



(12) 发明专利

(10) 授权公告号 CN 102855441 B

(45) 授权公告日 2015.12.16

(21) 申请号 201210162765.6

审查员 张琳

(22) 申请日 2003.03.20

(30) 优先权数据

10/112,169 2002.03.29 US

(62) 分案原申请数据

03811454.2 2003.03.20

(73) 专利权人 英特尔公司

地址 美国加利福尼亚州

(72) 发明人 詹姆斯·萨顿二世 戴维·格劳罗克

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 毛力

(51) Int. Cl.

G06F 21/57(2013.01)

(56) 对比文件

CN 1203394 A, 1998. 12. 30,

CN 1185609 A, 1998. 06. 24.

CN 1159630 A, 1997. 09. 17.

WO 01/16772 A1, 2001. 03. 08.

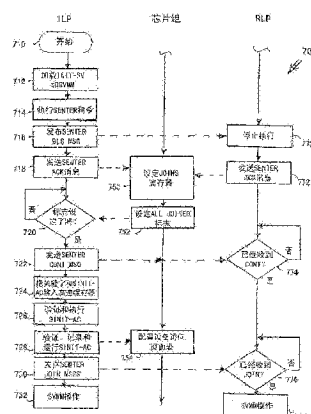
权利要求书2页 说明书10页 附图8页

(54) 发明名称

用于执行安全环境初始化指令的系统和方法

(57) 摘要

描述了在微处理器系统内启动安全操作的方法和装置。在一个实施方案中,一个启动逻辑处理器通过停止其它逻辑处理器的执行,然后把初始化和安全虚拟机监控软件载入存储器,来启动该过程。初始化处理器然后把初始化软件载入安全存储器进行验证和执行。初始化软件然后在安全系统操作之前验证和登记安全虚拟机监控软件。



1. 一种用于执行安全环境初始化指令的方法,包括:
同步第一逻辑处理器和第二逻辑处理器;
验证初始化代码模块;
验证安全虚拟机监控程序;
执行所述安全虚拟机监控程序;以及
把专用总线消息发送给所述第二逻辑处理器,以便在所述第二逻辑处理器上以响应方式执行所述安全虚拟机监控程序。
2. 如权利要求1所述的方法,其中所述同步包括:发送专用总线消息以使所述第二逻辑处理器暂停执行并且发送确认。
3. 如权利要求2所述的方法,其中所述同步包括:在芯片组内设定标志以响应所述确认。
4. 如权利要求1所述的方法,其中所述验证初始化代码模块包括把所述初始化代码模块的拷贝和公钥移到安全存储器中。
5. 如权利要求4所述的方法,其中所述验证初始化代码模块包括将所述初始化代码模块的第一摘要与所述初始化代码模块的第二摘要进行比较。
6. 如权利要求1所述的方法,其中所述验证安全虚拟机监控程序包括执行所述初始化代码模块。
7. 如权利要求6所述的方法,其中所述验证安全虚拟机监控程序包括把所述安全虚拟机监控程序登记在平台配置寄存器内。
8. 一种用于执行安全环境初始化指令的设备,包括:
同步单元,被配置为用于同步第一逻辑处理器和第二逻辑处理器;
初始化代码模块验证单元,被配置为用于验证初始化代码模块;
安全虚拟机监控程序验证单元,被配置为用于验证安全虚拟机监控程序;
执行单元,被配置为用于在所述第一逻辑处理器内执行所述安全虚拟机监控程序;以及
发送单元,被配置为用于把第一专用总线消息发送给所述第二逻辑处理器,以便在所述第二逻辑处理器上执行所述安全虚拟机监控程序。
9. 如权利要求8所述的设备,还包括:
移动单元,被配置为用于把所述初始化代码模块的拷贝和公钥移到安全存储器中。
10. 如权利要求9所述的设备,还包括:
比较单元,被配置为用于将所述初始化代码模块的第一摘要与所述初始化代码模块的第二摘要进行比较。
11. 如权利要求8所述的设备,还包括:
登记单元,被配置为用于登记所述安全虚拟机监控程序。
12. 如权利要求8所述的设备,还包括:
专用总线消息发送单元,被配置为用于发送专用总线消息;
验证单元,被配置为用于在所述第一逻辑处理器内对初始化代码进行验证。
13. 如权利要求12所述的设备,还包括:
确认单元,被配置为用于发送确认以响应所述专用总线消息。

14. 如权利要求 12 所述的设备,还包括:
暂停单元,被配置为用于暂停第二逻辑处理器中的执行;以及
用于发送确认的装置。
15. 如权利要求 13 或 14 所述的设备,还包括:
标志设定单元,被配置为用于在芯片组内设定标志以响应所述确认。
16. 如权利要求 12 所述的设备,所述验证单元进一步被配置为用于把所述初始化代码的拷贝和公钥移到安全存储器中。
17. 如权利要求 12 所述的设备,所述验证单元进一步被配置为用于执行所述初始化代码。
18. 一种用于执行安全环境初始化指令的方法,包括:
发送专用总线消息;
验证第一逻辑处理器内的初始化代码;
验证安全虚拟机监控程序;
在所述第一逻辑处理器内执行所述安全虚拟机监控程序;以及
把专用总线消息发送给第二逻辑处理器,以便在所述第二逻辑处理器上以响应方式执行所述安全虚拟机监控程序。
19. 如权利要求 18 所述的方法,还包括发送确认以响应所述专用总线消息。
20. 如权利要求 18 所述的方法,还包括暂停所述第二逻辑处理器内的执行和发送确认。
21. 如权利要求 20 所述的方法,还包括在芯片组内设定标志以响应所述确认。
22. 如权利要求 18 所述的方法,其中所述验证初始化代码包括把所述初始化代码的拷贝和公钥移到安全存储器中。
23. 如权利要求 22 所述的方法,其中所述验证初始化代码包括将所述初始化代码的第一摘要与所述初始化代码的第二摘要进行比较。
24. 如权利要求 18 所述的方法,其中所述验证安全虚拟机监控程序包括执行所述初始化代码。
25. 如权利要求 24 所述的方法,其中所述验证安全虚拟机监控程序包括把所述虚拟机监控程序登记在平台配置寄存器内。

用于执行安全环境初始化指令的系统和方法

[0001] 本申请是 PCT 国际申请号为 PCT/US03/08762、国际申请日为 2003 年 3 月 20 日、中国国家申请号为 03811454.2、题为“用于执行安全环境初始化指令的系统和方法”的原申请的分案申请(申请号 201110030876.7)的分案申请。

技术领域

[0002] 本发明通常涉及微处理器系统,更具体地说,涉及可以在可信或安全环境中运行的微处理器系统。

背景技术

[0003] 在本地或远程微型计算机上执行的金融和个人事务的增加量已经推动了“可信”或“安全”微处理器环境的建立。这些环境试图要解决的问题是个人隐私的泄露或者数据被破坏或滥用。用户不想公开他们的私人数据。他们也不想不恰当的事务改变或使用他们的数据。这样的例子包括非故意地泄露医疗记录或者从在线银行或其它存款处的资金电子失窃。类似地,内容供给者设法保护数字内容(例如,音乐、其它音频、视频或其它类型的一般数据)不会在未经授权的情况下被复制。

[0004] 现有的可信系统可以利用一套完全封闭的可信软件。这个方法实施起来相对简单,但缺点是不允许同时使用市场上可买到的普通操作系统和应用软件。这个缺点限制了对上述可信系统的认可程度。

附图说明

[0005] 本发明是以实施例的方式来说明的,而不是以限定的方式来说明的,附图中相近的附图标记表示类似的部件,其中:

[0006] 图 1 是在微处理器系统中执行的示例性软件环境的图。

[0007] 图 2 是依据本发明一个实施方案的某些示例性可信或安全软件模块和示例性系统环境的图。

[0008] 图 3 是依据本发明一个实施方案的示例性可信或安全软件环境的图。

[0009] 图 4A 是依据本发明一个实施方案,适合于支持图 3 的安全软件环境的示例性微处理器系统示意图。

[0010] 图 4B 是依据本发明另一实施方案,适合于支持图 3 安全软件环境的示例性微处理器系统示意图。

[0011] 图 5 是依据本发明另一实施方案,适合于支持图 3 安全软件环境的示例性微处理器系统示意图。

[0012] 图 6 是依据本发明一个实施方案的软件组件的执行的时线图。

[0013] 图 7 是依据本发明一个实施方案的软件和其它过程块的流程图。

具体实施方式

[0014] 下面的说明描述了在微处理器系统内初始化可信或安全环境的技术。在下面说明中,为了更彻底地理解本发明,阐述了许多具体细节,例如逻辑实现、软件模块分配、加密技术、总线信令技术,以及操作细节。然而,本领域技术人员将能理解,没有上述具体细节也可以实施本发明。在其它情况下,为了不混淆本发明,没有详细表示控制结构、门电平电路和全部软件指令序列。获悉本文所包含的说明的本领域普通技术人员无需超出常规的试验就能够实现恰当的功能性。本发明是以微处理器系统的形式公开的。然而,以其它处理器的形式也可以实施本发明,例如数字信号处理器、小型计算机或大型计算机。

[0015] 现在参考图 1,图中所示为在微处理器系统中执行的一个示例性软件环境。图 1 中所示的软件不是可信的(非可信的)。当在高特权级下运行时,操作系统 150 的大小和持续更新使得按照适时方式进行任何信任分析是非常困难的。许多操作系统位于特权环(ring)零(0)内,即最高特权级。应用 152,154 和 156 具有降低了很多的特权,典型地位于特权环三(3)内。基于作出的决策以信任由操作系统 150 提供的设备,不同特权环的存在以及将操作系统 150 和应用 152,154,156 分开为这些不同特权环似乎允许图 1 的软件按照可信模式运行。然而,实际上进行上述信任决策经常是不切实际的。影响这个问题的因素包括运行系统 150 的大小(代码行的数目),操作系统 150 可以是许多更新(新代码模块和补丁)的接收者的事实,以及操作系统 150 也可以包含由用户而不是操作系统开发者提供的代码模块(例如设备驱动器)的事实。操作系统 150 可以是通用操作系统,例如Microsoft®Windows®,Linux 或Solaris®,或者可以是任何其它适当已知或另外可获得的操作系统。应用或操作系统运行或正在运行的具体类型或名称不是关键的。

[0016] 现在参考图 2,图中所示为依据本发明一个实施方案的某些示例性可信或安全软件模块和示例性系统环境 200。在图 2 的实施方案中,处理器 202、处理器 212、处理器 222 和可选的其它处理器(未图示)被图示为单独硬件实体。在其它实施方案中,正如不同部件和功能单元的边界可以变化,处理器的数量也可以不同。在某些实施方案中,可以用在一个或多个物理处理器上运行的单独硬件执行线程(thread)或“逻辑处理器”来替换这些处理器。

[0017] 处理器 202,212,222 可以包含某些专用电路或逻辑元件以支持安全或可信操作。例如,处理器 202 可以包含安全输入(SENTER)逻辑 204 以支持执行专用 SENTER 指令,所述 SENTER 指令可以初始化可信操作。处理器 202 也可以包含总线消息逻辑 206 以支持系统总线 230 上的支持专用 SENTER 操作的专用总线消息。在另外的实施方案中,芯片组 240 的存储控制功能可以分配给处理器内的电路,对于多个处理器而言,可以包括在单个管芯上。在这些实施方案中,专用总线消息也可以在这些处理器内部的总线上发送。由于几个原因,使用专用总线消息可以增加系统安全性或可信任性。如果电路元件例如处理器 202,212,222 或芯片组 240 包含本发明公开的实施方案的适当逻辑元件,则它们可以只发布或响应上述消息。因此专用总线消息的成功交换可以有助于确保适当的系统配置。专用总线消息也可以允许通常应该被禁止的活动,例如复位平台配置寄存器 278。通过允许专用总线消息只响应于专用安全指令而发布,可以限制潜在的不友善的非可信代码对某些总线事务进行侦测的能力。

[0018] 另外,处理器 202 可以包含安全存储器 208 以支持安全初始化操作。在一个实施方案中,安全存储器 208 可以是处理器 202 的内部高速缓存器,或许按照专用模式运行。在

另外的实施方案中,安全存储器 208 可以是专用存储器。其它处理器,例如处理器 212 和处理器 222,也可以包括 SENTER 逻辑 214,224、总线消息逻辑 216,226、以及安全存储器 218,228。

[0019] “芯片组”可被定义为一组电路和逻辑,它们支持针对连接的一个或多个处理器所进行存储器和输入/输出(I/O)操作。芯片组的诸多单独元件可以被集合在单个芯片上、在一对芯片上或分散在多个芯片中,包括处理器。在图 2 的实施方案中,芯片组 240 可以包括支持存储器和 I/O 操作的电路和逻辑,以支持处理器 202,212 和 222。在一个实施方案中,芯片组 240 可以与许多存储页面 250-262 和设备访问页面表 248 进行交互,页面表 248 包含指示非处理器设备是否可以访问存储页面 250-262 的控制信息。芯片组 240 可以包括设备访问逻辑 247,所述设备访问逻辑可以允许或拒绝从 I/O 设备到存储页面 250-262 的所选部分的直接存储器存取(DMA)。在某一实施方案中,设备访问逻辑 247 可以包含允许或拒绝上述访问所需要的所有相关信息。在其它实施方案中,设备访问逻辑 247 可以访问保存在设备访问页面表 248 内的上述信息。存储页面的实际数量不是重要的,并且将根据系统需求而变化。在其它实施方案中,存储器访问功能可以在芯片组 240 的外部。在另外实施方案中,芯片组 240 的功能还可以在一个或多个物理设备中分配。

[0020] 为支持专用 SENTER 操作,芯片组 240 可以另外包括它自己的总线消息逻辑 242 来支持系统总线 230 上的专用总线消息。这些专用总线消息中的某些可以包括:把密钥(key)寄存器 244 的内容传递给处理器 202,212 或 222,或者允许通过处理器 202,212 或 222 检验专用的 ALL-JOINED 标志 274。总线消息逻辑 242 的附加特征可以是把多个处理器的总线活动登记在“EXISTS”寄存器 272 中以及把多个处理器的某一专用总线消息活动保存在“JOINS”寄存器 272 中。EXISTS 寄存器 272 和 JOINS 寄存器 272 的内容的等同性可以用来设定专用的 ALL-JOINED 标志,以指示系统内所有处理器都在参与安全输入过程。

[0021] 芯片组 240 可以支持 I/O 总线上的标准 I/O 操作,所述 I/O 总线例如外设部件接口(PCI)、加速图形接口(AGP)、通用串行总线(USB)、低引线数(LPC)总线或任何其它类型 I/O 总线(未示出)。接口 290 可以用来使芯片组 240 与令牌 276 连接,令牌 276 包含一个或多个平台配置寄存器(PCR)278,279。在一个实施方案中,接口 290 可以通过修改增加了某些安全上的增强的 LPC 总线(低引线数(LPC)接口规范,英特尔公司 1997 年 12 月 29 日的修订版 1.0)。上述安全上的增强的一个实施例是位置确认消息,利用以前保存的消息头和地址信息,把令牌 276 内的平台配置寄存器(PCR)278 作为目标。在一个实施方案中,令牌 276 可以包含专用安全特征,在一个实施方案中,可以包括可信平台模块(TPM)281,该模块在 2001 年 12 月 1 日由 TCPA 出版的版本为 1.1a 的可信计算平台联合(TCPA)主要规范中被公开(在本申请递交时从 www.trustedpc.com 可得到)。

[0022] 在系统环境 200 内确定的两个软件组件是安全虚拟机监控程序(SVMM)282 模块和安全初始化授权码(SINIT-AC)280 模块。SVMM 282 模块可以被保存在系统盘或其它大容量存储设备上,并且根据需要被移动或复制到其它位置。在一个实施方案中,在开始安全启动过程之前,SVMM 282 可以被移动或复制到一个或多个存储页面 250-262。安全输入过程之后,可以创建虚拟机环境,其中:SVMM282 可以作为系统内最高特权代码来运行,并可以被用来允许或拒绝在已创建的虚拟机内的操作系统或应用直接访问某些系统资源。

[0023] 安全输入过程所需要的某些动作可能超出简单硬件实施的范围,并且相反可以有

利地使用软件模块,其中所述软件模块的执行可以默认是可信的。在一个实施方案中,通过安全初始化(SINIT)代码可以执行这些动作。这里确定三个典型动作,但这些动作不应理解为是限定性的。一个动作可能要求对各种表示系统配置关键部分的控制进行检验,以确保所述配置支持正确的安全环境实例化。在一个实施方案中,一个要求的检验可以是,芯片组 240 提供的存储控制器配置不允许两个或多个不同系统总线地址接触存储页面 250-262 内的相同位置。第二个动作可以是配置设备访问页面表 248 和设备访问逻辑 247,以保护 SVM 282 的存储驻留拷贝使用的那些存储页面不受非处理器设备干扰。第三个动作可以是计算和登记 SVM 282 模块的身份,并且把系统控制传递给它。这里“登记(register)”是指把 SVM 282 的信任测量结果放入寄存器,例如放入 PCR278 或放入 PCR279。当进行了这个最后的动作,潜在的系统用户可以检查 SVM 282 的可信度。

[0024] 处理器或芯片组的制造商可以生成 SINIT 代码。为此,可以信任 SINIT 代码来帮助芯片组 240 的安全启动。为了分配 SINIT 代码,在一个实施方案中,众所周知的加密散列由全部 SINIT 代码构成,生成一个被称为“摘要”的值。一个实施方案生成一个 160 位的值来作为摘要。然后通过在一个实施方案中由处理器制造商拥有的私钥(private key)对摘要进行加密,以形成数字签名。当 SINIT 代码与相应数字签名捆绑在一起时,这个组合可以称为 SINIT 授权码(SINIT-AC) 280。如下所述, SINIT-AC 280 的拷贝可以在后面被验证。

[0025] SINIT-AC 280 可以被保存在系统盘或其它大容量存储器上或者被保存在固定媒介中,并且根据需要被移动或复制到其它位置。在一个实施方案中,在开始安全启动过程之前, SINIT-AC 280 可以被移动或复制到存储页面 250-262 以形成 SINIT-AC 存储驻留拷贝。

[0026] 任何逻辑处理器可以开始安全启动过程,并且因而可以被称为初始化逻辑处理器(ILP)。在本实施例中,处理器 202 为 ILP,尽管系统总线 230 上的任何处理器能够成为 ILP。此时, SINIT-AC 280 的存储驻留拷贝或 SVM 282 的存储驻留拷贝都不被认为是可信的,因为除了其它原因之外,另外的处理器或 DMA 设备可以重写存储页面 250-262。

[0027] 然后, ILP(处理器 202)执行专用指令。这个专用指令可以称为安全输入(SENTER)指令,并且可以由 SENTER 逻辑 204 支持。SENTER 指令的执行可以使 ILP(处理器 202)在系统总线 230 上发布专用总线消息,然后为随后的系统动作等待相当长的时间间隔。SENTER 执行开始之后,这些专用总线消息之一,即 SENTER BUS MESSAGE 在系统总线 230 上广播。除了 ILP 之外的那些逻辑处理器可以称为响应逻辑处理器(RLP),它们用内部非屏蔽事件响应 SENTER BUS MESSAGE。在本实施例中, RLP 包括处理器 212 和处理器 222。 RLP 必须各自终止当前操作,在系统总线 230 上发送 RLP 确认(ACK)专用总线消息,然后进入等待状态。应该注意, ILP 也在系统总线 230 上发送它自己的 ACK 消息。

[0028] 芯片组 240 可以包含一对寄存器,即“EXISTS”寄存器 270 和“JOINS”寄存器 272。这些寄存器可以被用来检验 ILP 和所有 RLP 正在适当地响应 SENTER BUS MESSAGE。在一个实施方案中,通过在逻辑处理器所进行的任何系统总线事务中把“1”写入 EXISTS 寄存器 270 的相应位,芯片组 240 可以始终跟踪在系统内的所有操作逻辑处理器。在本实施方案中,系统总线 230 上的每个事务必须包含标识字段(field),所述标识字段包含逻辑处理器标识符。在一个实施方案中,这是由物理处理器标识符和每个物理处理器内硬件执行线程的标识符构成的。例如,如果在处理器 222 上执行的线程在系统总线 230 上引起任何总线事务,则芯片组 240 将在该事务中发现这个逻辑处理器标识符,并且把“1”写入 EXISTS 寄

寄存器 270 内的相应位置 286。安全启动过程期间,当处理器 222 上的那个同一线程在系统总线 230 上发送它自己的 ACK 信息时,芯片组 240 也将发现此情况,并且把“1”写入 JOINS 寄存器 272 内的相应位置 288。(在图 2 的实施例中,为了清楚起见,每个物理处理器被图示为只带有单个执行线程。在另外实施方案中,物理处理器可以支持多个线程,因而支持多个逻辑处理器。)当 JOINS 寄存器 272 的内容与 EXISTS 寄存器 270 的内容匹配时,则芯片组 240 可以设置 ALL-JOINED 标志 246,该标志表示所有处理器已经适当地响应了 SENTER BUS MESSAGE。

[0029] 在另一实施方案中,在 ALL-JOINED 标志 246 设置之后,EXISTS 寄存器 270 和 JOINS 寄存器 272 可以继续有助于安全性。在 ALL-JOINED 标志 246 设置之后直到可信或安全操作结束期间,芯片组 240 可以继续监控并将总线周期与 JOINS 寄存器 272 相比较。在这期间,如果芯片组 240 在任何时候从不是在 JOINS 寄存器 272 内当前所确定的逻辑处理器中发现总线事务,则芯片组 240 可以假设这个逻辑处理器不知何故已经“出现”晚了。这将暗示上述逻辑处理器没有参加过安全启动过程,因此可能代表攻击者(安全威胁)。在这样的情况下,芯片组 240 可以适当地响应以把这个攻击者保持在安全环境之外。在一个实施方案中,芯片组 240 可以在这样的情况下强制系统复位。在第二个实施方案中,可通过每个逻辑处理器在 ACK 总线消息断言之后的每个事务中在系统总线上断言专用的保留信号,来实现类似的“晚到”处理器检测。在本实施方案中,在 ALL-JOINED 标志 246 设置之后,如果芯片组 240 观察到处理器所初始化的总线事务没有专用的断言信号,则芯片组 240 可以再次假设这个逻辑处理器不知何故已经“出现”晚了,并且可能代表攻击者。

[0030] 发布 SENTER BUS MESSAGE 之后,ILP (处理器 202) 轮询 ALL-JOINED 标志 246 以发现所有处理器何时且是否已经用它们的 ACK 适当地进行了响应。如果从未设置标志 246,几种实现是可能的。在 ILP 或芯片组内、或其它地方的监控定时器可以使系统复位。可选地,系统可能中止并需要操作员复位。在任一情况下,尽管系统可能不继续运行,但安全环境断言得到保护(其中如果不是所有的处理器都参与,安全启动过程就不结束)。在正常操作中,在短时间之后,ALL-JOINED 标志 246 被设置,并且 ILP 可以确保所有其它逻辑处理器已经进入等待状态。

[0031] 当 ALL-JOINED 标志 246 被设置时,为了验证和随后执行包含在 SINIT-AC280 内的 SINIT 代码,ILP (处理器 202) 可以把 SINIT-AC 280 的拷贝和密钥 284 都移入安全存储器 208。在一个实施方案中,这个安全存储器 208 可以是 ILP (处理器 202) 的内部高速缓存器,或许按照专用模式运行。密钥 284 表示与私钥对应的公钥(public key),私钥被用来加密包含在 SINIT-AC 280 模块内的数字签名,并且密钥 284 被用来检验数字签名和由此验证 SINIT 代码。在一个实施方案中,密钥 284 可能已经被保存在处理器内,或许作为 SENTER 逻辑 204 的一部分。在另一实施方案中,密钥 284 可以被保存在芯片组 240 的只读密钥寄存器 244 内,只读密钥寄存器 244 由 ILP 读取。在又一实施方案中,不是处理器就是芯片组密钥寄存器 244 可以实际保存密钥 284 的加密摘要,其中密钥 284 本身被包含在 SINIT-AC 280 模块内。在最后这个实施方案中,ILP 从密钥寄存器 244 中读取摘要,在嵌入在 SINIT-AC 280 内的密钥 284 之上计算等同加密散列(hash),并且比较这两个摘要以确保所提供的密钥 284 是确实可信的。

[0032] 然后,SINIT-AC 拷贝和公钥拷贝可以在安全存储器 208 内存在。通过使用公钥拷

贝解密包含在 SINIT-AC 拷贝内的数字签名, ILP 现在可以验证 SINIT-AC 拷贝。所述解密产生加密散列摘要的原始拷贝。如果新计算出的摘要与这个原始摘要匹配, 则 SINIT-AC 拷贝和它包含的 SINIT 代码可以被认为是可信的。

[0033] ILP 现在可以经由系统总线 230 发布另一专用总线消息, 即 SENTER CONTINUE MESSAGE, 向等待的 RLP (处理器 212, 处理器 222) 和芯片组 240 信令将要初始化安全操作。如下所概述的那样, 通过把 SINIT-AC 模块的加密摘要值写入安全令牌 276 内的平台配置寄存器 272 中, ILP 现在可以登记 SINIT-AC 模块的唯一身份。通过把执行控制传递给保存在 ILP 安全存储器 208 内的 SINIT 代码的可信拷贝, ILP 对其 SENTER 指令的执行现在可以终止。可信 SINIT 代码然后可以执行它的系统测试和配置动作, 并且依照上述“登记”的定义, 可以登记 SVMM 存储驻留拷贝。

[0034] 可以按照几种方式完成 SVMM 存储驻留拷贝的登记。在一个实施方案中, 运行在 ILP 上的 SENTER 指令把计算出的 SINIT-AC 摘要写入安全令牌 276 内的 PCR 278 中。随后, 可信 SINIT 代码可以把计算出的存储驻留 SVMM 摘要写入同一 PCR 278 或安全令牌 276 内的另一 PCR 279 中。如果把 SVMM 摘要写入同一 PCR 278 中, 则安全令牌 276 用新值 (SVMM 摘要) 对原始内容 (SINIT 摘要) 进行散列化操作, 并且把结果写回 PCR278。在对 PCR278 的第一次 (初始化) 写入被限制在 SENTER 指令的这些实施方案中, 结果摘要可以被用作系统的信任根 (root of trust)。

[0035] 一旦可信 SINIT 代码已经结束它的执行, 并且已经把 SVMM 的身份登记在 PCR 内, SINIT 代码就可以把 ILP 执行控制传递给 SVMM。在典型的实施方案中, ILP 执行的最初的 SVMM 指令可代表 SVMM 的自初始化例程。在一个实施方案中, ILP 可以把单独的 RLP JOIN MESSAGE 专用总线消息发送给每个 RLP, 使每个 RLP 在现在执行的 SVMM 拷贝的监督下加入操作。根据前面这个观点, 如下面图 3 的讨论中所概述的那样, 整个系统运行在可信模式下。

[0036] 现在参考图 3, 图中所示为依据本发明一个实施方案的示例性可信或安全软件环境。在图 3 的实施方案中, 可以同时加载可信或非可信软件, 并且可以在单个计算机系统上同时执行。SVMM350 可选择地允许或防止来自一个或多个非可信操作系统 340 和非可信应用 310-330 的对硬件资源 380 的直接访问。在上下文中, “非可信” 不是必定意味着操作系统或应用正在行为不端, 但是意味着相互作用的代码的大小和多样性使得无法可靠地断言软件正在按要求运行, 并且不存在干扰它执行的病毒或其它外来代码。在典型的实施方案中, 非可信代码是由在当今个人计算机上可以找到的普通操作系统和应用所组成的。

[0037] SVMM350 也可选择地允许或防止来自一个或多个可信或安全核心程序 360 和一个或多个可信应用 370 的对硬件资源 380 的直接访问。可以限制上述可信或安全核心程序 360 和可信应用 370 的大小和功能性, 从而有助于在其上面完成信任分析的能力。可信应用 370 可以是在安全环境中可执行的任何软件代码、程序、例程或例程组。因此, 可信应用 370 可以是各种应用或代码序列, 或者可以是相对小的应用, 例如 Java 程序。

[0038] 由操作系统 340 或核心程序 360 正常执行的能改变系统资源保护或特权的指令或操作可以被 SVMM350 俘获, 并且可选择地被允许、部分允许或拒绝。作为实施例, 在典型的实施方案中, 由操作系统 340 或核心程序 360 正常执行的更改处理器页面表的指令将被 SVMM350 俘获, 这将确保所述请求不试图要求改变在它的虚拟机范围之外的页面特权。

[0039] 现在参考图 4A, 图中所示为适合于支持图 3 的安全软件环境的微处理器系统 400 的一个实施方案。CPU A 410、CPU B 414、CPU C 418 和 CPU D 422 可被配置以附加微代码或逻辑电路以支持专门指令的执行。在一个实施方案中, 这个附加微代码或逻辑电路可以是图 2 的 SENTER 逻辑 204。这些专用指令可以支持专用总线消息在系统总线 420 上的发布, 系统总线 420 可以使这些处理器在启动安全环境期间能够适当同步。在一个实施方案中, 专用总线消息的发布可以由例如图 2 的总线消息逻辑 206 之类的电路来支持。类似地, 芯片组 430 可以类似于芯片组 420, 并且可以支持上述系统总线 420 上的专用周期。物理处理器的数量可以根据具体实施方案的实施而变化。在一个实施方案中, 处理器可以是 Intel®Pentium®级的微处理器。经由 PCI 总线 446, 或者可选择地, 经由 USB442、集成控制器电路 (IDE) 总线 (未图示)、小型计算机系统接口 (SCSI) 总线 (未图示)、或任何其它 I/O 总线, 芯片组 430 可以与大容量存储设备进行交互, 例如固定媒介 444 或可移动媒介 448。固定媒介 444 或可移动媒介 448 可以是磁盘、磁带、磁碟、磁光驱动器、CD-ROM、DVD-ROM、闪存卡, 或许多其它形式的大容量存储器。

[0040] 在图 4A 的实施方案中, 四个处理器 CPU A 410、CPU B 414、CPU C 418 和 CPU D 422 图示为四个单独硬件实体。在其它实施方案中, 处理器的数量可以不同。实际上, 物理上离散的处理器可以用在一个或多个物理处理器上运行的分立的硬件执行线程来替换。在后者的情况下, 这些线程拥有许多附加物理处理器的特征。为了具有一般的表述来讨论使用多个物理处理器和多个在处理器上的线程的任何混合, 表述“逻辑处理器”可以用来描述一个物理处理器或在一个或多个物理处理器内操作的线程。因此, 一个单线程处理器可以被认为是一个逻辑处理器, 且多线程或多核心处理器可以被认为是一个或多个逻辑处理器。

[0041] 在一个实施方案中, 芯片组 430 与改进的 LPC 总线 450 进行交互。改进的 LPC 总线 450 可以用来把芯片组 430 与安全令牌 454 连接。在一个实施方案中, 令牌 454 可以包括由可信计算平台联合 (TCPA) 设想的 TPM471。

[0042] 现在参考图 4B, 图中所示为适合于支持图 3 的安全软件环境的另一微处理器系统 490 的实施方案。与图 4A 的实施方案不同, CPU A 410 和 CPU B 414 可使用系统总线 A 402 而连接到芯片组 428, 而 CPU C 418 和 CPU D 422 可使用系统总线 B 404 而连接到芯片组 428。在其它实施方案中, 可以使用两个以上的系统总线。在另一替代实施方案中, 可以使用点对点总线。专用指令可以支持专用总线消息在系统总线 A 402 和系统总线 B 404 上的发布, 专用总线消息在系统总线 A402 和系统总线 B 404 上的发布可以使这些处理器在启动安全环境期间能够适当地同步。在一个实施方案中, 专用总线消息的发布可以由例如图 2 的总线消息逻辑 206 之类的电路来支持。

[0043] 在一个实施方案中, 芯片组 428 负责维护系统总线 A 402 和系统总线 B 404 上的一致性和相干性。如果在系统总线 A 402 上发送标准或专用的总线消息, 芯片组 428 就会把这个信息 (在适当时) 反映在系统总线 B 404 上, 反之亦然。

[0044] 在可替代的实施方案中, 芯片组 428 把系统总线 A 402 和系统总线 B 404 看作独立子系统。在系统总线 A 402 上发布的任何专用总线消息只应用于该总线上的处理器, 类似地, 在系统总线 B 404 上发布的任何专用总线消息只应用于该总线上的处理器。针对系统总线 A 402 建立的任何受保护的存储器只可被连接到系统总线 A 402 的处理器所访问, 而系统总线 B 404 上的处理器可以被看作非可信设备。为了获得对为系统总线 A 402 上的

CPU A 410 和 CPU B 414 建立的任何受保护的存储器的访问,系统总线 B 404 上的处理器 CPU C 418 和 CPU D 422 必须执行它们自己的 SENTER 过程,创建一个被登记在案的环境,该环境等同于为系统总线 A 402 上的处理器建立的环境。

[0045] 现在参考图 5,表示依据本发明另一实施方案,适合于支持图 3 的安全软件环境的示例性微处理器系统 500 的示意图。与图 4A 的实施方案不同,每个处理器(例如 CPU A 510)可以包括某些芯片组功能(例如芯片组功能 593),所述芯片组功能执行例如存储控制器功能和设备访问逻辑功能。由此,这些芯片组功能允许存储器(例如存储器 A 502)直接连接到处理器。其它芯片组功能可以保留在独立的芯片组 530 中。在系统总线 520 上可以发布专用总线消息。

[0046] 每个处理器可以间接访问连接到其它处理器的存储器,然而,与对处理器本身存储器的访问相比,这些访问可能相当慢。在开始 SENTER 过程之前,软件可以把 SINIT-AC 566 和 SVMM 574 的拷贝从固定媒介 544 移入本地存储器 504,形成 SINIT-AC 556 拷贝和 SVMM 572 拷贝。在一个实施方案中,可以选择存储器 504,因为它由确定为 ILP 的处理器直接访问,在图 5 实施例中,这是 CPU B 514。可选择地, SINIT-AC 566 和 SVMM 574 的拷贝可以放在连接到其它(非 ILP)处理器的其它存储器中,只要 ILP 514 能够访问那些存储器。如图 2 中已经描述的那样, CPU B ILP 514 通过发布 SENTER 指令开始安全输入过程,并且具有类似的结果和发布的总线周期。如上结合图 2 所述,芯片组 530 可以利用 EXISTS 寄存器 576、JOINS 寄存器 580 和 ALL-JOINED 标志 584,以确定所有处理器是否已经适当地响应了 SENTER BUS MESSAGE,并且把这个信息发送给 ILP。ILP (CPU B 514)可以再次把 SINIT-AC 556 的存储驻留拷贝连同公钥 564 的拷贝一起移入安全存储器 560。在 SINIT-AC 556 确认和登记后, ILP 就可以继续进行 SVMM 572 的存储驻留拷贝的确认和登记。

[0047] 现在参考图 6,表示依据本发明一个实施方案的各种操作的时线图。图 6 的时线表示结合图 2 所述的示例性系统所论述的全部操作的调度安排。当软件决定安全或可信操作是所期望的,在时间 610,任何软件定位 SINIT-AC 280 和 SVMM 282,并复制 SINIT-AC 280 和 SVMM 282 的拷贝以用于随后的 SENTER 指令。在本实施例中,软件把 SINIT-AC 280 拷贝和 SVMM 282 拷贝加载到一个或多个存储页面 250-262 中。然后选择一个处理器(在本实施例中是处理器 202)作为 ILP, ILP 在时间 612 发布 SENTER 指令。在时间 614, ILP 的 SENTER 指令发布 SENTER BUS MESSAGE 616。然后,在时间 628 进入“等待芯片组标志”状态之前, ILP 在时间 618 发布它本身的 SENTER ACK 608。

[0048] 每个 RLP,例如处理器 222,通过在时间 620 期间完成当前指令来响应 SENTER BUS MESSAGE 616。然后,RLP 发布它的 SENTER ACK 622,然后进入状态 634,其中它等待 SENTER CONTINUE MESSAGE。

[0049] 芯片组 240 花费时间 624 来设定 JOINS 寄存器 272 以响应在系统总线 230 上观察到的 SENTER ACK 信息。当 JOINS 寄存器 272 的内容与 EXISTS 寄存器 270 的内容匹配时,芯片组 240 在时间 626 设定 ALL-JOINED 标志 246。

[0050] 在此期间, ILP 在轮询 ALL-JOINED 标志 246 时可以保持在循环状态下。当 ALL-JOINED 标志 246 被设定,并且 ILP 在时间 630 确定 ALL-JOINED 标志 246 被设定,然后, ILP 可以在时间 632 期间发布 SENTER CONTINUE MESSAGE。当 SENTER CONTINUE MESSAGE 在时间 636 在系统总线 230 上传播时,RLP 可以进入“等待加入”的状态。例如,处理器 222

的 RLP 在时间周期 638 期间进入“等待加入”的状态。

[0051] 一旦发布 SENTER CONTINUE MESSAGE, ILP 于是(在时间周期 640 内)就可以把芯片组 240 的密钥寄存器 244 的公钥和 SINIT-AC 的拷贝加入它的安全存储器 208,以形成所述公钥拷贝和 SINIT-AC 拷贝。在另一实施方案中,密钥寄存器 244 可以包含公钥摘要,实际公钥可以被包含在 SINIT-AC 内或与 SINIT-AC 包含在一起。如上结合图 2 所述,一旦验证 SINIT-AC 的拷贝,ILP 于是就可以在安全存储器 208 内实际执行 SINIT-AC 的拷贝。

[0052] SINIT-AC 在安全存储器 208 内的拷贝开始执行之后,它随后(在时间周期 640 期间)确认并且登记 SVMM 的存储驻留拷贝。SVMM 的拷贝登记在安全令牌 276 的 PCR 278 内之后,SVMM 的存储驻留拷贝本身开始执行。此时,在正在进行的时间周期 650 期间,SVMM 操作被建立在 ILP 内。

[0053] ILP SVMM 操作最初要做的事情之一是在系统总线 230 上发布单独的 RLP JOIN MESSAGES。一个实施例就是处理器 222 的 JOIN MESSAGE 644。这个消息可以包括存储器内的一位置,在该位置上 RLP 处理器 222 可以加入被登记的 SVMM 存储驻留拷贝的执行。可选择地,ILP SVMM 操作可能已经把存储器位置登记在芯片组或存储器内的预定位置中,一旦接收到 JOIN MESSAGE,RLP 就从所述位置取回它的开始地址。在接收到处理器 222 的 JOIN MESSAGE 并且确定它的开始地址之后,在时间周期 646 期间,RLP 处理器 222 跳转到这个位置,并且加入被登记的 SVMM 存储驻留拷贝的执行。

[0054] 在所有 RLP 已经加入被登记的 SVMM 存储驻留拷贝之后,安全操作在整个微型计算机系统 200 上被建立起来。

[0055] 现在参考图 7,表示依据本发明一个实施方案的软件和其它过程块(process block)的流程图。为了清楚,图 7 只表示用于单个有代表性的 RLP 的过程块。在其它实施方案中,可以存在几个响应逻辑处理器。

[0056] 过程 700 从过程块 710 开始,这时,逻辑处理器复制 SINIT-AC 和 SVMM 模块的拷贝,该 SINIT-AC 和 SVMM 模块的拷贝能由随后的 SENTER 指令所访问。在这个实施例中,在过程块 712 中,ILP 把 SINIT-AC 和 SVMM 代码从大容量存储器加载到物理存储器内。在替代的实施方案中,任何逻辑处理器可以这样做,而不只是 ILP。如在过程块 714 中所注,通过执行 SENTER 指令,处理器成为 ILP。在过程块 716 中,ILP SENTER 指令在过程块 716 中发布 SENTER BUS MESSAGE。然后,ILP 在过程块 718 中把它本身的 SENTER ACK 消息发送给芯片组。如判断过程块 720 所示,ILP 于是进入等待状态,等待芯片组设定其 ALL-JOINED 标志。

[0057] 在每个 RLP 在过程块 770 中接收到 SENTER BUS MESSAGE 之后,它以当前指令的结束来暂停执行,然后在过程块 772 中发布它自己的 SENTER ACK。如判断过程块 774 所示,每个 RLP 于是进入等待状态,等待从 ILP 到来的 SENTER CONTINUE MESSAGE。

[0058] 当接收到 SENTER ACK 信息时,芯片组设定 JOINS 寄存器内的相应位。当 JOINS 寄存器的内容等于 EXISTS 寄存器的内容时,芯片组设定 ALL-JOINED 标志,给 ILP 发送信号以从判断过程块 720 继续进行。

[0059] 一旦在 YES 路径上离开判断过程块 720,ILP 于是就在过程块 722 中发布 SENTER CONTINUE MESSAGE。这给每个 RLP 发送信号以从判断过程块 774 继续进行。如判断过程块 776 所示,然后每个 RLP 进入第二个等待状态,等待 SENTER JOIN MESSAGE。

[0060] 同时,ILP 在过程块 724 中把芯片组公钥和 SINIT-AC 存储驻留拷贝移入它自己的

安全存储器用于安全执行。ILP 在过程块 726 中使用所述密钥来验证 SINIT-AC 的安全存储驻留拷贝,然后执行它。SINIT-AC 的执行可以进行对系统配置和 SVMM 拷贝的测试,然后登记 SVMM 身份,最后在过程块 728 中开始执行 SVMM。作为在过程块 728 中执行的动作的一部分,ILP SINIT 代码可以配置设备访问页面表 248 以及存储器和芯片组的设备访问逻辑 247,以保护 SVMM 282 存储驻留拷贝所使用的那些存储页面不受非处理器设备的干扰,如过程块 754 中所示。

[0061] ILP 在 SVMM 控制下开始执行之后,在过程块 730 中,ILP 给每个 RLP 发送单独的 SENTER JOIN MESSAGE。发送 SENTER JOIN MESSAGE 之后,ILP 随后在过程块 732 中开始 SVMM 操作。

[0062] SENTER JOIN MESSAGE 的接收使每个 RLP 沿着 YES 路径离开由判断过程块 776 表示的等待状态,且在过程块 780 中开始 SVMM 操作。SENER JOIN MESSAGE 可以包含 SVMM 入口点,RLP 在加入 SVMM 操作时分支进入该入口点。可选择地,ILP SVMM 代码可以把适当的 RLP 入口点登记在系统位置内(例如在芯片组内),RLP 一旦接收到 SENTER JOIN MESSAGE 就重新取回它。

[0063] 虽然公开的不同实施方案包括两个或多个处理器(逻辑或物理处理器),但应该理解,以更详细的方式描述这样的多个处理器和 / 或多个线程系统以解释与对带有多个逻辑或物理处理器的系统进行安全防护相关联的增加的复杂性。在复杂程度较底的系统中可能有利的实施方案可以只使用一个处理器。在某些情况下,一个物理处理器可以是多个线程,从而可以包括多个逻辑处理器(因此具有所述的 ILP 和 RLP)。然而,在其它情况下,可以使用单个处理器、单线程系统,并仍然利用所公开的安全处理技术。在上述情况下,可以没有 RLP;然而,所述安全处理技术仍然起作用来减少以未经授权方式能够窃取或操纵数据的可能性。

[0064] 在前述说明书中,已经参考本发明的示例性实施方案对其进行了描述。然而,显然可以对本发明进行各种修改和变化而不脱离附属权利要求书所提出的本发明更宽的本质和范围。因此把说明书和附图看作例证性的而不是限制性的。

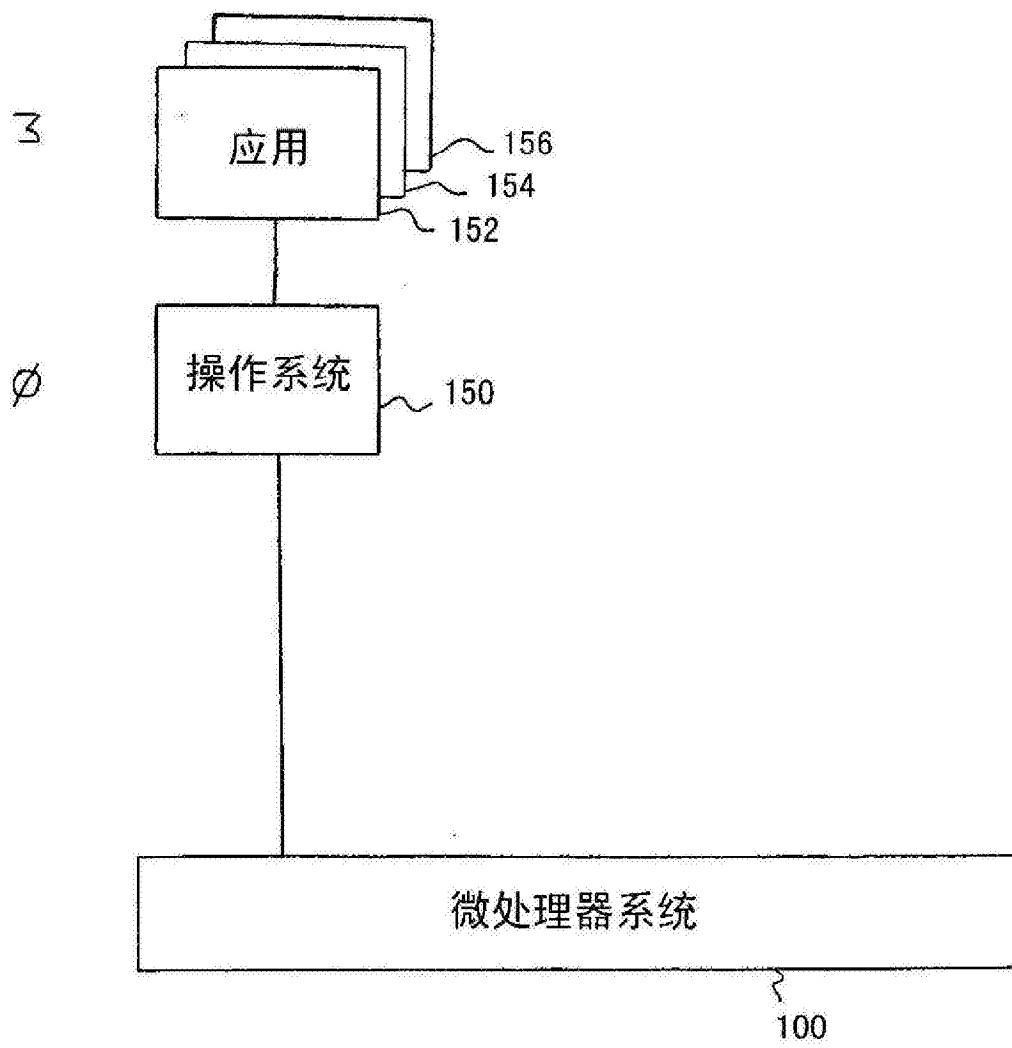


图 1

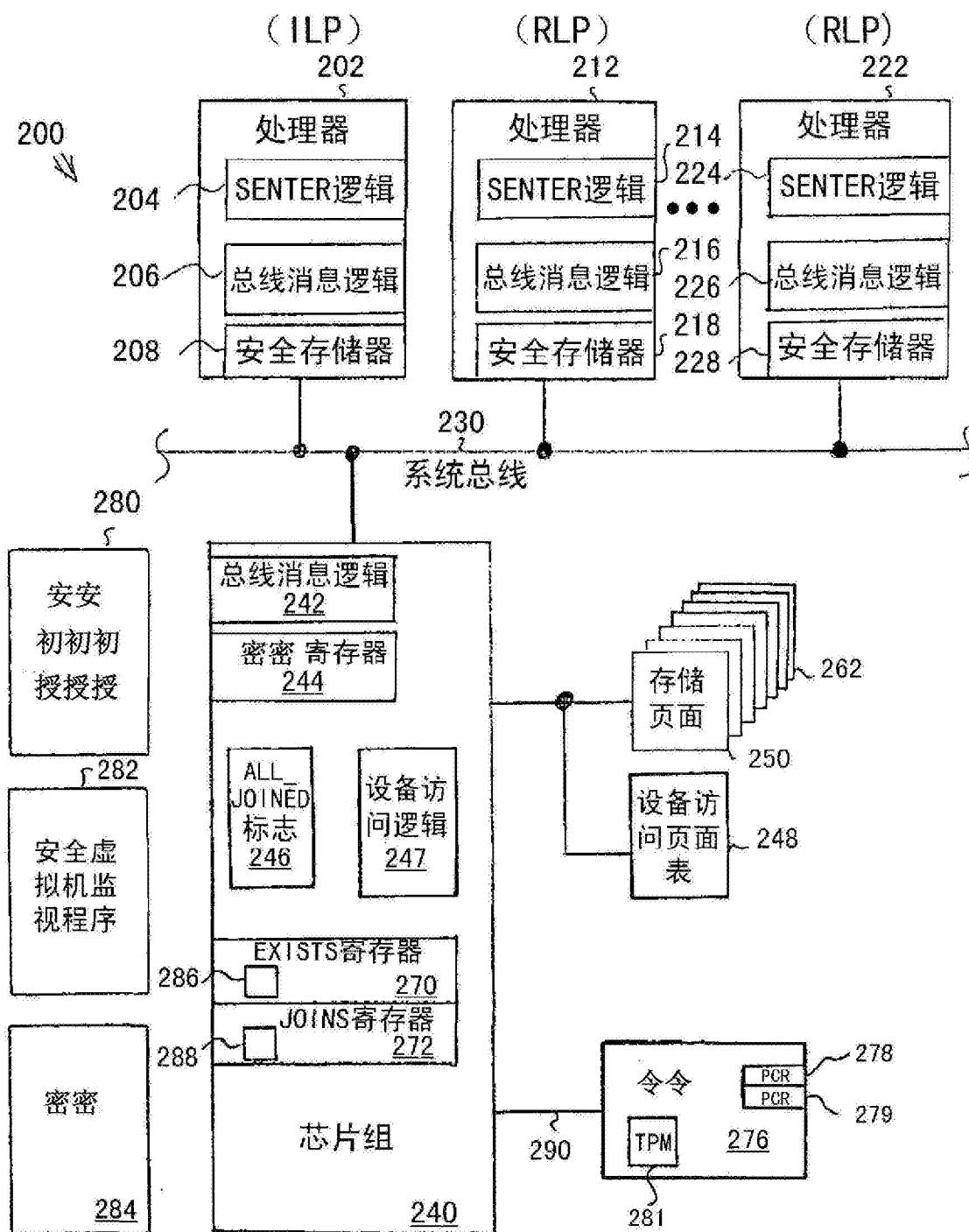


图 2

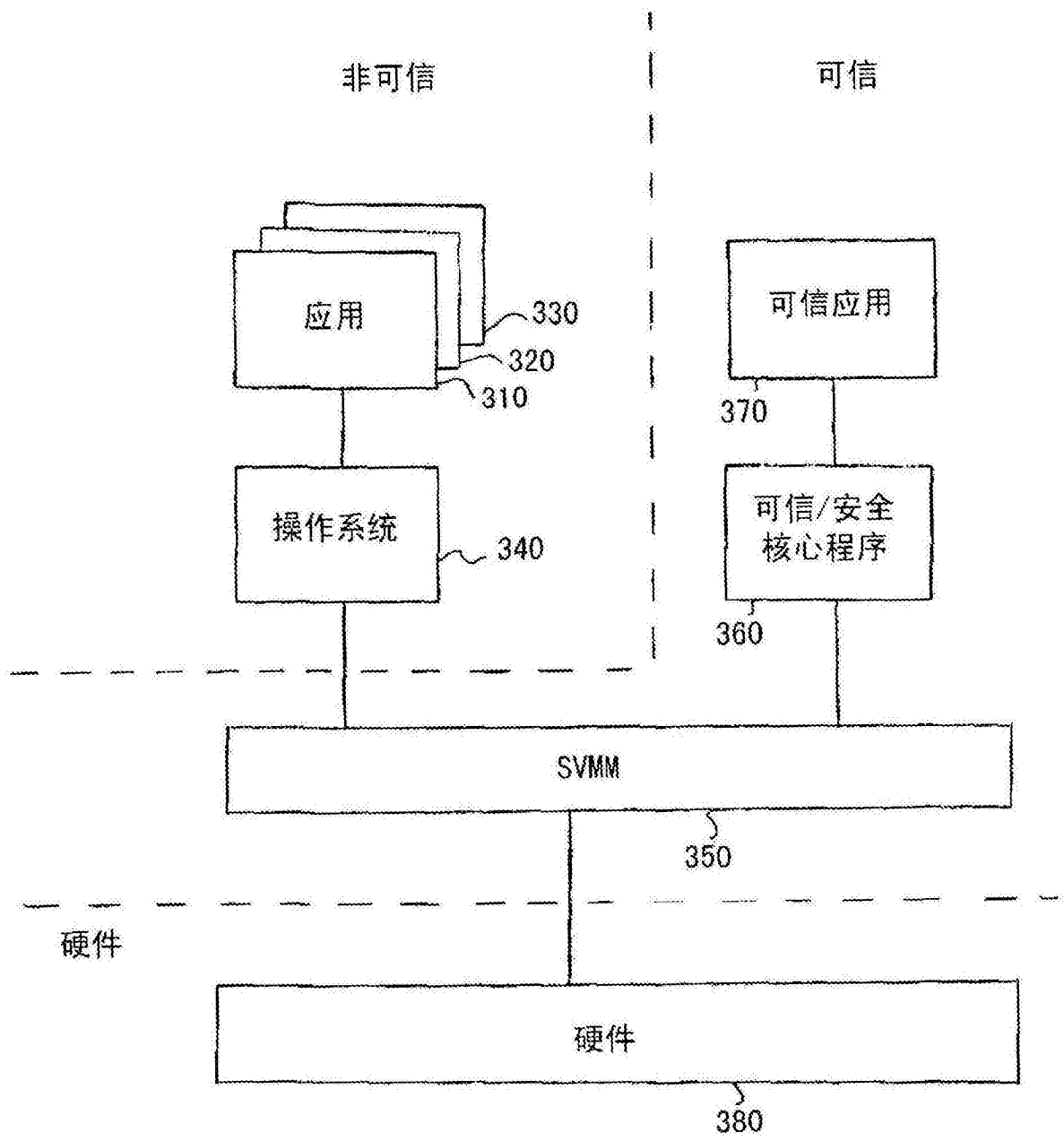


图 3

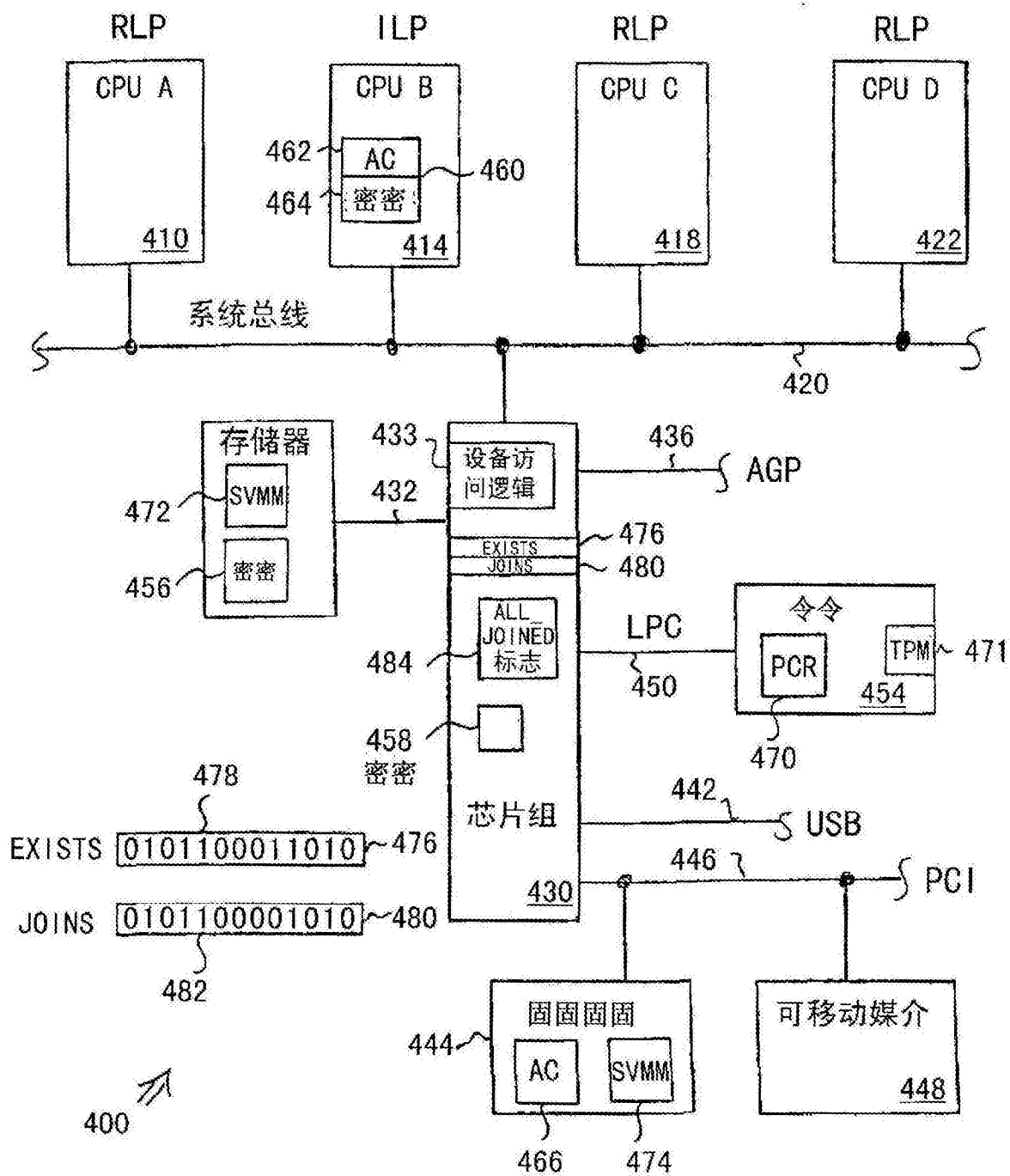


图 4A

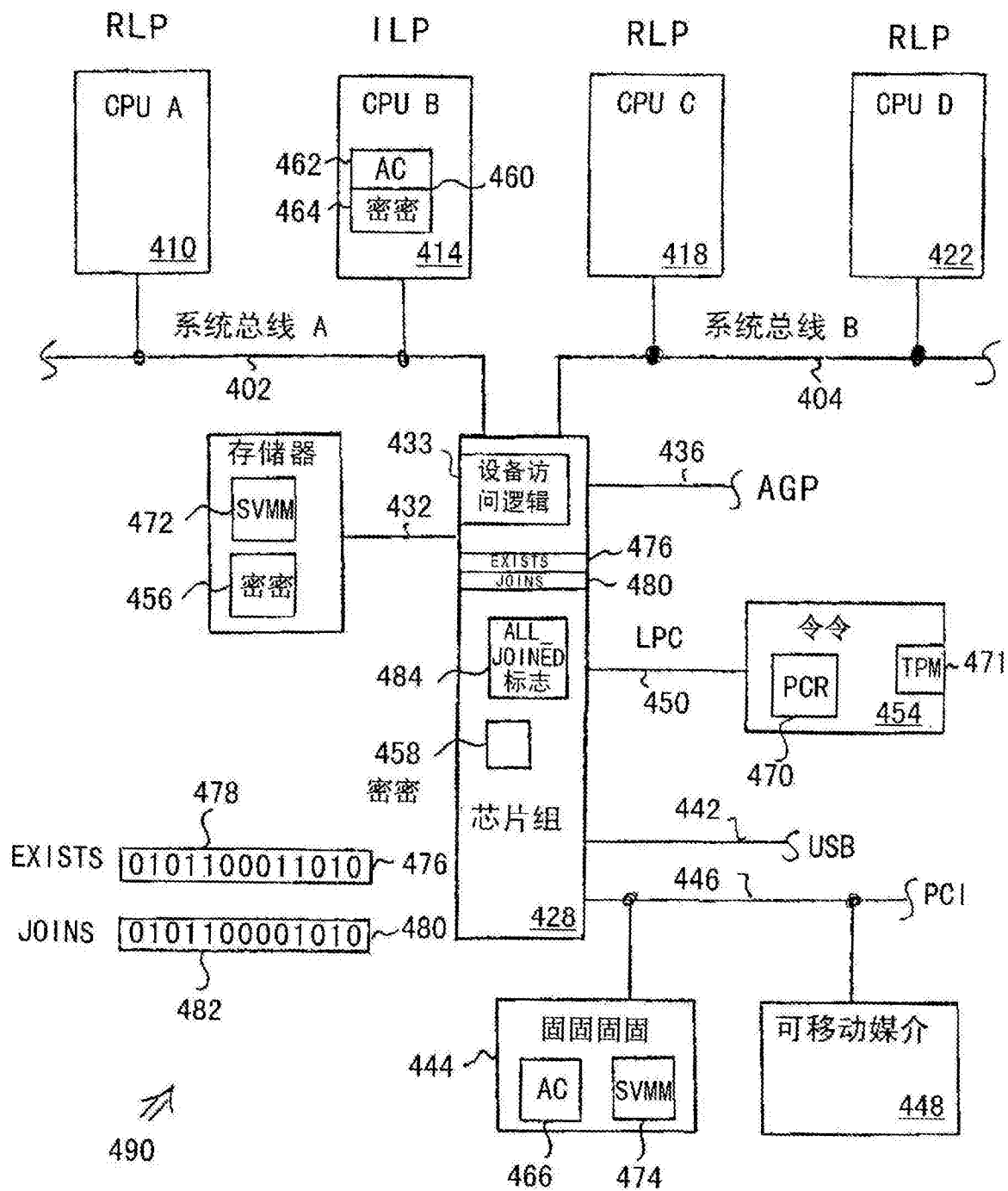


图 4B

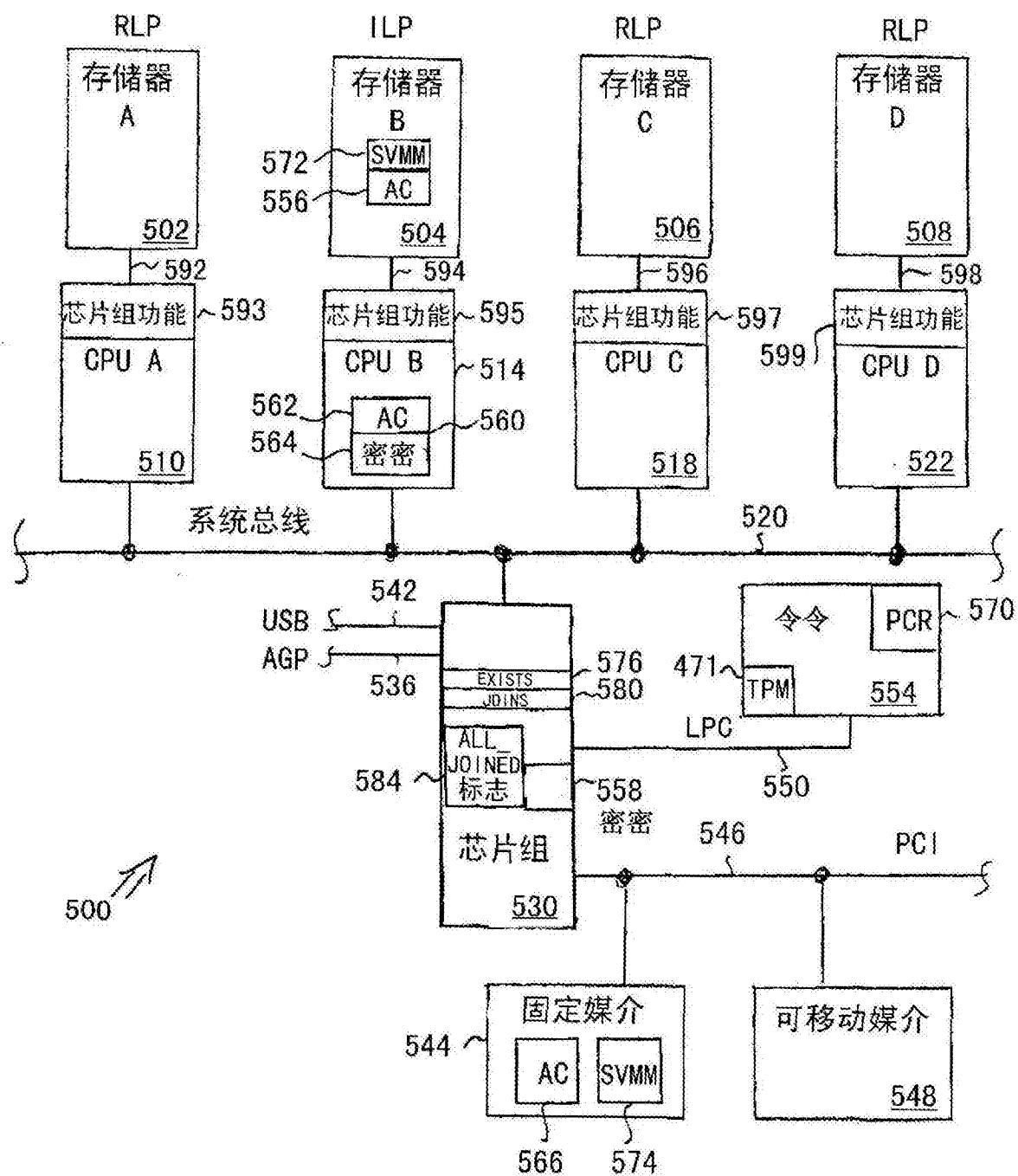


图 5

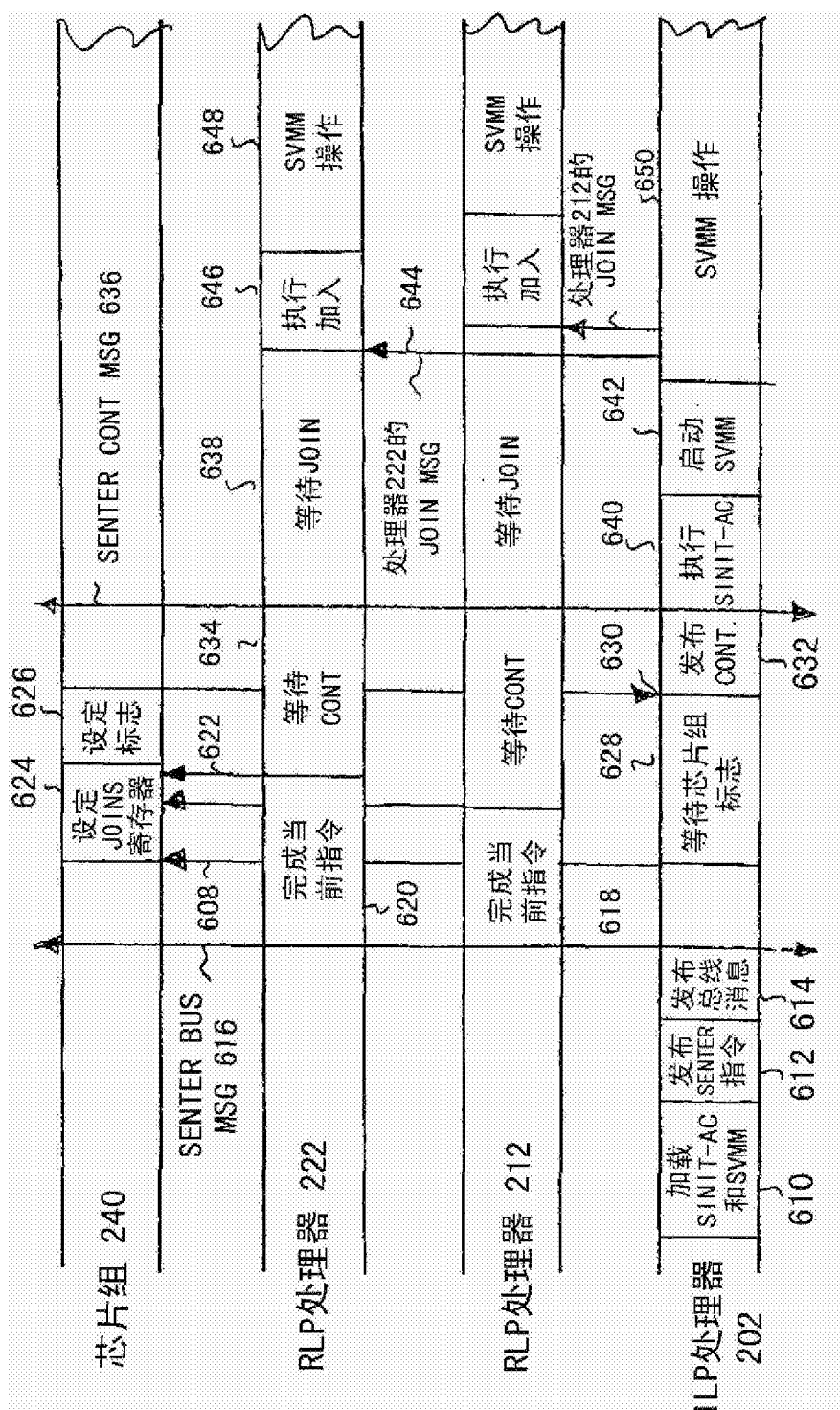


图 6

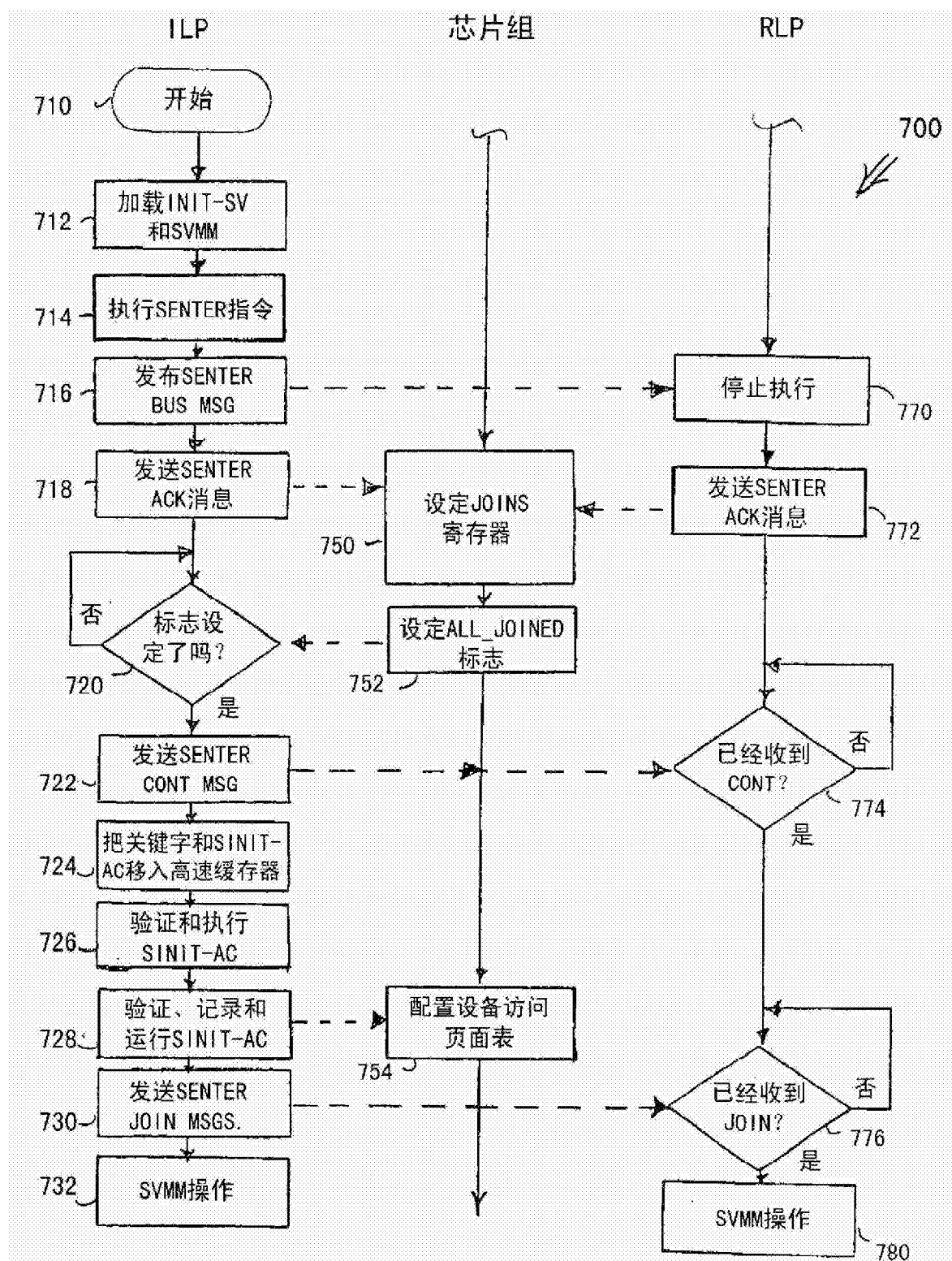


图 7