



## [12] 发明专利申请公开说明书

[21] 申请号 03812033.X

[43] 公开日 2005 年 8 月 17 日

[11] 公开号 CN 1656432A

[22] 申请日 2003.3.20 [21] 申请号 03812033.X

[30] 优先权

[32] 2002. 3. 29 [33] US [31] 10/112,124

[86] 国际申请 PCT/US2003/008764 2003. 3. 20

[87] 国际公布 WO2003/085498 英 2003. 10. 16

[85] 进入国家阶段日期 2004. 11. 26

[71] 申请人 英特尔公司

地址 美国加利福尼亚州

[72] 发明人 J·苏顿二世 D·格劳罗克

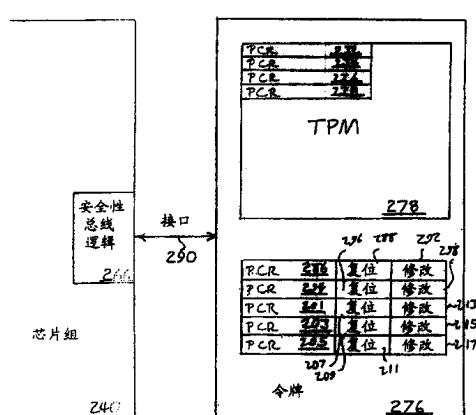
[74] 专利代理机构 中国专利代理(香港)有限公司  
代理人 吴立明 张志醒

权利要求书 4 页 说明书 13 页 附图 6 页

[54] 发明名称 用于复位平台配置寄存器的系统和方法

## [57] 摘要

本发明公开了用于复位和修改安全性令牌中的特殊寄存器的方法和装置。在一个实施例中，当总线上的特殊传送证明了相关联的处理器和芯片组的相互本地性时，当复位标志为真，则可以复位寄存器。也可以使用修改标志来指明是否可以修改寄存器内容。所述修改也可以取决于所述相互本地性的证明。



1、一种系统，包括：

包括第一寄存器的电路；和

第一总线，耦合至所述电路以传送本地确认消息。

2、如权利要求 1 所述的系统，还包括耦合至所述第一寄存器的第一标志。

3、如权利要求 2 所述的系统，其中所述第一标志指明允许复位所述第一寄存器。

4、如权利要求 3 所述的系统，其中在处理器执行特权指令期间能够复位所述第一寄存器。

5、如权利要求 3 所述的系统，还包括耦合至所述第一寄存器的第二标志，以指明允许修改所述第一寄存器的内容。

6、如权利要求 1 所述的系统，还包括耦合至所述第一寄存器的第一标志，以指明允许修改所述第一寄存器的内容。

7、如权利要求 6 所述的系统，其中在处理器执行特权指令期间，能够修改所述第一寄存器的所述内容。

8、如权利要求 1 所述的系统，其中所述第一寄存器可以包含第一代码的第一摘要。

9、如权利要求 8 所述的系统，其中所述第一寄存器可以包含扩展到所述第一摘要上的第二代码的第二摘要。

10、如权利要求 8 所述的系统，还包括第二寄存器，用来包含第二代码的第二摘要。

11、如权利要求 1 所述的系统，还包括芯片组，用来通过所述第一总线发送所述本地确认消息。

12、如权利要求 11 所述的系统，其中所述芯片组响应于特殊总线消息发送所述本地确认消息。

13、如权利要求 12 所述的系统，其中所述特殊总线消息在第二总线上传送。

14、如权利要求 13 所述的系统，还包括处理器，用来在所述第二总线上发送所述特殊总线消息。

15、如权利要求 14 所述的系统，其中所述处理器在执行特权指令期间发送所述特殊总线消息。

16、一种装置，包括：

第一寄存器，用来存储第一摘要；和

耦合至所述第一寄存器的第一标志，用来指明是否允许所述第一寄存器响应于本地确认消息而复位。

17、如权利要求 16 所述的装置，其中可以在制造时设置所述第一标志。

18、如权利要求 16 所述的装置，还包括耦合至所述第一寄存器的第二标志，用来指明是否允许所述第一寄存器的内容响应于所述本地确认消息而改变。

19、如权利要求 16 所述的装置，还包括第二寄存器，用来存储第二摘要。

20、如权利要求 19 所述的装置，还包括耦合至所述第二寄存器的第三标志，用来指明是否允许所述第二寄存器响应于所述本地确认消息而复位。

21、如权利要求 20 所述的装置，还包括耦合至所述第二寄存器的第四标志，用来指明是否允许所述第二寄存器的内容响应于所述本地确认消息而改变。

22、如权利要求 16 所述的装置，还包括一个电路，用于支持所述第一寄存器响应于写指令而以第二摘要来扩展所述第一摘要。

23、一种芯片组，包括：

总线消息逻辑，用来从处理器接收特殊总线消息；和

安全性总线逻辑，用来响应于所述特殊总线消息发送本地确认消息。

24、如权利要求 23 所述的芯片组，其中在执行安全进入指令期间内，发布所述特殊总线消息。

25、如权利要求 23 所述的芯片组，其中所述本地确认消息包括复位寄存器的指令。

26、如权利要求 23 所述的芯片组，其中本地确认消息包括修改寄存器内容的指令。

27、一种方法，包括：

在第一总线上发送第一本地确认消息；和

确定是否响应于所述第一本地确认消息复位第一寄存器。

28、如权利要求 27 所述的方法，还包括确定是否响应于第一标志复位所述第一寄存器。

29、如权利要求 28 所述的方法，还包括在所述第一总线上发送第二本地确认消息；并且还包括确定是否响应于所述第二本地确认消息来修改所述第一寄存器的第一内容。

30、如权利要求 29 所述的方法，还包括确定是否响应于第二标志修改所述第一寄存器的所述第一内容。

31、如权利要求 30 所述的方法，还包括在所述第一总线上发送第三本地确认消息；并且还包括确定是否响应于所述第三本地确认消息来复位第二寄存器。

32、如权利要求 31 所述的方法，还包括在所述第一总线上发送第四本地确认消息；并且还包括确定是否响应于所述第四本地确认消息来修改所述第二寄存器的第二内容。

33、如权利要求 30 所述的方法，还包括在所述第一总线上发送第三本地确认消息；并且还包括确定是否响应于所述第三本地确认消息来修改所述第一寄存器的第二内容。

34、如权利要求 31 所述的方法，还包括使用与摘要的加密组合来修改所述第一寄存器的所述第二内容。

35、一种装置，包括：

第一寄存器，用来存储第一摘要；和

耦合至所述第一寄存器的第一标志，用来指明是否允许响应于本地确认消息而修改所述第一寄存器的内容。

36、如权利要求 35 所述的装置，其中可以在制造时设置所述第一标志。

37、如权利要求 35 所述的装置，还包括耦合至所述第一寄存器的第二标志，用来指明是否允许所述第一寄存器响应于本地确认消息而复位。

38、一种方法，包括：

在第一总线上发送第一本地确认消息；和

确定是否响应于所述第一本地确认消息而修改第一寄存器的内容。

39、如权利要求 38 所述的方法，还包括确定是否响应于第一标

志而修改所述第一寄存器的内容。

## 用于复位平台配置寄存器的系统和方法

### 技术领域

本发明一般地涉及微处理器系统，更具体地，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

在本地或远程微计算机上进行的金融和个人交易数量的增加，推动了“可信”或“安全”微处理器环境的建立。这些环境努力解决的问题是隐私的丢失，或者数据被破坏或滥用。用户不希望其个人数据被公开。他们也不希望其数据被改变或被用于不当的交易中。这些情况的例子包括医疗记录的非故意公开，或者对在线银行或其他储蓄机构的资金的电子盗窃。类似地，内容提供者希望保护数字内容（例如，音乐，其他音频，视频，或者其他类型的数据）防止未经授权而进行拷贝。

这种可信微处理器系统的一个组件可以是由可信计算平台联盟(TCPA)于2001年12月1日发布的可信计算平台联盟主规范，1.1a版（在本申请提交时可以从[www.trustedpc.com](http://www.trustedpc.com)获得）中公开的可信平台模块( TPM )。TPM 可以包括几个称为平台配置寄存器(PCR)的特殊寄存器。PCR 可以用来存储加密的文件摘要，包括软件程序，以后可以对其进行检索以进一步用于认证处理。由于这些 PCR 的内容对于系统的安全或可信的性质是非常重要的，PCR 不应该被系统的通用操作复位。目前的设计将 PCR 限制为仅在总的系统复位（例如开机复位）时才被复位。

### 附图说明

附图中以举例而非限制的方式示出了本发明，其中相似的附图标记表示类似的部件，其中：

图 1 示出了根据本发明一个实施例的示例性可信或安全软件环境。

图 2 示出了根据本发明一个实施例的特定的示例性可信或安全

软件模块和示例性系统环境。

图 3 详细示出了根据本发明一个实施例的图 2 所示系统的安全性令牌。

图 4 示出了根据本发明一个实施例的、用于支持图 1 所示安全软件环境的示例性微处理器系统。

图 5 是根据本发明一个实施例对平台配置寄存器进行复位的流程图。

图 6 是根据本发明另一个实施例对平台配置寄存器进行复位的流程图。

10

### 具体实施方式

下面的说明描述了用于对使用在微处理器系统中的可信或安全环境中的特殊寄存器进行复位或修改的技术。在下面的说明中阐述了大量的具体细节，例如逻辑电路实现、软件模块的分配、加密技术、总线信令技术和操作的细节，以提供对本发明的更透彻的理解。但是，本领域的技术人员能够理解，没有这些具体细节也可以实施本发明。在其他一些情况下，为了避免混淆本发明，没有详细示出控制结构、门级电路和全部的软件指令序列。本领域的技术人员借助这里的详细说明，能够实现适当的功能，而不需要过多的实验。本发明以微处理器系统的形式公开。但是，本发明可以用其他形式的处理器例如数字信号处理器、微型计算机或大型计算机来实施。

在一个实施例中，将一种新型的平台配置寄存器（PCR）与安全性令牌（security token）内的可信平台模块（TPM）结合使用。每个新的 PCR 可以具有一对与其相关联的标志，所述标志可用来指示是否可以对 PCR 进行复位或者对其内容进行修改。安全性令牌可以具有一个总线接口，该总线接口可用来接收特殊的本地确认消息。这些本地确认消息可以表示安全性令牌与其他计算机资源在物理上是直接连接的，并且可信软件希望复位或修改 PCR 内容。

现参见图 1，其中示出了根据本发明一个实施例的示例性可信或安全软件环境。在图 1 的实施例中，可信和不可信软件可以同时加载并且可以在一个计算机系统上同时执行。安全虚拟机监视器（SVMM）  
350 有选择地允许或阻止一个或多个不可信操作系统 340 和不可信应

用程序 310 至 330 直接访问硬件资源 380。在本申请的上下文中，“不可信”并非必须表示操作系统或应用程序是在有意地进行不当的活动，而是表示，交互代码的大小和多样性使得可靠地断言该软件是按照希望的方式工作、并且没有病毒或其他外来代码干扰其执行是不切实际的。在一个典型的实施例中，不可信代码可以由当前个人计算机上普通操作系统和应用程序组成。

SVMM 350 也有选择地允许或阻止一个或多个可信或安全内核 360 和一个或多个可信应用程序 370 直接访问硬件资源 380。可以对这种可信或安全内核 360 和可信应用程序 370 的大小和功能进行限制，以有助于对其进行可信度分析的能力。可信应用程序 370 可以是能够在安全环境中执行的任何软件代码、程序、例行程序或例行程序组。因此，可信应用程序 370 可以是各种应用程序，或代码序列，或比较小的应用程序如 Java 小程序。

通常由操作系统 340 或内核 360 执行的、可能改变系统资源保护或特权的指令或操作可以被 SVMM 350 捕获，并被有选择地允许、部分地允许或拒绝。例如，在一个典型的实施例中，通常由操作系统 340 或内核 360 执行的、改变处理器的页面表的指令相反会被 SVMM 350 捕获，这确保该请求不是在企图改变其虚拟机之外的页面特权。

现参见图 2，其中示出了根据本发明一个实施例的特定的示例性可信或安全软件模块和示例性系统环境 200。在图 2 的实施例中，示出了处理器 202、处理器 212、处理器 222 和其他可选的处理器（未示出）。在其他实施例中，处理器的数目可以不同，如同各种组件和功能单元的边界范围可以不同。

处理器 202、212、222 可以包含特定的特殊电路或逻辑元件，来支持安全或可信操作。例如，处理器 202 可以包含安全进入 (SENTER) 逻辑 204，来支持可以启动可信操作的特殊 SENTER 指令的执行。SENTER 是特权指令的一个例子，被定义为机器指令，该机器指令通常仅在特殊模式下才被执行，并且通常可由操作系统获得而不能由其他用户获得。一旦执行了特权指令，则可能发生特权操作。处理器 202 也可以包含总线消息逻辑 206，来支持对特殊 SENTER 操作提供支持的系统总线 230 上的特殊总线消息。在其他的实施例中，芯片组

240 的存储器控制功能可以分配给处理器内的电路，并且对于多处理器来说可以包括在一个管芯上。在这些实施例中，也可以在处理器内部的总线上发送特殊总线消息。由于下面几个原因，使用特殊总线消息可以提高系统的安全性和可信度。电路元件如处理器 202、212 和 222 或芯片组 240 可以仅在其包含本发明实施例的适当逻辑元件或者其等同物的情况下，才发布或响应这种消息。因此，特殊总线消息的成功交换可以帮助确保适当的系统配置。特殊总线消息也可以允许通常会被禁止的活动，例如复位平台配置寄存器 286。通过使特殊总线消息仅响应特殊安全指令而被发布，能够削弱潜在的敌意不可信代码对特定总线事务进行监视的能力。

另外，处理器 202 可以包含安全存储器来支持安全初始化操作。在一个实施例中，安全存储器 208 可以是处理器 202 的内部高速缓存，其可能工作在特殊模式下。

“芯片组”可以定义为一组电路和逻辑，其支持对于所连接的处理器的存储器和输入/输出 (I/O) 操作。芯片组的各个元件可以组合在一个芯片上、一对芯片上或分布在包括处理器的多个芯片上。在图 2 的实施例中，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。在一个实施例中，芯片组 240 可以与多个存储器页面 250 至 262 和设备访问页面表 248 接口，所述设备访问页面表 248 包含指明非处理器设备是否可以访问存储器页面 250 至 262 的控制信息。芯片组 240 可以包括设备访问逻辑 247，其可以允许或拒绝 I/O 设备对存储器页面 250 至 262 的所选择部分的直接存储器访问 (DMA)。在一些实施例中，设备访问逻辑 247 可以包含允许或拒绝这种访问所需的所有相关信息。在其他的实施例中，设备访问逻辑 247 可以访问保存在设备访问页面表 248 中的这种信息。在其他实施例中，芯片组 240 的功能还可以在一个或多个物理设备中进行分配。芯片组 240 还可以包括其自己的总线消息逻辑 242，来支持对特殊 SENTER 操作提供支持的系统总线 230 上的特殊总线消息。

芯片组 240 可以支持 I/O 总线例如外围元件互连 (PCI)、加速图形端口 (AGP)、通用串行总线 (USB)、低管脚数量 (LPC) 总线或其他类型的 I/O 总线 (未示出) 上的标准 I/O 操作。可以采用接口 290 将芯片组 240 与包含一个或多个平台配置寄存器 (PCR) 286, 294 的令牌 276

相连。在一个实施例中，接口 290 可以是以附加了特定的安全性增强而修改的 LPC 总线（低管脚数量 (LPC) 接口规范，英特尔公司，修订版 1.0，1997 年 12 月 29 日）。这种安全性增强的一个例子是本地确认消息，利用了预先保存的消息报头和目标指向令牌 276 内的平台配置寄存器 (PCR) 286 的地址信息。在一个实施例中，令牌 276 可以包含特殊的安全性特征，在一个实施例中，可以包括由可信计算平台联盟 (TCPA) 于 2001 年 12 月 1 日发布的 TCPA 主规范，1.1a 版（在本申请提交时可以从 [www.trustedpc.com](http://www.trustedpc.com) 获得）中公开的可信平台模块 (TPM) 281。接口 290 可以为芯片组 240 与令牌 276 之间的各种双向通信提供数据通路。芯片组 240 的安全性总线逻辑 266 可以包括产生并向令牌 276 发送本地确认消息的电路。

系统 200 中标出的两个软件组件为安全虚拟机监视器 (SVMM) 282 模块和安全初始化认证代码 (SINIT-AC) 280 模块。SVMM 282 模块可以存储在系统盘或其他大容量存储器上，并可以在需要时移动或复制到其他位置。在一个实施例中，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器页面 250 至 262。在安全进入过程之后，可以生成一个虚拟机环境，其中，SVMM 282 可以作为系统内最有特权的代码运行，并可以被用来允许或拒绝所生成的虚拟机内的操作系统或应用程序对特定系统资源的直接访问。

安全进入过程所需的一些动作可能会超出简单的硬件实现方案的范围，而最好替代地采用可以绝对信任其执行的软件模块。在一个实施例中，这些动作可以由安全初始化 (SINIT) 代码执行。这种动作可以包括检验系统配置的关键部分，以确保其支持安全环境的正确实例化 (instantiation)，配置设备访问逻辑 247 和设备访问页面表 248，以及在其转移系统控制之前，计算并注册 SVMM 282 身份。这里“注册”表示将 SVMM 282 的信用量度置于寄存器中，例如 PCR 286 或 PCR 294 中。当作出所述最后一个动作时，潜在的系统用户可以检查 SVMM 282 的可信度。

SINIT 代码可以由处理器或芯片组的制造商生成。因此，可以信任 SINIT 代码来辅助芯片组 240 的安全启动过程。为了分配 SINIT 代码，在一个实施例中，公知的密码杂凑由全部的 SINIT 代码形成，产生一个称为“摘要”的值。在一个实施例中，为摘要生成 160 位的值。

然后可以用私人密钥对该摘要进行加密，在一个实施例中私人密钥由处理器的制造商持有，以形成一个数字签名。当 SINIT 代码与相应的数字签名捆绑在一起时，该组合可以称为 SINIT 认证代码(SINIT-AC) 280。然后可以如下面所述对 SINIT-AC 280 的拷贝进行验证。

任何逻辑处理器可以开始安全启动过程，于是可以被称为启动逻辑处理器 (ILP)。在本例中，处理器 202 成为 ILP，不过系统总线 230 上的任何处理器都可以成为 ILP。此时，无论是 SINIT-AC 280 的存储器驻留拷贝还是 SVMM 282 的存储器驻留拷贝都不能被认为是可信的，这是因为除了别的原因以外，其他的处理器或 DMA 设备可以重写存储器页面 250-262。

然后 ILP(处理器 202)执行一个特权指令。在一个实施例中该特权指令可以称为安全进入(SENTER)指令，并可以由 SENTER 逻辑 204 支持。SENTER 指令的执行可以引起 ILP(处理器 202)在系统总线 230 上发布特殊总线消息。在其他实施例中，可以执行其他的特权指令。

发布特殊总线消息后，ILP(处理器 202)可以检验被称为响应逻辑处理器(RLP)的其他处理器何时以及是否准备好进入安全操作。当 RLP 已经准备好时，ILP(处理器 202)可以将 SINIT-AC 280 的拷贝和密钥 284 都移入安全存储器，以便用于认证并随后执行 SINIT-AC 280 中包括的 SINIT 代码。在一个实施例中，该安全存储器可以是 ILP(处理器 202)的内部高速缓存，其可能操作在特殊模式下。密钥 284 表示与用来对 SINIT-AC 280 模块中包括的数字签名进行加密的私人密钥相对应的公共密钥，并用来验证数字签名，从而认证 SINIT 代码。在一个实施例中，密钥 284 可能已经存储在处理器中，可能作为 SENTER 逻辑 204 的一部分。在另一个实施例中，密钥 284 可以存储在由 ILP 读取的芯片组 240 的只读密钥寄存器 244 中。在又一个实施例中，处理器或者芯片组的密钥寄存器 244 实际上可以保存密钥 284 的加密摘要，其中密钥 284 本身包含在 SINIT-AC 280 模块中。在所述最后一个实施例中，ILP 从密钥寄存器 244 中读取摘要，计算嵌入在 SINIT-AC 280 中的密钥 284 之上的等效密码杂凑值，并对这两个摘要进行比较，以保证所提供的密钥 284 确实是可信的。

SINIT-AC 的拷贝和公共密钥的拷贝可以存在于安全存储器中。ILP 现在可以通过使用公共密钥的拷贝，将 SINIT-AC 的拷贝中包含

的数字签名进行解密，来验证 SINIT-AC 的拷贝。该解密产生了密码杂凑摘要的原始拷贝。如果新计算出来摘要与该原始摘要匹配，则 SINIT-AC 的拷贝及其包含的 SINIT 代码可以认为是可信的。

ILP 现在可以通过将 SINIT-AC 模块的加密摘要值写入安全性令牌 276 中的平台配置寄存器 286 中，来注册 SINIT-AC 模块的唯一身份，如下所述。现在可以通过将执行控制转移给保存在 ILP 的安全存储器内的 SINIT 代码的可信拷贝，来结束 ILP 对其 SENTER 指令的执行。然后该可信 SINIT 代码可以执行其系统测试和配置动作，并可以按照上面对“注册”的定义来注册 SVMM 的存储器驻留拷贝。

可以以几种方式来进行 SVMM 的存储器驻留拷贝的注册。在一个实施例中，运行在 ILP 上的 SENTER 指令将计算出的 SINIT-AC 的摘要写入安全性令牌 276 内的 PCR 286 中。接着，可信 SINIT 代码可以将计算出的 SVMM 的存储器驻留摘要写入安全性令牌 276 内的同一 PCR 286 或另一 PCR 294 中。如果 SVMM 摘要被写入同一 PCR 286，安全性令牌 276 用新值(SVMM 摘要)对原始内容(SINIT 摘要)进行杂凑化，并将结果写回 PCR 286 中。在一些实施例中，对 PCR 286 的第一次(初始化)写入被限制为 SENTER 指令，则所得到的摘要可以被用作系统的信用根源。

一旦可信 SINIT 代码结束其执行，并在 PCR 中注册 SVMM 的身份，则 SINIT 代码可以将 ILP 执行控制移交给 SVMM。在一个典型的实施例中，ILP 执行的第一 SVMM 指令可以表示 SVMM 的自初始化例行程序。在一个实施例中，ILP 可以使每个 RLP 加入处于现在正在执行的 SVMM 拷贝的监视下的运行。从这点向前，整个系统运行在可信模式，如上面对图 1 的讨论中所述的那样。

现在参见图 3，其中根据本发明的一个实施例详细示出了图 2 系统的安全性令牌 276。令牌 276 可以包括含有几个现有技术中的 PCR 232、234、236、238 的 TPM 278。这些现有技术中的 PCR 只有在总的系统复位事件的情况下才被复位。该 TPM 278 可以包含 16 个或更多的现有技术中的通用 PCR。

另外，在一个实施例中，令牌 276 可以包括本发明的 PCR 286、294。每个 PCR 286、294 可以与一些标志相关联，所述标志指出该 PCR 在不同条件下的行为。例如，PCR 286 可以具有与其相关联的复

位标志 288 和修改标志 292，而 PCR 294 可以具有与其相关联的复位标志 296 和修改标志 298。当复位标志 288 的逻辑值为真，PCR 286 可以在到达接口 290 的本地确认消息的命令下被复位。当修改标志 292 的逻辑值为真，PCR 286 可以在到达接口 290 的本地确认消息的命令下修改其内容。在一个实施例中，要将 SINIT-AC 280 的加密摘要写入 PCR 286，而将 SVMM 282 的加密摘要写入 PCR 294。这些摘要可以在请求之下出示作为证明过程的一部分，该证明过程可以向本地或远程用户证明，系统环境 200 正以安全或可信的方式运行。

在另一个实施例中，SINIT-AC 280 和 SVMM 282 的摘要可以被写入单个 PCR，如 PCR 286 中。在该实施例中，SINIT-AC 280 的摘要首先被写入 PCR 286。然后，PCR 286 的内容与 SVMM 282 的摘要被加密地组合，所得到的组合摘要被写入 PCR 286。在一个实施例中，被配置用来支持 PCR 286 的令牌 276 内的电路，可以采用接口 290 上的输入写请求的值，将该值与 PCR 286 内的当前值加密地组合，然后将所得到的新值置于 PCR 286 内。该过程可以称为“扩展”操作。可以将输入写请求的值说成是被“扩展到”PCR 286 内的当前值上。PCR 286 内的该扩展值，其表示组合摘要，然后可以在请求之下被出示作为证明过程的一部分，该证明过程可以向本地或远程用户证明，系统环境 200 正以安全或可信的方式运行。

可以在制造令牌 276 时将复位标志 288、296 和修改标志 292、298 设置为逻辑真值。这将允许一旦在接口 290 上接收到本地确认总线消息时，对 PCR 286、294 进行复位和内容修改。在其他的实施例中，取决于在 TPM 278 的控制下采取的特定的安全性动作，复位标志 288、296 和修改标志 292、298 的值可以在真与假之间改变。

在图 3 所示的实施例中，示出了根据一个实施例的示例性附加 PCR 201、203 和 205。在替换的实施例中，可以采用根据本发明一个实施例的较少的或者附加的 PCR。在其他的实施例中，附加的 PCR，如 PCR 201、203 和 205 可以用来存储与安全性相关的信息，如代码的摘要。任何附加的 PCR，如示例性的 PCR 201、203 和 205，可以分别具有相关的复位标志 207、209 和 211，以及相关的修改标志 213、215 和 217。

当系统环境 200 使本地确认消息从芯片组 240 发送到令牌 276

时，PCR 286、294 可以在除了总的系统复位以外的时间上被复位或修改。在一个实施例中，响应于总线消息逻辑 242 在系统总线 230 上接收到 SENTER 特殊总线消息，可以由芯片组 240 的安全性总线逻辑 266 产生本地确认消息。接着通过在处理器中执行 SENTER 指令产生 SENTER 特殊总线消息。这种 SENTER 指令可以在任何总的系统复位事件之后的数天、数周或数月执行。下面结合图 3 至图 7 描述将系统操作从不安全操作转换为安全操作的 SENTER 过程的细节。

在图 3 的实施例中，示出了在一个示例性的可信或安全个人计算机环境中的令牌 276。具体地，令牌 276 通过接口 290 与芯片组 240 连接。在替换的实施例中，根据本发明一个实施例的令牌 276 可以用在其他系统环境中。这种环境的例子可以包括数据服务器环境、蜂窝式或其他形式的移动电话、移动数据端口、汽车系统、有线或卫星电视系统、或个人数字助理 (PDA)。这些例子不应该被理解为是限制性的，因为很多其他的系统环境也可以利用本实施例的令牌。接口 290 或者其他的等效接口可以将根据其他实施例的令牌连接到上述的示例性系统环境之一中的电路，其中芯片组 240 例如可以是集成蜂窝电话控制器或 PDA 控制器的一部分。这些实施例符合上面对图 2 的讨论中包括的“芯片组”的定义。

现在参见图 4，其中示出了用于支持图 1 的安全软件环境的微处理器系统 400 的一个实施例。CPU A 410、CPU B 414、CPU C 418 和 CPU D 422 可以与图 1 的处理器类似，但可以用额外的微代码或逻辑电路来配置，以支持特殊指令的执行。在一个实施例中，所述额外的微代码或逻辑电路可以是图 2 的 SENTER 逻辑 204。这些特殊的指令可以支持特殊总线消息在系统总线 420 上的发布。在一个实施例中，可以由例如图 2 的总线消息逻辑 206 的电路来支持特殊总线消息的发布。类似地，芯片组 430 可以与芯片组 130 类似，但可以支持系统总线 420 上的上述特殊周期。另外，芯片组 430 可以具有额外的页面保护电路，用于保护物理存储器 434 内的页面表。在一个实施例中，页面保护电路可以是图 2 的设备访问逻辑 247。

在一个实施例中，芯片组 430 通过 LPC 总线逻辑 (LPC-BL) 452 与被称为修改的低管脚数量 (LPC) 总线 450 的额外数据总线进行接口。修改的 LPC 总线 450 可以用来连接芯片组 430 与安全性令牌 454。在

一个实施例中，令牌 454 可以包括 TCPA 所设想的可信平台模块 (TPM) 460。在一个实施例中，令牌 454 可以包括 SINIT PCR 470 和 SVMM PCR 480。每个 PCR 470、480 可以与一些标志相关联，所述标志指明 PCR 在不同条件下的行为。例如，SINIT PCR 470 可以具有与其相关联的复位标志 472 和修改标志 474，而 SVMM PCR 480 可以具有与其相关联的复位标志 482 和修改标志 484。当复位标志 472 的逻辑值为真时，SINIT PCR 470 可以在到达修改的 LPC 总线 450 的本地确认消息的命令下被复位。当修改标志 474 的逻辑值为真时，SINIT PCR 470 可以在到达修改的 LPC 总线 450 的本地确认消息的命令下修改其内容。在一个实施例中，要将 SINIT-AC 的加密摘要写入 SINIT PCR 470，而要将 SVMM 的加密摘要写入 SVMM PCR 480。这些摘要可以在请求之下出示作为证明过程的一部分，所述证明过程可以向本地或远程用户证明，微处理器系统 400 正以安全或可信的方式运行。

在另一个实施例中，SINIT-AC 和 SVMM 的摘要可以写入单个 PCR，如 SINIT PCR 470。在该实施例中，SINIT-AC 的摘要首先被写入 SINIT PCR 470。然后，SINIT PCR 470 的内容与 SVMM 的摘要被加密地组合，所得到的组合摘要被写入 SINIT PCR 470。在一个实施例中，被配置用来支持 SINIT PCR 470 的令牌 276 内的电路，可以采用修改的 LPC 总线 450 上的输入写请求的值，将该值与 SINIT PCR 470 内的当前值加密地组合，然后将所得到的新值置于 SINIT PCR 470 内。该组合摘要然后可以在请求之下出示作为证明过程的一部分，该证明过程可以向本地或远程用户证明，微处理器系统 400 正以安全或可信的方式运行。

可以在制造令牌 254 时将复位标志 472、482 和修改标志 474、484 设置为逻辑真值。这将允许一旦在修改的 LPC 总线 450 上接收到本地确认总线消息时，对 SINIT PCR 470 和 SVMM PCR 480 进行复位和内容修改。在其他的实施例中，取决于在微处理器系统 400 内采取的特定的安全性动作，例如 SENTER 或其他特权指令执行，复位标志 472、482 和修改标志 474、484 的值可以在真与假之间改变。

为了在例如微处理器系统 400 的微处理器系统中使用 SVMM 350，可能需要采取几个动作。一个动作可能是，在硬件配置即 SVMM 350 本身和加载并启动 SVMM 350 的任何软件这两方面，应当对系统配置

的可信度进行测试。另一个动作可能是，注册 SVMM 350 的身份并将系统控制移交给它。当采取该最后一个动作时，SVMM 350 的可信度可以由潜在的系统用户进行检查。

在一个实施例中，可以将 SINIT-AC 的拷贝存储在固定的介质 444 中，用于当微处理器系统 400 需要进入安全操作时使用。当逻辑处理器，例如物理处理器 CPU 414，希望使用 SVMM 350 初始化系统宽安全操作时，它可以将 SINIT-AC 的拷贝加载到存储器 434 中，以形成 SINIT-AC 的存储器驻留拷贝。以这种方式启动安全操作的任何逻辑处理器可以被称为启动逻辑处理器 (ILP)。在本例中，CPU B 414 成为 ILP。然后，作为 ILP 的 CPU B 414 可以将 SVMM 的拷贝加载到存储器中，以形成 SVMM 的存储器驻留拷贝。SVMM 的该拷贝可以位于存储器的非连续页面上，但芯片组 430 将为它维持一个页面表。此时，无论是 SINIT-AC 还是 SVMM 都不能被认为是值得信赖的，由于除了别的原因之外，其他的处理器或 DMA 设备可以重写存储器 434。

然后 ILP 可以执行以微代码或其他处理器逻辑如 SENTER 实现的特殊或特权指令。SETER 指令的执行可以使 ILP 在系统总线 420 上发布特殊总线消息，然后等待相当长的时间间隔，为了后续的系统动作。在其他实施例中，可以执行其他的特权指令。

发布特殊总线消息后，ILP (CPU B 414) 检验何时以及是否所有的 RLP 已经正确地被响应。当 RLP 已经准备好进入安全操作时，ILP 可以将 SINIT-AC 的存储器驻留拷贝和芯片组 430 的经加密的公共密钥移入安全存储器中。在一个实施例中，该安全存储器可以是 ILP，即 CPU B 414 的内部高速缓存。于是 SINIT-AC 的拷贝和公共密钥的拷贝可以存在于内部高速缓存内。现在 ILP 可以通过使用公共密钥的高速缓存内驻留拷贝，对 SINIT-AC 的该拷贝的数字签名进行解密，来验证 SINIT-AC 的高速缓存内驻留拷贝。该解密产生了密码杂凑的“摘要”的原始拷贝。如果新计算的“摘要”与该原始“摘要”匹配，则可以认为 SINIT-AC 的高速缓存内驻留拷贝是可信的。

在 SENTER 指令的执行过程中的该点处，可以由 LPC-BL 452 产生一对本地确认消息。这允许复位 SINIT PCR 470，并将 SINIT 摘要写入 SINIT PCR 470。

现在 ILP 可以发布另一条特殊总线消息，借助系统总线 420 通知

等待的 RLP (CPU A 410, CPU C 418, CPU D 422) 和芯片组 430 安全操作将要被启动。然后可信的 SINIT-AC 的高速缓存内驻留拷贝可以测试并验证驻留在物理存储器 434 内的 SVMM 的拷贝。在一个实施例中, SINIT-AC 可以读取页面表条目以寻找位于存储器 434 中的 SVMM 的拷贝。然后 SINIT-AC 可以命令芯片组 430 阻止 I/O 设备对这些页面表条目的 DMA 访问。可以以几种方式来进行 SVMM 的存储器驻留拷贝的验证和后续注册。同样, 在 SENTER 指令的这部分执行过程中, 可以由 LPC-BL 452 生成另一对本地确认消息。这就允许 SVMM PCR 480 的复位, 并将 SVMM 摘要写入 SVMM PCR 480 中。

在 SENTER 指令执行结束时, SVMM 的存储器驻留拷贝可以在该点开始执行。在一个实施例中, ILP 可以使每个 RLP 在现在正在执行的 SVMM 存储器驻留拷贝的监视下加入运行中。从这点向前, 整个系统运行在可信模式下, 如上面对图 1 的讨论中所述的那样。

现在参见图 5, 其中示出了根据本发明一个实施例的对平台配置寄存器进行复位的流程图。图 5 中所示的方法可以用于图 1 和图 3 的安全或可信计算机环境。该方法也可以用于其他环境, 如蜂窝式或其他形式的移动电话、汽车或其他安全系统, 或用于数字或卫星电视或媒体交换系统。在后面这些实施例中, 所提到的设备中的电路可以作为芯片组 240 或芯片组 430 的等同物, 用于向安全性令牌发送本地确认消息的目的。

图 5 的实施例在方框 520 开始, 其中通过接口发送第一本地确认消息, 被寻址到第一 PCR。第一本地确认消息中的数据指明要进行复位, 在方框 530 中, 检查第一 PCR 的复位标志, 以查看是否可以允许复位。如果是, 则复位第一 PCR。然后, 在方框 540, 发送第二本地确认消息, 被寻址到第一 PCR。第二本地确认消息中的数据指明要进行摘要(或其他数据)的写, 在方框 550 中, 检查第一 PCR 的修改标志, 以查看是否可以允许对包含在第一 PCR 内的值进行修改。如果是, 则以包含在第二本地确认消息中的摘要对第一 PCR 进行写操作。

然后, 在方框 560, 可以发送第三本地确认消息, 其被寻址到第二 PCR。第三本地确认消息中的数据指明要进行复位, 在方框 570 中, 检查第二 PCR 的复位标志, 以查看是否可以允许复位。如果是, 则复位第二 PCR。然后, 在方框 580, 发送第四本地确认消息, 被寻址到

第二 PCR。第四本地确认消息中的数据指明要进行摘要(或其他数据)的写，在方框 590 中，检查第二 PCR 的修改标志，以查看是否可以允许对包含在第二 PCR 内的值进行修改。如果是，则以包含在第四本地确认消息中的摘要对第二 PCR 进行写操作。

现在参见图 6，其中示出了根据本发明另一个实施例的对平台配置寄存器进行复位的流程图。图 6 中所示的方法可以用于上面关于图 5 的方法所列举的各种环境。图 6 的实施例在方框 620 开始，其中通过接口发送第一本地确认消息，其被寻址到第一 PCR。第一本地确认消息中的数据指明要进行复位，在方框 630 中，检查第一 PCR 的复位标志，以查看是否可以允许复位。如果是，则复位第一 PCR。然后，在方框 640，发送第二本地确认消息，其被寻址到第一 PCR。第二本地确认消息中的数据指明要进行摘要(或其他数据)的写，在方框 650 中，检查第一 PCR 的修改标志，以查看是否可以允许对包含在第一 PCR 内的值进行修改。如果是，则以包含在第二本地确认消息中的摘要对第一 PCR 进行写操作。

然后，在方框 660，发送第三本地确认消息，其被再次寻址到第一 PCR。第三本地确认消息中的数据指明要进行摘要(或其他数据)的写，在方框 670 中，检查第一 PCR 的修改标志，以查看是否仍然允许对包含在第一 PCR 内的值进行修改。如果是，则在方框 680，令牌将第三本地确认消息中包含的摘要进行加密性地杂凑到第一 PCR 的现有内容之上。然后，在方框 690，所得到的组合摘要可以被写入第一 PCR。该过程可以是上面结合图 2 描述的扩展过程。

在上文中，已经参考特定的示例性实施例对本发明进行了描述。但很明显，可以对其作出各种修改和变动而不偏离后附权利要求中所阐明的本发明的精神和范围。因此，说明书和附图应该被认为是说明性的而非限制性的。

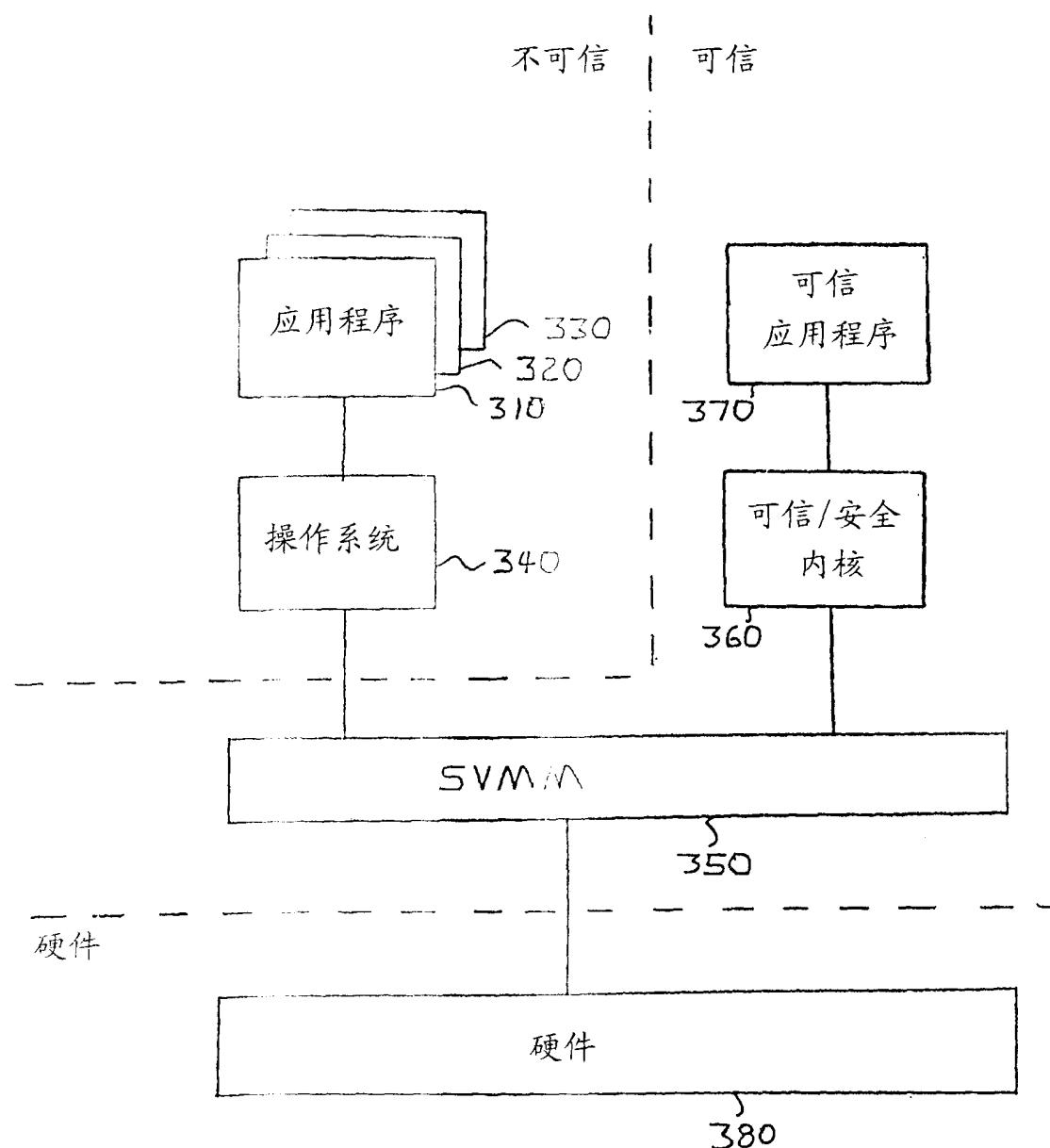


图 1

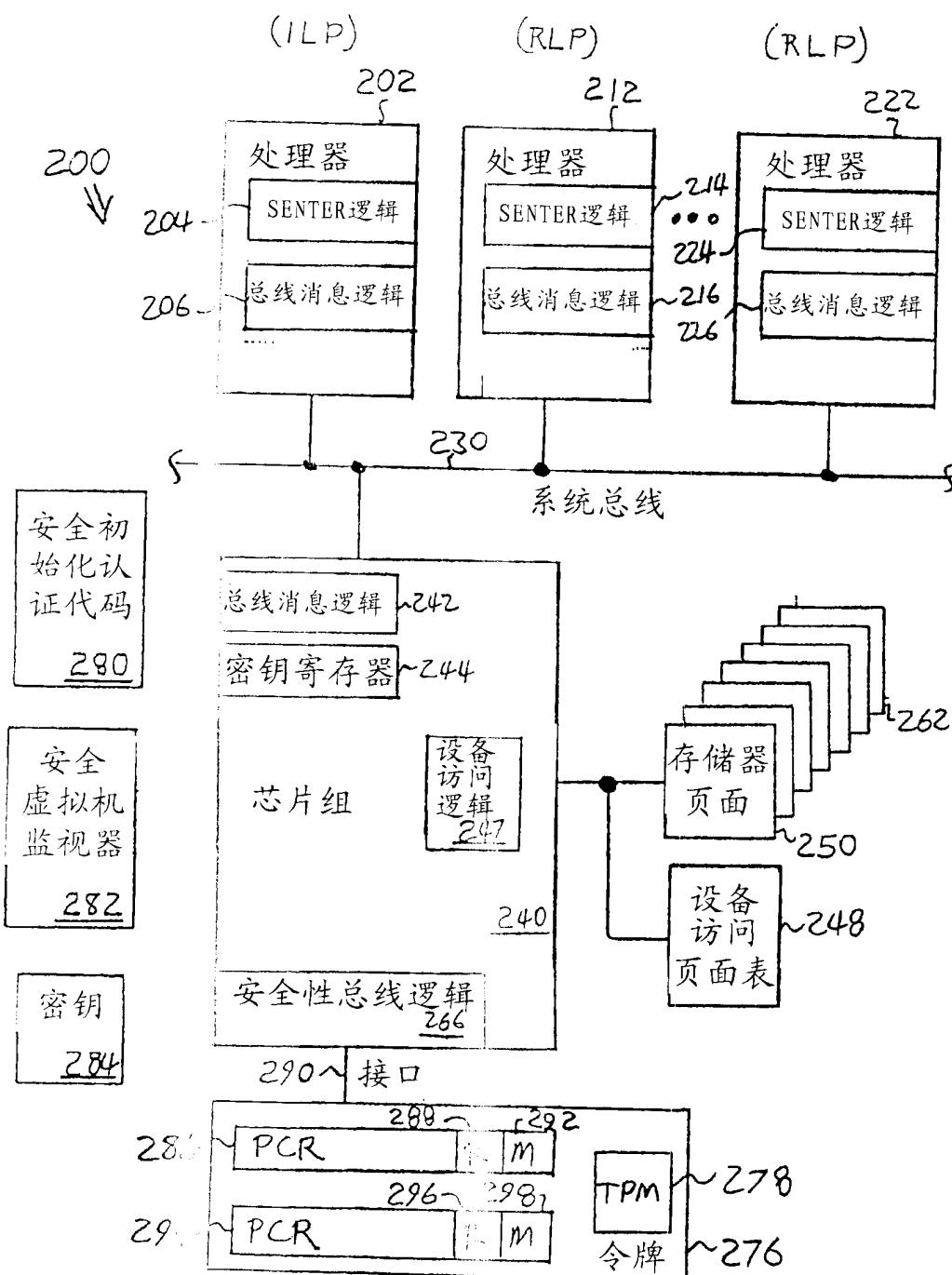


图 2

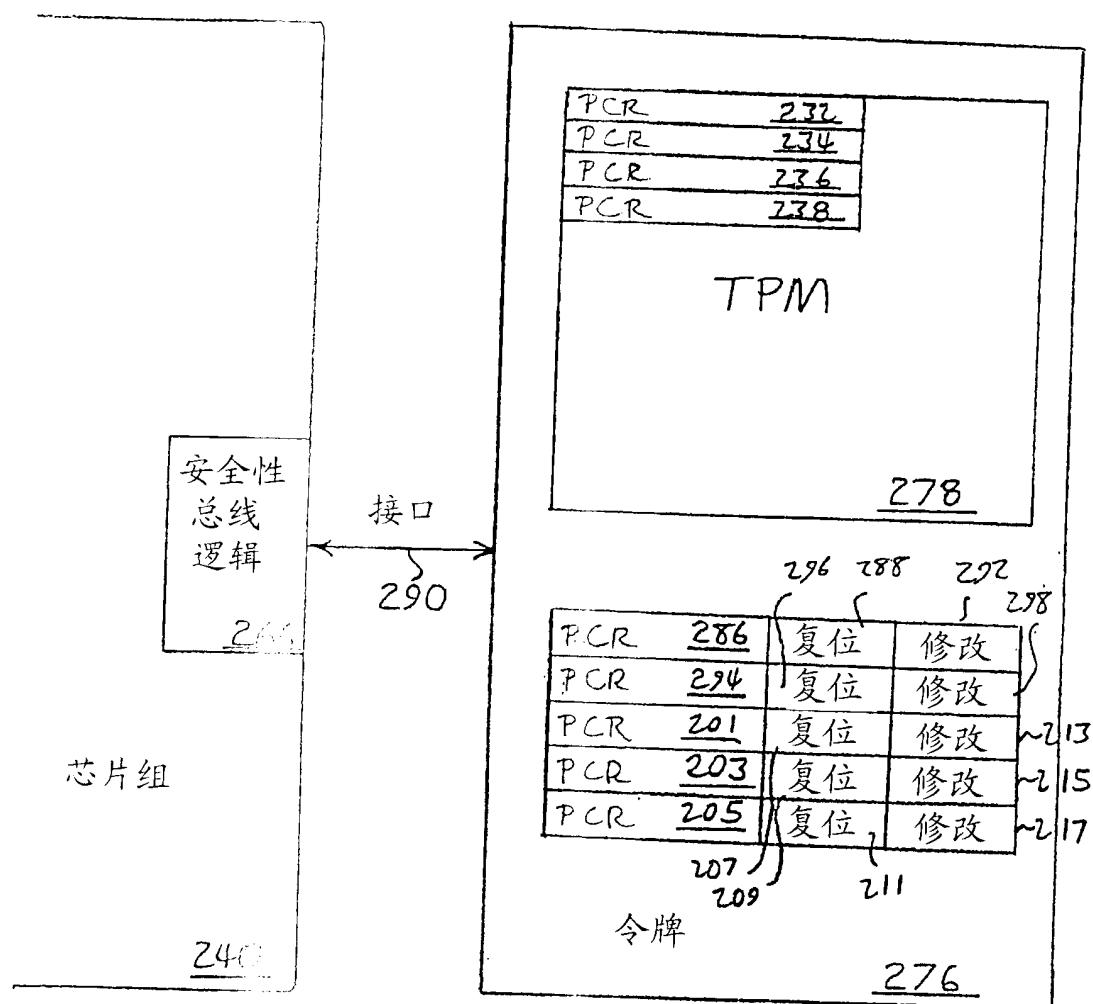


图 3

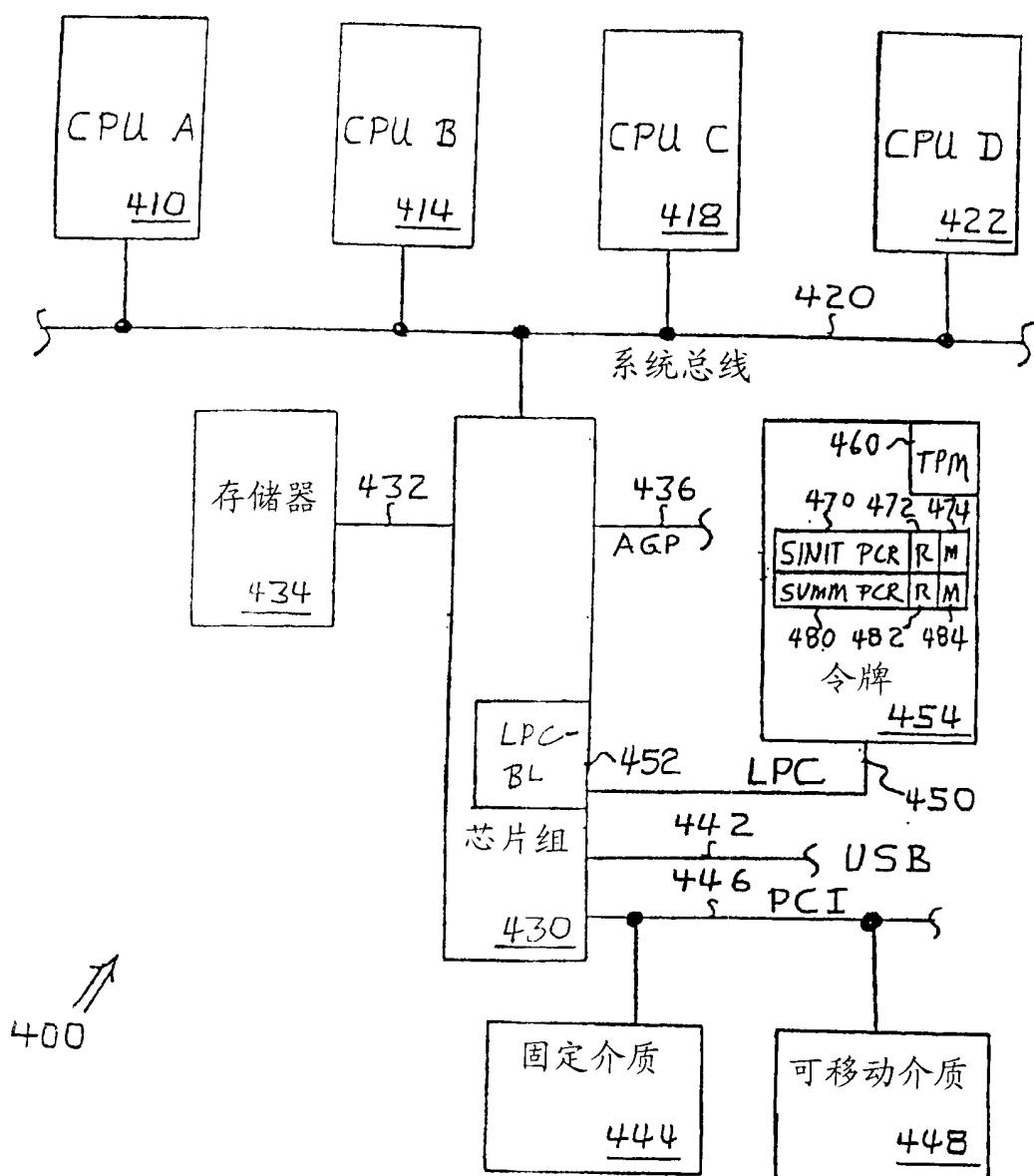


图 4

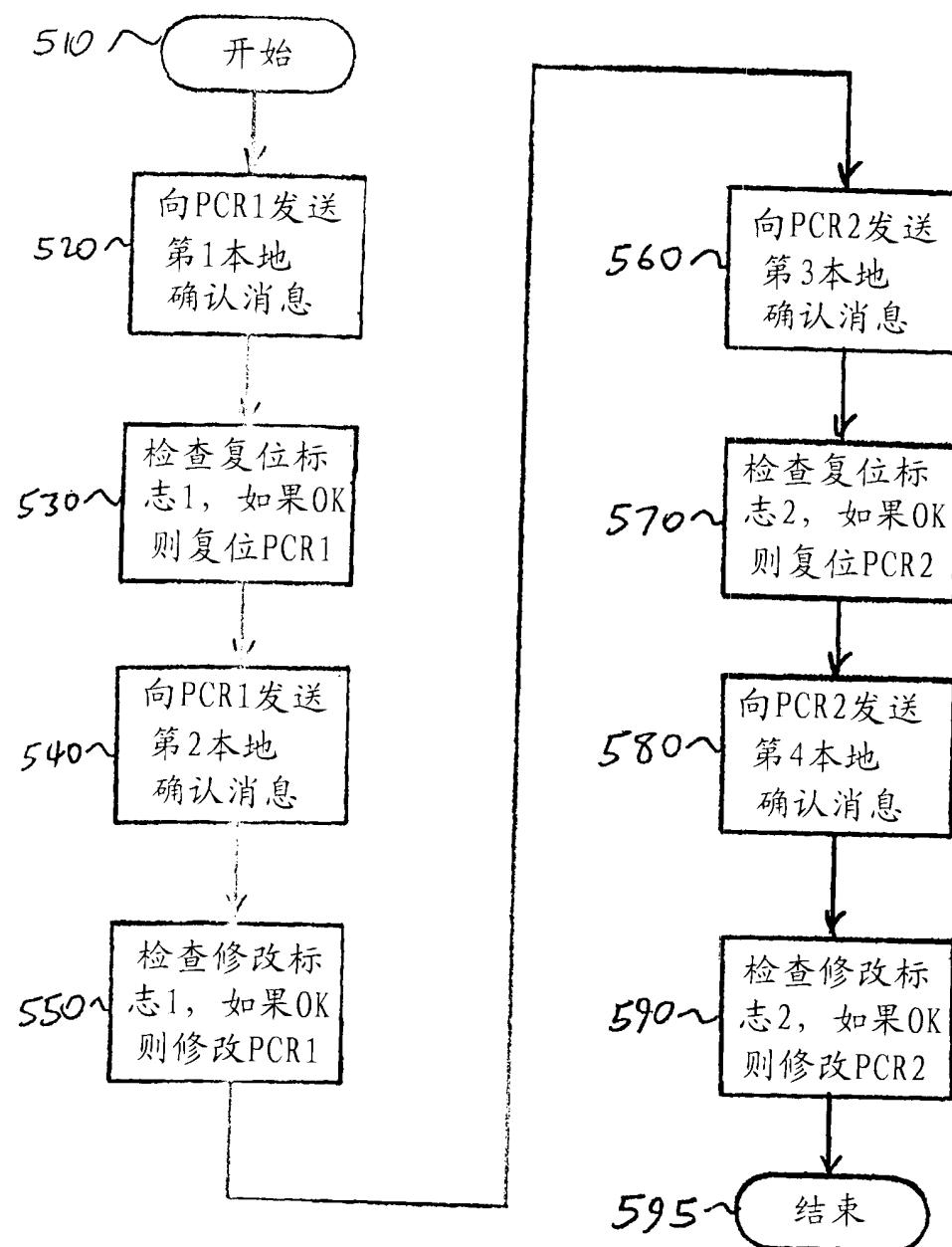


图 5

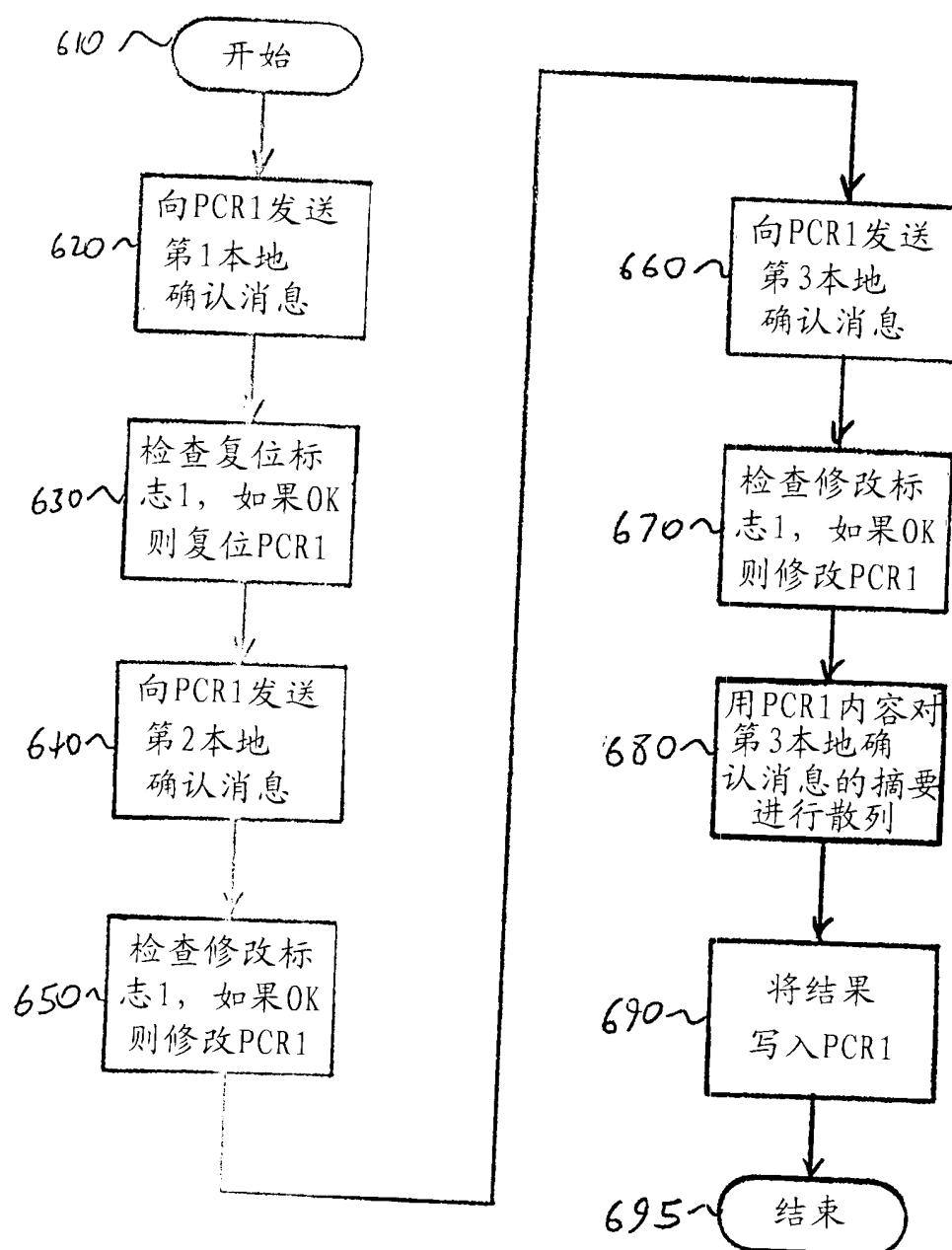


图 6