



[12] 发明专利申请公布说明书

[21] 申请号 200580035526.2

[43] 公开日 2007 年 9 月 26 日

[11] 公开号 CN 101044458A

[22] 申请日 2005.10.31

[21] 申请号 200580035526.2

[30] 优先权

[32] 2004.10.29 [33] US [31] 10/976,970

[86] 国际申请 PCT/US2005/040450 2005.10.31

[87] 国际公布 WO2006/050534 英 2006.5.11

[85] 进入国家阶段日期 2007.4.17

[71] 申请人 英特尔公司

地址 美国加利福尼亚州

[72] 发明人 S·贝内特 G·奈格尔

A·安德森

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 曾祥凌 梁永

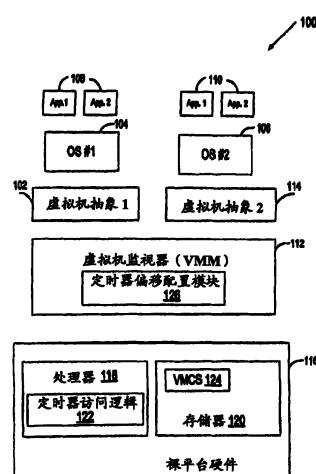
权利要求书 5 页 说明书 14 页 附图 5 页

[54] 发明名称

虚拟机环境中的定时器偏移机制

[57] 摘要

在一个实施例中，一种方法包括从虚拟机监视器(VMM)接收将控制转移到虚拟机(VM)的请求，计算偏移值，在VM的操作期间接收对定时器的当前值的请求，基于偏移值调整定时器的当前值，并将已调整的定时器值提供给VM。



1. 一种方法，包括：

从虚拟机监视器（VMM）接收将控制转移到虚拟机（VM）的请求；

计算偏移值；

在所述 VM 的操作期间接收对定时器的当前值的请求；

基于所述偏移值调整所述定时器的当前值；以及

将已调整的定时器值提供给所述 VM。

2. 如权利要求 1 所述的方法，还包括：

当从所述 VMM 接收到所述请求时将控制转移到所述 VM，所述转移包括计算所述偏移值。

3. 如权利要求 1 所述的方法，其特征在于，计算所述偏移值包括：

确定在接收到将控制转移到所述 VM 的请求时读取的定时器值与检测到与将控制从所述 VM 转移到所述 VMM 关联的先前事件时保存的定时器值之间的差值；以及

将定时器偏移字段的值减去所述差值。

4. 如权利要求 3 所述的方法，还包括将计算的定时器偏移值存储到所述定时器偏移字段中。

5. 如权利要求 3 所述的方法，还包括：

在确定所述差值之前，确定调整定时器偏移指示符被设为启用值。

6. 如权利要求 1 所述的方法，还包括：

确定调整定时器偏移指示符被设为已禁用值；以及

读取定时器偏移字段，此后使用所述定时器偏移字段的值作为所述偏移值。

7. 如权利要求 3 所述的方法，其特征在于，如果保存定时器指

示符被启用，则在检测到与将控制转移到所述 VMM 关联的先前事件时保存所述定时器值。

8. 如权利要求 1 所述的方法，还包括：

确定定时器访问控制指示符被设为退出值；以及
响应所述请求将控制转移到所述 VMM。

9. 如权利要求 1 所述的方法，还包括：

确定所述定时器的偏移被禁用；以及
以零加载偏移寄存器。

10. 一种设备，包括：

虚拟机监视器（VMM）；

由所述 VMM 控制的数据结构，所述数据结构存储虚拟机（VM）
的偏移值；以及

定时器访问逻辑，所述定时器访问逻辑在从所述 VMM 接收到
将控制转移到所述 VM 的请求时计算所述偏移值，并在所述 VM 的
操作期间将所述定时器的值提供给所述 VM，所述定时器的值基于所
述偏移值进行了调整。

11. 如权利要求 10 所述的设备，其特征在于，所述定时器访问
逻辑还在从所述 VMM 接收到所述请求时将控制转移到所述 VM，所
述转移包括计算所述偏移值。

12. 如权利要求 10 所述的设备，其特征在于，所述定时器访问
逻辑通过如下步骤计算所述偏移值：确定在接收到将控制转移到所
述 VM 的请求时读取的定时器值与检测到与将控制从所述 VM 转移
到所述 VMM 关联的先前事件时保存的定时器值之间的差值，并将
定时器偏移字段的值减去所述差值。

13. 如权利要求 12 所述的设备，其特征在于，所述数据结构还
包括存储所计算的定时器偏移值的定时器偏移字段。

14. 如权利要求 12 所述的设备，其特征在于，所述数据结构还
包括调整定时器偏移指示符。

15. 如权利要求 14 所述的设备，其特征在于，所述定时器访问逻辑还在确定所述差值之前确定所述调整定时器偏移指示符被设为启用值。

16. 如权利要求 12 所述的设备，其特征在于，所述定时器访问逻辑还确定调整定时器偏移指示符被设为已禁用值，并且读取定时器偏移字段，此后使用所述定时器偏移字段的值作为所述偏移值。

17. 如权利要求 12 所述的设备，其特征在于，所述数据结构存储所述定时器访问逻辑使用的保存定时器指示符，以确定是否要在检测到与将控制转移到所述 VMM 关联的先前事件时保存所述定时器值。

18. 如权利要求 10 所述的设备，还包括：

偏移寄存器，所述偏移寄存器用于存储所述偏移值。

19. 一种系统，包括

存储器，所述存储器用于存储虚拟机（VM）的与定时器偏移关联的一组字段；以及

与所述存储器耦合的处理器，所述处理器用于使用与定时器偏移关联的一组字段计算偏移值，并响应所述 VM 对定时器的当前值的请求，在所述 VM 的操作期间将基于所述偏移值调整的定时器值提供给所述 VM。

20. 如权利要求 19 所述的系统，其特征在于，所述处理器在将控制转移到所述 VM 时计算所述偏移值。

21. 如权利要求 19 所述的系统，其特征在于，所述处理器还以所述偏移值加载偏移寄存器。

22. 如权利要求 19 所述的系统，其特征在于，所述处理器通过如下步骤计算所述偏移值：确定在接收到将控制转移到所述 VM 的请求时读取的定时器值与检测到与将控制从所述 VM 转移到所述 VMM 关联的先前事件时保存的定时器值之间的差值，并将定时器偏移字段的值减去所述差值。

23. 如权利要求 22 所述的系统，其特征在于，如果调整定时器偏移指示符被启用，则所述处理器在接收到将控制转移到所述 VM 的请求时读取所述定时器值。

24. 如权利要求 22 所述的系统，其特征在于，如果保存定时器指示符被启用，则所述处理器在检测到与将控制转移到所述 VMM 关联的先前事件时保存所述定时器值。

25. 一种包含指令的机器可读媒体，当所述指令被处理系统执行时导致所述处理系统执行一种方法，所述方法包括：

从虚拟机监视器（VMM）接收将控制转移到虚拟机（VM）的请求；

计算偏移值；

在所述 VM 的操作期间接收对定时器的当前值的请求；

基于所述偏移值调整所述定时器的当前值；以及

将已调整的定时器值提供给所述 VM。

26. 如权利要求 25 所述的机器可读媒体，其特征在于，所述方法还包括：

当从所述 VMM 接收到所述请求时将控制转移到所述 VM，所述转移包括计算所述偏移值。

27. 如权利要求 25 所述的机器可读媒体，其特征在于，计算所述偏移值包括：

确定在接收到将控制转移到所述 VM 的请求时读取的定时器值与检测到与将控制从所述 VM 转移到所述 VMM 关联的先前事件时保存的定时器值之间的差值；以及

将定时器偏移字段的值减去所述差值。

28. 如权利要求 27 所述的机器可读媒体，其特征在于，如果调整定时器偏移指示符被启用，则在接收到将控制转移到所述 VM 的请求时读取所述定时器值。

29. 如权利要求 27 所述的机器可读媒体，其特征在于，如果保

存定时器指示符被启用，则在检测到与将控制转移到所述 VMM 关联的先前事件时保存所述定时器值。

虚拟机环境中的定时器偏移机制

技术领域

本发明的实施例一般涉及虚拟机，更具体地说涉及虚拟机环境中的定时器偏移机制。

背景技术

定时器通常被操作系统和应用软件用于调度活动。例如，操作系统内核可以使用定时器来使多个用户级应用程序对系统资源（例如中央处理单元（CPU））实施时间共享。个人计算机（PC）平台上使用的定时器的示例是 8254 可编程间隔定时器。该定时器可以配置成在指定的间隔之后或周期性地发出中断。

定时器的另一个示例是 Intel® Pentium® 4 的指令集体系统结构（ISA）（本文称为 IA-32 ISA）中使用的时间戳计数器（TSC）。TSC 是一种 64 位计数器，它在处理器的硬件复位之后被设为 0，然后即使是在处理器被 HLT 指令暂停时，仍按每个处理器时钟周期递增。TSC 无法用于生成中断。它仅是时间参考，仅对于测量时间间隔有用。IA-32 ISA 提供指令（RDTSC）来读取 TSC 的值，以及提供指令（WRMSR）来写入 TSC。当使用 WRMSR 来写入时间戳计数器时，仅可以写入 32 个低阶位，而将 32 个高阶位清零。因为这些写限制，软件一般无法将 TSC 向前设置到任意值，也无法将 TSC 向后设置到任意值。

在多处理器（MP）系统中，使用 TSC 来恰当地将处理器同步并适合地调度进程。在引导时间，将所有处理器上 TSC 的值同步，并且大多数操作系统假定 TSC 按相同速率计数。如果 TSC 值在处理器之间偏离（例如如果一个处理器按与其他处理器不同的速率计数），则可能使操作系统对进程的调度紊乱。

附图说明

通过示例形式而非限定形式在附图中图示了本发明，其中相似的引用号指代相似的元素，其中：

图 1 图示本发明可以工作的虚拟机环境的一个实施例。

图 2 是用于控制 VM 对定时器的访问的过程的一个实施例的流程图；

图 3 是用于配置与定时器偏移关联的字段的过程的一个实施例的流程图；以及

图 4 和 5 是用于计算 VM 的定时器偏移值的过程的两个备选实施例的流程图。

发明内容

描述用于控制虚拟机对定时器的访问的方法和设备。在上文说明中，为了解释的目的，提出了许多特定细节，以便提供对本发明的透彻理解。但是对于本领域人员来说，显然本发明可以在没有这些特定细节的情况下实施。

下文的详细描述的一些部分是依据对计算机系统的寄存器或存储器内的数据比特的操作的算法和符号表示来给出的。这些算法描述和表示是数据处理领域的技术人员将他们的工作实质最有效地传达给该领域的其他技术人员所用方式。在本文中以及一般情况下，将算法设想为导致期望结果的独立次序的操作。这些操作是需要对物理量进行物理操纵的那些操作。通常但非一定，这些量采用能够被存储、传输、组合、比较以及以其他方式操纵的电或磁信号的形式。以位、值、元素、符号、字符、术语、数字等提及这些信号时常被证明是方便的，主要是因为常用。

但是，应该记住的是，所有这些和相似术语均要与适合的物理量关联，并且仅仅是应用于这些量的便利标号。除非下文论述中显

著地另外专门指明，否则要认识到在本发明中，采用诸如“处理”或“计算”或“运算”或“确定”等术语的论述可以指计算机系统或相似的电子计算装置的动作和进程，这些动作和进程操纵计算机系统的寄存器和存储器内表示为物理（电子）量的数据并将其转换成计算机系统存储器或寄存器或其他此类信息存储器、传输或显示装置内以相似方式表示为物理量的其他数据。

在下文对这些实施例的详细描述中，参考了附图，这些附图以说明方式示出可以实施本发明的特定实施例。在这些附图中，多个图中相似的引用号指代基本相似的组件。对这些实施例给予了充分详细的描述，以使本领域技术人员能够实施本发明。可以利用其他实施例，在不背离本发明范围的前提下可以进行结构、逻辑和电方面的更改。而且，要理解本发明的多个不同实施例虽然不同但是并不一定是相互排斥的。例如，在一个实施例中描述的特定功能特征、结构或特征可以被包括在另一个实施例内。因此下文的详细描述不应视为限定意义的，本发明的范围仅由所附权利要求以及对此类权利要求赋予权利的等效物的完整范围来限定。

虽然下文示例描述在执行单元和逻辑电路的环境中控制虚拟机对定时器的访问，但是本发明的其他实施例可以通过软件来实现。例如，在一些实施例中，本发明可以作为计算机程序产品或软件来提供，这些计算机程序产品或软件可以包括其上存储有指令的机器或计算机可读媒体，这些指令可以用于对计算机（或其他电子装置）编程以执行根据本发明的过程。在其他实施例中，本发明的过程可以由包含用于执行这些过程的硬连线逻辑的特定硬件组件来执行，或通过将编程的计算机组件与定制的硬件组件的任何组合来执行。

因此，机器可读媒体可以包括用于存储或传输以机器（例如计算机）可读形式的信息的任何机制，但是不限于软盘、光盘、压缩光盘、只读存储器（CD-ROM）以及磁光盘、只读存储器（ROM）、随机存取存储器（RAM）、可擦写可编程只读存储器（EPROM）、电

可擦写可编程只读存储器（EEPROM）、磁卡或光卡、闪速存储器、通过因特网的传输、电、光、声波或其他形式的传播信号（例如载波、红外线信号、数字信号等）等。

另外，设计可能经历多个不同阶段，从建立到仿真到制造。表示设计的数据能以多种方式来表示该设计。首先，正如有利于仿真的那样，可以使用硬件描述语言或另一种功能描述语言来表示硬件。此外，还可以在设计过程的一些阶段制作具有逻辑和/或晶体管门电路的电路级模型。而且，在某个阶段，大多数设计达到表示多种装置在硬件模型中的物理布置的数据级。在使用常规半导体制造技术的情况下，表示硬件模型的数据可以是指定制造集成电路所用的掩模的不同掩模层上存在或不存在各种特征的数据。在设计的任何表示中，数据能以任何形式的机器可读媒体来存储。经调制或以其他方式生成以传输此类信息的光波或电波、诸如盘的存储器或磁或光存储器均可以是机器可读媒体。这些媒体的任何一个可以“承载”或“指示”设计或软件信息。当传送指示或承载代码或设计的电载波，且范围涉及到执行电信号的复制、缓冲或重新传送时，制作新副本。因此，通信提供商或网络提供商可以制作实施本发明技术的物品（载波）的副本。

图 1 图示本发明可以工作的虚拟机环境 100 的一个实施例。在该实施例中，裸平台硬件 116 包括计算平台，该计算平台能够例如执行标准操作系统（OS）或例如 VMM 112 的虚拟机监视器（VMM）。

通常以软件形式实现的 VMM 112 可以仿真并导出与较高级软件的裸机器接口。此类较高级软件可以包括标准或实时 OS，并可以是具有有限操作系统功能的高度剥离的操作环境，可以不包括传统的 OS 实用程序等。或者，例如 VMM 112 可以在另一个 VMM 内运行或在另一个 VMM 上运行。VMM 能以例如硬件、软件、固件或多种技术组合的形式来实现。

平台硬件 116 可以是个人计算机（PC）、主机、手持装置、便携

式计算机、机顶盒或任何其他计算系统。平台硬件 116 包括处理器 118 和存储器 120。

处理器 118 可以是能够执行软件的任何类型的处理器，例如微处理器、数字信号处理器、微控制器等。处理器 118 可以包括用于执行实现本发明方法实施例的的微代码、可编程逻辑或硬编码逻辑。虽然图 1 仅示出一个此类处理器 118，但是在系统中可以有一个或多个处理器。

存储器 120 可以是硬盘、软盘、随机存取存储器 (RAM)、只读存储器 (ROM)、闪速存储器、上面这些装置的任何组合或处理器 118 可读的任何其他类型的机器媒体。存储器 120 可以存储用于实现本发明方法实施例的指令和/或数据。

VMM 112 向其他软件（即“访客”软件）呈示一个或多个虚拟机 (VM) 的抽象，这样可以向各种访客提供相同或不同的抽象。图 1 示出 VM 102 和 114。运行于每个 VM 上的访客软件可以包括例如访客 OS 104 或 106 的访客 OS 以及各种访客软件应用 108 和 110。访客 OS 104 和 106 的每一个期望访问访客 OS 104 或 106 在其上运行的 VM 102 和 114 内的物理资源（例如处理器寄存器、存储器和 I/O 装置）并执行其他功能。例如，根据 VM 102 和 114 中呈示的处理器和平台的体系结构，访客 OS 104 和 106 期望对所有寄存器、高速缓存、结构、I/O 装置、存储器等具有访问权。访客软件可以访问的资源可以分类为“特权”或“非特权”资源。对于特权资源，VMM 112 协助实现访客软件期望的功能同时保持对这些特权资源的最终控制。非特权资源不需要被 VMM 112 控制，并且可以被访客软件访问。

而且，每个访客 OS 期望能够处理各种错误事件，例如异常（例如，页错误、一般性保护错误等）、中断（例如硬件中断、软件中断）以及平台事件（例如初始化 (INIT) 和系统管理中断 (SMI)）。这些错误事件的其中一些是“特权的”，因为它们必须由 VMM 112 来处理以便确保 VM 102 和 114 的正确操作以及针对访客软件和访客软件

之间的保护。

当发生特权错误事件或访客软件试图访问特权资源时，可以将控制转移到 VMM 112。将控制从访客软件转移到 VMM 112 在本文称为 VM 退出。在协助资源访问或适当地处理事件之后，VMM 112 可以将控制返回给访客软件。将控制从 VMM 112 转移到访客软件在本文称为 VM 进入。

在一个实施例中，处理器 118 根据存储在虚拟机控制结构 (VMCS) 124 中的数据控制 VM 102 和 114 的操作。VMCS 124 是一种可以包含访客软件的状态、VMM 112 的状态、指示 VMM 112 期望如何控制访客软件的操作的执行控制信息、VMM 112 和 VM 之间的信息控制转移等的结构。处理器 118 从 VMCS 124 读取信息以确定 VM 的执行环境并约束它的行为。在一个实施例中，VMCS 被存储在存储器 120 中。在一些实施例中，使用多个 VMCS 结构来支持多个 VM。

VMM 112 可能需要使用定时器来调度资源、提供服务质量、确保安全性并执行其他功能。例如，在 Intel® Pentium® 4 的指令集体体系结构 (ISA) (本文称为 IA-32 ISA) 中，VMM 112 可以使用时间戳计数器 (TSC) 来执行这些功能。VM 102 和 114 的每一个还可能需要使用定时器来校准定时循环并执行性能优化。因为 VM 102 和 114 彼此不知道对方或不知道 VMM 112，所以可能需要调整提供给 VM 102 或 114 的定时器的值以呈示客户 OS 104 或 106 正在专用硬件平台而非虚拟平台上运行的假象。提供定时器偏移机制以适当地将定时器虚拟化，由此为访客 OS 104 和 106 保留这种假象。在一个实施例中，定时器偏移机制包括定时器偏移配置模块 126 和定时器访问逻辑 122。

定时器偏移配置模块 126 负责在请求将控制转移到 VM 102 或 114 之前为与定时器偏移关联的字段提供值。在一个实施例中，这些值可以包括指定将定时器值提供给 VM 102 或 114 时处理器 118 要使

用的偏移量的偏移值以及指定是否为 VM 102 或 114 启用定时器偏移的定时器偏移指示符。在一个实施例中，定时器偏移值是带符号的值，这样使 VMM 112 能够向访客软件呈示是小于还是大于实际硬件定时器值的定时器值。在一个实施例中，在将值返回给 VM 102 或 114 之前将定时器偏移值加上定时器值。在一个实施例中，与定时器偏移关联的字段还包括定时器访问控制指示符，它指定访问定时器的 VM 请求是否与将控制转移到 VMM 关联（例如访问定时器的 VM 请求是否应该导致 VM 退出）。

在一个实施例中，VM 102 或 114 的偏移值考虑了此 VM 因 VMM 112 和其他 VM 的执行而未运行期间的时间间隔的累积。例如，假定当定时器的值是 1000 个节拍 (tick) 时，VM 102 期望定时器的值是 1000。这样，在 1500 个节拍时，VM 102 可能因 VM 退出被中断，然后是 VMM 112 执行持续 100 个节拍 (1600 个节拍的定时器值)，之后 VMM 112 可以请求进入 VM 114。然后 VM 114 可以执行持续 600 个节拍 (2200 个节拍的定时器值) 直到 VM 退出，VM 退出导致 VMM 112 执行持续 200 个节拍 (定时器值 2400)，然后可以请求重新进入 VM 102。在重新进入的时间，VM 102 预期定时器具有 1500 个节拍的值。相反，该时间的实际定时器值是 2400 个节拍。VMM 112 为 VM 102 提供的偏移值将是 900 个节拍，这是 VM 112 因 VM 114 和 VMM 112 的执行而未运行期间的时间间隔的累积。因此，VMM 112 会将 900 的偏移值存储到定时器偏移字段，以便当 VM 102 尝试读取定时器时，将通过将当前定时器值减去 900 来计算提供给 VM 102 的值。还可以通过计算实际定时器值与 VM 102 预期的定时器值之间的差来计算偏移值。在一个备选实施例中，通过将当前定时器值加上 VMM 112 配置的偏移值来计算 VM 102 尝试读取定时器时提供给 VM 102 的值。在该实施例中，VMM 存储的偏移值是负数。在上文描述的示例中，所存储的值是 -900。

如上文论述的，在一个实施例中，偏移值由 VMM 112 确定。在

一个备选实施例中，偏移值由处理器 118 确定，与定时器偏移关联的字段和控制可以包括定时器偏移指示符、调整偏移指示符、访客定时器字段、保存定时器指示符和定时器偏移字段。在一个实施例中，能以多种方式组合这些三个指示符值。例如，调整偏移指示符和保存定时器指示符可以是相同的控制（即启用定时器偏移的调整隐含地启用定时器的保存），下文将对此予以更详细的描述。在本发明的一些实施例中，可以不存在上面这些指示符的其中一些。例如，可以不存在定时器偏移指示符，并且假定定时器偏移始终是启用的，下文将对此予以更详细的论述。

在一个实施例中，将与定时器偏移关联的字段和控制存储在 VMCS 124 中。或者，与定时器偏移关联的字段和控制可以驻留在处理器 118、存储器 120 和处理器 118 的组合或任何其他一个或多个存储单元中。在一个实施例中，为 VM 102 和 114 的每一个维护与定时器偏移关联的单独字段和控制。或者，为 VM 102 和 144 维护与定时器偏移关联的相同字段和控制，并由 VMM 112 在每次 VM 进入之前对它们进行更新。

在一个实施例中，在系统 100 包括多个处理器、多个核或多个线程处理器的情况下，将多个逻辑处理器的每一个与定时器偏移关联的单独字段和控制关联，VMM 112 为这些多个逻辑处理器的每一个配置与定时器偏移关联的字段和控制。

在一个实施例中，处理器 118 包括定时器访问逻辑 122，定时器访问逻辑 122 负责基于定时器偏移值将 VM 102 和 114 对定时器的访问虚拟化。具体来说，如果定时器访问逻辑 122 确定启用了定时器偏移，则它将已调整的定时器值提供给 VM 102 或 114。在一个实施例中，定时器访问逻辑 122 通过检查定时器偏移指示符值来确定定时器偏移是否被启用。在一个实施例中，当定时器访问逻辑 122 从 VM 102 或 114 接收到对定时器的当前值的请求时，它读取定时器的当前值，并将偏移值加定时器的当前值，并将结果值返回给 VM 102

或 114，由此向 VM 102 或 114 呈示它正在专用的硬件平台上运行的假象。在一个实施例中，偏移值是带符号的值。

可以由 VMM 112 为 VM 102 和 114 确定偏移值（例如通过定时器偏移配置模块 126）。下文将结合图 3 和图 4 更详细地论述由 VMM 112 确定偏移值的过程的一个实施例。或者，可以由处理器 118 来确定偏移值（例如通过定时器访问逻辑 122）。下文将结合图 5 更详细地论述由处理器 118 确定偏移值的过程的一个实施例。

图 2 是用于控制 VM 对定时器的访问的过程 200 的一个实施例的流程图。该过程可以由处理逻辑来执行，该处理逻辑可以包括硬件（例如电路、专用逻辑、可编程逻辑、微代码等）、软件（如通用计算机系统或专用机器上运行的软件）或二者的组合。在一个实施例中，过程 200 由图 1 的定时器访问逻辑 122 来执行。

参考图 2，过程 200 开始于处理逻辑从 VMM 接收到将控制转移到 VM 的请求（即 VM 进入的请求）（处理框 202）。在一个实施例中，通过 VMM 执行的 VM 进入指令接收 VM 进入请求。

接下来，处理逻辑确定定时器偏移是否被启用（处理框 204）。在一个实施例中，作为向 VM 转移的一部分，处理逻辑作出此确定（例如当检查并加载存储在 VMCS 中的 VM 状态并执行控制信息时）。在一个实施例中，该确定是基于对应于正在进入的 VM 的存储在 VMCS 中的定时器偏移指示符的当前值。

如果定时器偏移被启用，则处理逻辑在响应 VM 对当前定时器值的请求时使用定时器偏移值。在一个实施例中，在发出将控制转移到该 VM 的请求之前由 VMM 确定该定时器偏移值。下文将结合图 3 和图 4 更详细地论述由 VMM 确定偏移值的过程的一个实施例。或者，可以在将控制转移到此 VM 时由处理器自动地确定偏移值。下文将结合图 5 更详细地论述由处理器确定偏移值的过程的一个实施例。

在一个实施例中，如果定时器偏移被启用，则处理逻辑以存储

在 VMCS 中的定时器偏移值加载偏移寄存器（处理框 206）。或者，如果定时器偏移被禁用，则处理逻辑以 0 加载偏移寄存器（处理框 206）。接下来，处理逻辑开始 VM 中的执行（处理框 210）。

在 VM 的执行期间，处理逻辑可能接收到 VM 对定时器的当前值的请求。例如在 IA-32 ISA 中，VM 可能通过执行 RDMSR 指令或 RDTSC 指令发出对 TSC 的当前值的请求以读取 TSC。

在处理框 212 中，处理逻辑确定 VM 是否请求定时器的当前值，如果是这样的话，则在一个实施例中，处理逻辑确定该请求是否与将控制转移到 VMM 关联（处理框 214）。在一个实施例中，可以将定时器访问控制指示符设为“退出”值，以在 VM 访问定时器的每次请求时使该 VM 退出。例如，在一个实施例中，定时器访问控制指示符为一个比特，如果设为 1，则指示 VM 访问定时器的请求使 VM 退出。在一个实施例中，可以将该指示符存储在 VMCS 中。如果请求不与将控制转移到 VMM 关联，则处理逻辑继续进行到处理框 216。如果请求与将控制转移到 VMM 关联，则处理逻辑将控制转移到 VMM，向 VMM 指示尝试访问定时器导致了 VM 退出（处理框 220）。在一个实施例中，在将控制转移到 VMM 之前，处理逻辑以 0 加载偏移寄存器（处理框 218）以使 VMM 能够获取定时器的实际值。

在另一个实施例中，不使用定时器访问控制指示符，并且处理逻辑响应 VM 对定时器的当前值的请求并不检查将控制转移到 VMM。相反，处理逻辑跳过处理框 214，并直接继续进行到处理框 216。

在处理框 216 中，处理逻辑读取定时器的当前值，将偏移值加定时器的当前值，并将结果返回给 VM。在一个实施例中，定时器偏移值是使用带符号的加法与定时器的内容组合的带符号的值。

在 VM 的执行期间，可能发生与 VM 退出关联的多种其他事件（例如页错误、中断等）。在一个实施例中，如果处理逻辑检测到与 VM 退出关联的事件（处理框 222），则处理逻辑以 0 加载偏移寄存

器（处理框 218）并将控制转移到 VMM，由此指示处理框 222 中检测到的 VM 退出的起因（处理框 220）。如果处理逻辑未检测到与 VM 退出关联的任何事件，则处理逻辑返回到处理框 212。

图 3 是用于配置与定时器偏移关联的字段的过程 300 的一个实施例的流程图。该过程可以由处理逻辑来执行，处理逻辑可以包括硬件（例如电路、专用逻辑、可编程逻辑、微代码等）、软件（例如在通用计算机系统或专用机器上运行的软件）或而二者的组合。在一个实施例中，过程 300 由图 1 中定时器偏移配置模块 216 来执行。

参考图 3，过程 300 开始于处理逻辑确定需要将控制转移到 VM。在发出将控制转移到 VM 的请求之前，处理逻辑确定 VM 的定时器偏移值（处理框 302）并将定时器偏移值存储在 VMCS 中（处理框 304）。在一个实施例中，定时器偏移值是正在进入的 VM 因 VMM 和其他 VM 的执行而未运行期间的时间间隔的累积。下文将接合图 4 更详细地论述计算定时器偏移值的过程的一个实施例。

接下来，处理逻辑将定时器偏移指示符设为启用值（处理框 306），并发出将控制转移到 VM 的请求（例如 VM 进入请求）（处理框 308）。

此后，当生成从 VM 的 VM 退出时，处理逻辑收回控制（处理框 310），确定定时器的当前值（处理框 312），并按需要处理 VM 退出（例如执行找出 VM 退出原因的操作）（处理框 314）。如下文描述的，在将控制返回给 VM 之前，可以使用 VM 退出时的定时器的值来计算定时器偏移。

图 4 是用于计算 VM 的定时器偏移值（例如参考图 3 的处理框 302）的过程 400 的一个实施例的流程图。该过程可以由处理逻辑来执行，处理逻辑可以包括硬件（例如电路、专用逻辑、可编程逻辑、微代码等）、软件（例如在通用计算机系统或专用机器上运行的软件）或而二者的组合。在一个实施例中，过程 400 由图 1 中定时器偏移配置模块 216 来执行。

参考图 4，过程 400 开始于处理逻辑计算自上次 VM 进入起在此 VM 中度过的时间（处理框 402）。在一个实施例中，该时间通过如下步骤计算：确定 VM-进入时间（即刚好发出进入 VM 的请求之前的定时器值）和 VM-退出时间（例如接收从 VM 返回的控制时的定时器值），并将 VM-退出时间减去 VM-进入时间。

在处理框 404 中，处理逻辑通过如下步骤计算在 VM 中度过的累计时间：将上次进入期间 VM 中度过的时间加上先前计算的累计时间。在一个实施例中，在处理逻辑接收从 VM 返回的控制时计算在 VM 中度过的累计时间。或者，在处理逻辑发出将控制返回给 VM 的请求时计算在此 VM 中度过的累计时间。

当处理逻辑决定将控制返回给 VM 时，它读取定时器的当前值（处理框 406），并作为在 VM 中度过的累计时间与定时器的当前值之间的差值计算定时器偏移值（处理框 408）。在一个实施例中，定时器偏移值是带符号的值。

图 5 是用于计算 VM 的定时器偏移值的过程的备选实施例的流程图。该过程可以由处理逻辑来执行，处理逻辑可以包括硬件（例如电路、专用逻辑、可编程逻辑、微代码等）、软件（例如在通用计算机系统或专用机器上运行的软件）或而二者的组合。在一个实施例中，过程 500 由图 1 的定时器偏移逻辑 122 来执行。

参考图 5，过程 500 开始于处理逻辑检测到与将控制从 VM1 转移到 VMM 关联的事件（处理框 502）。

在处理框 504，处理逻辑确定保存定时器指示符是否被启用。在一个实施例中，保存定时器指示符由 VMM 来配置并存储在 VMCS 中。

如果保存定时器指示符被启用，则处理逻辑将当前定时器值作为 VM1 定时器值保存到访客定时器字段（处理框 506），并继续进行到处理框 508。在一个实施例中，将访客定时器字段存储在 VMCS 中。

如果保存定时器指示符被禁用，则处理逻辑跳过处理框 506，并直接继续进行到处理框 508。

在处理框 508，处理逻辑将控制转移到 VMM。

此后，在处理框 510 中，处理逻辑接收到将控制返回给 VM1 的请求。对此响应，处理逻辑确定调整偏移指示符是否被启用（处理框 512）。在一个实施例中，调整偏移指示符由 VMM 来配置并存储在 VMCS 中。在另一个实施例中，保存定时器指示符和调整偏移指示符由在处理框 504 和 512 被检查的同一个指示符来表示。在另一个实施例中，可以评估定时器偏移指示符以确定是否应该使用定时器偏移。在一个实施例中，定时器偏移指示符由 VMM 来配置并存储在 VMCS 中。

如果调整偏移指示符被启用，则处理逻辑读取定时器值（处理框 514），并通过将当前定时器值减去保存的 VM1 定时器值来确定当前定时器值与处理框 506 中保存的 VM1 定时器值之间的差值（处理框 516）。进一步地，处理逻辑通过将定时器偏移字段中的定时器偏移值减去该差值来计算已调整的定时器偏移值（处理框 518）。在一个实施例中，将此已调整的定时器偏移值存储在定时器偏移字段中。接下来，处理逻辑将控制转移到 VM1（处理框 520）。

如果调整偏移指示符被禁用，则在一个实施例中，处理逻辑跳过处理框 514，并直接继续进行到处理框 520。在另一个实施例中，如果调整偏移指示符被禁用，则处理逻辑检索定时器偏移字段的值以用作已调整的定时器偏移值，然后继续进行到处理框 520。

在备选实施例中，在 VM 退出时，计算虚拟访客定时器值（通过计算当前定时器偏移值与定时器的当前值之和来计算），并将其存储到虚拟访客定时器字段中。在一个实施例中，将虚拟访客定时器字段存储在 VMCS 中。此后，在 VM 进入时，通过将虚拟访客定时器值减去 VM 进入时的定时器值来计算偏移值。在 VM 执行时，读取定时器的尝试将返回由偏移值调整的定时器的当前值。

由此，已经描述了用于控制 VM 对定时器的访问的方法和设备。要理解上文描述旨在说明而非限制。对于本领域技术人员来说，阅读并理解上文描述时将显见到许多其他实施例。因此，本发明的范围应参考所附权利要求以及对此类权利要求赋予权利的等效物的完整范围来确定。

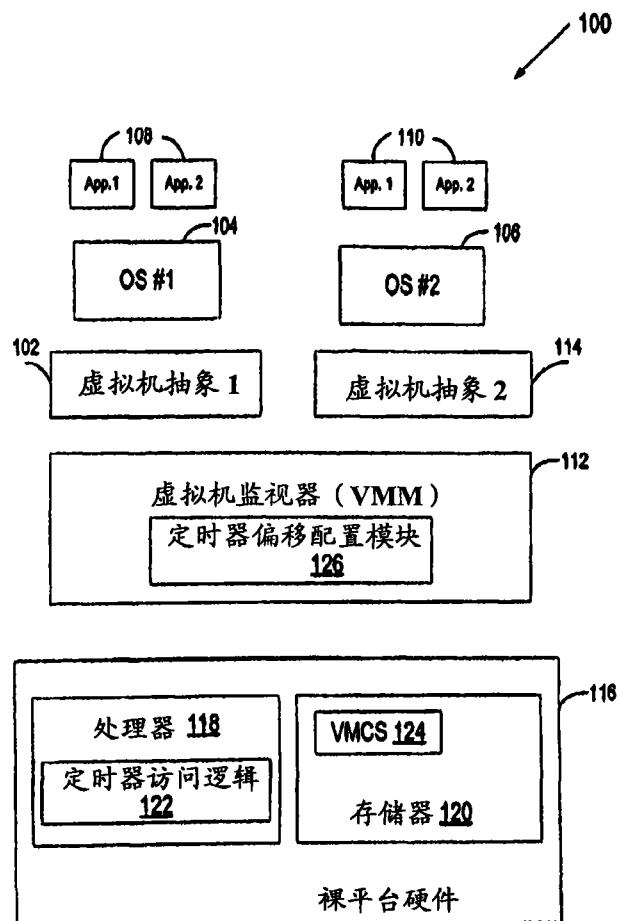


图 1

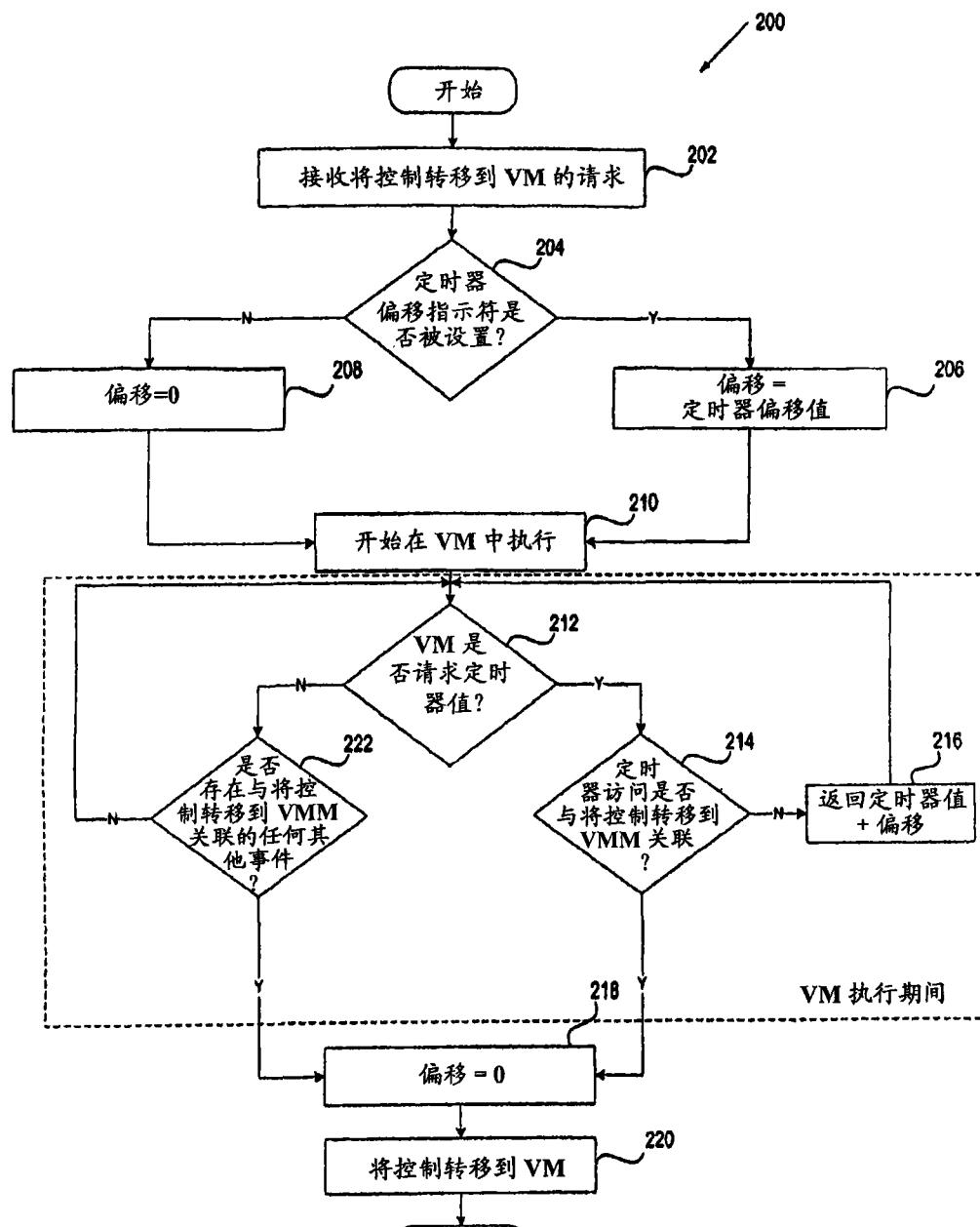


图 2

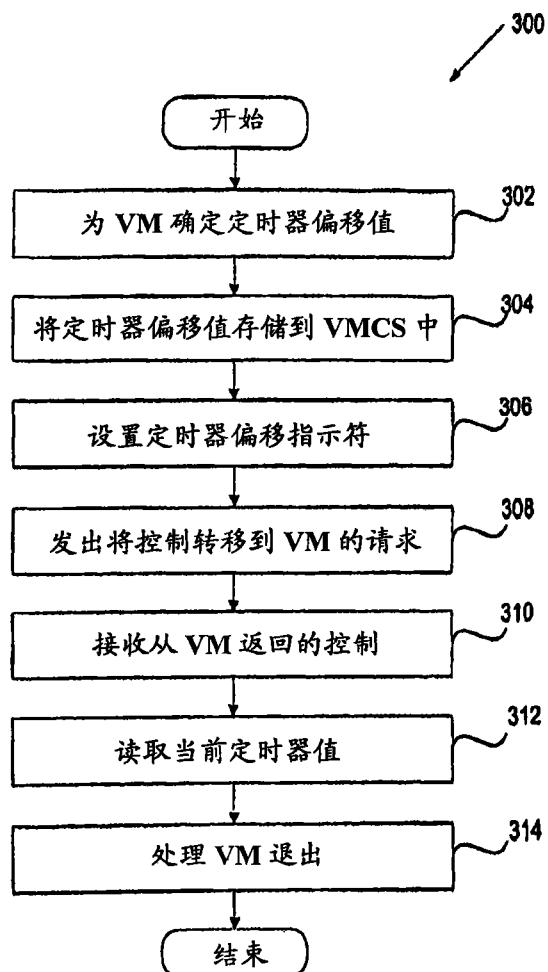


图 3

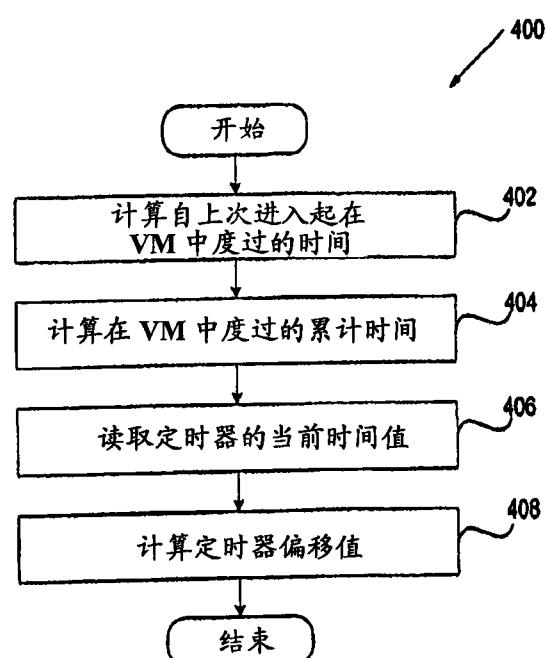


图 4

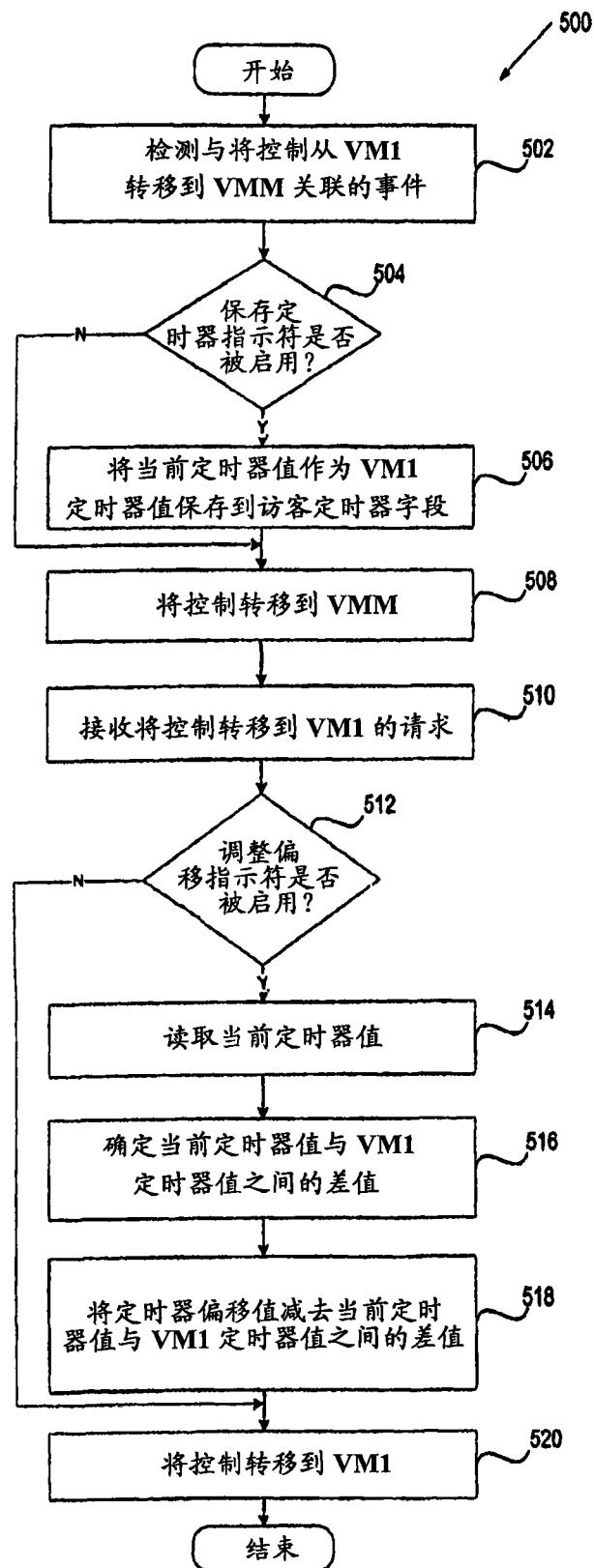


图 5