。系统软件有责任用段存储管理技术在虚拟存储空间中重定位应用程序和数据。系统软件也有责任通过使用页转换机制将虚拟存储器映射到物理存储器。x86-64体系结构用以下的地址转换模式支持不同的虚拟存储器尺寸：

- 保护模式——此模式用32位虚拟地址支持4GB的虚拟地址空间。
- 长模式——此模式用64位虚拟地址支持2^{64}字节的虚拟地址空间。

2. 物理存储器

物理地址用于直接访问主存。对于一个具体的计算机系统，可用的物理地址空间的尺寸等于安装在系统中的主存的数量。可访问的物理存储器的最大容量，取决于处理器的实现和地址转换模式。x86-64体系结构用以下地址转换模式支持可变的物理存储器尺寸：

- 实地址模式——此模式也称为实模式，用20位物理地址支持1MB的物理地址空间。
- 传统保护模式——此模式支持若干不同的地址空间尺寸，取决于所用的转换机制以及是否启用这些机制的扩展。

  传统保护模式用32位物理地址支持4GB的物理地址。当处理器运行在传统保护模式时，段转换和页转换都能用于访问此物理地址空间。

  当启用物理地址尺寸扩展时，页转换机制能扩展以支持52位物理地址。52位物理地址允许支持多至4×10^{15}B的物理地址空间（当前，x86-64体系结构在此模式下支持40位地址，允许支持多至1×10^{12}B的物理地址空间）。
- 长模式——此模式是x86-64体系结构独有的。此模式用52位物理地址支持多至4×10^{15}B的物理地址。长模式要求使用页转换和物理地址尺寸扩展（PAE）。

1.1.3 规范地址形式

长模式定义64位虚拟地址空间，但处理器实现的支持常常少于64位。虽然某些处理器实现不用全部64位虚拟地址，但它仍检查位63至实现的最高有效位，查这些位是否为全0或全1。这样的地址形式即为规范地址形式。在大多数情况下，引用的虚拟地址不以规范形式表示会引起通用保护异常（#GP）。然而，隐含的堆栈引用，若堆栈地址不是规范形式，则会引起堆栈异常（#SS）。隐含的堆栈引用包括所有的push和pop指令

以及任何以 RSP 或 RBP 作为基寄存器的指令。

由检查规范地址形式，x86-64 体系结构防止软件为了其他目的使用未实现的地址高位。在规定的处理器实现中遵循规范地址形式的软件能不改变地，在支持更大的虚拟地址空间的长模式中运行。

1.2　存储管理

存储管理由这样的方法组成，这些方法使由软件生成的地址用段或页转换为物理地址。存储管理对于应用软件是不可见的。它由系统软件和处理器硬件处理。

1.2.1　段

段最初是为了增加系统的可靠性，在同时运行多个进程的系统上，使软件进程（任务）和这些进程的数据间相互隔离。

x86-64 体系结构支持传统的段的所有形式。然而，大多数现代系统软件不使用在传统的 x86 体系结构中可用的段特性。典型情况下的替代方法是，系统软件用页级保护处理程序和数据的隔离。为此，x86-64 体系结构在 64 位模式下不用多段而用平面存储模型。取消段机制，允许新的 64 位系统软件编码更简单，这样对多进程支持比在传统的 x86 体系结构中的管理更有效。

在兼容模式和传统模式下会使用段。此处，段是基本的存储寻址方式，它允许在虚拟地址空间中软件和数据重定位。软件和数据可用一个或多个可变尺寸的存储段，从而能在虚拟地址空间中重定位。传统的 x86 体系结构提供若干种限制段之间访问的方法，所以，软件和数据能防止相互干扰。

在兼容和传统模式中，能定义多至 16 383 个不同的段。每个段的基地址值、段尺寸（称为段界限）、保护和其他属性包含在称为段描述符的数据结构中。段描述符的集合保持在描述符表中。规定的段描述符用段选择子寄存器从描述符表中引用或选择。有 6 个段选择子寄存器可用，一次可访问多至 6 个段。

图 1-1 显示的是分段存储器的例子。

分段存储器的一种特殊情况是平面存储模型。在传统的平面存储模型中，所有段基地址的值都是 0，段界限固定在 4GB。不能禁止分段，但使用平面存储模型有效地屏蔽了段转换。结果是虚拟地址等于有效地址。图 1-2 显示的是平面存储模型的例子。

1.2.2　分页

分页允许软件和数据用称为物理页的固定尺寸块在物理地址空间中重定位。传统的 x86 体系结构支持 4KB、2MB 和 4MB 3 种不同的物理页尺寸。像段转换，分页能限制较低特权级的软件访问物理页。

页转换用称为页转换表的分层数据结构将虚拟页转换为物理页。页转换表层次的级数最少为 1 级最多为 4 级，取决于物理页尺寸和处理器操作模式。物理页必须在 4KB、

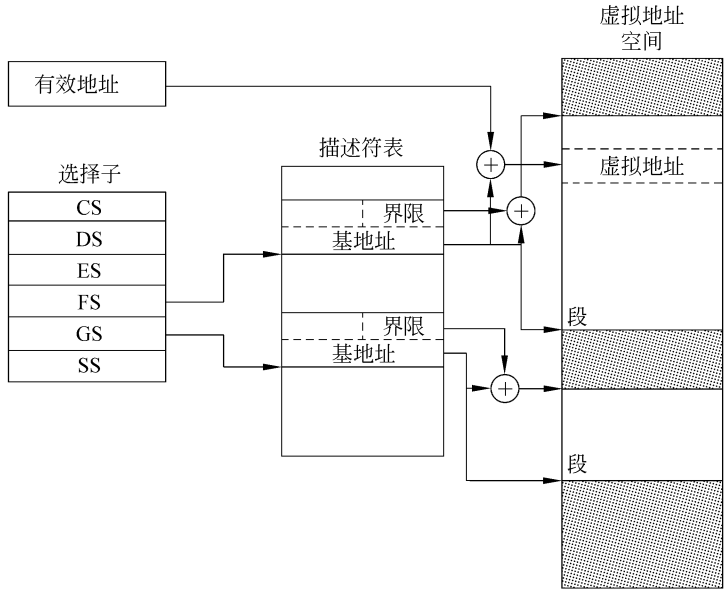

图 1-1 分段的存储器

注：该图取自参考文献[2]，第一篇中的所有图表均取自该文献，以下图、表不再一一注明。

2MB 或 4MB 边界上对齐。

在转换层次中的每个表都由虚拟地址位的一部分作为索引。由表索引引用的项包括在转换层次中下一级表的基地址的指针。在最低级的表中，它的项指向物理页基地址。然后，物理页以虚拟地址的最低有效地址部分作为索引，产生物理地址。

图 1-3 显示的是在转换表层次中有 3 级表的物理存储器的例子。

在长模式中运行的软件要求启动页转换。

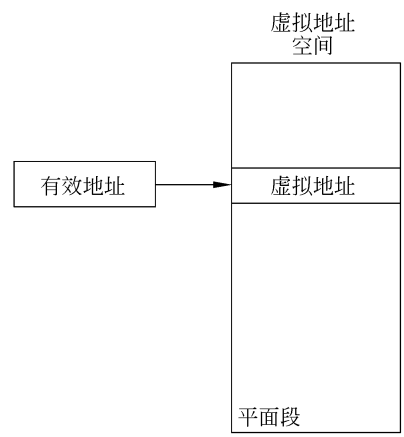

图 1-2 平面存储模型

1.2.3 混合分段和分页

存储管理软件能组合使用分段存储器和分页存储器。因为分段不能被屏蔽，所以分页的存储管理要求分段最少。分页能完全被屏蔽，所以分段存储管理不要求分页资源的初始化。

段长度尺寸的范围能从单个字节至 4GB。因此，可以将多个段映射至单个物理页或将多个物理页映射至单个段。在段和物理页边界之间不要求对齐，但当段和物理页的边界对齐时，能简化存储管理软件。

存储管理的最简单和最有效的方法是平面存储模型。在平面存储模型中，所有段基地址的值都是 0，段界限固定在 4GB。每次在引用内存时，仍使用段机制，但因为在此模型中，虚拟地址与有效地址相同，所以有效地忽略了段机制。虚拟(有效)地址至物理地址的转换由页机制实现。

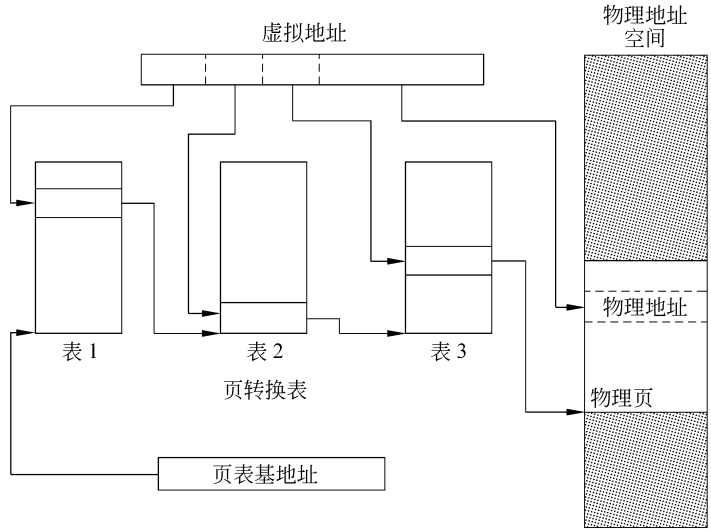

图 1-3　分页存储模型

因为 64 位模式屏蔽分段，所以对于存储管理，它使用平面分页存储模型。在 64 位模式中，忽略了 4GB 段界限。图 1-4 显示了此模型的例子。

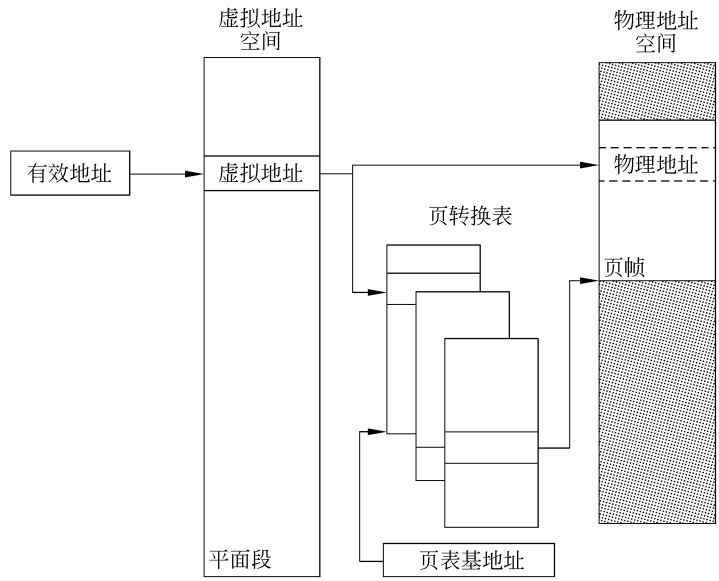

图 1-4　64 位、平面、分页存储模型

1.2.4　实寻址

实寻址是在实模式中使用的用于地址转换的传统方式。地址转换的这种最简单方式与 8086 处理器的有效地址向物理地址的转换后向兼容。在此模式中，16 位有效地址映射至 20 位物理地址，可提供 1MB 物理地址空间。

在实地址转换中用段选择子，但不是作为描述符表的索引。而是此16位段选择子值左移4位形成20位段基地址。16位有效地址加至此20位段基地址，产生20位物理地址。若段基地址和有效地址之和进位至位20，此位可被任意地截断，用A20M#输入信号屏蔽A20地址位，以模仿8086处理器的20位地址环绕。

实地址转换用在16B边界上对齐的段，支持1MB物理地址空间。每个段精确地为64KB长。图1-5显示的是实地址转换的例子。

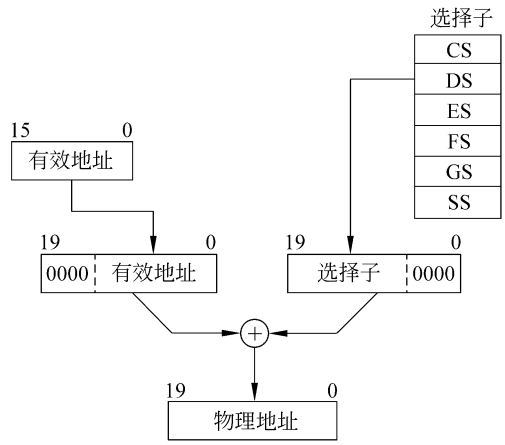

图1-5 实地址存储模型

1.3 操作模式

传统的x86体系结构提供4种操作模式或环境，它们支持各种格式的存储管理、虚拟存储器和物理存储器尺寸和保护：

- 实模式。
- 保护模式。
- 虚拟8086模式。
- 系统管理模式。

x86-64体系结构支持所有这些传统模式，它还增加了一种新的操作模式称为长模式。表1-1显示了长模式和传统模式之间的区别。软件能在所有系统支持的操作模式之间变动，如图1-6所示。以下各节将详细介绍每种操作模式。

1.3.1 长模式

长模式由两种子模式组成：64位模式和兼容模式。64位模式支持许多新的特性，包括寻址64位虚拟地址空间的能力。当运行在64位系统软件上时，兼容模式提供与已存在的16位和32位应用程序的二进制兼容。

表 1-1　操作模式

模　　式		要求的操作系统	应用程序重编译要求	默认①		寄存器扩展②	最大的 GPR 宽度(位)
				地址尺寸(位)	操作数尺寸(位)		
长模式③	64 位模式	新的 64 位操作系统	是	64	32	是	64
	兼容模式		否	32		否	32
				16	16		
传统模式	保护模式	传统的 32 位操作系统	否	32	32	否	32
				16	16		
	虚拟 8086 模式			16	16		
	实模式	传统的 16 位操作系统					

① 默认表示在大多数模式中都能用指令前缀或系统控制位超越；

② 寄存器扩展包括 8 个新 GPR 和 8 个新 XMM 寄存器(也称为 SSE 寄存器)；

③ 长模式只支持 x86 保护模式，不支持 x86 实模式或虚拟 8086 模式。

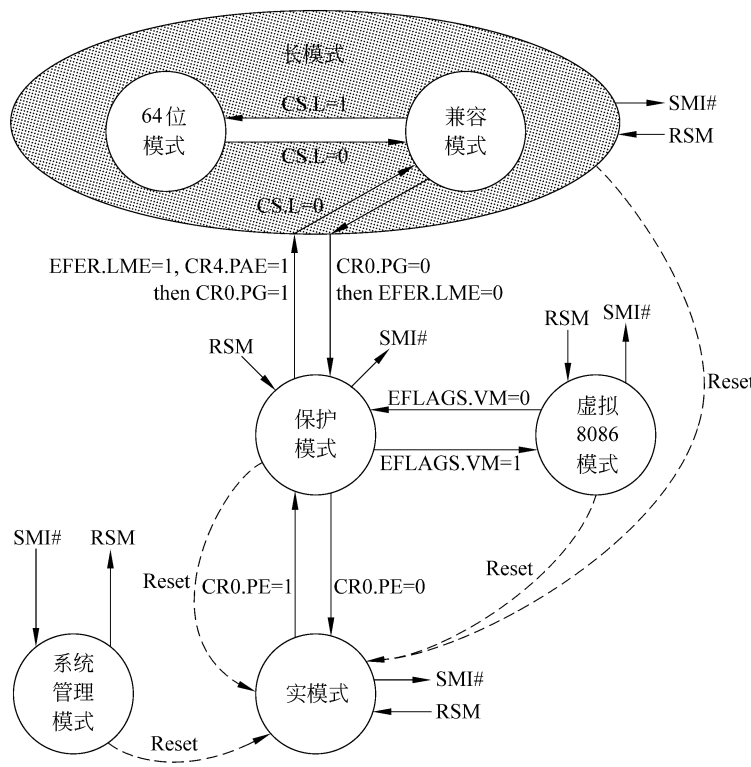

图 1-6　x86-64 体系结构的操作模式

在本书中涉及长模式的内容也同时涉及64位模式和兼容模式。若某一功能是特定用于64位模式或兼容模式，则用特定名代替长模式。

在启用和激活长模式前，系统软件必须首先启用保护模式。

1. 64位模式

64位模式是长模式的一种子模式，通过增加以下新特性为64位系统软件和应用程序提供支持。

- 64位虚拟地址(处理器实现能少于64位)。
- 通过一个新的指令前缀(REX)提供的寄存器扩展：
  - 加8个GPR(R8～R15)。
  - 扩展GPR至64位。
  - 加8个128位流SIMD扩展(SSE)寄存器(XMM8～XMM15)。
- 64位指令指针(RIP)。
- 新的与RIP相对的数据寻址模式。
- 带有单个代码、数据和堆栈空间的平面段地址空间。

此模式由系统软件可在分别的码段基础上启用。为了存储管理要求页转换。因为64位模式支持64位虚拟地址空间，所以它要求64位系统软件和开发工具。

在64位模式中，默认的地址尺寸是64位，默认的操作数尺寸是32位。默认能用指令前缀在逐条指令的基础上超越。为了规定64位操作数尺寸和新的寄存器，引入一个新的REX前缀。

2. 兼容模式

兼容模式也是长模式的一种子模式，允许系统软件实现与已存在的16位和32位x86应用程序的二进制兼容。这些应用程序无须重编译就可在长模式中64位系统软件下运行。

在兼容模式中，应用程序只能访问虚拟地址空间的前4GB。用标准的x86指令前缀可在16位和32位地址和操作数之间切换。

兼容模式与64位模式相同的是，由系统软件可在分别的码段基础上启用。与64位模式不同的是，段功能用16位或32位保护模式语法，这一点与在传统的x86体系结构中相同。从应用程序的角度看，兼容模式看上去很像传统的保护模式环境。从系统软件的角度看，地址转换、中断和异常处理以及系统数据结构，用长模式机制。

1.3.2 传统模式

传统模式由3种子模式组成：实模式、保护模式和虚拟8086模式。保护模式可以是分页的或不分页的。传统模式不只与已存在的16位和32位应用程序二进制兼容，也与已存在的16位和32位系统软件二进制兼容。

1. 实模式

实模式也称为实地址模式，在此模式中，处理器支持1MB的物理存储器空间和16位

(默认)或 32 位(用指令前缀)的操作数尺寸。该模式不支持分页。所有软件都运行在特权级 0。

在复位或处理器上电后进入实模式。当处理器在长模式下操作时不支持此模式,因为长模式要求启用页保护模式。

2. 保护模式

在此模式下,处理器支持 4GB 的虚拟存储器和物理存储器空间以及 16 位或 32 位操作数尺寸。所有的段转换、段保护和硬件多任务功能都是可用的。系统软件能通过分段技术在虚拟地址空间中重定位有效地址。若不启用分页机制,虚拟地址与物理地址相等。在这种模式下可任意启用分页机制以允许虚拟地址至物理地址的转换和基于页的存储保护机制。

在保护模式下,软件运行在特权级 0、1、2 或 3。典型地,应用软件运行在特权级 3,系统软件运行在特权级 0 和 1,特权级 2 可用于系统软件或其他应用。

3. 虚拟 8086 模式

虚拟 8086 模式允许系统软件在虚拟的 8086 处理器上运行 16 位实模式软件。在此模式下,为 8086、8088、80186 或 80188 处理器编写的软件能在保护模式下作为特权级 3 的任务运行。处理器支持 1MB 的虚拟存储器空间和 16 位(默认)或 32 位(用指令前缀)操作数尺寸,以及采用实模式地址转换。

虚拟 8086 模式由设置在 EFLAGS 寄存器(EFLAGS. VM)中的虚拟机器位启用。只能当 EFLAGS 寄存器作为任务切换的结果装入,或由特权软件执行 IRET 指令从 TSS 装入时,EFLAGS. VM 位才能被设置或清除。POPF 指令不能用于设置或清除 EFLAGS. VM 位。

当处理器运行在长模式下时,虚拟 8086 模式是不支持的。

1.3.3　系统管理模式

系统管理模式(SMM)是一种设计用于系统控制活动的运行模式,它在通常情况下对于常规的系统软件是透明的。电源管理是系统管理模式的一种流行应用。SMM 的最初目标由基本输入输出系统(BIOS)和特殊的低级设备驱动程序使用。对于 SMM 的代码和数据存储在 SMM 内存区,它由 SMM 输出信号与主存隔离。

SMM 由系统管理中断(SMI)进入,直至处理器识别为 SMM,处理器才进入和切换至对应的 SMM 地址空间,然后处理程序存放和执行。在 SMM 中,处理器支持实模式寻址,具有 4GB 段界限和 16 位的默认的操作数、地址和堆栈尺寸(能用前缀超越这些默认值)。

1.4　系统寄存器

图 1-7 显示的是 x86-64 体系结构定义的系统寄存器。系统软件将这些寄存器用于管理处理器操作环境、定义系统资源特性和监视软件执行。除了 RFLAGS 寄存器外,系