



[12] 发明专利申请公布说明书

[21] 申请号 200710109071.5

[43] 公开日 2007年12月26日

[11] 公开号 CN 101093415A

[22] 申请日 2007.6.15

[21] 申请号 200710109071.5

[30] 优先权

[32] 2006.6.21 [33] US [31] 11/425,499

[71] 申请人 国际商业机器公司

地址 美国纽约阿芒克

[72] 发明人 C·R·约翰斯 王帆

[74] 专利代理机构 北京市金杜律师事务所

代理人 朱海波

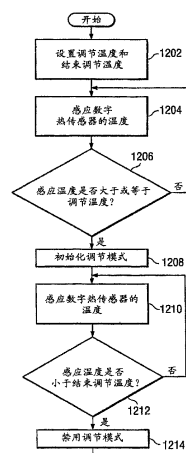
权利要求书 3 页 说明书 38 页 附图 10 页

[54] 发明名称

用于热调节中的滞后的方法、系统和处理器

[57] 摘要

提供了一种用于热调节中的滞后的计算机实现的方法、数据处理系统和处理器。数字热传感器感应集成电路中的温度。确定感应温度是否大于或等于调节温度。作为对感应温度达到或超过调节温度的响应，初始化调节模式。数字热传感器感应新的温度。确定该新感应温度是否小于结束调节温度。作为对该新感应温度小于结束调节温度的响应，禁用调节模式。



1. 一种用于在集成电路中使用滞后来进行热调节的计算机实现的方法，包括：

用数字热传感器感应所述集成电路中的温度；

确定所述感应温度是否大于或等于调节温度；

作为对所述感应温度达到或超过所述调节温度的响应，初始化调节模式；

用所述数字热传感器感应新的温度；

确定所述新感应温度是否小于结束调节温度；以及

作为对所述新感应温度小于所述结束调节温度的响应，禁用所述调节模式。

2. 根据权利要求1所述的方法，其中所述确定步骤、初始化步骤和禁用步骤由驻留在所述集成电路内的热管理控制状态机执行。

3. 根据权利要求1所述的方法，其中所述调节温度大于所述结束调节温度。

4. 根据权利要求1所述的方法，其中所述调节模式包括待初始化的调节模式类型。

5. 根据权利要求4所述的方法，其中所述调节模式类型是以下至少一种：禁用动态调节、正常操作、始终调节 Power 处理器单元或协处理器单元、禁用内核停止安全性、或始终调节 Power 处理器单元或协处理器单元并且禁用内核停止安全性。

6. 根据权利要求1所述的方法，其中所述集成电路是异质多内核处理器。

7. 根据权利要求6所述的方法，其中所述数字热传感器驻留在所述异质多内核处理器中的一个内核内。

8. 根据权利要求6所述的方法，其中所述数字热传感器驻留在所述异质多内核处理器内，但不是在内核内。

9. 根据权利要求1所述的方法，其中所述调节模式阻止指令的分

派。

10. 根据权利要求1所述的方法,其中所述调节模式减缓时钟频率。

11. 根据权利要求1所述的方法,其中所述调节温度是可编程的温度。

12. 根据权利要求1所述的方法,其中所述结束调节温度是可编程的温度。

13. 一种数据处理系统,包括:

总线;

连接到所述总线的存储器,其中所述存储器包括一组指令;以及

连接到所述总线的集成电路,其中所述集成电路执行所述一组指令

以便:

用数字热传感器感应集成电路中的温度;

确定所述感应温度是否大于或等于调节温度;

作为对所述感应温度达到或超过所述调节温度的响应,初始化调节模式;

用所述数字热传感器感应新的温度;

确定所述新感应温度是否小于结束调节温度;以及

作为对所述新感应温度小于所述结束调节温度的响应,禁用所述调节模式。

14. 根据权利要求13所述的系统,其中所述调节温度大于所述结束调节温度。

15. 根据权利要求13所述的系统,其中所述调节模式包括待初始化的调节模式类型。

16. 根据权利要求13所述的系统,其中所述调节模式执行阻止指令的分派和减缓时钟频率中的至少一项。

17. 一种处理器,包括:

至少一个处理内核;

热管理控制状态机;以及

数字热传感器,其中所述处理器执行一组指令以便:

用数字热传感器感应集成电路中的温度；
使用所述状态机确定所述感应温度是否大于或等于调节温度；
作为对所述感应温度达到或超过所述调节温度的响应，使用所述状态机初始化调节模式；
用所述数字热传感器感应新的温度；
使用所述状态机确定所述新感应温度是否小于结束调节温度；以及
作为对所述新感应温度小于所述结束调节温度的响应，使用所述状态机禁用所述调节模式。

18. 根据权利要求 13 所述的处理器，其中所述调节温度大于所述结束调节温度。

19. 根据权利要求 13 所述的处理器，其中所述调节模式包括待初始化的调节模式类型。

20. 根据权利要求 13 所述的处理器，其中所述调节模式执行阻止指令的分派和减缓时钟频率中的至少一项。

用于热调节中的滞后的方法、系统和处理器

技术领域

本申请一般地涉及对热管理的使用。更特别地，本申请涉及一种用于实现热调节中的滞后的计算机实现的方法、数据处理系统和处理器。

背景技术

第一代异质 Cell Broadband Engine™ (BE) 处理器是包括一个 64 比特的 Power PC® 处理器内核和八个单指令多数据 (SIMD) 协处理器内核的多内核芯片，能够进行大规模浮点处理，针对运算密集型工作负载和宽带富媒体应用而进行了优化。高速存储控制器和高带宽总线接口也集成到芯片上。Cell BE 的突破性多内核体系结构和超高速通信能力在很多情况下以最新 PC 处理器性能的 10 倍递送大大改善的实时响应。Cell BE 是操作系统中立的并同时支持多个操作系统。这种类型的处理器的应用的范围可以从具有显著增强的真实感的下一代游戏系统，到形成家庭数字媒体和流式传送内容中心 (hub) 的系统，到用于开发和分布数字内容的系统，并且到加速可视化和超级计算应用的系统。

现在的多内核处理器常常受到热考虑的限制。典型的解决方案包括冷却和电源管理。冷却可能是昂贵的并且/或者难以整合。功率管理一般是粗略的措施，作为对达到热限度的响应，对处理器的很大一部分或整个处理器进行“调节”。诸如热管理之类的其他技术通过只调节超过给定温度的单元来帮助实现这些粗略措施。但是，多数热管理技术会影响应用的实时保证。因此，提供热管理解决方案是有益的，该解决方案为处理器提供一种方法，用以在即使出现需要调节处理器的热状况的情况下也保证应用的实时性。在不能满足实时保证的情况下，通知应用管理者使得可以实现纠正措施。

发明内容

说明性实施例的不同方面提供了一种用于记录集成电路中的最大温度的计算机实现的方法、数据处理系统和处理器。说明性实施例用数字热传感器感应集成电路中的温度。说明性实施例确定感应温度是否大于或等于调节温度。作为对感应温度达到或超过调节温度的响应，说明性实施例初始化调节模式。说明性实施例用数字热传感器感应新的温度。说明性实施例确定该新感应温度是否小于结束调节温度。作为对该新感应温度小于结束调节温度的响应，说明性实施例禁用调节模式。

附图说明

在所附权利要求中阐明了确信是说明性实施例所特有的新颖特征。但是，在结合附图阅读时，参考下面对说明性实施例的详细描述，可以最好地理解说明性实施例本身以及其优选的使用模式、进一步的目的及优势，其中：

图 1 描述了可以实现说明性实施例各方面的数据处理系统的网络的图示；

图 2 描述了可以实现说明性实施例各方面的数据处理系统的框图；

图 3 描述了可以实现说明性实施例各方面的 Cell BE 芯片的示例性示图；

图 4 示出了根据说明性实施例的示例性热管理系统；

图 5 描述了根据说明性实施例的温度曲线图以及可能发生中断和动态调节的各个点；

图 6 描述了根据说明性实施例的用于记录最大温度的操作的流程图；

图 7 描述了根据另一个说明性实施例的用于通过性能监控来跟踪热数据的操作的流程图；

图 8A 和图 8B 描述了根据另外的说明性实施例的针对高级热中断产生的操作的流程图；

图 9 描述了根据另外的说明性实施例的用于在热管理系统中支持深度节能模式和部分良好的操作的流程图；

图 10 描述了根据另外的说明性实施例的针对使热感知软件应用的实时测试能够与温度相独立的热调节控制特征的操作的流程图；

图 11 描述了根据另外的说明性实施例的用于实现对中断等待时间影响最小的热调节控制的操作的流程图；

图 12 描述了根据另外的说明性实施例的用于热调节中的滞后的操作的流程图；以及

图 13 描述了根据另外的说明性实施例的用于实现热调节逻辑的操作的流程图。

具体实施方式

说明性实施例涉及热调节中的滞后。图 1-图 2 被提供为可以实现说明性实施例的数据处理环境的示例性示图。应当理解，图 1-图 2 只是示例性的，并非旨在明确或暗示任意关于可以实现实施例各方面的环境的限制。在不偏离说明性实施例的精神和范围的情况下，可以对所描述的环境进行很多修改。

现在参考附图，图 1 描述了可以实现说明性实施例各方面的数据处理系统的网络的图示。网络数据处理系统 100 是可以实现说明性实施例的计算机网络。网络数据处理系统 100 包含网络 102，该网络 102 是一种用于在网络数据处理系统 100 内连接在一起的各种设备和计算机之间提供通信链路的媒介。网络 102 可以包括诸如线缆、无线通信链路或光纤电缆之类的连接。

在所描述的示例中，服务器 104 和服务器 106 连接到网络 102 和随之的存储单元 108。另外，客户端 110、112 和 114 连接到网络 102。这些客户端 110、112 和 114 可以是例如个人计算机或网络计算机。在所描述的示例中，服务器 104 向客户端 110、112 和 114 提供诸如启动文件、操作系统映像和应用之类的数据。在该示例中，客户端 110、112 和 114 是服务器 104 的客户端。网络数据处理系统 100 可以包括附加的

服务器、客户端和没有示出的其他设备。

在所描述的示例中，网络数据处理系统 100 是具有网络 102 的因特网，网络 102 表示使用传输控制协议/网际协议 (TCP/IP) 协议组来相互通信的网络和网关的全球集合。在因特网的中心是主节点或主机之间的高速数据通信线路骨干，包括数以千计的商业计算机系统、政府计算机系统、教育计算机系统和其他对数据和消息进行路由的计算机系统。当然，网络数据处理系统 100 还可以实现为多种不同类型的网络，诸如内网、局域网 (LAN) 或广域网 (WAN)。图 1 旨在作为一个示例，而不是作为对不同说明性实施例的体系结构限制。

现在参考图 2，示出了可以实现说明性实施例各方面的数据处理系统的框图。数据处理系统 200 是诸如图 1 中的服务器 104 或客户端 110 之类的计算机的示例，实现说明性实施例的处理的计算机可用代码或指令可以位于该计算机中。

在所描述的示例中，数据处理系统 200 采用中心体系结构，包括北桥和存储控制器中心 (MCH) 202 以及南桥和输入/输出 (I/O) 控制器中心 (ICH) 204。处理单元 206、主存储器 208 和图形处理器 210 连接到北桥和存储控制器中心 202。图形处理器 210 可以通过加速图形端口 (AGP) 连接到北桥和存储控制器中心 202。

在所描述的示例中，LAN 适配器 212 连接到南桥和 I/O 控制器中心 204。音频适配器 216、键盘和鼠标适配器 220、调制解调器 222、只读存储器 (ROM) 224、硬盘驱动器 (HDD) 226、CD-ROM 驱动器 230、通用串行总线 (USB) 端口和其他通信端口 232、以及 PCI/PCIe 设备 234 通过总线 238 和总线 240 连接到南桥和 I/O 控制器中心 204。PCI/PCIe 设备可以包括例如以太网适配器、插入卡和笔记本电脑的 PC 卡。PCI 使用卡总线控制器，而 PCIe 则不使用。ROM 224 可以是例如闪速二进制输入/输出系统 (BIOS)。

硬盘驱动器 226 和 CD-ROM 驱动器 230 通过总线 240 连接到南桥和 I/O 控制器中心 204。硬盘驱动器 226 和 CD-ROM 驱动器 230 可以使用例如集成驱动电子设备 (IDE) 或串行高级技术附件 (SATA) 接口。

超级 I/O (SIO) 设备 236 可以连接到南桥和 I/O 控制器中心 204。

操作系统在处理单元 206 上运行并且调整和提供对图 2 中的数据处理系统 200 内的各种组件的控制。作为客户端, 操作系统可以是市面上有售的操作系统, 诸如 Microsoft® Windows® XP (Microsoft 和 Windows 是微软公司在美国、其他国家或同时在美国和其他国家的商标)。诸如 Java™ 编程系统之类的面向对象编程系统可以结合操作系统而运行, 并提供从在数据处理系统 200 上执行的 Java 程序或应用对操作系统的调用 (Java 是 Sun 微系统公司在美国、其他国家或同时在美国和其他国家的商标)。

作为服务器, 数据处理系统 200 可以是例如运行高级交互执行 (AIX®) 操作系统或 LINUX 操作系统的 IBM eServer™ pSeries® 计算机系统 (eServer、pSeries 和 AIX 是国际商业机器公司在美国、其他国家或同时在美国和其他国家的商标, 而 Linux 是 Linux Torvalds 在美国、其他国家或同时在美国和其他国家的商标)。数据处理系统 200 可以是在处理单元 206 中包括多个处理器的对称多处理器 (SMP) 系统。作为选择, 可以采用单处理器系统。

操作系统、面向对象的编程系统以及应用或程序的指令位于诸如硬盘驱动器 226 之类的存储设备上, 并且可以被载入主存储器 208 以供处理单元 206 执行。说明性实施例的处理由处理单元 206 使用计算机可用程序代码来执行, 这些代码可以位于诸如主存储器 208、只读存储器 224 之类的存储器中, 或位于一个或多个外围设备 226 和 230 中。

本领域的普通技术人员应当理解, 根据不同的实现, 图 1-图 2 中的硬件可以变化。可以使用诸如闪速存储器、等效非易失性存储器或光盘驱动器等其他内部硬件或外围设备来补充或代替图 1-图 2 中所描述的硬件。同样, 说明性实施例的处理还可以应用于多处理器数据处理系统。

在一些说明性示例中, 数据处理系统 200 可以是个人数字助理 (PDA), 该个人数字助理配置有闪速存储器以提供用于存储操作系统文件和/或用户生成的数据的非易失性存储器。

总线系统可以包括一个或多个总线, 诸如图 2 中示出的总线 238 或

总线 240。当然，可以使用任意类型的通信架构或体系结构来实现总线系统，该架构或体系结构提供对附在该架构或体系结构上的不同组件或设备之间的数据的传送。通信单元可以包括一个或多个用于发送或接收数据的设备，诸如图 2 的调制解调器 222 或网络适配器 212。存储器可以是例如主存储器 208、只读存储器 224 或诸如在图 2 中的北桥和存储控制器中心 202 中见到的高速缓存。图 1-图 2 中所描述的示例及上述示例并非意味着暗示体系结构限制。例如，除了采取 PDA 的形式之外，数据处理系统 200 还可以是写字板计算机、膝上型计算机或电话设备。

图 3 描述了可以实现说明性实施例各方面的 Cell BE 芯片的示例性示图。Cell BE 芯片 300 是针对分布式处理的单芯片多处理器实现，该分布式处理的目标在于诸如游戏控制台、桌面系统和服务器之类的富媒体应用。

Cell BE 芯片 300 可以在逻辑上分成以下功能组件：Power PC®处理器单元（PPE）301、协处理器单元（SPU）310、311 和 312、以及存储器流控制器（MFC）305、306 和 307。尽管通过示例示出了协处理器单元（SPE）302、303 和 304 以及 PPE 301，但是可以支持任意类型的处理器单元。尽管图 3 只示出了三个 SPE 302、303 和 304，但示例性的 Cell BE 芯片 300 的实现包括一个 PPE 301 和八个 SPE。CELL 处理器的 SPE 是设计为加速媒体和数据流工作负载的新处理器体系结构的第一实现。

Cell BE 芯片 300 可以是片上系统，使得可以在单个微处理器芯片上提供图 3 所示的每一个单元。此外，Cell BE 芯片 300 是一种异质处理环境，其中每个 SPU 310、311 和 312 可以从系统中的每个其他 SPU 接收不同的指令。此外，SPU 310、311 和 312 的指令集与 Power PC®处理器单元（PPU）308 的指令集不同，例如，PPU 308 可以在 Power™体系结构中执行基于精简指令集计算机（RISC）的指令，而 SPU 310、311 和 312 执行向量化的指令。

每个 SPE 包括一个 SPU 310、311 或 312，它自己的本地存储（LS）区域 313、314 或 315 和具有关联的存储管理单元（MMU）316、317

或 318 的专用 MFC 305、306 或 307 保存并处理存储器保护和访问许可信息。同样，尽管通过示例示出了 SPU，但是可以支持任意类型的处理器单元。另外，Cell BE 芯片 300 实现单元互连总线（EIB）319 和其他 I/O 结构以实现片上和外部数据流。

EIB 319 用作 PPE 301 以及 SPE 302、303 和 304 的主片上总线。另外，EIB 319 与专用于片外（off-chip）访问的其他片上接口控制器进行接口连接。片上接口控制器包括存储接口控制器（MIC）320 和 Cell BE 接口单元（BEI）323，其中 MIC 320 提供两个极速数据速率 I/O（XIO）存储通道 321 和 322，BEI 323 为 Cell BE 300 提供两个高速外部 I/O 通道和内部中断控制。BEI 323 实现为总线接口控制器（BIC，标注为 BIC0 和 BIC1）324 和 325 以及 I/O 接口控制器（IOC）326。两个高速外部 I/O 通道连接到 Redwood Rambus® Asic Cell（RRAC）接口的一端，该接口为 Cell BE 300 提供灵活的输入和输出（FlexIO_0 和 FlexIO_1）353。

每个 SPU 310、311 或 312 都有对应的 LS 区域 313、314 或 315 以及协执行单元（SXU）354、355 或 356。每个单独的 SPU 310、311 或 312 只能从与它关联的 LS 区域 313、314 或 315 内执行指令（包括数据加载和存储操作）。由于这个原因，MFC 直接存储器访问（DMA）操作通过 SPU 的 310、311 和 312 专用的 MFC 305、306 和 307 来执行所有需要的去往或来自系统中其他地方的存储器的数据传送。

在 SPU 310、311 或 312 上运行的程序只使用 LS 地址来引用它自己的 LS 区域 313、314 或 315。但是，还为每个 SPU 的 LS 区域 313、314 或 315 分配一个在整个系统的存储映射内的真实地址（RA）。该 RA 是设备将响应的地址。在 Power PC® 中，应用通过有效地址（EA）来引用存储位置（或设备），然后该 EA 被映射成存储位置（或设备）的虚拟地址（VA），然后该 VA 被映射为 RA。EA 是由应用用来引用存储器和/或设备的地址。这种映射使操作系统能够分配比系统中物理上更多的存储器（也就是称为 VA 的虚拟存储器）。存储映射是系统中所有设备（包括存储器）和它们对应的 RA 的列表。存储映射是对标识设备或存储器将响应的 RA 的真实地址空间的映射。

这使特权软件能够将 LS 区域映射到处理器的 EA 以在一个 SPU 的 LS 和另一个 SPU 的 LS 区域之间实现直接存储器访问传送。PPE 301 还可以使用 EA 来直接访问任意 SPU 的 LS 区域。在 Power PC®中有三个状态（问题、特权和管理）。特权软件是在特权或管理状态下运行的软件。这些状态有不同的访问特权。例如，特权软件可以访问用于将真实存储器映射成应用的 EA 的数据结构寄存器。问题状态是在运行应用并通常被禁用访问系统管理资源（诸如用于映射真实存储器的数据结构）时处理器通常所处的状态。

MFC DMA 数据命令始终包括一个 LS 地址和一个 EA。DMA 命令将存储内容从一个位置复制到另一个位置。在这种情况下，MFC DMA 命令在 EA 和 LS 地址之间复制数据。LS 地址直接指向与 MFC 命令队列对应的关联的 SPU 310、311 或 312 的 LS 区域 313、314 或 315。命令队列是 MFC 命令的队列。有一个队列用来保存来自 SPU 的命令，一个队列用来保存来自 PXU 或其他设备的命令。但是可以安排或映射 EA 以访问系统中的任意其他存储器存储区域，包括其他 SPE 302、303 和 304 的 LS 区域 313、314 和 315。

主存储器（没有示出）由诸如图 2 所示出的系统之类的系统中的 PPU 308、PPE 301、SPE 302、303 和 304 以及 I/O 设备（没有示出）共享。所有保存在主存储器中的信息对系统中的所有处理器和设备来说是可见的。程序使用 EA 来引用主存储器。由于 MFC 代理命令队列、控制和状态设施具有 RA，并且使用 EA 来映射 RA，所以 Power 处理器单元可能在关联的 SPE 302、303 和 304 的主存储器和本地存储器之间使用 EA 来初始化 DMA 操作。

作为示例，当在 SPU 310、311 或 312 上运行的程序需要访问主存储器时，SPU 程序生成具有适当的 EA 和 LS 地址的 DMA 命令并将其放置到它的 MFC 305、306 或 307 的命令队列中。在命令被 SPU 程序放置到队列中之后，MFC 305、306 或 307 执行该命令并在 LS 区域和主存储器之间传送所需要的数据。MFC 305、306 或 307 为由诸如 PPE 301 之类的其他设备生成的命令提供第二代理命令队列。MFC 代理命令队

列典型地用于在启动 SPU 之前将程序存储到本地存储中。MFC 代理命令还可以用于上下文存储操作。

EA 地址为 MFC 提供了一个可以由 MMU 转换为 RA 的地址。转换处理考虑到系统存储器的虚拟化和对在真实地址空间中的存储器和设备的访问保护。由于 LS 区域被映射成真实地址空间，因此 EA 还可以指向所有的 SPU LS 区域。

Cell BE 芯片 300 上的 PPE 301 包括 64 比特的 PPU 308 和 Power PC® 存储子系统 (PPSS) 309。PPU 308 包含处理器执行单元 (PXU) 329、一级 (L1) 高速缓存 330、MMU 331 和替换管理表 (RMT) 332。PPSS 309 包括可高速缓存接口单元 (CIU) 333、不可高速缓存单元 (NCU) 334、二级 (L2) 高速缓存 328、RMT 335 和总线接口单元 (BIU) 327。BIU 327 将 PPSS 309 连接到 EIB 319。

SPU 310、311 或 312 以及 MFC 305、306 和 307 通过具有容量的单向通道相互通信。通道实质上是使用 34 个 SPU 指令中的一个指令来访问的 FIFO，读通道 (RDCH)、写通道 (WRCH) 和读通道计数 (RDCHCNT)。RDCHCNT 返回通道中的信息量。容量是 FIFO 的深度。通道对去往和来自 MFC 305、306 和 307，SPU 310、311 和 312 的数据进行传送。BIU 339、340 和 341 将 MFC 305、306 和 307 连接到 EIB 319。

MFC 305、306 和 307 为 SPU 310、311 和 312 提供两个主要功能。MFC 305、306 和 307 在 SPU 310、311 或 312，LS 区域 313、314 或 315 和主存储器之间移动数据。另外，MFC 305、306 和 307 在 SPU 310、311 和 312 与系统中的其他设备之间提供同步设施。

MFC 305、306 和 307 的实现具有四个功能单元：直接存储器访问控制器 (DMAC) 336、337 和 338，MMU 316、317 和 318，原子单元 (ATO) 342、343 和 344，RMT 345、346 和 347 以及 BIU 339、340 和 341。DMAC 336、337 和 338 维护并处理 MFC 命令队列 (MFC CMDQ) (没有示出)，其包括 MFC SPU 命令队列 (MFC SPUQ) 和 MFC 代理命令队列 (MFC PrxyQ)。十六条目的 MFC SPUQ 处理从 SPU 通道接口接收的 MFC 命令。八条目的 MFC PrxyQ 通过存储映射输入和输出

(MMIO) 加载以及存储操作来处理来自诸如 PPE 301 或 SPE 302、303 和 304 之类的其他设备的 MFC 命令。典型的直接存储器访问命令在 LS 区域 313、314 或 315 与主存储器之间移动数据。MFC DMA 命令的 EA 参数用于指向主存储设备，包括主存储器、本地存储器和所有具有 RA 的设备。MFC DMA 命令的本地存储器参数用于指向关联的本地存储器。

在虚拟模式中，MMU 316、317 和 318 提供地址转换和存储器保护设施以处理来自 DMAC 336、337 和 338 的 EA 转换请求并送回已转换的地址。每个 SPE 的 MMU 维护区段监视缓冲器 (SLB) 和转换监视缓冲器 (TLB)。SLB 将 EA 转换为 VA，TLB 将从 SLB 出来的 VA 转换为 RA。EA 由应用使用并且通常是 32 比特或 64 比特的地址。不同的应用或一个应用的多个副本可以使用相同的 EA 来引用不同的存储位置 (例如，均使用相同 EA 的一个应用的两个副本，需要两个不同的物理存储位置)。为了完成这一点，EA 首先被转换成一个更大的 VA 空间，其对在操作系统下运行的所有应用来说是公共的。EA 到 VA 的转换由 SLB 执行。然后使用 TLB 来将 VA 转换成 RA，该 TLB 是包含 VA 到 RA 的映射的页表或映射表的高速缓存。该表由操作系统维护。

ATO 342、343 和 344 提供了维护与系统中其他处理单元的同步所必需的数据高速缓存的级别。原子直接存储器访问命令提供了使协处理器单元执行与其他单元的同步的手段。

BIU 339、340 和 341 的主要功能是为 SPE 302、303 和 304 提供到 EIB 的接口。EIB 319 在 Cell BE 芯片 300 上的所有处理器内核和附在 EIB 319 上的外部接口控制器之间提供通信路径。

MIC 320 在 EIB 319 与 XIO 321 和 322 中的一个或两个之间提供接口。极速数据速率 (XDR™) 动态随机存取存储器 (DRAM) 是由 Rambus® 提供的高速高度串行存储器。由 Rambus 提供的宏访问极速数据速率动态随机存取存储器，该存储器在本文中称为 XIO 321 和 322。

MIC 320 只是 EIB 319 上的从设备。MIC 320 对在它的配置地址范围内的命令进行确认，该范围与所支持的中心中的存储器对应。

BIC 324 和 325 管理片上或片外的从 EIB 319 到两个外部设备中的任意一个的数据传送。BIC 324 和 325 可以和 I/O 设备交换非一致性业务，或者它可以将 EIB 319 扩展到另一个设备，该设备甚至可以是另一个 Cell BE 芯片。当用于扩展 EIB 319 时，总线协议维护 Cell BE 芯片 300 中的高速缓存和附带的外部设备中的高速缓存之间的一致性，该外部设备可以是另一个 Cell BE 芯片。

IOC 326 处理在 I/O 接口设备中发起并发往相关 EIB 319 的命令。I/O 接口设备可以是附到 I/O 接口上的任意设备，诸如附带多个 I/O 设备或用非一致性的方式访问的另一个 Cell BE 芯片 300 的 I/O 桥芯片。IOC 326 还截取 EIB 319 上针对存储映射寄存器的访问，并将它们路由到正确的 I/O 接口，这些寄存器驻留在 I/O 桥芯片或非一致性 Cell BE 芯片 300 之中或之后。IOC 326 还包括内部中断控制器 (IIC) 349 和 I/O 地址转换单元 (I/O Trans) 350。

普适逻辑 (pervasive logic) 351 是为 Cell BE 芯片 300 提供时钟管理、测试特征和上电序列的控制器。普适逻辑可以为处理器提供热管理系统。普适逻辑包含通过本领域公知的联合测试行动小组 (JTAG) 或 SPI (串行外围接口) 接口到系统中其他设备的连接。

尽管已经提供了如何实现不同组件的特定示例，但这并不意味着对可以使用说明性实施例各方面的体系结构构成限制。可以结合任意多内核处理器系统而使用说明性实施例的各方面。

在应用或软件的执行期间，Cell BE 芯片内的区域的温度可能上升。如果不加抑制，温度可能上升到最大指定结温度 (junction temperature) 之上，导致不正确的运行或物理损害。为了避免这些情形，Cell BE 芯片的数字热管理单元在运行期间监控并试图控制 Cell BE 芯片内的温度。数字热管理单元包括此处描述的一个热管理控制单元 (TMCU) 和十个分布式数字热传感器 (DTS)。

一个传感器位于八个 SPE 中的一个 SPE 中，一个传感器位于 PPE 中，并且一个传感器与线性热二极管相邻。线性热二极管是计算温度的片上二极管。这些传感器位于与关联的单元内的各区域相邻的位置，该

关联的单元典型地在执行大多数应用期间经历温度的最大上升。热控制单元监控来自每个这些传感器的反馈。如果传感器的温度上升到可编程的点之上，热控制单元就可以配置为引起对 PPE 或一个或多个 SPE 的中断并且动态地调节关联的 PPE 或 SPE 的执行。

将 PPE 或 SPE 停止和运行可编程数量的周期提供了必需的调节。中断使特权软件能够采取正确的措施，同时动态调节在没有软件介入的情况下试图将 broadband engine（宽带引擎）芯片内的温度保持在可编程级别以下。特权软件将调节级别设置为等于或低于应用所提供的推荐设置。每个应用可能会不同。

如果调节 PPE 或 SPE 没有有效地管理温度并且温度继续上升，则普适逻辑 351 在温度达到热过载温度（由可编程配置数据限定）时停止 Cell BE 芯片的时钟。热过载特征保护 Cell BE 芯片不受物理损害。从这种情形恢复需要硬重启（hard reset）。由 DTS 监控的区域的温度并不一定是关联的 PPE 或 SPE 内的最热点。

图 4 示出了根据说明性实施例的示例性热管理系统。热管理系统可以实现为集成电路，诸如图 3 的普适逻辑单元 351 所提供的集成电路。热管理系统可以是专用集成电路、处理器、多处理器或异质多内核处理器。热管理系统被分在十个分布式 DTS 和热管理控制单元 (TMCU) 402 之间，为简单起见只示出了 DTS 404、406、408 和 410。每个在 SPU 传感器 440 中的 DTS 404 和 406、在 PPU 传感器 442 中的 DTS 408、以及在与线性热二极管（没有示出）相邻的传感器 444 中的 DTS 410 提供当前温度检测信号。该信号表明温度等于或小于由 TMCU 402 设置的当前温度检测范围。TMCU 402 使用来自 DTS 404、406、408 和 410 的信号的状态来连续地跟踪每个 PPE 或 SPE 的 DTS 404、406、408 或 410 的温度。随着对温度的跟踪，TMCU 402 将当前温度提供为表示关联的 PPE 或 SPE 内的温度的数值。校准单独的传感器的制造厂设置内部校准存储器 428。

除了上述 TMCU 402 的单元之外，TMCU 402 还包括复用器 446 和 450、工作寄存器 448、比较器 452 和 454、串行器 456、热管理控制状

态机 458 和数据流 (DF) 单元 460。复用器 446 和 450 对各种发出的和进入的信号进行组合以便在单个媒介上传送。工作寄存器 448 保存在 TMCU 402 中执行的乘法的结果。比较器 452 和 454 提供了对两个输入的比较功能。比较器 452 是大于或等于比较器。比较器 454 是大于比较器。串行器 456 将来自源的低速并行数据转换为用于传送的高速串行数据。串行器 456 结合 SPU 传感器 440 上的解串器 462 和 464 而工作。解串器 462 和 464 将接收的高速串行数据转换为低速并行数据。热管理控制状态机 458 启动 TMCU 402 的内部初始化。DF 单元 460 控制去往和来自热管理控制状态机 458 的数据。

TMCU 402 可以配置为使用中断逻辑 416 来引发对 PPE 的中断以使用调节逻辑 418 来动态地调节 PPE 或 SPE 的执行。

TMCU 402 将表示温度的数值与可编程的中断温度和可编程的调节点进行比较。每个 DTS 具有一个独立的可编程的中断温度。如果该温度在已编程的中断温度范围之内,那么如果已启用,则 TMCU 402 产生对 PPE 的中断。如果温度在依赖于下述方向比特的已编程级别以上或以下,则产生中断。另外,第二可编程的中断温度可以引发对系统控制器的注意信号。系统控制器在系统面板上并且连接到在 SPI 端口上的 Cell BE。

如果由与 PPE 或 SPE 关联的 DTS 感应的温度等于或高于调节点,则 TMCU 402 通过独立地开始和停止 PPE 或一个或多个 SPE 来调节该 PPE 或 SPE 的执行。软件可以使用诸如热管理停止时间寄存器和热管理比例寄存器之类的热管理寄存器来控制调节的比率和频率。

图 5 描述了根据说明性实施例的温度曲线图以及可能发生中断和动态调节的各个点。在图 5 中,线 500 可以表示 PPE 或 SPE 的温度。如果 PPE 或 SPE 正常运行,那么在用“N”标记的区域中不进行调节。当 PPE 或 SPE 的温度达到调节点时,TMCU 开始调节关联的 PPE 或 SPE 的执行。调节发生的区域用“T”标记。当 PPE 或 SPE 的温度下降到结束调节点以下时,执行返回到正常操作。

如果由于任意原因温度继续上升并达到在全面调节点处或之上的

温度时，TMCU 402 停止 PPE 或 SPE 直到温度下降到全面调节点以下。停止 PPE 或 SPE 的区域用“S”标记。在温度在全面调节点处或之上时停止 PPE 或 SPE 称为内核停止安全性。

在该示例性图示中，将中断温度设置在调节点以上；因此，TMCU 402 产生中断，该中断是对软件的通知，即关于因为温度曾经在或仍在内核停止温度以上所以相应的 PPE 或 SPE 被停止的通知；假设热中断屏蔽寄存器（TM_ISR）设置为活动的，参见图 4 中的 422，使 PPE 或 SPE 在未决中断（pending interrupt）期间能够继续。如果动态调节被禁用，则特权软件管理热状况。不管理热状况可能导致关联的 PPE 或 SPE 的不正确运行或由热过载功能引起的热关闭。

返回到图 4，热传感器状态寄存器包括热传感器当前温度状态寄存器 412 和热传感器最大温度状态寄存器 414。这些寄存器使软件能够读取每个 DTS 的当前温度，确定在一段时间期间达到的最高温度，并在温度达到可编程的温度时引发中断。热传感器状态寄存器具有关联的可以标记为享有管理特权的真实地址页。

热传感器当前温度状态寄存器 412 包含每个 DTS 的当前温度的编码值或数字值。由于传感器温度检测中的等待时间、读取这些寄存器的等待时间和正常的温度波动，在这些寄存器中报告的温度是较早时间点的温度，可能不能反映软件接收数据时的实际温度。由于每个传感器都有专用的控制逻辑，因此 DTS 404、408 和 410 内的控制逻辑并行地对所有的传感器进行采样。TMCU 402 在采样周期结束时更新热传感器当前温度状态寄存器 412 的内容。TMCU 402 将热传感器当前温度状态寄存器 412 中的值改变为当前温度。TMCU 402 每个 SenSampTime 周期都轮询新的当前温度。SenSampTime 配置字段控制采样周期的长度。

热传感器最大温度状态寄存器 414 包含从热传感器最大温度状态寄存器 414 最后被读取的时间开始，每个传感器达到的数字地编码的最大温度。通过软件或诸如片外设备 472 或片外 I/O 设备 474 之类的任意片外设备读取这些寄存器使 TMCU 402 将每个传感器的当前温度复制进寄存器。在读取之后，TMCU 402 从该点开始继续跟踪最大温度。每个

寄存器的读取是独立的。对一个寄存器的读取不影响另一寄存器的内容。

每个传感器都具有专用的控制逻辑，因此 DTS 404、406、408 和 410 内的控制逻辑并行地对所有的传感器进行采样。TMCU 402 将热传感器最大温度状态寄存器 414 中的值改变为当前温度。TMCU 402 每个 SenSampTime 周期都轮询新的当前温度。SenSampTime 配置字段控制采样周期的长度。

中断逻辑 416 中的热传感器中断寄存器控制对 PPE 的热管理中断的产生。这组寄存器包括热传感器中断温度寄存器 420 (TS_ITR1 和 TS_ITR2)、热传感器中断状态寄存器 422 (TS_ISR)、热传感器中断屏蔽寄存器 424 (TS_IMR) 和热传感器全局中断温度寄存器 426 (TS_GITR)。热传感器中断温度寄存器 420 和热传感器全局中断温度寄存器 426 包含引起对 PPE 的热管理中断的温度的编码。

当热传感器当前温度状态寄存器 412 中的针对传感器的用数字格式编码的温度大于或等于热传感器中断温度寄存器 420 中的相应传感器的中断温度编码时，TMCU 402 设置热传感器中断状态寄存器 422 中的相应状态比特 (TS_ISR[Sx])。当热传感器当前温度状态寄存器 412 中的针对任意传感器的温度编码大于或等于热传感器全局中断温度寄存器 426 中的全局中断温度编码时，TMCU 402 设置热传感器中断状态寄存器 422 中的相应状态比特 (TS_ISR[Gx])。

如果设置了任意热传感器中断状态寄存器 422 比特 (TS_ISR[Sx]) 并且还设置了热传感器中断屏蔽寄存器 424 中的相应屏蔽比特 (TS_IMR[Mx])，那么 TMCU 402 引发对 PPE 的热管理中断信号。如果设置了任意热传感器中断状态寄存器 422 比特 (TS_ISR[Gx]) 并且还设置了热传感器中断屏蔽寄存器 424 中的相应屏蔽比特 (TS_IMR[Cx])，那么 TMCU 402 引发对 PPE 的热管理中断信号。

为了清除中断条件，特权软件应当将热传感器中断屏蔽寄存器中的任意相应的屏蔽比特设置为“0”。为了启用热管理中断，特权软件保证温度在相应传感器的中断温度以下，然后执行以下序列。在温度不在中

断温度以下时启用中断可能导致产生立即热管理中断。

1. 将“1”写到热传感器中断状态寄存器 422 中的相应状态比特。
2. 将“1”写到热传感器中断屏蔽寄存器 424 中的相应屏蔽比特。

热传感器中断温度寄存器 420 包含位于 SPE、PPE 中且与线性热二极管相邻的传感器的中断温度级别。TMCU 402 将该寄存器中已编码的中断温度级别与热传感器当前温度状态寄存器 412 中的相应中断温度编码进行比较。这些比较的结果产生热管理中断。每个传感器的中断温度级别是独立的。

除了在热传感器中断温度寄存器 420 中设置的独立的中断温度级别之外，热传感器全局中断温度寄存器 426 包含第二中断温度级别。该级别适用于 Cell BE 芯片中的所有传感器。TMCU 402 将该寄存器中的已编码的全局中断温度级别与每个传感器的当前温度编码进行比较。这些比较的结果产生热管理中断。

全局中断温度的目的是提供对 Cell BE 芯片中的温度上升的早期指示。特权软件和系统控制器可以使用该信息来启动措施以控制温度，例如，增加扇入速度、在单元之间重新平衡应用软件等等。

热传感器中断状态寄存器 422 标识哪些寄存器满足中断条件。中断条件是指每个热传感器中断状态寄存器 422 比特所具有的特定条件，当满足该特定条件时中断可能发生。如果设置了相应的屏蔽比特，那么实际中断只提交给 PPE。

热传感器中断状态寄存器 422 包含三组状态比特，即数字传感器全局门限中断状态比特 (TS_ISR[Gx])、数字传感器门限中断状态比特 (TS_ISR[Sx]) 和数字传感器全局门限以下中断状态比特 (TS_ISR[Gb])。

当热传感器当前温度状态寄存器 412 中的传感器温度编码大于或等于热传感器中断温度寄存器 420 中的相应的传感器的中断温度编码并且热传感器中断屏蔽寄存器 424 中的相应方向比特 TM_IMR[Bx]='0'时，TMCU 402 设置热传感器中断状态寄存器 422 中的状态比特 (TS_ISR[Sx])。另外，当热传感器当前温度状态寄存器 412 中的传感

器温度编码低于热传感器中断温度寄存器 420 中的相应的传感器的中断温度编码并且热传感器中断屏蔽寄存器 424 中的相应方向比特 $TM_IMR[Bx]=1$ 时, TMCU 402 设置热传感器中断状态寄存器 422, 即 $TS_ISR[Sx]$ 。

当任意参与的传感器的当前温度大于或等于热传感器全局中断温度寄存器 426 的当前温度并且热传感器中断屏蔽寄存器 424 $TM_IMR[B_G]=0$ 时, TMCU 402 设置热传感器中断状态寄存器 422, 即 $TS_ISR[Gx]$ 。单独的热传感器中断状态寄存器 422 的 $TS_ISR[Gx]$ 比特表明哪些单独的传感器满足这些条件。

当热传感器中断屏蔽寄存器 424 $TM_IMR[Cx]$ 中的所有参与的传感器的当前温度低于热传感器全局中断温度寄存器 426 的当前温度并且热传感器中断屏蔽寄存器 424 $TM_IMR[B_G]=1$ 时, TMCU 402 设置热传感器中断状态寄存器 422, 即 $TS_ISR[Gb]$ 。由于所有参与的传感器的当前温度低于热传感器全局中断温度寄存器 426 的当前温度, 因此对于全局门限以下中断条件, 只出现热传感器中断状态寄存器 422 中的一个状态比特 ($TS_ISR[Gb]$)。

一旦将热传感器中断状态寄存器 422 中的一个状态比特 ($TS_ISR[Sx]$ 、 $TS_ISR[Gx]$ 或 $TS_ISR[Gb]$) 设置为 '1', TMCU 402 就维护该状态直到由特权软件重置为 '0'。特权软件通过将 '1' 写到热传感器中断状态寄存器 422 中的相应比特而将状态比特重置为 '0'。

热传感器中断屏蔽寄存器 424 包含单独传感器的两个字段和全局中断条件的多个字段。中断条件是指每个热传感器中断状态寄存器 422 比特所具有的特定条件, 当满足该特定条件时中断可能发生。如果设置了相应的屏蔽比特, 那么实际中断只提交给 PPE。

单独传感器的两个热传感器中断屏蔽寄存器的数字热门限中断字段是 $TS_IMR[Mx]$ 和 $TS_IMR[Bx]$ 。热传感器中断屏蔽寄存器 424 的屏蔽比特 $TS_IMR[Mx]$ 防止中断状态比特产生对 PPE 的热管理中断。热传感器中断屏蔽寄存器 424 的方向比特 $TS_IMR[Bx]$ 将中断条件的温度方向设置为高于或低于热传感器中断温度寄存器 420 中的相应温度。将热

传感器中断屏蔽寄存器 424 的 TS_IMR[Bx] 设置为‘1’将中断条件的温度设置为低于热传感器中断温度寄存器 420 中的相应温度。将热传感器中断屏蔽寄存器 424 的 TS_IMR[Bx] 设置为‘0’将中断条件的温度设置为等于或高于热传感器中断温度寄存器 420 中的相应温度。

针对全局中断条件的热传感器中断屏蔽寄存器 424 字段是 TS_IMR[Cx]、TS_IMR[B_G]、TS_IMR[C_{gb}] 和 TS_IMR[A]。热传感器中断屏蔽寄存器 424 的屏蔽比特 TS_IMR[Cx] 防止全局门限中断并且选择哪些传感器参与全局门限以下中断条件。热传感器中断屏蔽寄存器 424 的方向比特 TS_IMR[B_G] 选择针对全局中断条件的温度方向。热传感器中断屏蔽寄存器 424 的屏蔽比特 TS_IMR[C_{gb}] 防止全局门限以下中断。热传感器中断屏蔽寄存器 424 TS_IMR[A] 引发对系统控制器的注意信号。注意信号是一种对系统控制器的信号，表明普适逻辑需要注意或具有针对系统控制器的状态。可以将注意信号映射到系统控制器中的中断。系统控制器在系统面板 (planer) 上并且连接到在 SPI 端口上的 Cell Broadband Engine。

将热传感器中断屏蔽寄存器 424 的 TS_IMR[B_G] 设置为‘1’，就将针对全局中断条件的温度范围设置为当在热传感器中断屏蔽寄存器 424 的 TS_IMR[Cx] 中设置的所有参与传感器的温度都低于全局中断温度级别时发生。将热传感器中断屏蔽寄存器 424 的 TS_IMR[B_G] 设置为‘0’，就将针对全局中断条件的温度范围设置为当任意参与传感器的温度大于或等于热传感器全局中断温度寄存器 426 中的相应温度时发生。如果热传感器中断屏蔽寄存器 424 的 TS_IMR[A] 设置为‘1’，那么当任意热传感器中断屏蔽寄存器 424 TS_IMR[Cx] 比特和它相应的热传感器中断状态寄存器 422 状态比特 (TS_ISR[Gx]) 都设置为‘1’时 TMCU 402 引发注意信号。另外，当热传感器中断屏蔽寄存器 424 的 TS_IMR[C_{gb}] 和热传感器中断状态寄存器 422 的 TS_ISR[G_b] 都设置为‘1’时，TMCU 402 引发注意信号。

当任意热传感器中断屏蔽寄存器 424 TS_IMR[Mx] 比特和它相应的热传感器中断状态寄存器 422 状态比特 (TS_ISR[Sx]) 都设置为‘1’时，

TMCU 402 将热管理中断提交给 PPE。当任意热传感器中断屏蔽寄存器 424 TS_IMR[Cx]比特和它相应的热传感器中断状态寄存器 422 状态比特 (TS_ISR[Gx]) 都设置为‘1’时, TMCU 402 产生热管理中断。另外, 当热传感器中断屏蔽寄存器 424 的 TS_IMR[Cgb]和热传感器中断状态寄存器 422 的 TS_ISR[Gb]都设置为‘1’时, TMCU 402 将热管理中断提交给 PPE。

调节逻辑 418 中的动态热管理寄存器包含用于控制 PPE 或 SPE 的执行调节的参数。动态热管理寄存器是一组寄存器, 包括热管理控制寄存器 430(TM_CR1 和 TM_CR2)、热管理调节点寄存器 432(TM_TPR)、热管理停止时间寄存器 434(TM_STR1 和 TM_STR2)、热管理调节比例寄存器 436(TM_TSR) 和热管理系统中断屏蔽寄存器 438(TM_SIMR)。

热管理调节点寄存器 432 设置传感器的调节温度点。可以在热管理调节点寄存器 432 中设置两个独立的调节温度点, 即 ThrottlePPE 和 ThrottleSPE, 一个用于 PPE 并且另一个用于 SPE。该寄存器中还包含用于禁用调节和停止 PPE 或 SPE 的温度点。PPE 或 SPE 的执行调节在温度等于或高于调节点时开始。调节在温度下降到用以禁用调节的温度 (TM_TPR[EndThrottlePPE/EndThrottleSPE]) 以下时停止。如果温度达到全面调节温度或停止温度 (TM_TPR[FullThrottlePPE/FullThrottleSPE]), 则 TMCU 402 停止 PPE 或 SPE 的执行。热管理控制寄存器 430 控制调节行为。

热管理停止时间寄存器 434 和热管理调节比例寄存器 436 控制调节频率和调节量。当温度达到调节点时, TMCU 402 将相应的 PPE 或 SPE 停止一定时钟数, 该时钟数由热管理停止时间寄存器 434 中的相应值中的停止时间乘以热管理调节比例寄存器 436 中的相应比例值来指定。然后 TMCU 402 使 PPE 或 SPE 能够运行一定的时钟数, 该时钟数由运行时间乘以相应的比例值来指定, 其中运行时间是依赖于实现方式的固定时间量减去停止时间之间的差值。热管理调节比例寄存器 436 中的可编程的比例值是停止时间和运行时间的乘数。一个示例可以是 (Stop×Scale)

$/(Run \times Scale) ((停止时间 \times 比例) / (运行时间 \times 比例))$ 。内核停止的时间百分比保持相同，但是周期增大或频率减小。该序列继续直到温度下降到禁用调节 (TM_TPR[EndThrottlePPE/EndThrottleSPE]) 以下。

热管理系统中断屏蔽寄存器 438 选择哪个 PPE 中断将使 TMCU 402 禁用调节。当这些中断仍然未决并且屏蔽仍然选择未决中断时 TMCU 402 将继续阻止调节。如果取消选定屏蔽或中断不再是未决的，则 TMCU 402 将不再阻止中断。

热管理控制寄存器 430 独立地为每个 PPE 或 SPE 设置调节模式。在两个寄存器之间拆分控制比特。下面是可以独立地为每个 PPE 或 SPE 设置的五个不同的模式：

- 禁用动态调节 (包括内核停止安全性)；
- 正常操作 (启用动态调节和内核停止安全性)；
- 始终调节 PPE 或 SPE (启用内核停止安全性)；
- 禁用内核停止安全性 (启用动态调节并且禁用内核停止安全性)；
- 始终调节 PPE 或 SPE 并且禁用内核停止安全性。

特权软件应当针对运行应用或操作系统的 PPE 或 SPE 而将控制比例设置为正常操作。如果 PPE 或 SPE 没有运行应用代码，则特权软件应当将控制比特设置为禁用。“始终调节 PPE 或 SPE”模式拟用于应用开发。这些模式对确定应用是否能够在极限调节条件下运行是有用的。应当只在特权软件主动地管理热事件时才使 PPE 或 SPE 能够在禁用动态调节或内核停止安全性的情况下执行。

热管理系统中断屏蔽寄存器 438 控制哪个 PPE 中断使热管理逻辑暂时停止调节 PPE。TMCU 402 在中断未决的同时暂时挂起对这两个线程的调节，而不管中断所指向的线程。当中断不再未决时，只要调节条件仍然存在调节就可以继续。从不基于系统中断条件而禁用对 SPE 的调节。可以优先于调节条件的 PPE 中断条件如下：

外部

减量器

管理程序 (Hypervisor) 减量器

系统错误

热管理

热管理调节点寄存器 432 包含 PPE 或 SPE 的执行调节开始和结束时的已编码温度点。该寄存器还包含 PPE 或 SPE 的执行被全面调节时的已编码温度点。

软件使用热管理调节点寄存器中的值来设置用于在三个热管理状态之间改变的三个温度点，这三个状态是：正常运行 (N)、调节 PPE 或 SPE (T) 以及停止 PPE 或 SPE (S)。TMCU 402 支持针对 PPE 和 SPE 的独立温度点。

当热传感器当前温度状态寄存器 412 中的已编码的传感器当前温度等于或大于调节温度 (ThrottlePPE/ThrottleSPE) 时，如果已经启用，那么相应的 PPE 或 SPE 的执行调节就会开始。执行调节继续到相应的传感器的已编码当前温度小于结束调节 (EndThrottlePPE/EndThrottleSPE) 的已编码温度为止。作为一种安全措施，如果已编码的当前温度等于或大于全面调节点 (FullThrottlePPE/FullThrottleSPE)，那么 TMCU 402 停止相应的 PPE 或 SPE。

热管理停止时间寄存器 434 控制在热管理调节状态下应用于特定 PPE 或 SPE 的调节量。热管理停止时间寄存器 434 中由软件设置的值表示内核将停止的时间量相对于允许内核运行的时间量的比值 (stop/run) 或者内核停止的时间百分比。热管理调节比例寄存器 436 控制 PPE 或 SPE 停止和运行的实际时钟数 (NCIks)。

热管理调节比例寄存器 436 控制 PPE 或 SPE 在热管理调节状态期间停止和运行的实际周期数。该寄存器中的值是配置环设置 TM_config[MinStopSPE] 的倍数。下面的等式计算实际的停止和运行周期数：

SPE 运行和停止时间：

$$\begin{aligned} \text{SPE_StopTime} &= (\text{TM_STR1}[\text{StopCore}(x)] * \\ &\quad \text{TM_Config}[\text{MinStopSPE}]) * \text{TM_TSR}[\text{ScaleSPE}] \\ \text{SPE_RunTime} &= (32 - \text{TM_STR1}[\text{StopCore}(x)]) * \\ &\quad \text{TM_Config}[\text{MinStopSPE}] * \text{TM_TSR}[\text{ScaleSPE}] \end{aligned}$$

Power PC® 单元运行和停止时间:

$$\begin{aligned} \text{PPE_StopTime} &= (\text{TM_STR2}[\text{StopCore}(8)] * \\ &\quad \text{TM_Config}[\text{MinStopPPE}]) * \text{TM_TSR}[\text{ScalePPE}] \\ \text{PPE_RunTime} &= (32 - \text{TM_STR2}[\text{StopCore}(8)]) * \\ &\quad \text{TM_Config}[\text{MinStopPPE}] * \text{TM_TSR}[\text{ScalePPE}] \end{aligned}$$

运行和停止时间可以通过中断和对各种热管理寄存器进行写入的特权软件来改变。

片上性能监控器 466 可以提供可以跟踪由诸如 DTS 404、406、408 和 410 之类的温度感应设备提供的热数据的性能监控。可以将热数据存储在存储器 470 中或者写到诸如图 2 的主存储器 208 之类的片外设备 472 或写到诸如图 2 的南桥和输入/输出 (I/O) 控制器中心 (ICH) 204 之类的片外 I/O 设备 474。位于性能监控器 466 中的控制器 468 控制确定将热数据发送到哪里。

尽管以下描述针对一个指令流和一个处理器，但是该指令流可以是一组指令流并且该处理器可以是一组处理器。也就是说，一组可以是单个指令流和单个处理器或者两个或更多指令流和处理器。

利用上述体系结构，针对 Cell BE 芯片的热管理和热调节进行了很多改进并添加了可编程性。这些改进和所添加的可编程性中的一些使得可以实现关键特征而另一些增强了可用性。

图 6 描述了根据说明性实施例的用于记录最大温度的操作的流程图。随着操作开始，包含诸如图 3 的 Cell BE 芯片 300 之类的 Cell Be 芯片的计算机系统启动或重启 (步骤 602)。如以前所述，Cell BE 芯片包括通过图 3 的普适逻辑单元 351 提供的热管理系统。对于诸如图 4 的 DTS 404、406、408 和 410 之类的每个 DTS，该热管理系统包括一组最大温度状态寄存器和一组当前温度状态寄存器，诸如图 4 的最大温度状态寄存器 414 和当前温度状态寄存器 412。当前温度状态寄存器存储它的目标 DTS 自诸如图 4 的热管理控制状态机 458 之类的热管理控制状态机最后一次感应 DTS 之后的当前温度。最大温度状态寄存器存储它的目标 DTS 自计算机系统最后一次读取最大温度状态寄存器或计算机系统重启之后的最大温度。可以使用任意数量的诸如处理器、集成电路

之类的设备或通过使用串行外围接口 (SPI) 端口或联合测试行动小组 (JTAG) 端口的设备来读取最大温度状态寄存器。但是, 通过 JTAG 端口读取寄存器不会导致重启。

说明性地将以下讨论限于一个 DTS, 计算机系统启动或重启 (步骤 602) 之后的最大温度是零。一旦热管理控制状态机感应到 DTS 的温度, 该热管理控制状态机就将感应的 DTS 的温度发送到诸如图 4 的比较器 454 之类的比较器 (步骤 604)。该比较器将感应温度与最大温度状态寄存器中存储的针对该 DTS 的当前最大温度进行比较 (步骤 606)。如果在步骤 606 感应温度高于最大温度状态寄存器中存储的当前最大温度, 那么感应温度变成新的最大温度并且热管理控制状态机将新的最大温度记录到最大温度状态寄存器中 (步骤 608)。也就是说, 热管理控制状态机覆盖或替换最大温度状态寄存器中存储的当前最大温度。如果在步骤 606 感应温度低于或等于最大温度状态寄存器中存储的当前最大温度, 那么最大温度状态寄存器保持最大温度状态寄存器中现有的当前最大温度 (步骤 610)。

最大温度状态寄存器中的当前最大温度停留在最大温度直到计算机系统以读取请求的形式读取最大温度状态寄存器 (步骤 612) 或计算机系统重启为止。如果没有读取当前最大温度, 那么操作返回到步骤 604。如果在步骤 612 计算机系统读取当前最大温度, 那么热管理控制状态机将当前最大温度重置为当前温度状态寄存器中的当前温度 (步骤 614), 然后操作返回到步骤 604。

对于该操作的一个示例, 如果诸如处理器内核或处理器本身之类的特定单元的 DTS 在一段时间上要感应 67°C 、 70°C 、 75°C 、 72°C 和 74°C 的温度, 那么最大温度状态寄存器中的最大温度将是 75°C 。如果在对 DTS 的第四次感应之后, 计算机系统发出读取请求, 那么返回的最大温度将是 75°C 。但是, 此时热管理控制状态机将最大温度重置为当前温度, 并且在由 DTS 执行的最后一次感应之后, 最大温度状态寄存器中的最大温度将是 74°C 。

这样, 最大温度状态寄存器的目的是记录 DTS 自最大温度寄存器

最后一次被读取之后达到的最大温度。该最大温度信息帮助操作系统在不用连续轮询当前温度寄存器的情况下确定 DTS 在应用或程序执行期间达到的最大温度。连续轮询将影响系统的性能，因此可能影响最大温度。另外，轮询当前温度不能保证读取到最大温度。如果最大温度发生在对当前温度的多次读取之间，就属于这种情况。

图 7 描述了根据另一个说明性实施例的用于通过性能监控来跟踪热数据的操作的流程图。如前所述，Cell BE 芯片包括通过图 3 的普适逻辑单元 351 提供的热管理系统。可以通过诸如图 4 的性能监控器 466 之类的性能监控器提供性能监控。性能监控可以跟踪由诸如图 4 的 DTS 404、406、408 和 410 之类的温度感应设备提供的在诸如图 4 的存储器 470 之类的其内部存储器中的热数据，写到诸如图 2 的主存储器 208 之类的主存储器或图 4 的片外设备 472，或写到诸如图 2 的南桥和输入/输出 (I/O) 控制器中心 (ICH) 204 或图 4 的片外 I/O 设备 474 之类的 I/O 设备。

性能监控支持两种主要的跟踪模式：跟踪固定时间段和持续跟踪。对热性能的跟踪可以是诸如图 5 的跟踪 500 之类的跟踪。性能监控还可以规定采样频率的配置以控制两个连续的采样之间的时间段。此外，可以使用热信息压缩来增大采样间隔。一种压缩技术是只在发生变化时存储热信息。对相同的热采样的数量的计数还可以与热信息一起存储。因为热信息典型地变化缓慢，所以这是一种有用的技术。

随着用于通过性能监控器来跟踪热数据的操作开始，诸如图 4 的热管理控制状态机 458 之类的热管理控制状态机将性能监控器设置为跟踪模式 (步骤 702)。说明性地将以下讨论限于一个 DTS，热管理控制状态机感应 DTS 的温度 (步骤 704) 并将感应到的 DTS 的温度发送到当前温度状态寄存器和/或其他数据结构以进行存储 (步骤 706)。此时热管理控制状态机确定性能监控器是否还在运行 (步骤 708)。一旦性能监控器在步骤 702 中启动，则该性能监控器将运行用户指定的时间段或运行到由用户通过用户输入而停止。但是，性能监控器还可以基于特定的热状况而停止。该特定的热状况称为触发器，诸如在一组信号上寻找特

定条件的逻辑分析器。触发器的使用在软件调试中很有用。例如，用户可以将性能监控器设置为在达到热状况时停止或检停 (checkstop) 系统。这可以使用户能够准确地确定哪条代码或代码组合在引发热状况。如果性能监控器在步骤 708 仍然在运行，则操作返回到步骤 704。

返回到步骤 708，如果性能监控器不再运行，则热管理控制状态机读取存储在存储器中的温度信息并以图形形式为用户显示所存储的信息 (步骤 710)，之后操作结束。在步骤 706 发送到当前温度状态寄存器和其他数据结构的感应温度还可以在操作仍然在箭头 712 所表明的处理 (步骤 710) 中时同时显示，而不是等待跟踪结束。

这样，性能监控器跟踪由 DTS 提供的热数据。自动跟踪热数据消除了对软件持续轮询当前温度寄存器的需要。性能监控对于收集工作负载的热数据很重要，原因是性能监控不需要插入附加代码来轮询热数据，这种插入可能会改变工作负载的行为。换句话说，性能监控提供非侵入的方法来实时跟踪软件应用的热特征数据。将热信息发送到性能监控器的另外的好处是能够触发或停止对预先指定的热状况上的热信息的记录。另外，性能监控器还可以用于在满足热状况时停止系统 (或检停)。这样做使用户能够确定哪个代码段或代码段组合正在产生热状况。然后用户可以重写代码段或避免特定的组合，从而避免了热事件。

图 8A 和图 8B 描述了根据另外的说明性实施例的针对高级热中断产生的操作的流程图。如前所述，Cell BE 芯片包括通过图 3 的普适逻辑单元 351 提供的热管理系统。高级热中断产生是帮助操作系统处理热事件的另一个特征。高级热中断逻辑是诸如图 4 的 TMCU 402 之类的热管理控制单元的一部分。当有热状况时 (也就是芯片温度上升到一定的门限以上)，热中断警告操作系统。在这种情况下，操作系统应当采取正确措施来降低芯片温度。正确措施可以由软件中断处理程序处理，软件中断处理程序是一条处理热状况并初始化正确措施的代码。然后操作系统等待热状况在继续正常操作之前消失。这通常需要操作系统等待特定的时间量，然后轮询处理器的温度以确定继续正常操作是否安全。使用高级热中断产生，操作系统可以设置中断以检测温度何时下降到一定

的门限以下，从而消除了对轮询当前温度寄存器的需要。图 4 的热传感器中断屏蔽寄存器 424 (TS_IMR) 和热传感器中断状态寄存器 422 (TS_ISR) 的组合使操作系统处理热事件更加容易。

高级热中断产生可以在局部级别或全局级别上执行。也就是说，高级热中断产生可以单独地（局部）在特定 DTS 上执行或在诸如图 4 的 DTS 404、406、408 和 410 之类的所有（全局）DTS 上执行。热传感器中断屏蔽寄存器的方向比特是 B_G 和 B_X 。中断方向定义了产生中断的条件。中断可以在温度从低于中断温度变到等于或高于中断温度时，或在温度从高于或等于中断温度变到低于中断温度时产生。热管理控制状态机用中断屏蔽寄存器中的方向比特 B_G 和 B_X 来标识条件。 B_G 是全局方向比特。当 B_G 设置为‘0’时，热管理控制状态机在任意 DTS 的温度大于或等于全局中断温度时产生中断。当 B_G 设置为‘1’时，热管理控制状态机在所有 DTS 的温度都低于全局中断温度时产生中断。 B_X 是局部方向比特，其中 X 是单独关联的 DTS 的数量。当 B_X 设置为‘0’时，热管理控制状态机在单独 DTS 的温度大于或等于 DTS 中断温度时产生中断。当 B_X 设置为‘1’时，热管理控制状态机在单独 DTS 的温度低于 DTS 中断温度时产生中断。热中断状态寄存器 (TS_ISR) 记录哪个传感器引起了高级热中断。软件读取该寄存器以确定发生了哪种状况以及哪个传感器或哪些传感器引起了中断。一旦被软件读取，热管理控制状态机就重置热中断状态寄存器中的状态比特。

因此，针对高级热中断产生的操作可以从全局和局部角度示出。图 8A 描述了全局高级热中断产生，图 8B 描述了局部高级热中断产生。随着操作在图 8A 的全局高级热中断产生中开始，热管理控制状态机将全局中断温度 T 设置为温度 T_1 并将全局中断方向 B_G 设置为‘0’（步骤 802）。热管理控制状态机感应 DTS 的温度（步骤 804）。热管理控制状态机确定是否有任何从 DTS 感应的温度大于或等于温度 T_1 （步骤 806）。如果没有感应温度大于或等于温度 T_1 ，那么操作返回到步骤 804。如果在步骤 806 任意一个感应温度大于或等于温度 T_1 ，那么热管理控制状态机产生中断并设置热中断状态寄存器中的相应状态比特以记录哪个

传感器或哪些传感器引起了中断(步骤 808)。然后操作系统将为中断提供服务并且可以减缓处理器上的工作负载或将处理器的部分工作负载卸载到系统中的另一个处理器。

在产生中断之后,热管理控制状态机将全局中断温度 T 设置为温度 T₂ 并将全局中断方向 B_G 设置为‘1’(步骤 810)。温度 T₂ 应当设置为小于或等于温度 T₁。热管理控制状态机再次感应 DTS 的温度(步骤 812)。热管理控制状态机确定是否所有从 DTS 感应的温度都低于温度 T₂(步骤 814)。如果没有感应温度低于温度 T₂,那么操作返回到步骤 812。如果在步骤 814 所有感应温度都低于温度 T₂,那么热管理控制状态机产生中断并设置热中断状态寄存器中的相应状态比特以记录哪个传感器或哪些传感器引起了中断(步骤 816)。此时,操作系统继续正常操作现在是安全的。然后操作系统将为中断提供服务并且将系统恢复到正常操作。下一步,操作返回到步骤 802,其中全局中断温度 T 设置为温度 T₁ 并且全局中断方向 B_G 设置为‘0’。

该操作的一个示例是所有 DTS 都具有全局中断温度 80°C 和全局中断方向‘0’。一旦诸如处理器内核或处理器本身之类的关联的单元的任意 DTS 感应到大于或等于 80°C 的温度,热管理控制状态机就产生中断并设置热中断状态寄存器中的相应状态比特以记录哪个传感器或哪些传感器引起了中断。然后操作系统将为中断提供服务并且可以减缓处理器上的工作负载或将处理器的部分工作负载卸载到系统中的另一个处理器。同样,此时热管理控制状态机可以将全局中断温度重置为示例性的 77°C 并将全局中断方向设置为‘1’。工作负载将继续在慢模式下操作或保持不被处理器处理直到 DTS 感应到对于所有 DTS 都低于 77°C 的温度。一旦热管理控制状态机确定感应温度低于 77°C,该热管理控制状态机就产生另一个中断。热管理控制状态机将全局中断温度设置为 80°C,将全局中断方向设置为‘0’,然后操作系统继续对工作负载的正常操作。

转到图 8B,将说明性实施例限于一个 DTS,但该说明性实施例对于每个 DTS 都是相同的。随着针对局部高级热中断产生的操作开始,

热管理控制状态机将局部中断温度 T 设置为温度 T3 并将局部中断方向 B_x 设置为‘0’（步骤 852）。热管理控制状态机感应 DTS 的温度（步骤 854）。热管理控制状态机确定从 DTS 感应的温度是否大于或等于温度 T3（步骤 856）。如果感应温度并不大于或等于温度 T3，那么操作返回到步骤 854。如果感应温度大于或等于温度 T3，那么热管理控制状态机产生中断并设置热中断状态寄存器中的相应状态比特以记录哪个传感器或哪些传感器引起了中断（步骤 858）。然后操作系统将为中断提供服务并且可以减缓处理器上的工作负载或将处理器的部分工作负载卸载到处理器内其他单元或卸载到系统中的另一个处理器。

在热管理控制状态机产生中断之后，热管理控制状态机将局部中断温度 T 设置为温度 T4 并将全局中断方向 B_x 设置为‘1’（步骤 860）。温度 T4 应当设置为小于或等于温度 T3。热管理控制状态机再次感应 DTS 的温度（步骤 862）。热管理控制状态机确定从 DTS 感应的温度是否低于温度 T4（步骤 864）。如果感应温度不低于温度 T4，那么操作返回到步骤 862。如果感应温度低于温度 T4，那么热管理控制状态机产生中断并设置热中断状态寄存器中的相应状态比特以记录哪个传感器或哪些传感器引起了中断（步骤 866）。此时，操作系统继续正常操作现在是安全的。然后操作系统将为中断提供服务并且将系统恢复到正常操作。下一步，操作返回到步骤 852，其中热管理控制状态机将全局中断温度 T 设置为温度 T3 并且全局中断方向 B_x 设置为‘0’。

该操作的一个示例是给定的 DTS 具有局部中断温度 80°C 和局部中断方向‘0’。一旦关联的单元的 DTS 感应到大于或等于 80°C 的温度，热管理控制状态机就产生中断并设置热中断状态寄存器中的相应状态比特以记录哪个传感器或哪些传感器引起了中断。然后操作系统将为中断提供服务并且可以减缓处理器上的工作负载或将处理器的部分工作负载卸载到系统中的另一个处理器。同样，此时热管理控制状态机可以将局部中断温度重置为示例性的 77°C 并将局部中断方向设置为‘1’。工作负载将继续在慢模式下运行或保持在处理器单元之外直到 DTS 感应到低于 77°C 的温度。一旦热管理控制状态机确定感应温度低于 77°C，该

热管理控制状态机就产生另一个中断。热管理控制状态机将局部中断温度设置为 80°C，将局部中断方向设置为‘0’，然后操作系统继续对工作负载的正常操作。

这样，高级热中断产生使操作系统能够对中断产生进行编程以跟随温度变化的方向，并且消除了对中断处理程序的需要以在热中断的情况下持续轮询当前温度。

图 9 描述了根据另外的说明性实施例的用于在热管理系统中支持深度节能模式和部分良好的操作的流程图。如前所述，Cell BE 芯片包括通过图 3 的普适逻辑单元 351 提供的热管理系统。在图 3 的 Cell BE 芯片 300 中，存在多种节能模式。根据每种节能模式的实现方式，一些节能模式可以限制诸如图 4 的 DTS 404、406、408 和 410 之类的 DTS 的可访问性。例如，如果诸如图 3 的 SPU 310、311 和 312 之类的 SPU 处于时钟关闭的节能模式，也就是说诸如图 4 的解串器 462 之类的解串器被禁用，那么诸如图 4 的串行器 456 之类的串行器与诸如图 4 的 DTS 404 之类的 DTS 之间的路径不起作用。节能模式的另一个示例可以是电源关闭的情况。在这种情况下，实际的 DTS 可能被禁用。另一个示例是热管理控制状态机确定处理器内的传感器或单元在制造测试期间是否坏掉的情况。如果传感器或单元是多余的，制造商可以将该传感器或单元标记为不合格，从而产生将只有有限数量的单元或传感器起作用的部分良好的处理器。在任一情况下，诸如图 4 的热管理控制状态机 458 之类的热管理控制状态机需要监控这些电源模式的状态并屏蔽掉不起作用的 DTS 使其不能参与热管理任务（调节、中断等）。

返回到图 9，其描述了用于在热感应和热管理系统中支持深度节能模式和部分良好的操作的流程图。随着操作开始，热管理控制状态机使用来自各 DTS 的数据跟踪 DTS 的状态（步骤 902）。热管理控制状态机将这些数据存储到内部校准存储器中，诸如图 4 的内部校准存储器 428 中。如前所述，节能模式、不合格的 DTS 或通过诸如图 4 的数据流 460 之类的数据流与热管理控制状态机通信的 SPU 可以禁止特定 DTS 的操作。由制造工艺报告的部分良好状况的效果与节能模式类似，只不过部

分良好是永久条件并且应当永久地屏蔽掉 DTS。在 SPU 被标记为不合格的情况下，热管理控制状态机关闭整个 SPU，并禁用串行器。在 DTS 被标记为不合格的情况下，热管理控制状态机屏蔽掉该 DTS。热管理控制状态机确定 DTS 或 SPU 是不合格还是在起作用（步骤 904）。如果 DTS 或 SPU 不合格，则热管理控制状态机屏蔽掉 DTS（步骤 906），之后操作结束。

为了屏蔽掉处于电源管理状态的 DTS，热管理控制状态机将诸如图 4 的当前温度状态寄存器 412 之类的当前温度状态寄存器中的相关的当前温度状态寄存器重置为 0x0，0x0 是最低温度设置。另一种方法还可以是通过设置状态比特来分配相关的当前温度状态寄存器的编码，以表明 DTS 被屏蔽，这可以比只重置传感器读数更加精确。然后热管理控制状态机从当前温度状态寄存器停止去往和来自 DTS 的通信。停止通信是一个可选步骤，主要用于节能和不执行无用的开销工作。然后热管理控制状态机产生表明 DTS 现在被屏蔽并且不应当参与热管理任务的信号。最后，热管理控制状态机重置 DTS 的状态。当诸如处理器内核或处理器本身之类的与 DTS 相关的单元退出节能模式时，热管理控制状态机继续与 DTS 通信，继续对当前温度状态寄存器进行更新，并发送 DTS 可以参与热管理任务的信号。

返回步骤 904，如果 DTS 和 SPU 都起作用，则热管理控制状态机开始与 DTS 通信（步骤 908）。热管理控制状态机监控 SPU 的电源管理状态以确定 SPU 何时进入节能模式（步骤 910）。在 SPU 进入节能模式之前，操作返回到步骤 908。如果 SPU 进入节能模式并且 DTS 被禁用，那么热管理控制状态机用上面结合步骤 906 而讨论的方法屏蔽掉 DTS（步骤 912）。由于表明了 DTS 是禁用还是在起作用，热管理控制状态机继续监控 SPU 的电源管理状态（步骤 914）。在 SPU 退出节能模式之前，操作返回到步骤 912。当 SPU 退出节能模式并且 DTS 不再被禁用时，热管理控制状态机开始与 DTS 通信，继续对当前温度状态寄存器进行更新，并发送 DTS 可以参与热管理任务的信号（步骤 916），然后操作返回到步骤 908。

这样，对部分良好、不合格或处于节能模式的 DTS 的温度读数的屏蔽隔离了不工作的或禁用的 DTS 使其不能参与热管理任务。

图 10 描述了根据另外的说明性实施例的针对使热感知软件应用的实时测试能够与温度相独立的热调节控制特征的操作的流程图。如前所述，Cell BE 芯片包括通过图 3 的普适逻辑单元 351 提供的热管理系统。诸如图 4 的热管理控制寄存器 430 之类的热管理控制寄存器为各种热调节控制特征提供访问和配置。将热调节设计为通过在使用调节的热事件的情况下消减性能来降低温度。

诸如图 4 的热管理停止时间寄存器 434 之类的热管理停止时间寄存器，和诸如图 4 的热管理调节比例寄存器 436 之类的热管理调节比例寄存器一起设置调节量和调节行为。在实时系统中，需要保证实时期限。软件开发者和质量保证团队知道并测试最大调节量很重要，最大调节量是程序或代码段能够容忍并仍然保证实时系统的实时期限的热管理停止时间寄存器和热管理调节比例寄存器的最大设置。作为对调节硬件的实际温度以引发热事件并且因此触发调节条件的替代，热管理控制状态机提供了不管温度如何都始终提供调节的模式。热管理控制状态机在热管理控制寄存器中设置该模式，这将芯片设置为恒定调节状态。该特征帮助软件开发者进行测试并确保他们的代码满足实时标准。

随着操作开始，接收热管理停止时间寄存器和热管理调节比例寄存器的热控制设置（步骤 1002）。热管理控制状态机使用热管理停止时间寄存器和热管理调节比例寄存器的设置来确定如何执行调节。然后，热管理控制状态机设置测试模式并将热管理控制寄存器设置为始终调节设置（步骤 1004）。然后程序运行以进行实时确认，即软件或程序将在热管理停止时间寄存器和热管理调节比例寄存器的热控制设置下满足实时期限（1006）。测试模式可以是任意类型的调节模式，诸如始终调节或随机调节。然后热管理控制状态机确定是否满足实时期限（步骤 1008）。如果不满足实时期限，则热管理控制状态机将当前热管理停止时间寄存器和热管理调节比例寄存器的热控制设置记录为失败（步骤 1010）。然后热管理控制状态机确定是否有将降低调节量的任意新的热

管理停止时间寄存器和热管理调节比例寄存器的热控制设置（步骤 1012）。如果有新的热管理停止时间寄存器和热管理调节比例寄存器的热控制设置，则操作返回到步骤 1002。如果在步骤 1002 没有任何新的热管理停止时间寄存器和热管理调节比例寄存器的热控制设置，则操作结束。

返回到步骤 1008，如果满足实时期限，则热管理控制状态机将当前热管理停止时间寄存器和热管理调节比例寄存器的热控制设置记录为通过（步骤 1014）。热管理控制状态机确定是否有将增加调节量的任意新的热管理停止时间寄存器和热管理调节比例寄存器的热控制设置（步骤 1016）。如果有新的热管理停止时间寄存器和热管理调节比例寄存器的热控制设置，则操作返回到步骤 1002。如果在步骤 1016 没有任何新的热管理停止时间寄存器和热管理调节比例寄存器的热控制设置，则操作结束。

这样，提供始终调节的操作模式帮助软件开发者进行测试并确保他们的代码在最坏情况的热状况下也能满足实时期限。软件开发者和质量保证团队可以使用该特征来确定程序或代码段能够容许并仍然保证满足实时系统的实时期限的最大调节量。一旦热管理控制状态机确定并确认了最大调节量，软件就可以将中断设置为在全面调节发生的状况下发生。如果热管理控制状态机总是产生该中断，那么热管理控制状态机通知应用可能存在违反或不满足实时保证的情况。

除了始终调节控制设置，实现方式还可以提供注入随机热事件或定向随机热事件的模式以对调节与软件执行的更具有真实感的交互进行仿真。该技术类似于在总线上随机注入错误以测试错误恢复代码。

图 11 描述了根据另外的说明性实施例的用于实现对中断等待时间影响最小的热调节控制的操作的流程图。如前所述，Cell BE 芯片包括通过图 3 的普适逻辑单元 351 提供的热管理系统。当计算机系统的任意部分被放置在调节条件下，该调节条件会降低整个系统的性能。性能的降低会根据要多久才能为该中断提供服务以及将为该中断提供多久的服务而增加中断等待时间。中断等待时间的增加在总体上对系统有严重

的影响，因此期望并且有必要最小化热调节对中断等待时间的影响。最小化热调节因中断等待时间而产生的影响是针对诸如由图 3 的 PPU 308 进行的 PPU 调节控制的特征。诸如图 3 的 SPU 310、311 和 312 之类的 SPU 不会获得中断，因此不会被该特征影响。

随着操作开始，诸如图 4 的热管理控制状态机 458 之类的热管理控制状态机监控所有的 PPU 中断状态比特和热管理系统中断屏蔽寄存器，诸如图 4 的热管理系统中断屏蔽寄存器 438（步骤 1102）。热管理系统中断屏蔽寄存器控制对中断的屏蔽。热管理控制状态机确定是否有任何未屏蔽的未决中断（步骤 1104）。如果没有未决中断或者有未决中断但是被屏蔽了，则操作返回到步骤 1102。

如果在步骤 1104 有未屏蔽的未决中断，则热管理控制状态机暂时禁用任何调节模式，不管是部分调节还是全面调节状态（步骤 1106）。禁用调节模式使 PPU 能够在没有热调节效果引起的任何延迟的情况下暂时以全性能运行并处理任意未决中断。同样，热管理控制状态机监控所有的 PPU 中断状态和热管理系统中断屏蔽寄存器（步骤 1108）。热管理控制状态机确定是否有任何未屏蔽的未决中断（步骤 1110）。如果没有未决中断或者有未决中断但是被屏蔽了，则操作返回到步骤 1108。当在步骤 1110 中断状态清除时，热管理控制状态机将 PPU 恢复到初始调节模式（步骤 1112），并且操作返回到步骤 1102。

中断处理程序可以选择在中断处理程序例程的开头或结尾清除中断状态比特。中断处理程序可以位于诸如图 3 的 Power 处理器单元 301 之类的 Power 处理器单元或由 Power 处理器单元执行的软件中。如果中断处理程序选择在开头清除中断状态比特并且还希望避免 PPU 的任意性能降低，则中断处理程序可以在清除中断状态比特之前禁用热调节。也就是说，中断不引发控制寄存器中的变化。因此，调节仍然是启用的，但是在出现未屏蔽中断时由诸如图 4 的 TMCU 402 之类的热管理控制单元挂起。如果中断处理程序应当在对中断进行处理之前重置中断状态，则该处理程序应当将控制寄存器设置为禁用调节（或将调节量减小到可接受的级别），重置中断，为中断提供服务，然后重新启用调节或将调

节量设回以前的级别。可以通过将诸如图 4 的热管理控制寄存器 430 之类的热管理控制寄存器设置为 0XX 来执行对热调节的示例性禁用，其中 X 是“无关位”（does not care）。在中断例程的结尾，中断处理程序应当将热管理控制寄存器设回它的初始值。如果中断处理程序在中断例程结尾清除中断状态比特，那么就不需要额外的工作，并且只要中断状态比特是激活的，热管理控制状态机就会将 PPU 保持在调节模式之外。

图 12 描述了根据另外的说明性实施例的用于热调节中的滞后的操作的流程图。如前所述，Cell BE 芯片包括通过图 3 的普适逻辑单元 351 提供的热管理系统。热调节中的滞后是做出诸如调节或结束调节之类的改变与该变化的响应或效果之间的迟滞。例如，如果将调节点设置为 75°C 并将结束调节点设置为 72°C，那么滞后范围是从 75°C 到 72°C。图 5 描述了热调节滞后。

诸如图 4 的热管理调节点寄存器 432 之类的热管理调节点寄存器提供两个温度设置：调节温度和结束调节温度。调节温度应当设置为高于结束调节温度。温度差异定义了调节温度和结束调节温度之间的滞后量，从而提供了可编程的滞后量。

说明性地将以下讨论限于一个 DTS，随着滞后热调节的操作开始，热管理控制状态机设置热管理调节点寄存器中的调节温度和结束调节温度（步骤 1202）。热管理控制状态机感应 DTS 的温度（步骤 1204）。热管理控制状态机确定从 DTS 感应的温度是否大于或等于调节温度（步骤 1206）。如果感应温度并不大于或等于调节温度，则操作返回步骤 1204。如果在步骤 1206 感应温度大于或等于调节温度，则热管理控制状态机初始化调节模式（步骤 1208）。

同样，热管理控制状态机感应 DTS 的温度（步骤 1210）。热管理控制状态机确定从 DTS 感应的温度是否大于或等于结束调节温度（步骤 1212）。如果感应温度不小于结束调节温度，则操作返回步骤 1210。如果在步骤 1212 感应温度小于结束调节温度，则热管理控制状态机禁用调节模式（步骤 1214），操作返回到步骤 1204。

这样，假定正确配置了热管理控制寄存器以允许调节模式，当温度

上升到等于或高于调节温度时，热管理控制状态机使单元进入调节模式。热管理控制状态机将单元保持在调节模式下直到温度下降到结束调节温度以下。如果结束调节温度小于调节温度，那么所标识的滞后使单元能够在禁用调节模式之前充分冷却。没有滞后，单元可能会很频繁地进入和退出调节模式并降低调节的整体效率和处理器的效率。

可以通过阻断指令的分派来完成示例性处理器调节方法。如果调节被频繁地启动和禁用，那么可能会经常刷新处理器的流水线，从而降低处理能力。另一个示例性处理器调节方法可以通过减缓时钟频率来完成。

图 13 描述了根据另外的说明性实施例的用于实现热调节逻辑的操作的流程图。图 13 表示一个如以上附图所述的完整的热管理解决方案。如前所述，Cell BE 芯片包括通过图 3 的普适逻辑单元 351 提供的热管理系统。诸如图 4 的 TMCU 402 之类的 TMCU 包括多个动态热管理寄存器。动态热管理寄存器是热管理控制寄存器、热管理调节点寄存器、热管理停止时间寄存器、热管理调节比例寄存器和热管理系统中断屏蔽寄存器，诸如图 4 的热管理控制寄存器 430 (TM_CR1 和 TM_CR2)、热管理调节点寄存器 432 (TM_TPR)、热管理停止时间寄存器 434 (TM_STR1 和 TM_STR2)、热管理调节比例寄存器 436 (TM_TSR) 和热管理系统中断屏蔽寄存器 438 (TM_SIMR)。

热管理调节点寄存器设置针对 DTS 的调节点。可以在热管理调节点寄存器中设置两个独立的调节点，一个针对 PPE，一个针对 SPE。该寄存器中还包含用于启用调节和禁用调节或者停止 PPE 或 SPE 的温度点。当温度等于或关于调节点时开始对 PPE 或 SPE 的执行调节。当温度下降到禁用调节的温度以下时调节停止。如果温度达到全面调节温度或停止温度，则停止对 PPE 或 SPE 的执行。

热管理控制状态机使用热管理停止时间寄存器和热管理调节比例寄存器来控制调节频率和调节量。当温度达到调节点时，热管理控制状态机将相应的 PPE 或 SPE 停止由在热管理调节比例寄存器中的相应比例值指定的时钟数。然后热管理控制状态机使 PPE 或 SPE 能够运行由

在热管理停止时间寄存器中的运行值乘以相应的比例值所指定的时钟数。该序列持续到温度下降到禁用调节以下。

热管理控制状态机使用热管理系统中断屏蔽寄存器来选择哪些中断在中断未决的同时禁用对 PPE 的调节。

热管理控制寄存器独立地为每个 PPE 或 SPE 设置调节模式。下面是可以独立地为每个 PPE 或 SPE 设置的五种不同的模式：

禁用动态调节（包括内核停止安全性）；

正常操作（启用动态调节和内核停止安全性）；

始终调节 PPE 或 SPE（启用内核停止安全性）；

禁用内核停止安全性（启用动态调节并且禁用内核停止安全性）；

始终调节 PPE 或 SPE 并且禁用内核停止安全性。

作为用于实现热调节逻辑的操作，热管理控制状态机设置热管理调节节点寄存器中的调节温度和结束调节温度（步骤 1302）。热管理控制状态机感应 DTS 的温度（步骤 1304）。热管理控制状态机确定从 DTS 感应的温度是否大于或等于调节温度（步骤 1306）。如果感应温度并不大于或等于调节温度，则操作返回步骤 1304。如果感应温度大于或等于调节温度，则热管理控制状态机初始化调节模式（步骤 1308）。

然后，热管理控制状态机通过如热管理控制寄存器中所表明的值所表示的调节类型来控制调节（步骤 1310）。一旦表明了调节模式，热管理控制状态机就通过在热管理停止时间寄存器中所表明的调节量来限制调节（步骤 1312）。停止时间寄存器设置处理器将停止的时间与将允许处理器运行的时间之间的比值或调节百分比。最后，热管理控制状态机通过在热管理比例寄存器中指定的值来缩放停止的持续时间和运行时间（步骤 1314）。此时操作分为并发的操作，即步骤 1316 和步骤 1322。在步骤 1316，热管理控制状态机感应 DTS 的温度。热管理控制状态机确定从 DTS 感应的温度是否小于调节温度（步骤 1318）。如果感应温度不小于结束调节温度，则操作返回步骤 1316。如果 DTS 小于结束调节温度，则热管理控制状态机禁用调节模式（步骤 1320），操作返回到步骤 1304。

返回到步骤 1314, 在实现最终的调节限制之后, 热管理控制状态机并发地监控任意未决中断的所有 PPU 中断状态 (步骤 1322)。如果在实现调节时遇到中断, 则热管理控制状态机暂时禁用任何调节模式直到中断已被处理, 于是不管是部分调节状态还是全面调节状态都启用调节并且操作返回到步骤 1308。参考图 11 进行对监控中断状态的深入讨论。

这样, Cell BE 芯片所包括的热管理系统的热中断逻辑提供了一种动态手段来管理 Cell BE 芯片的热状态和保护 Cell BE 芯片及它的组件。

说明性实施例可以采取全硬件实施例、全软件实施例或既包含硬件单元又包含软件单元的实施例的形式。说明性实施例在软件中实现, 该软件包括但不限于固件、驻留软件、微代码等。

此外, 说明性实施例可以采取可以从计算机可用或计算机可读的介质访问的计算机程序产品的形式, 该计算机程序产品提供程序代码以供计算机或任意指令执行系统使用或结合计算机或任意指令执行系统而使用。为了该描述的目的, 计算机可用或计算机可读的介质可以是任意能够包含、存储、传送、传播或传输程序以供指令执行系统、设备或装置使用或结合指令执行系统、设备或装置而使用的有形设备。

介质可以是电子、磁、光、电磁、红外线或半导体系统 (或者设备或装置) 或传播媒介。计算机可读介质的示例包括半导体或固态存储器、磁带、可移动计算机磁盘、随机存取存储器 (RAM)、只读存储器 (ROM)、硬磁盘和光盘。光盘的当前示例包括压缩光盘-只读存储器 (CD-ROM)、光盘-读/写 (CD-R/W) 和 DVD。

适合于存储和/或执行程序代码的数据处理系统将包括至少一个直接地或通过系统总线间接地与存储单元连接的处理器。存储单元可以包括在程序代码的实际执行期间使用的本地存储器、大容量存储器和高速缓冲存储器, 为了减小在执行期间从大容量存储器获取代码的次数, 高速缓冲存储器提供对至少一部分程序代码的临时存储。

输入/输出或 I/O 设备 (包括但不限于键盘、显示器、指示设备等等) 能够直接地或通过中间 I/O 控制器与系统连接。

网络适配器也可以与系统连接以使数据处理系统能够通过中间专用网络或公共网络与其他数据处理系统或者远程打印机或存储设备连接。调制解调器、电缆调制解调器和以太网卡正是几种当前可用类型的网络适配器。

已经为了说明和描述的目的而提出了对说明性实施例的描述，并且该描述并非旨在穷举或将本发明限制为所公开形式的说明性实施例。很多修改和变更对本领域的普通技术人员来说将是很明显的。选择和描述实施例是为了最好地说明说明性实施例的原理、实际应用并使本领域的普通技术人员能够针对具有适合于所考虑的特定用途的各种修改的各种实施例来理解说明性实施例。

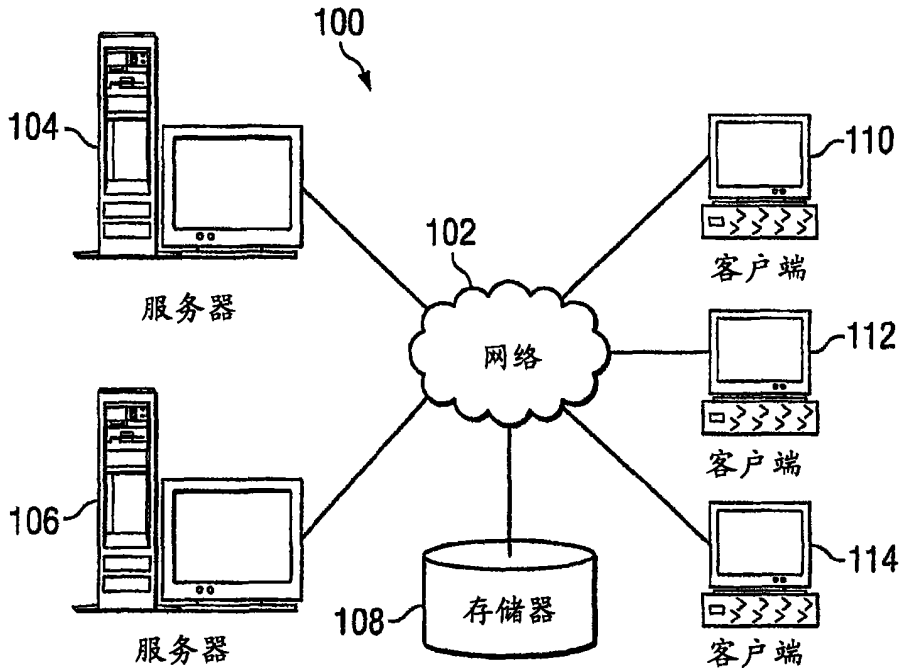


图 1

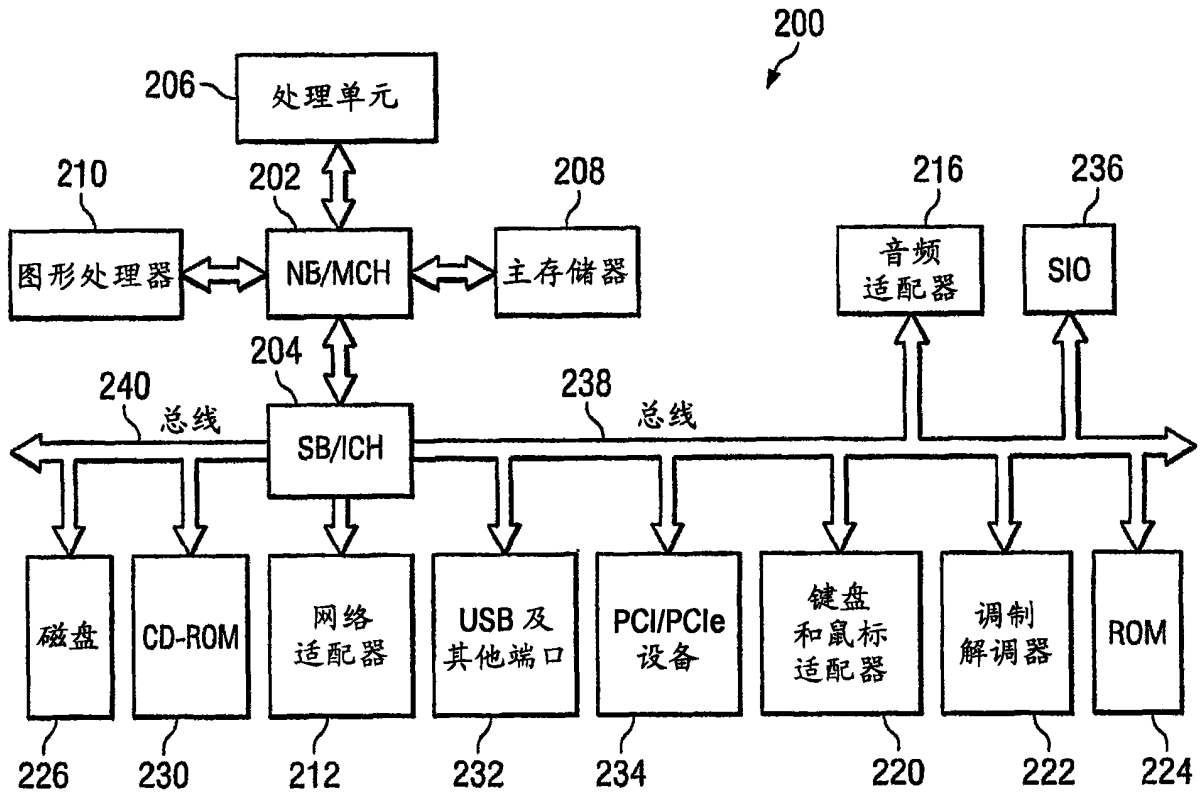


图 2

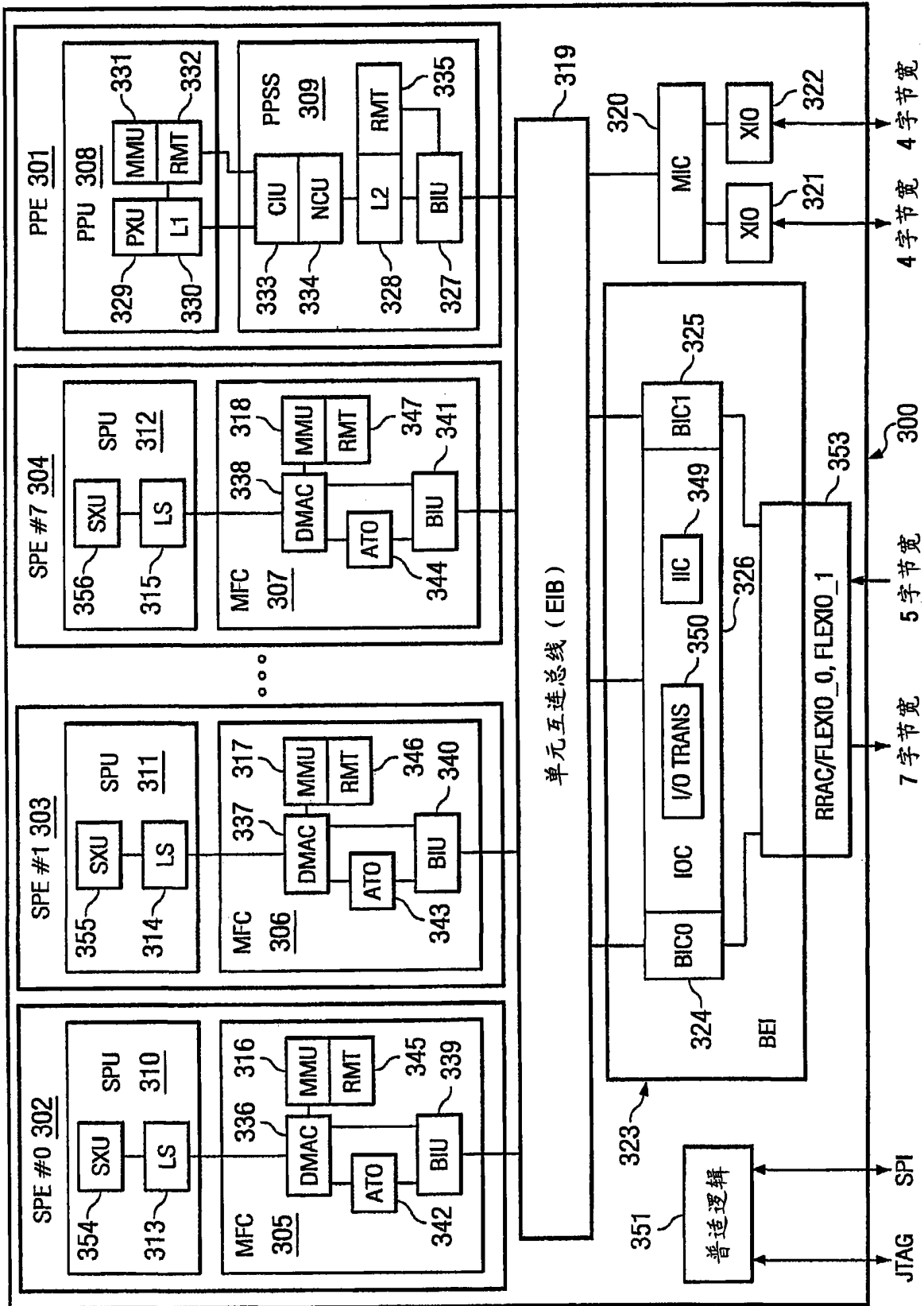


图 3

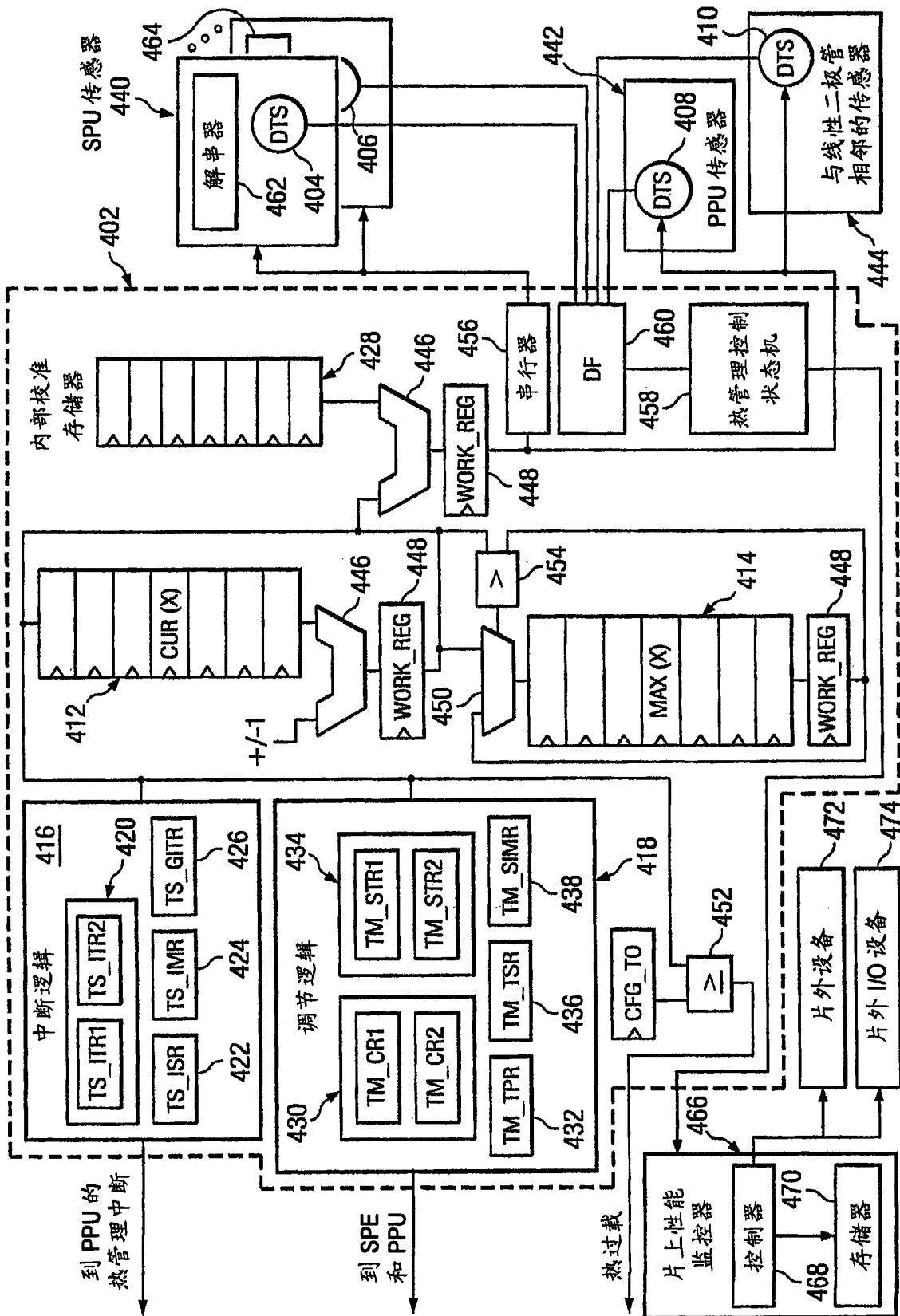


图 4

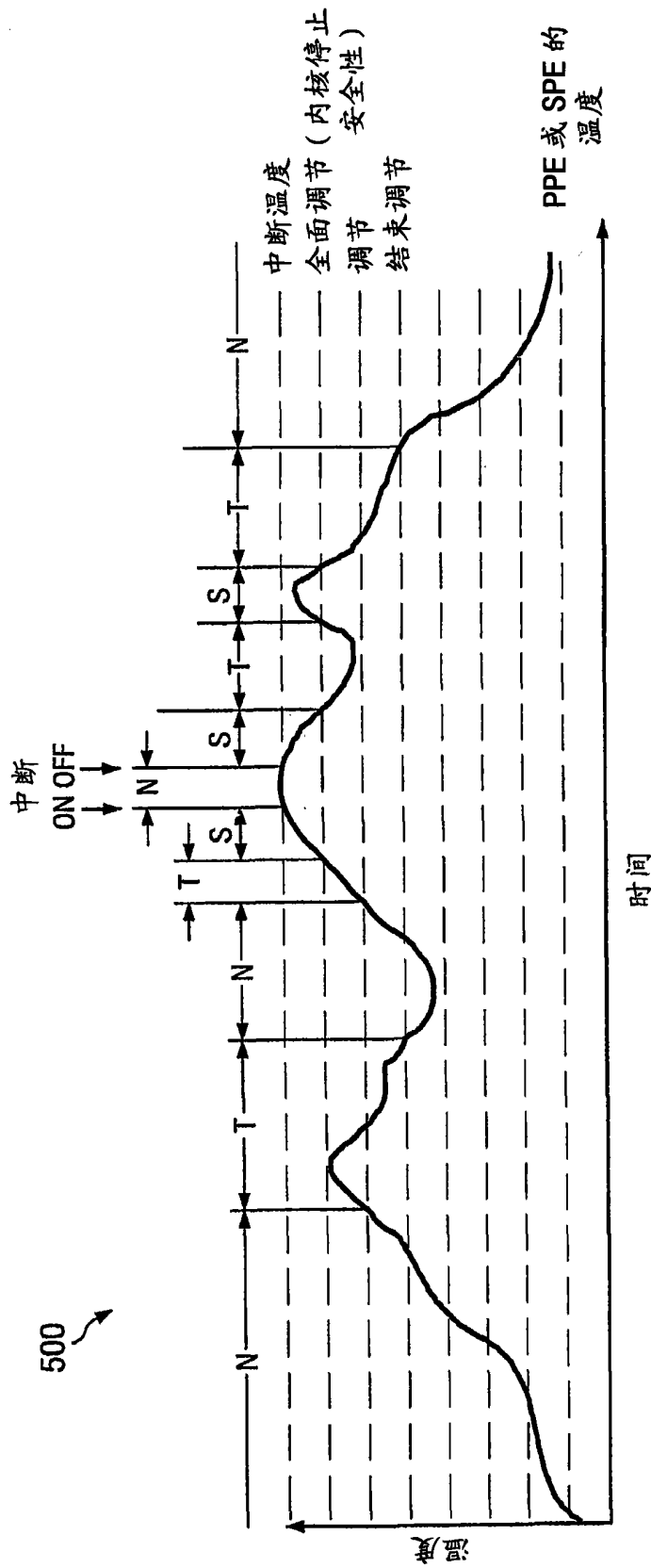


图 5

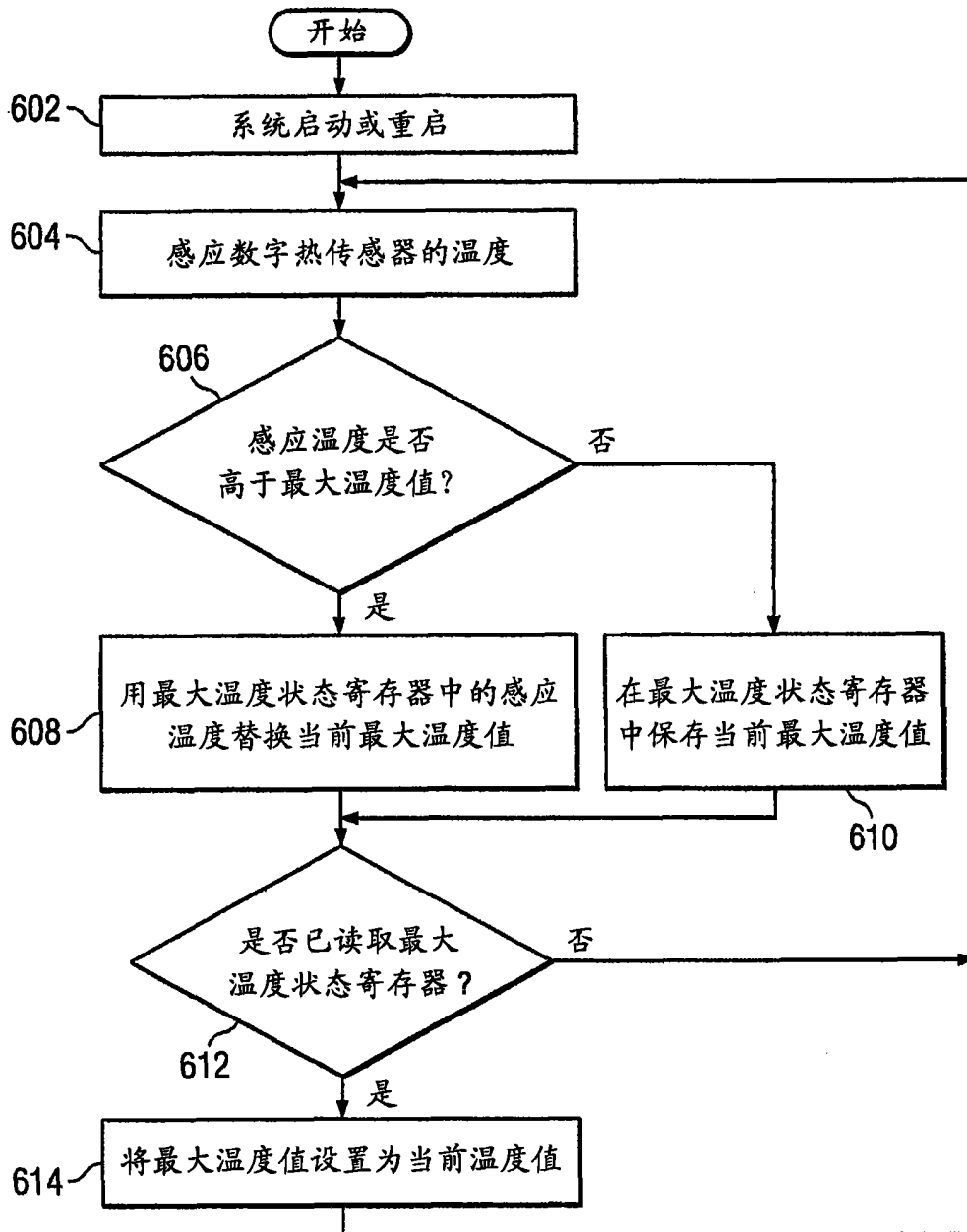


图 6

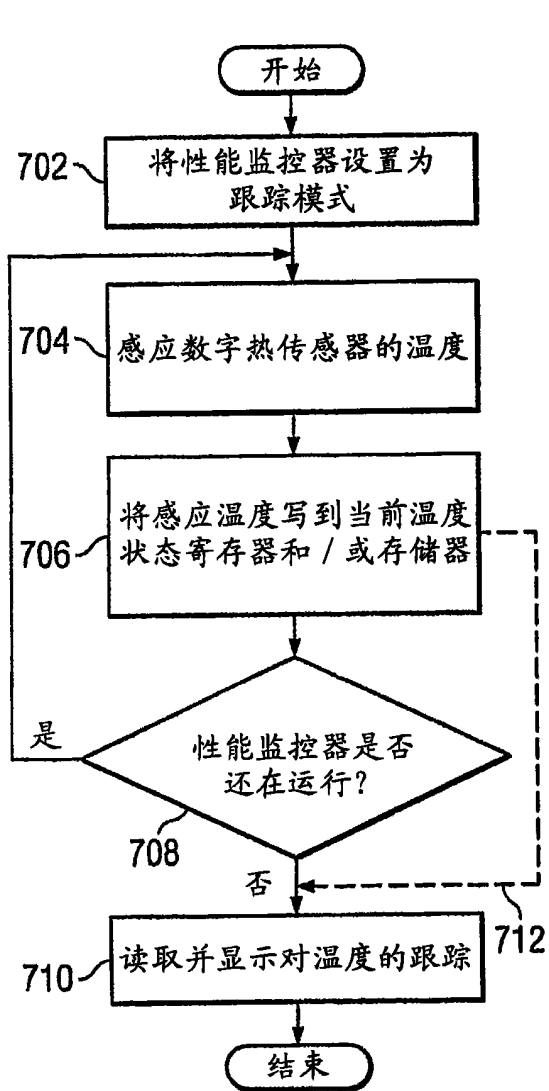


图 7

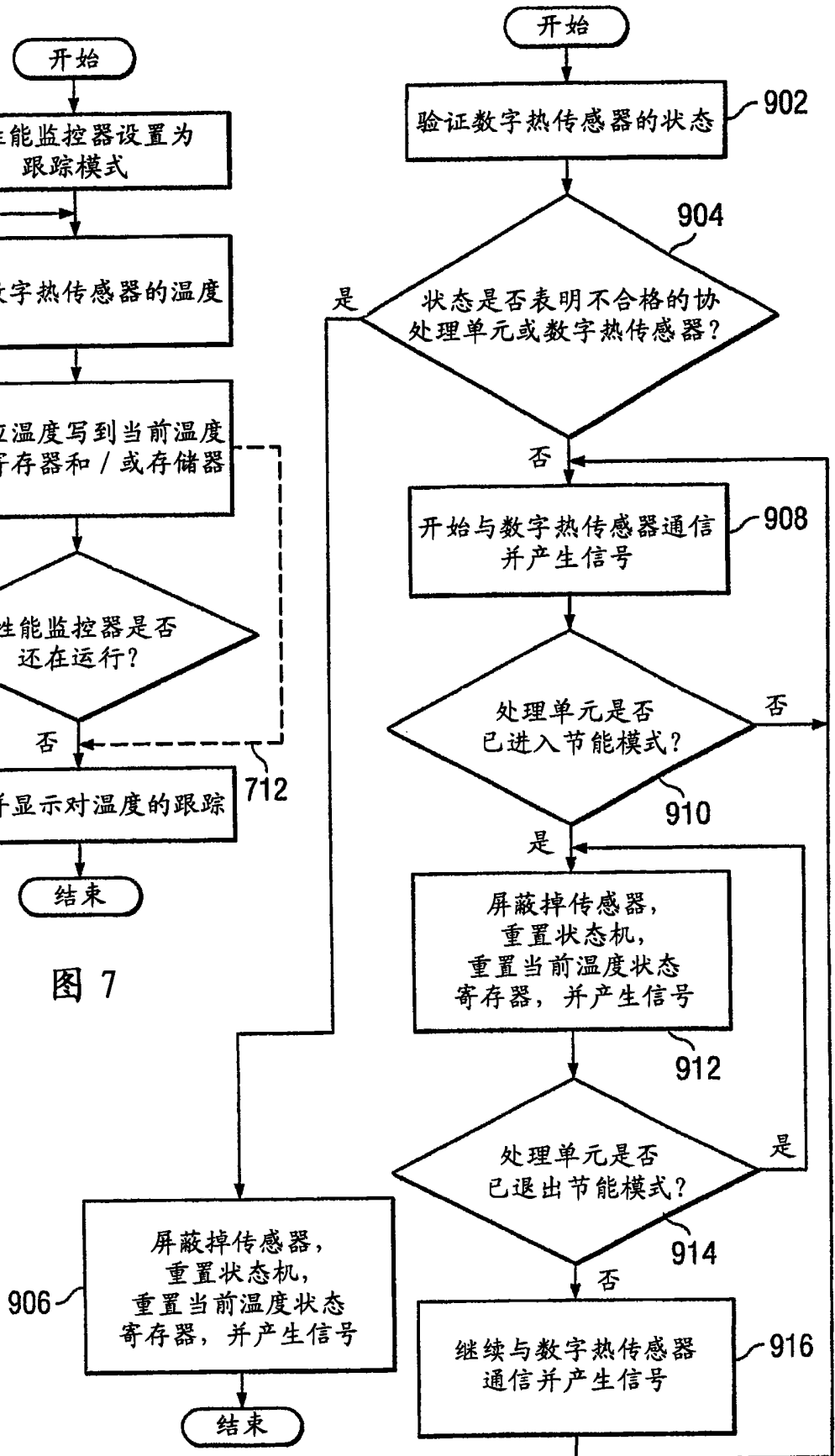


图 9

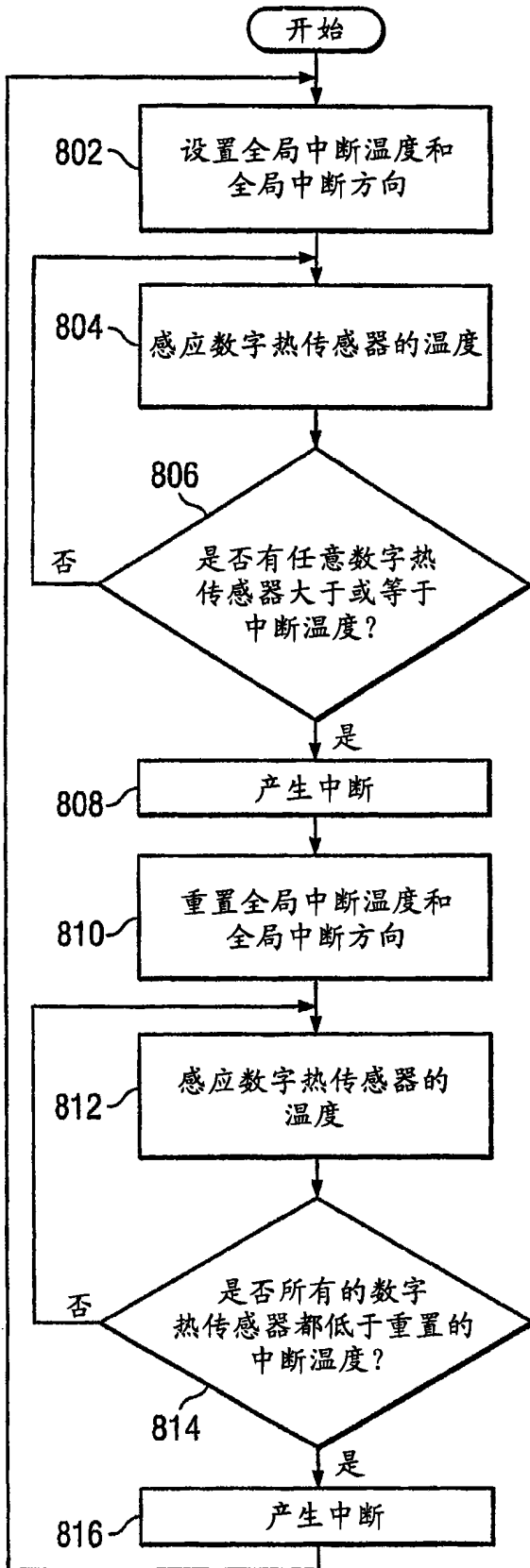


图 8A

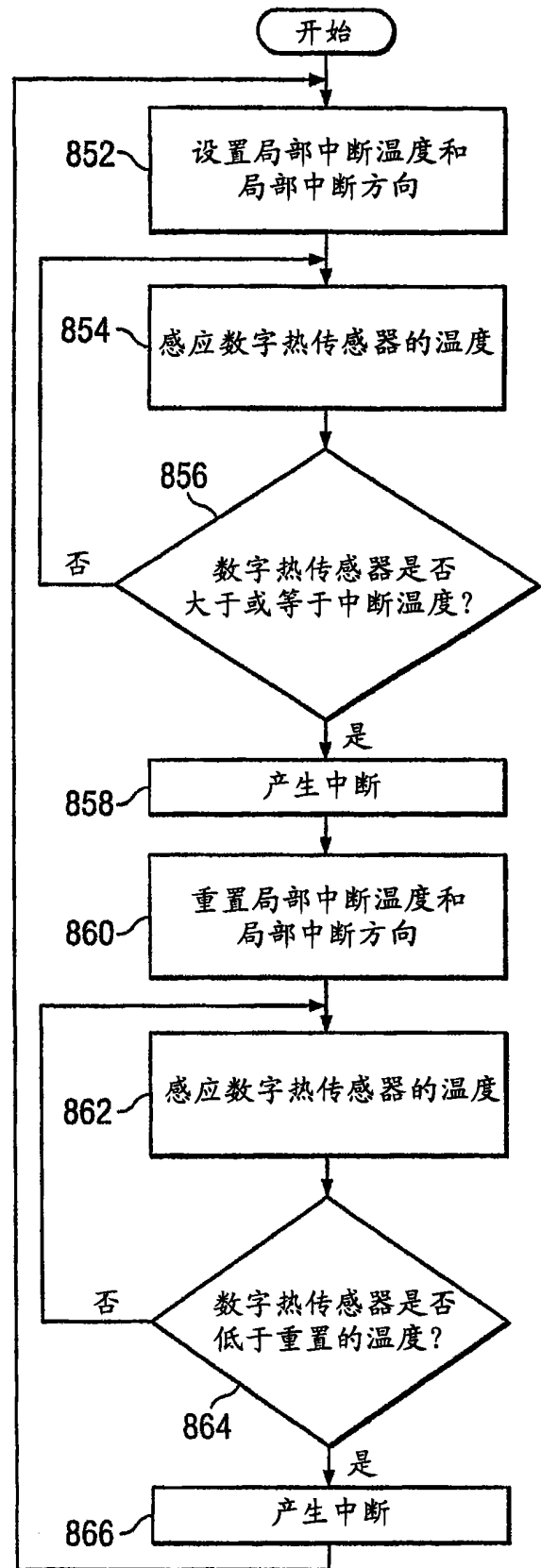


图 8B

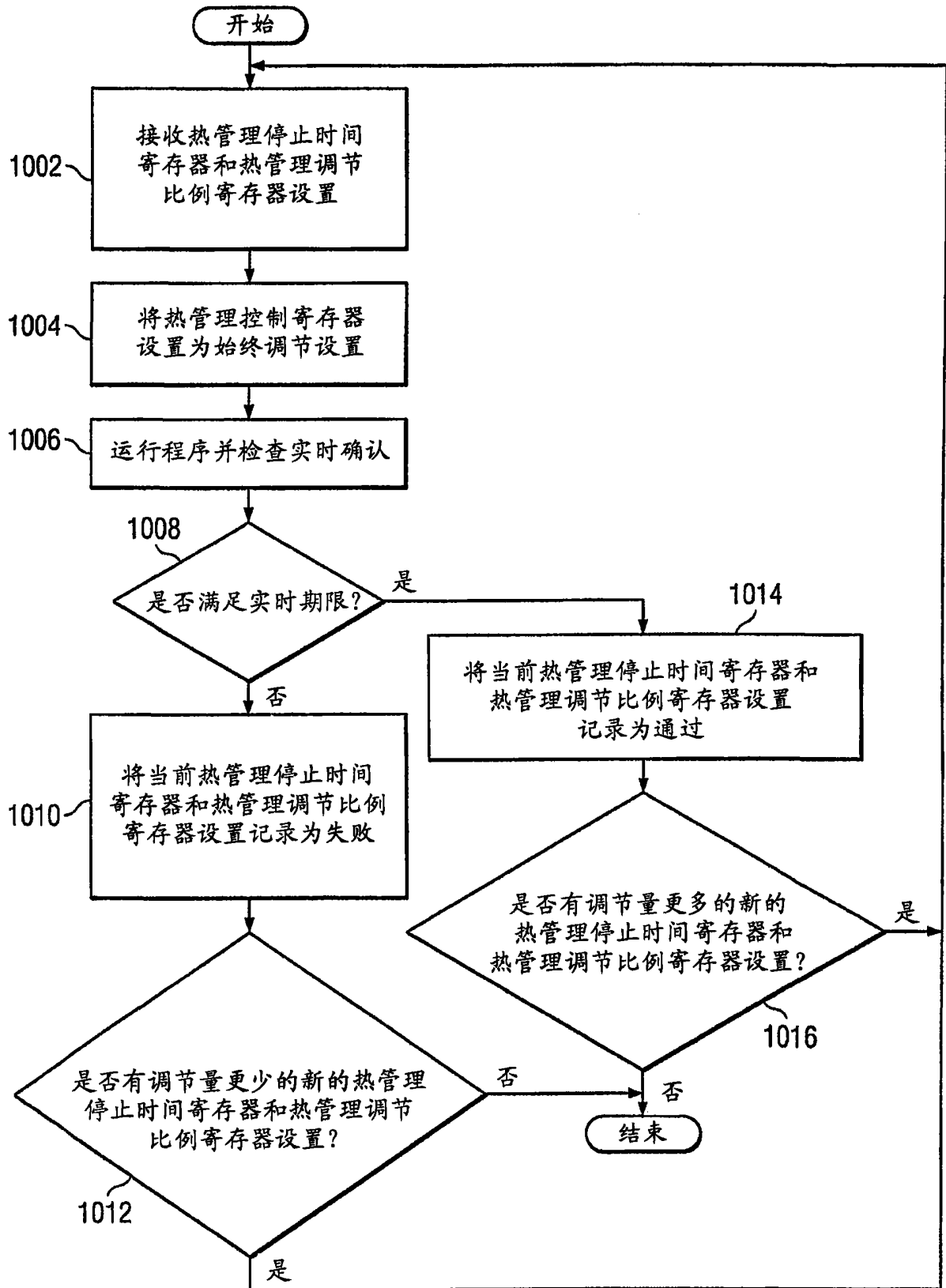


图 10

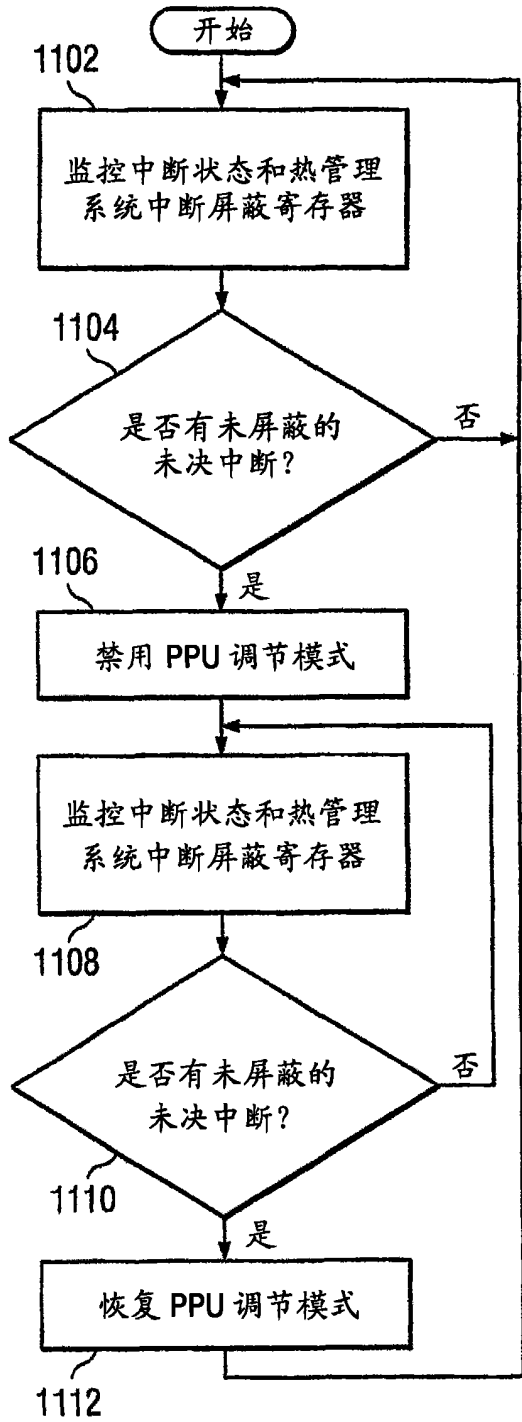


图 11

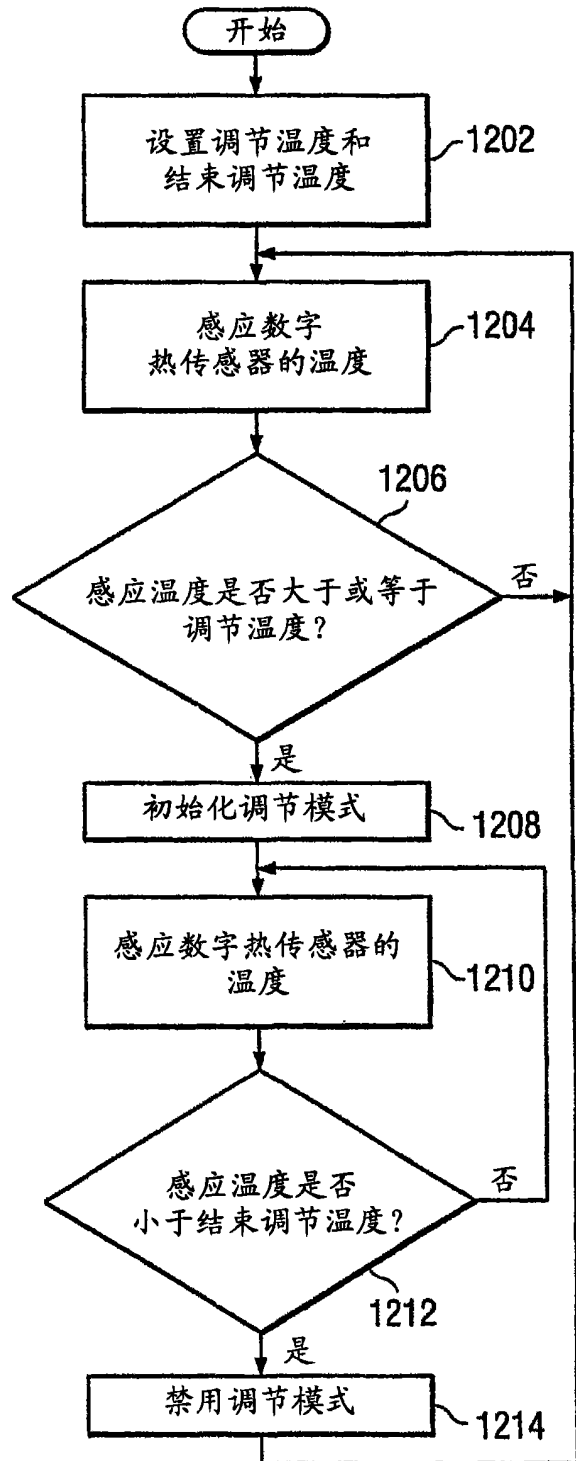


图 12

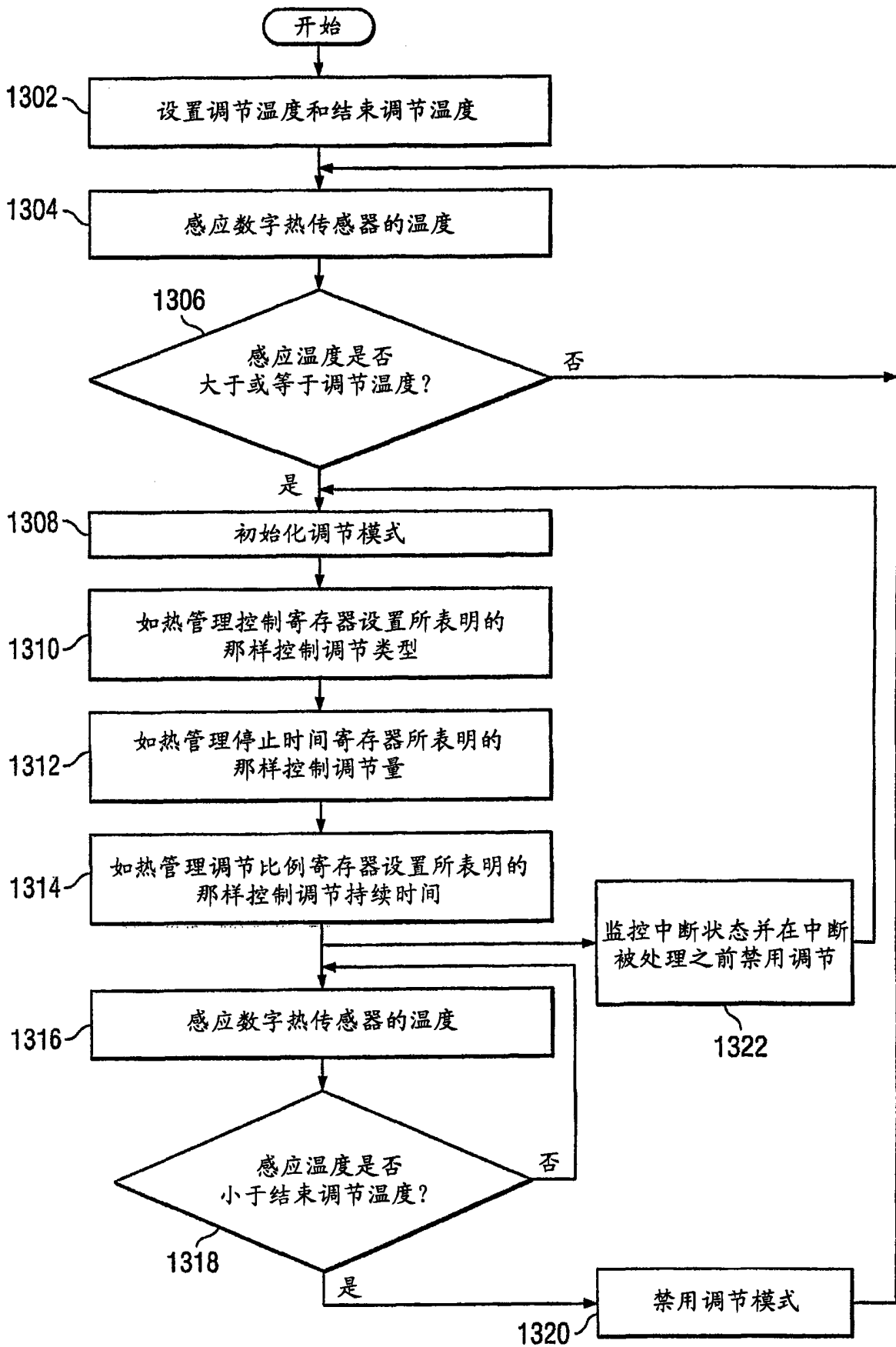


图 13