



[12] 发明专利申请公开说明书

[21] 申请号 00818055.5

[43] 公开日 2003 年 4 月 30 日

[11] 公开号 CN 1415085A

[22] 申请日 2000.10.23 [21] 申请号 00818055.5

[30] 优先权

[32] 1999.11.5 [33] US [31] 09/434,973

[86] 国际申请 PCT/US00/41489 2000.10.23

[87] 国际公布 WO01/33322 英 2001.5.10

[85] 进入国家阶段日期 2002.7.1

[71] 申请人 英特尔公司

地址 美国加利福尼亚州

[72] 发明人 S·简 S·-S·曹

[74] 专利代理机构 上海专利商标事务所

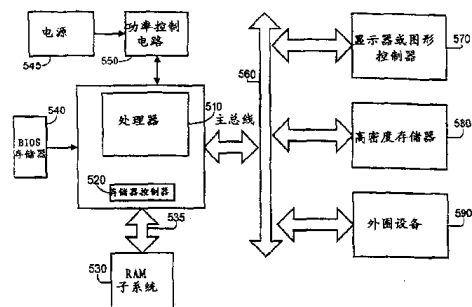
代理人 孙敬国

权利要求书 5 页 说明书 9 页 附图 5 页

[54] 发明名称 睡眠状态转换

[57] 摘要

一种系统具有具备多种状态(包括唤醒状态和睡眠状态)的处理器、包括存储器控制器和各种存储器设备的存储器子系统,以及第二存储器。当从睡眠状态转换为唤醒状态时,第二存储器中的系统使用软件来对存储器控制器进行初始化。系统检测到来自唤醒事件的触发,并响应于该来自唤醒事件的触发来执行被存储在第二存储器中的软件,以便对存储器控制器进行初始化,然后,在初始化之后执行第一存储器以外的软件。



1. 一种在包括处理器、第一存储器、第一存储器控制器和第二存储器的系统中用于在唤醒状态与睡眠状态之间转换的方法，其特征在于，包括检测从睡眠状态转换为唤醒状态的触发，
响应于该检测来初始化第一存储器控制器，所述初始化包括执行第二存储器中的软件，以及
在初始化之后执行第一存储器中的软件。
2. 如权利要求 1 所述的方法，其特征在于，还包括
在被存储在第一存储器中的软件控制下，为从唤醒状态转换为睡眠状态做准备，以及
转换为睡眠状态。
3. 如权利要求 2 所述的方法，其特征在于，
所述被存储在第一存储器中的软件包括操作系统软件。
4. 如权利要求 2 所述的方法，其特征在于，
所述第一存储器包括 RDRAM。
5. 如权利要求 4 所述的方法，其特征在于，
所述第一存储器控制器驻留在与处理器相同的芯片中。
6. 如权利要求 5 所述的方法，其特征在于，
所述第二存储器中的软件包括 BIOS 软件。
7. 如权利要求 5 所述的方法，其特征在于，
所述第二存储器中的软件包括不可由操作系统进行存取的软件。
8. 如权利要求 5 所述的方法，其特征在于，
所述第二存储器中的软件包括系统管理模式软件。

9. 如权利要求 1 所述的方法，其特征在于，
所述处理器和存储器控制器具有用于接收各自的时钟信号的输入，所述方法还包括防止在检测之前接收各自的时钟信号。

10. 如权利要求 2 所述的方法，其特征在于，
所述准备包括
将处理器配置成响应于来自操作系统的第一个睡眠触发信号而执行被存储在第二个存储器中的软件，
从操作系统接收第一睡眠触发信号，
响应于该接收来执行被存储在第二存储器中的软件，
将处理器重新配置成响应于睡眠触发信号而转换为睡眠状态，以及，
接收第二睡眠触发信号。

11. 如权利要求 10 所述的方法，其特征在于，
所述被存储在第二存储器中的软件包括系统管理模式软件，所述方法还包括响应于对第一睡眠触发的接收而生成一个系统管理中断。

12. 如权利要求 10 所述的方法，其特征在于，
所述初始化根据在转换为睡眠状态之前所执行的最后指令后面的指令来执行被存储在第二存储器中的软件。

13. 如权利要求 2 所述的方法，其特征在于，
所述转换使处理器转入 S1 状态。

14. 如权利要求 2 所述的方法，其特征在于，
所述转换使处理器转入 S2 状态。

15. 如权利要求 2 所述的方法，其特征在于，
所述准备包括刷新高速缓冲存储器。

16. 如权利要求 2 所述的方法，其特征在于，还包括在初始化之前重置处理器。

17. 一种在包括处理器、第一存储器、第一存储器控制器和第二存储器的一种系统中用于在唤醒状态与睡眠状态之间转换的方法，其中，所述处理器和存储器控制器具有用于接收各自的时钟信号的输入，所述第一存储器存储操作系统软件，其特征在于，包括

在操作系统软件的控制下，为从唤醒状态转变为睡眠状态做准备，所述准备包括将地址空间映射配置成在检测之后指向第二存储器，

防止对各自的时钟信号的接收，

转换为睡眠状态，

检测从睡眠状态转换为唤醒状态的触发，

响应于该检测来初始化第一个存储器控制器，初始化包括执行第二个存储器中的 BIOS 软件，

在初始化之后，执行操作系统软件。

18. 如权利要求 17 所述的方法，其特征在于，还包括

将一个 BIOS 恢复地址存储在第一存储器中，以及

在使用被存储在第一存储器中的恢复地址进行初始化之后，将控制从 BIOS 转移到操作系统。

19. 如权利要求 18 所述的方法，其特征在于，还包括

在检测之前切断对第一存储器控制器的供电。

20. 一种系统，其特征在于，包括

具有唤醒状态和睡眠状态的处理器，

第一存储器，

第一存储器控制器，

第二存储器，以及

被存储在第二存储器中的软件，所述软件响应于用信号通知从睡眠状态转换为唤醒状态的触发信号来执行对第一存储器控制器的初始化。

21. 如权利要求 20 所述的系统，其特征在于，
所述第一存储器存储操作系统软件，
所述第二存储器是非易失性存储器，
所述第二存储器存储 BIOS 软件，以及，
所述操作系统软件被配置成控制从唤醒状态到睡眠状态的转换。

22. 如权利要求 21 所述的系统，其特征在于，
所述操作系统在从唤醒状态转换为睡眠状态之前，将一个 BIOS 恢复地址
存储在所述第一存储器中，以及
在存储器控制器被初始化之后，BIOS 将控制返回到使用被存储的 BIOS 恢
复地址的操作系统。

23. 如权利要求 21 所述的系统，其特征在于，
所述第一存储器包括 RDRAM。

24. 如权利要求 21 所述的系统，其特征在于，
所述第二存储器包括 RAM。

25. 如权利要求 21 所述的系统，其特征在于，还包括
系统管理模式软件，所述系统管理模式软件不可由操作系统进行存取，其
中系统管理模式软件对第一存储器进行初始化。

26. 如权利要求 20 所述的系统，其特征在于，
所述处理器、第一存储器和第一存储器控制器具有时钟输入，所述系统包
括防止处理器、第一存储器和第一存储器控制器的内部计时的时钟禁止电路。

27. 如权利要求 20 所述的系统，其特征在于，
所述处理器和第一存储器控制器驻留在一个共同的芯片内。

28. 如权利要求 20 所述的系统，其特征在于，还包括

驻留在与所述处理器和第一存储器控制器相同的芯片中的高速缓冲存储器，其中操作系统被配置成在从唤醒状态转换为睡眠状态之前刷新高速缓冲存储器。

29. 一种便携式计算机系统，其特征在于，包括
功率存储介质，
显示器，
处理器，
处理器时钟，
第一存储器，
第一存储器控制器，
第二存储器，
其中，所述系统包括唤醒状态和睡眠状态，
其中，所述处理器和第一存储器控制器在睡眠状态中不被计时，以及
其中，所述第二存储器中的软件响应于从睡眠状态到唤醒状态的转换来对第一个存储器控制器进行初始化。

30. 如权利要求 29 所述的系统，其特征在于，还包括
使处理器、第一存储器控制器和第一存储器内部的功率流动不能进行的电路。

睡眠状态转换

发明领域

本发明涉及睡眠状态转换。

背景技术

为了在处理器系统中实施低功耗“睡眠”状态，英特尔(INTEL(TM))和其他公司已提出“高级配置与功率接口说明书”(“ACPI”)。ACPI定义了操作系统与硬件之间的一个接口，该接口允许操作系统和硬件相互作用，同时允许操作系统和硬件的设计独立演变。本说明书的附录中再现了于1999年2月2日发行的修订本1.0b的ACPI说明书中有关S1和S2睡眠状态的描述。

RAM子系统也可以具有低功耗状态。在一些RAM子系统中，存储器控制器与使用特定协议的存储器芯片进行通信。存储器控制器是一种智能设备，它在开始从存储器芯片读取数据和将数据写到存储器芯片这项正常的操作之前被初始化。在由加利福尼亚州的Mountainview的RAMBUS(TM)公司开发的RDRAM(TM)RAM子系统中，存储器控制器包括一个RAMBUS ASIC CELL(“RAC”)，该单元控制与存储器芯片的电接口，执行多路复用和多路分解功能，并在跟存储器芯片的高速专用串行接口与处理器所用的较低速度并行接口之间转换数据。可以减少对RDRAM子系统的供电，以节省功率。在被断电之后，必须重新初始化RDRAM子系统。

发明概述

系统有一个具有包括唤醒状态和睡眠状态的多种状态的处理器、包括一个存储器控制器和存储器设备的存储器子系统，以及第二存储器。当从睡眠状态转换为唤醒状态时，系统使用第二存储器中的软件来初始化存储器控制器。系统检测来自唤醒事件的触发，并响应于该唤醒事件的触发来执行被存储在第二存储器中的软件，以便初始化存储器控制器，然后在初始化之后执行第一存储器以外的软件。

本发明的另一个方面，存储器子系统是基于RAM并且存储一些或所有的操

作系统软件。初始化存储器控制器的软件被存储在 BIOS 存储设备中。在从唤醒状态转换为睡眠状态之前，操作系统控制对转换的准备。

附图简要说明

图 1 表示本发明的处理器系统的方框图。

图 2 是表示由图 1 中的系统执行的一组状态转换的流程图。

图 3 是表示由图 1 中的系统执行的到 S1 状态和从 S1 状态的转换的流程图。

图 4 是表示由图 1 中的系统执行的到 S2 状态和从 S2 状态的转换的流程图。

图 5 是表示另一种处理器系统的方框图。

图 6 表示另一种处理器系统。

详细描述

如图 1 所示，处理器 10 被连接到存储器的控制器中枢 20。处理器可以是“奔腾”II 型处理器、其他的通用处理器或专用控制器。处理器可以是工作站、桌面个人计算机、便携式计算机或电信、视频或图形设备的一部分。存储器的控制器中枢 20 被连接到主存储器 30 并对其加以控制。存储器的控制器中枢 20 也处理图形通信量，以及到和来自 I/O 控制器中枢的通信量。例如，主存储器 30 可以是 RAMBUS 存储器系统，该系统包括多个存储器模块，每个模块具有 RDRAM 存储器芯片。单独模块的尺寸与标准双列直插式存储器模块的尺寸同等。

存储器的控制器中枢 20 与使用被分组化的(packetized)协议的主存储器 30 相互作用。存储器控制器担任 RAM 总线与处理器 10 之间的翻译员，以便处理器不需要关注 RAM 结构或操作的细节。也可以采用其他的高速 RAM 技术，这些技术使用存储器的控制器来存取主存储器。

存储器的控制器中枢 20 和主存储器 30 由存储器时钟 40 计时。例如，可以使用双重相位计时按 400 MHz 来对主存储器进行有差别的计时，以便提供 800 MHz 的有效时钟比率。处理器由处理器时钟 50 来计时。非易失性存储器 60 也经由 I/O 控制器中枢 55 被耦合到处理器 50。非易失性存储器 60 可以是 ROM、EPROM、EEPROM、有备用电池的 RAM 和类似物。非易失性存储器 60，存储基本输入/输出软件(BIOS)，并可包括系统管理模式软件(SMM)。SMM 也可以驻留在主存储器中。

非易失性存储器 60 存储被用来初始化存储器的控制器中枢 20 的初始化软

件 70。如果存在的话，初始化软件 70 可以是 BIOS 的一部分或 SMM 软件的一部分。在一些应用程序中，初始化软件可以独立于 BIOS，例如在不具有处理器 10 以外的 BIOS 软件的系统中。存储器的控制器中枢 20 包括内部寄存器 90，这些寄存器控制地址空间映射（“PAM 寄存器”）。这些寄存器控制地址发生器是依赖非易失性存储器 60 获得指令和数据，还是依赖主存储器 30。另一方面，PAM 寄存器可以驻留在 I/O 控制器中枢 55 中或处理器中的一个分开的阱中，以便当处理器 10 被断电时，不会失去功率。显示器或图形控制器 95 被连接到处理器 10。

处理器 10 可以包括高速缓冲存储器 110，以加快存储器存取时间。高速缓冲存储器可以在处理器芯片或封装的内部，也可以在外部。I/O 控制器 55 包含一个唤醒触发状态机器 100，以便处理从处理器以外接收到的来自唤醒事件的触发。状态机器 100 也可以驻留在存储器的控制器中枢 20 或处理器 10 中。该状态机器使处理器能够在任何软件开始执行之前的某个时间响应于唤醒事件。

一旦系统运行，系统就处于唤醒状态，存储器的控制器中枢 20 被初始化，操作系统 80 的各个部分被载入主存储器 30，系统处于正常的操作状态。

参考图 2，操作系统可以确定应该节省功率，并且系统应该进入睡眠状态。可以根据例如，系统空闲超时、来自用户的请求、来自硬件设备的请求，以及例如，低电池或高温指示或来自应用程序的请求来引发这种确定。

在进入睡眠状态之前，操作系统在步骤 200 中准备转换。这种准备可以包括各种保管任务、高速缓冲存储器刷新、内容保存等。操作系统也可以确定哪些设备将被置于“睡眠”状态。在将系统设计成使功率节省最大化的情况下，整个系统可以被置于睡眠状态。在更简单的设计中，只有处理器和存储器子系统可被置于睡眠状态，而外围设备被充分供电或被关闭。操作系统也选择理想的睡眠状态，并在睡眠状态寄存器中设置一个或多个相应的位。例如，ACPI 说明书包括为低等待时间返回唤醒状态提供的 S1 和 S2 睡眠状态。

在步骤 210 中，处理器转换为睡眠状态。实现该转换的一种方法是在睡眠使能寄存器中设置相应的位。然后，软件或硬件处理检测到这个位被设置，并发布睡眠信号到相应的部件。处理器时钟 50 被断电。通过切断设备本身的电源，或通过引入的信号与每个芯片内部的内部分布线用电力断开，可以实现断电。例如，处理器时钟 50 可以在运行，但处理器可以用电力断开引入的时钟信号，所以，处理器的内部部件没有正在被计时。同样，单独的设备可以用

这些设备内部的电路来被断电，以防止电力流向设备内的一些或所有部件。在 RDRAM 系统中，存储器的控制器中枢 20、主存储器 30 和存储器时钟 40 被断电。当主存储器被断电时，其内容并没有失去，但主存储器设备转变为消耗极少功率的断电状态。当主存储器被断电时，主存储器 20 内的内部自我更新机制保存存储器内容。存储器时钟 40 也转换为低功耗状态。在低功耗状态中，可以或可以不移动物理功率。

在步骤 220 中，检测到一个来自唤醒事件的触发。该触发用信号通知处理器 10 应该恢复正常的操作。在一些应用程序中，这可能是返回全速、全功率模式。在其他的应用程序中，系统可能觉醒到一种更呆滞的状态，其中，处理器 10 可以不按全速运行。来自唤醒事件的触发可能由系统本身以外的一个来源例如，用户按“通电”或“恢复”键、从调制解调器引入的调用信号或其他电话接收器生成，或者，它可能由与一天中的某个特定时间或其他某个事件例如，预定的系统维修相联系的定时器生成。

响应于被检测到的来自唤醒事件的触发，系统在步骤 230 中初始化存储器控制器。在 RDRAM 系统中，这包括初始化 RAC 和 RDRAM 核心。在初始化期间所执行的其他功能可以是 RAM 总线驱动器的再校准、RAM 总线时钟的同步和存储器控制器的一般重置。该初始化并不是专门由硬件来执行，而是涉及执行来自非易失性存储器 60 的初始化软件 70。

在存储器的控制器中枢 20 被初始化之后，控制在步骤 240 中从初始化软件 70 被传递给存储在主存储器 30 中的操作系统 80。现在，操作系统 80 处理来自唤醒事件的触发。该处理可包括恢复处理器环境、执行快速系统诊断或执行通常在唤醒事件之后被执行的其他程序。

图 3 表示实施具有 RDRAM 的 S1 睡眠状态的一个实施例。在正常的操作中，设置睡眠使能位来使处理器转变为 S1 睡眠状态。但是，在这个实施例中，系统管理模式软件被用来在睡眠状态与 RDRAM 之间进行调停。系统管理模式软件的一个部分被存储在非易失性存储器 60 中，该非易失性存储器也存储了 BIOS (BIOS 存储设备)。但是，系统管理模式软件不能对操作系统进行存取，操作系统无法直接跳越到系统管理模式软件内的程序。

考虑到对有效并完全地从操作系统变换到系统管理模式软件的控制，处理器被配置成响应于具有系统管理中断 (SMI) 的睡眠触发。为了实现这一点，操作系统在步骤 300 中将一个位写到寄存器。该寄存器告诉硬件响应于睡眠使能

信号而不是响应于向睡眠状态的转换来生成一个 SMI。响应于该 SMI，处理器指导对系统管理模式软件的控制。在步骤 310 中，服务 SMI 的 SMI 处理程序刷新高速缓冲存储器。该高速缓冲存储器刷新避免了当指令提取时在 L2 高速缓冲存储器中发生统一的反写。如果执行这个步骤，则在发生从睡眠状态的处理器转换之前将没有进一步的存储器书写。接下来，如步骤 320 中所示，SMI 处理程序设置 PAM 寄存器，以指向 BIOS 存储设备。PAM 存有系统的地址空间映射。一旦 PAM 寄存器指向 BIOS 存储设备，就将从该设备而不是从 RDRAM 取出指令和数据。在步骤 330 中，SMI 处理程序执行跳越/分支指令，该指令指向 BIOS 存储设备中的一个项目。

在步骤 340 中，SMI 处理程序清除一个位，该位使处理器响应于睡眠使能而生成 SMI。现在，处理器被配置成响应于睡眠使能信号而进入睡眠状态。在步骤 350 中，第二次设置睡眠使能位。但是，这次设置该位的是 SMI 处理程序，而不是操作系统。SMI 处理程序也识别所需的睡眠模式。在这个实施例中，所需的睡眠模式是 S1 状态。处理器检测到睡眠使能位被设置，在步骤 360 中，系统转变为 S1 睡眠状态。处理器时钟和 RDRAM 时钟被断电。在这个实施例中，处理器和 RDRAM 子系统有其各自的时钟。在其他的实施例中，处理器和存储器子系统可使用与其各自的时钟相同的时钟。一旦 RDRAM 子系统被断电，它就要求重新初始化。

在步骤 370 中，一个来自唤醒事件的触发由硬件接收，该硬件用信号通知系统应该从睡眠状态返回唤醒状态。时钟返回到其通电状态。处理器恢复指令提取。在步骤 380 中，被取出的第一个指令是在转变为 S1 状态之后来自 SMI 处理程序的指令。然后，在步骤 385 中，SMI 处理程序执行这些指令，以初始化 RDRAM。在步骤 390 中，SMI 处理程序然后设置 PAM 寄存器，以指向 RDRAM 中的一个项目。然后，SMI 处理程序执行返回指令，控制转移到操作系统。在步骤 395 中，操作系统执行下一个指令，该指令在其中设置睡眠使能位的指令之后。系统已成功地从睡眠状态返回，并继续正常的操作。

图 4 表示使用 S2 状态的一个实施例。操作系统在步骤 410 中进入睡眠状态，并将 BIOS 所用的恢复地址存储在 RDRAM 中。操作系统在步骤 420 中刷新高速缓冲存储器，通过将 S2 状态写入睡眠类型的寄存器来识别睡眠状态，并通过将相应的信息写入睡眠使能寄存器来启动睡眠状态。在步骤 430 中，处理器和 RDRAM 时钟被断电。在 S2 状态中，功率实际上移到处理器 10，以便处理

器 10 不消耗有效功率或泄漏功率。

系统在步骤 440 中处于 S2 状态。在步骤 450 中，检测到一个来自唤醒事件的触发。功率被还原到时钟。处理器重置 (CPURST#) 也被宣称重置处理器。系统在步骤 460 中完成重置，并开始位置 FFFFFFF0h 处执行软件。RAM 寄存器被配置成指向 BIOS 存储设备，而不是遮盖 RDRAM 中的这个空间。换句话说，通过改变 PAM 寄存器来指向 BIOS 存储设备，硬件状态机器可以对应于唤醒事件。在步骤 470 中，BIOS 初始化 RDRAM。在步骤 480 中，BIOS 重新指导 PAM 寄存器执行来自 RDRAM 的软件。在步骤 410 中，BIOS 经由被存储在 RDRAM 中的恢复地址将控制传递到操作系统。在步骤 490 中，操作系统处理来自唤醒事件中中断。在步骤 495 中，完成从睡眠状态的恢复，并恢复处于唤醒状态的正常操作。

图 5 表示在较大系统的环境例如，可能在桌面系统、便携式计算机、便携式通信设备、置顶盒或视频与图形控制器内的处理器与存储器子系统。处理器 510 和存储器控制器 520 被并入相同的芯片内。处理器经由通过存储器总线 535 的存储器控制器 520 与存储器 530 (较佳的是 RDRAM) 相互作用。存储器控制器 520 醒来，并且通过执行来自 BIOS 存储设备 540 的软件来对其进行初始化。在一些应用程序中，可能需要将 BIOS 存储设备并入与处理器 510 和存储器控制器 520 相同的芯片。

系统由电源 545 供电。在便携式系统中，电源 545 可能是电池。在桌面或置顶设备中，电源可能是驱动 AC 电线电源的 DC 电源。功率由电源控制电路 550 分配。电源控制电路响应于处理器，以便减少或切断到系统的各个部分的电力。电源控制电路 550 也可以将低功耗条件通知处理器 510。如图所示，电源控制电路按独立于主总线 560 的一种方式与处理器连接。在其他的实施例中，电源控制电路可以被视作连接到主总线的任何其他的外围设备。在桌面系统中，主总线可能是 PCI 总线。在一些图形密集的系统，显示器或图形控制器 580 可以具有到处理器的其自己的专门路径或高速路径。显示器或图形控制器 580 可以通过一个分开的总线被连接到处理器 10 或存储器的控制器中枢 20，或者可以与处理器核心中的存储器控制器集成。高密度存储器 590 通常将会是硬盘驱动器。外围设备 590 将随特定的应用而变化。

参考图 6，示出核心芯片集的三种不同的配置。配置 1 具有处理器 610 (CPU)、图形控制器 620 (GFX) 和被集成到单个芯片 640 的存储器控制器 630 (也被称作“存储器的控制器中枢”或 MCH)。I/O 控制器中枢 650 (ICH) 和视

频控制器中枢 655 (VCH) 被示作截然不同的芯片。VCH 也可以被并入芯片 640。ICH 650 控制主总线例如，图 5 所示的主总线 560 的操作。ICH 650 具有重置芯片 640 的输出 (NRST)。ICH 650 具有重置主总线例如，PCI 总线的分开的输出 (PCIRST#)。在配置 2 中，处理器 610、GFX 620 和 MCH 630 都在分开的芯片中。在配置 3 中，处理器 610 在其自己的芯片中。GFX 620 和 MCH 630 在单个芯片中。在配置 2 和 3 中，CPU 610 在 ICH 650 的控制下具有其自己的重置输入。

在配置 1 中，芯片 640 和它所有的部件在睡眠状态例如，S2 状态中被断电。在配置 2 中，CPU 610 和 MCH 630 被断电。保持对 GFX 620 的供电，以维持显示。换句话说，GFX 620 可以被断电，以节省更多的功率。在配置 3 中，CPU 610、GFX 620 和 MCH 630 被断电。除了停止时钟以外，减少对各个部件的供电大大减少了泄露电流。此外，ICH 650 与其他部件之间的接口被隔离。这个接口不是 PCI 接口，而是基于通讯联系协议的接口。在每种配置中，为 ICH 供电。ICH 具有从睡眠状态恢复所必需的硬件。减少或消除来自 CPU 610、GFX 620 和 MCH 630 的 S2 状态中的泄露功率将用 0.18 微米处理技术等的一种实质的方法来延长电池的寿命。

所揭示的实施例只是起示范的作用。其他实施例在以下权利要求的范围内。

附录

在 1999 年 2 月 2 日发布的修订本 1.0b 的 ACPI 说明书中的 S1 和 S2 睡眠状态

9.1.1 S1 睡眠状态

S1 睡眠状态被定义为一种低唤醒等待时间睡眠状态。在这种状态中，不会失去系统环境 (CPU 或芯片集)，并且，硬件负责维持所有的系统环境，这包括 CPU、高速缓冲存储器、存储器和所有芯片集 I/O 的环境。以下是 S1 睡眠状态实施替换的一些例子。

9.1.1.1 S1 睡眠状态实施 (例 1)

这个例子提及 IA 处理器，该处理器通过维护 STPCLK# 信号来支持停止同意

状态。当 SLP_TYPx 被编程为 S1 值(OEM 选择一个值, 然后将其放在_S1 对象中)并且 SLP_Enx 位随后被设置时, 硬件可以通过维护 STPCLK#信号到处理器来实施 S1 状态, 从而使其进入停止同意状态。在这种情况下, 系统时钟(PCI 和 CPU)仍然在运行。任何激活的唤醒事件应该使硬件不维护 STPCLK#信号到处理器。

9.1.1.2 S1 睡眠状态实施(例 2)

当 SLP_TYPx 被编程为 S1 值并且 SLP_Enx 位随后被设置时, 硬件将通过以下行动来实施 S1 状态

1. 将处理器放入停止同意状态。
2. 停止处理器的输入时钟, 将处理器放入停止时钟状态。
3. 将系统存储器放入自我更新或暂停更新状态。由存储器本身或通过睡眠状态时期不被停止的其他某个参考时钟来维持更新。
4. 停止所有的系统时钟(维护备用信号到系统 PLL 芯片)。通常, RTC 将继续运行。

在这种情况下, 系统中的所有时钟已经被停止(除了 RTC 的时钟以外)。当发生任何激活的唤醒事件时, 硬件必须保存处理(重新启动系统时钟)。

9.1.2 S2 睡眠状态

S2 睡眠状态被定义为低唤醒等待时间睡眠状态。除了失去 CPU 和系统高速缓冲存储器环境(OS 负责维持高速缓冲存储器和 CPU 环境)以外, 该状态类似于 S1 睡眠状态。此外, 在唤醒事件之后, 控制从处理器的重置矢量开始。在设置 SLP_EN 位之前, ACPI 驱动器将刷新系统高速缓冲存储器。如果平台支持 WBINVD 指令(如 FACP 表格中的 WBINVD 和 WBINVD_FLUSH 标记所指出的), 则 OS 将执行 WBINVD 指令。如果平台不支持 WBINVD 指令以便刷新高速缓冲存储器, 那么, ACPI 驱动器将尝试使用 FACP 表格中的 FLUSH_SIZE 和 FLUSH_STRIDE 字段来手工刷新高速缓冲存储器。硬件负责维持芯片集和存储器环境。以下是 S2 睡眠状态实施的一个例子。

9.1.2.1 S2 睡眠状态实施例子

当 SLP_TYPx 被编程为 S2 值(在_S2 对象中)并且然后设置 SLP_EN 位时, 硬件将通过以下行动来实施 S2 状态

- 停止系统时钟(唯一运行的时钟是 RTC)。

- 将系统存储器放入自我或暂停更新状态。
- 切断 CPU 和高速缓冲存储器子系统的电源。

在这种情况下，当检测唤醒事件时，重置 CPU，但是，核心逻辑和存储器维持它们的环境。执行控制从 CPU 的引导矢量开始。要求 BIOS

- 编程 CPU 的初始引导配置(例如，CPU 的 MSR 和 MTRR 寄存器)。
- 将高速缓冲存储器控制器初始化为其初始的引导尺寸和配置。
- 使存储器控制器能够接受存储器存取。
- 调用醒着的矢量。

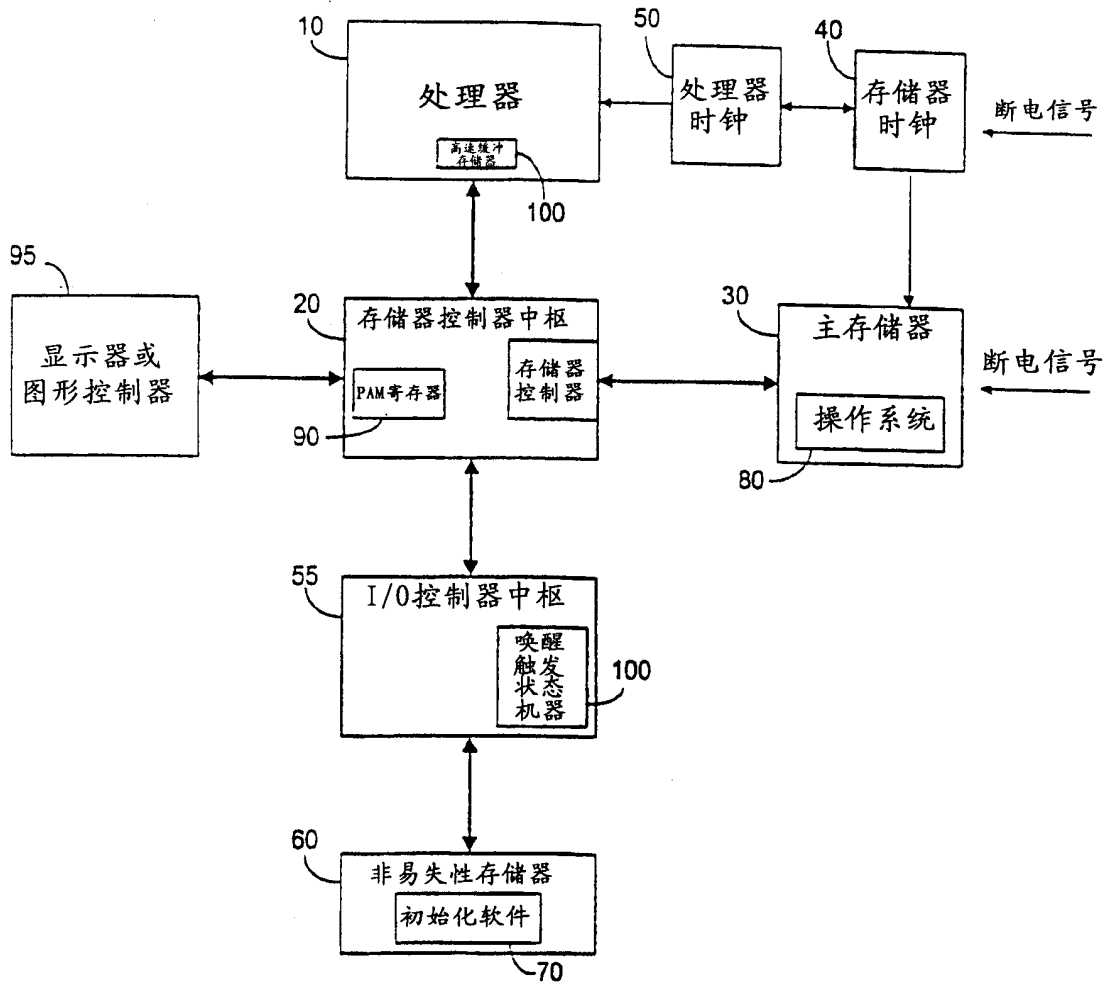


图 1

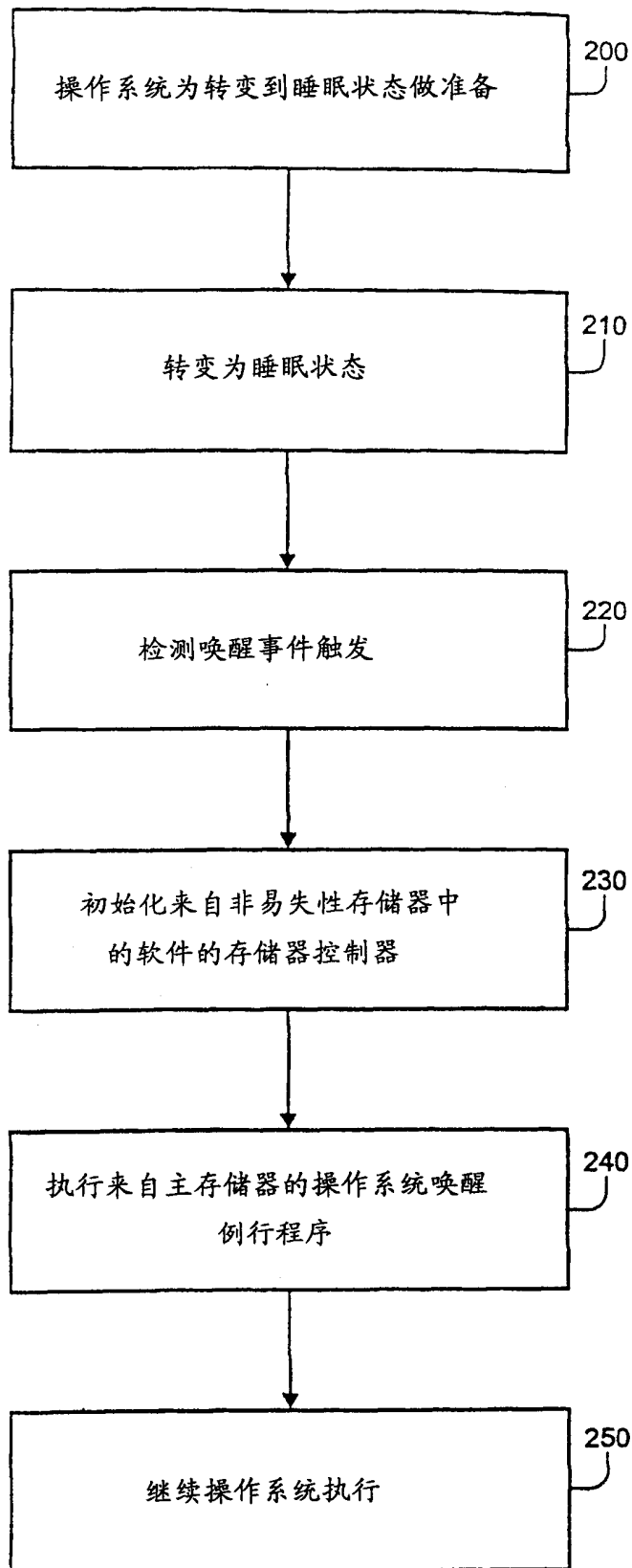


图 2

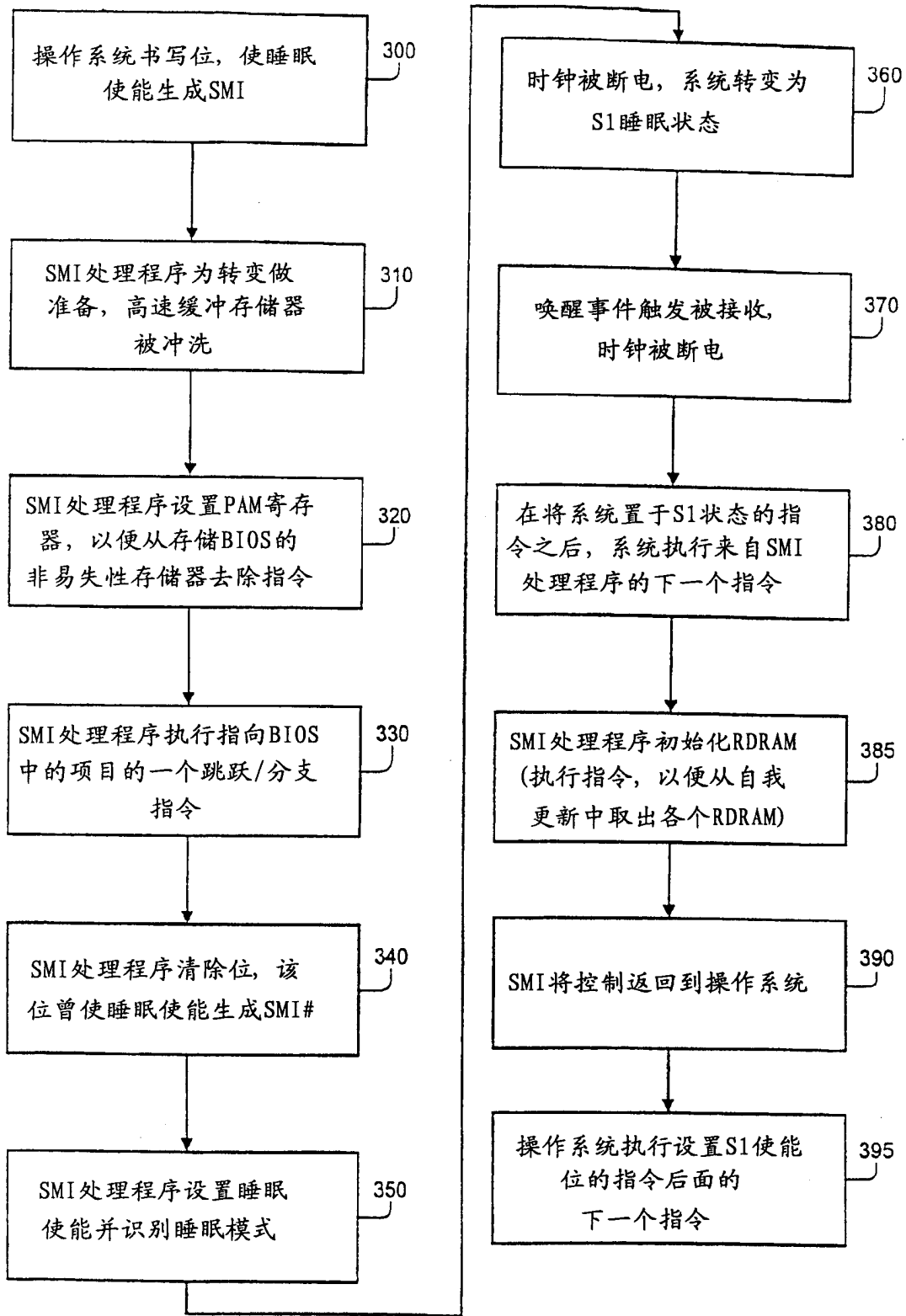


图 3

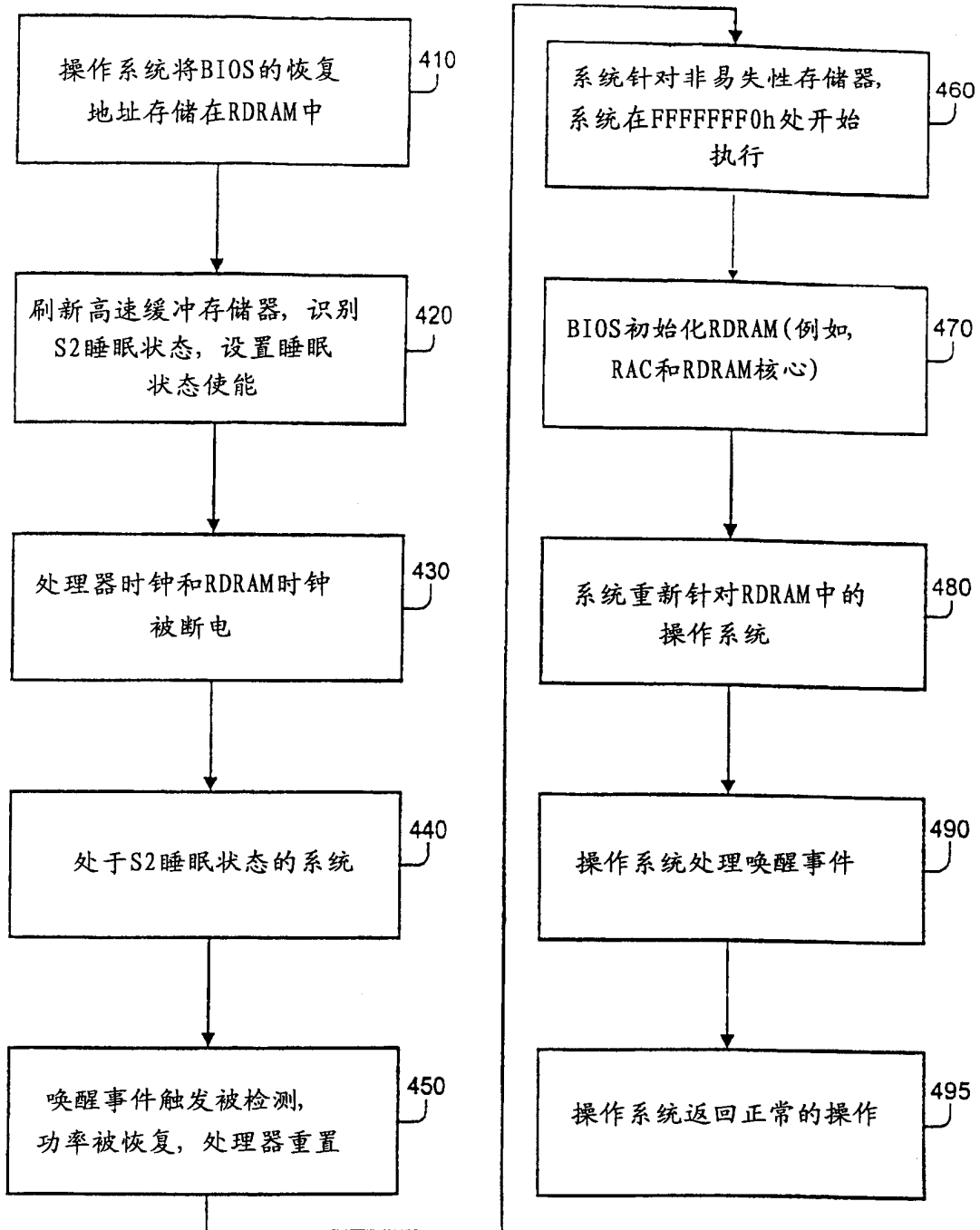


图 4

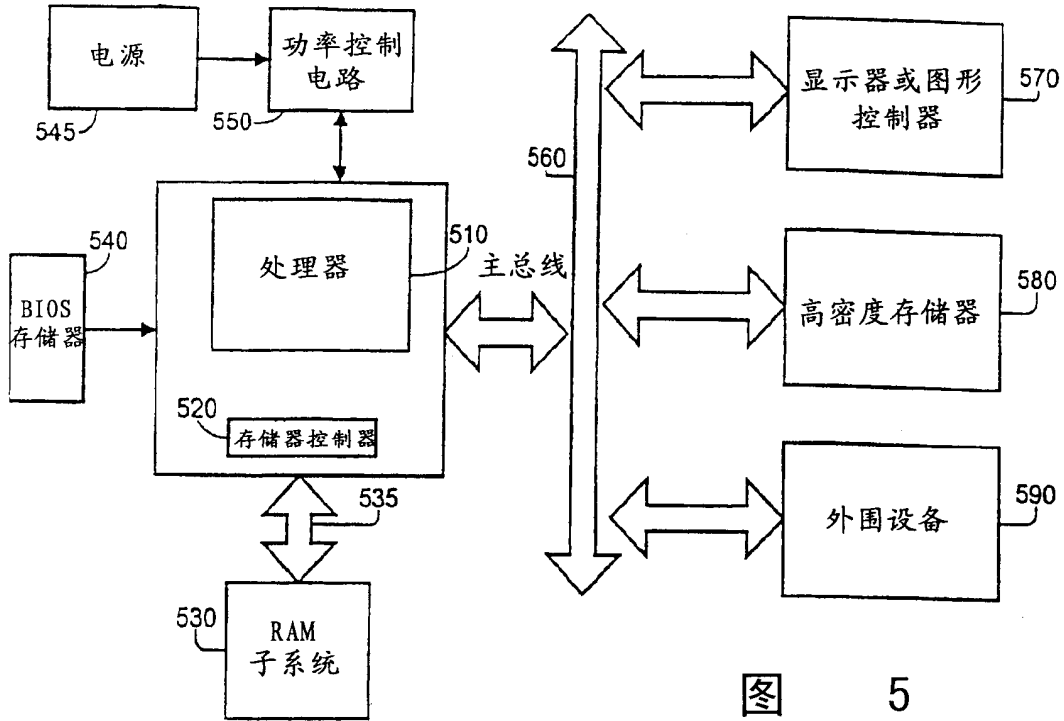


图 5

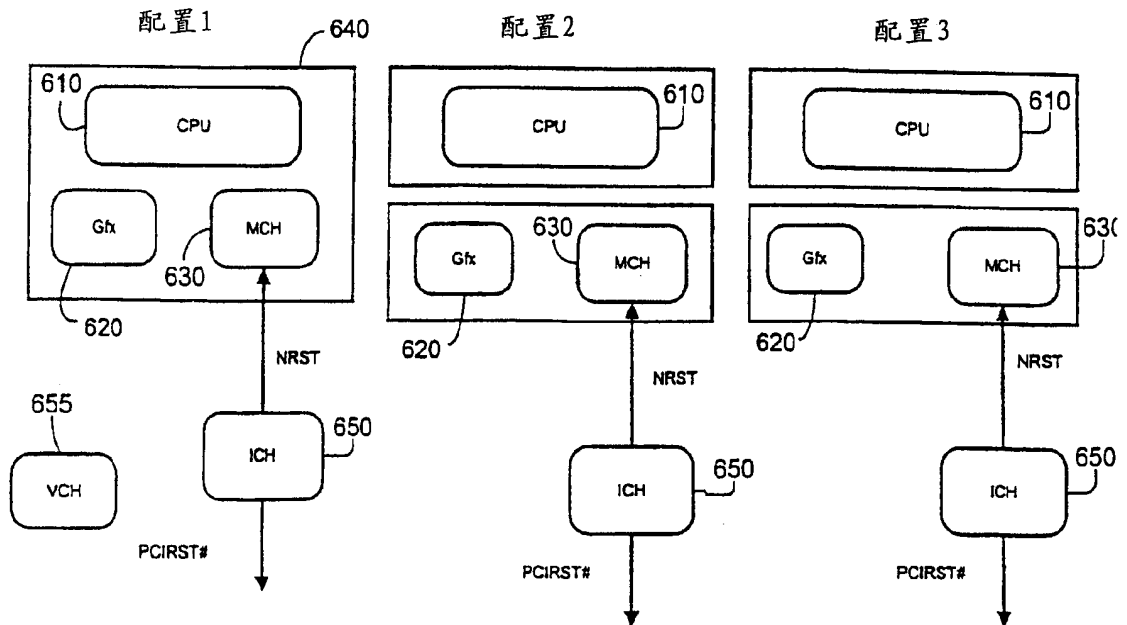


图 6