



(12)发明专利

(10)授权公告号 CN 106233224 B

(45)授权公告日 2019.09.03

(21)申请号 201580018319.X

(22)申请日 2015.04.08

(65)同一申请的已公布的文献号
申请公布号 CN 106233224 A

(43)申请公布日 2016.12.14

(30)优先权数据
61/977,013 2014.04.08 US
61/981,714 2014.04.18 US
14/280,630 2014.05.18 US

(85)PCT国际申请进入国家阶段日
2016.09.30

(86)PCT国际申请的申请数据
PCT/US2015/024845 2015.04.08

(87)PCT国际申请的公布数据
W02015/157371 EN 2015.10.15

(73)专利权人 高通股份有限公司
地址 美国加利福尼亚

(72)发明人 H·J·朴 Y·H·康
R·F·奥尔顿 C·L·梅德拉诺
J·J·安德森

(74)专利代理机构 永新专利商标代理有限公司
72002
代理人 张扬 王英

(51)Int.Cl.
G06F 1/20(2006.01)
G06F 1/26(2006.01)

(56)对比文件
CN 103649869 A, 2014.03.19,
US 2008115010 A1, 2008.05.15,
US 2010058078 A1, 2010.03.04,
US 2014075222 A1, 2014.03.13,
US 2004215987 A1, 2004.10.28,
US 2013073875 A1, 2013.03.21,

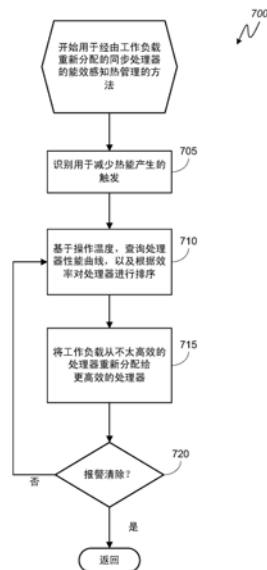
审查员 张诗伯
权利要求书5页 说明书21页 附图16页

(54)发明名称

多处理器片上系统中的能效感知热管理的方法和系统

(57)摘要

本文公开了用于包含异构的多处理器片上系统(“SoC”)的便携式计算设备中的能效感知热管理的方法和系统的各种实施例。由于该异构的多处理器SoC中的各个处理部件可能在给定的温度,呈现不同的处理效率,因此可以利用能效感知热管理技术(其对各个处理部件在它们测量的操作温度时的性能数据进行比较),以便通过调整针对最低能效处理部件的电源、将工作负载重新分配离开最低能效处理部件、或者转换最低能效处理部件的功率模式,来优化服务质量(“QoS”)。用这些方式,该解决方案的实施例对跨SoC用于处理一个MIPS的工作负载所消耗的平均功率量进行优化。



1. 一种用于在具有同步多处理器片上系统 (“SoC”) 的便携式计算设备中管理热能产生的方法, 所述方法包括:

监测与所述多处理器SoC中的多个独立处理部件中的每一个处理部件唯一关联的温度读数, 其中, 所述多个独立处理部件共享公共的电源电压和时钟发生器频率;

在所述便携式计算设备中所述多个独立处理部件外部的各个位置处监测所述便携式计算设备的至少一个热参数, 其中所述便携式计算设备的所述至少一个热参数包括所述便携式计算设备的皮肤温度;

响应于所监测的热参数超过预定门限, 来产生报警;

响应于所述报警, 对所监测的与所述多个独立处理部件中的每一个处理部件唯一关联的温度读数进行采样;

基于所采样的温度读数, 查询每一个处理部件的性能数据, 其中, 所述性能数据表示当给定的独立处理部件在给定的温度操作时, 其功耗和工作负载处理能力之间的关系;

对每一个处理部件的所述性能数据进行比较, 以识别所述多处理器SoC中的最低能效处理部件; 以及

将第一工作负载从所述最低能效处理部件重新分配给更高能效处理部件, 其中, 重新分配所述第一工作负载起到减少所述最低能效处理部件的所述功耗的作用。

2. 根据权利要求1所述的方法, 其中, 所述最低能效处理部件是: 每一处理的工作负载, 消耗最多功率量的处理部件。

3. 根据权利要求1所述的方法, 还包括:

确定所述报警还没有被清除;

对所监测的与所述多个独立处理部件中的每一个处理部件唯一关联的温度读数进行重新采样;

基于所重新采样的温度读数, 重新查询每一个处理部件的性能数据;

对每一个处理部件的所重新查询的性能数据进行比较, 以识别新的最低能效处理部件; 以及

将第二工作负载从所述最低能效处理部件重新分配给更高能效处理部件, 其中, 重新分配所述第二工作负载起到减少所述新的最低能效处理部件的所述功耗的作用。

4. 根据权利要求1所述的方法, 还包括:

确定所述报警已被清除; 以及

准许排队的工作负载被调度给所述最低能效处理部件。

5. 根据权利要求1所述的方法, 还包括:

将所述最低能效处理部件的功率模式从活动模式转换成空闲模式。

6. 根据权利要求5所述的方法, 还包括:

确定所述报警已被清除; 以及

准许所述最低能效处理部件返回到活动功率模式。

7. 根据权利要求1所述的方法, 其中, 所述热参数与以下各项中的一项相关联: 层叠封装存储器温度、结点温度和电池容量。

8. 根据权利要求1所述的方法, 其中, 所述便携式计算设备具有无线电话的形式。

9. 一种用于在具有同步多处理器片上系统 (“SoC”) 的便携式计算设备中管理热能产生

的计算机系统,所述系统包括:

监测模块,其用于:

监测与所述多处理器SoC中的多个独立处理部件中的每一个处理部件唯一关联的温度读数,其中,所述多个独立处理部件共享公共的电源电压和时钟发生器频率;

在所述便携式计算设备中所述多个独立处理部件外部的各个位置处监测所述便携式计算设备的至少一个热参数,其中所述便携式计算设备的所述至少一个热参数包括所述便携式计算设备的皮肤温度;

响应于所监测的热参数超过预定门限,来产生报警;以及

响应于所述报警,对所监测的与所述多个独立处理部件中的每一个处理部件唯一关联的温度读数进行采样;

效率管理器(“EM”)模块,其用于:

基于所采样的温度读数,查询每一个处理部件的性能数据,其中,所述性能数据表示当给定的独立处理部件在给定的温度操作时,其功耗和工作负载处理能力之间的关系;

对每一个处理部件的所述性能数据进行比较,以识别所述多处理器SoC中的最低能效处理部件;以及

将第一工作负载从所述最低能效处理部件重新分配给更高能效处理部件,其中,重新分配所述第一工作负载起到减少所述最低能效处理部件的所述功耗的作用。

10. 根据权利要求9所述的计算机系统,其中,所述最低能效处理部件是:每一处理的工作负载,消耗最多功率量的处理部件。

11. 根据权利要求9所述的计算机系统,其中:

所述监测模块还用于:

确定所述报警还没有被清除;以及

对所监测的与所述多个独立处理部件中的每一个处理部件唯一关联的温度读数进行重新采样;

所述EM模块还用于:

基于所重新采样的温度读数,重新查询每一个处理部件的性能数据;以及

对每一个处理部件的所重新查询的性能数据进行比较,以识别新的最低能效处理部件;以及

将第二工作负载从所述最低能效处理部件重新分配给更高能效处理部件,其中,重新分配所述第二工作负载起到减少所述新的最低能效处理部件的所述功耗的作用。

12. 根据权利要求9所述的计算机系统,其中:

所述监测模块还用于:

确定所述报警已被清除;以及

所述EM模块还用于:

准许排队的工作负载被调度给所述最低能效处理部件。

13. 根据权利要求9所述的计算机系统,其中:

所述监测模块还用于:

将所述最低能效处理部件的功率模式从活动模式转换成空闲模式。

14. 根据权利要求13所述的计算机系统,其中:

所述监测模块还用于：

确定所述报警已被清除；以及

所述EM模块还用于：

准许所述最低能效处理部件返回到活动功率模式。

15. 根据权利要求9所述的计算机系统，其中，所述热参数与以下各项中的一项相关联：层叠封装存储器温度、结点温度和电池容量。

16. 一种用于在具有同步多处理器片上系统（“SoC”）的便携式计算设备中管理热能产生的计算机系统，所述系统包括：

用于监测与所述多处理器SoC中的多个独立处理部件中的每一个处理部件唯一关联的温度读数的单元，其中，所述多个独立处理部件共享公共的电源电压和时钟发生器频率；

用于在所述便携式计算设备中所述多个独立处理部件外部的各个位置处监测所述便携式计算设备的至少一个热参数的单元，其中所述便携式计算设备的所述至少一个热参数包括所述便携式计算设备的皮肤温度；

用于响应于所监测的热参数超过预定门限，来产生报警的单元；

用于响应于所述报警，对所监测的与所述多个独立处理部件中的每一个处理部件唯一关联的温度读数进行采样的单元；

基于所采样的温度读数，用于查询每一个处理部件的性能数据的单元，其中，所述性能数据表示当给定的独立处理部件在给定的温度操作时，其功耗和工作负载处理能力之间的关系；

用于对每一个处理部件的所述性能数据进行比较，以识别所述多处理器SoC中的最低能效处理部件的单元；以及

用于将第一工作负载从所述最低能效处理部件重新分配给更高能效处理部件的单元，其中，重新分配所述第一工作负载起到减少所述最低能效处理部件的所述功耗的作用。

17. 根据权利要求16所述的计算机系统，其中，所述最低能效处理部件是：每一处理的工作负载，消耗最多功率量的处理部件。

18. 根据权利要求16所述的计算机系统，还包括：

用于确定所述报警还没有被清除的单元；

用于对所监测的与所述多个独立处理部件中的每一个处理部件唯一关联的温度读数进行重新采样的单元；

基于所重新采样的温度读数，用于重新查询每一个处理部件的性能数据的单元；

用于对每一个处理部件的所重新查询的性能数据进行比较，以识别新的最低能效处理部件的单元；以及

用于将第二工作负载从所述最低能效处理部件重新分配给更高能效处理部件的单元，其中，重新分配所述第二工作负载起到减少所述新的最低能效处理部件的所述功耗的作用。

19. 根据权利要求16所述的计算机系统，还包括：

用于确定所述报警已被清除的单元；以及

用于准许排队的工作负载被调度给所述最低能效处理部件的单元。

20. 根据权利要求16所述的计算机系统，还包括：

用于将所述最低能效处理部件的功率模式从活动模式转换成空闲模式的单元。

21. 根据权利要求20所述的计算机系统,还包括:

用于确定所述报警已被清除的单元;以及

用于准许所述最低能效处理部件返回到活动功率模式的单元。

22. 根据权利要求16所述的计算机系统,其中,所述热参数与以下各项中的一项相关联:层叠封装存储器温度、结点温度和电池容量。

23. 根据权利要求16所述的计算机系统,其中,所述便携式计算设备具有无线电话的形式。

24. 一种上面存储有计算机程序的计算机可读存储介质,所述计算机程序可由处理器执行,以实现用于在具有同步多处理器片上系统(“SoC”)的便携式计算设备中管理热能产生的方法,所述方法包括:

监测与所述多处理器SoC中的多个独立处理部件中的每一个处理部件唯一关联的温度读数,其中,所述多个独立处理部件共享公共的电源电压和时钟发生器频率;

在所述便携式计算设备中所述多个独立处理部件外部的各个位置处监测所述便携式计算设备的至少一个热参数,其中所述便携式计算设备的所述至少一个热参数包括所述便携式计算设备的皮肤温度;

响应于所监测的热参数超过预定门限,来产生报警;

响应于所述报警,对所监测的与所述多个独立处理部件中的每一个处理部件唯一关联的温度读数进行采样;

基于所采样的温度读数,查询每一个处理部件的性能数据,其中,所述性能数据表示当给定的独立处理部件在给定的温度操作时,其功耗和工作负载处理能力之间的关系;

对每一个处理部件的所述性能数据进行比较,以识别所述多处理器SoC中的最低能效处理部件;以及

将第一工作负载从所述最低能效处理部件重新分配给更高能效处理部件,其中,重新分配所述第一工作负载起到减少所述最低能效处理部件的所述功耗的作用。

25. 根据权利要求24所述的计算机可读存储介质,其中,所述最低能效处理部件是:每一处理的工作负载,消耗最多功率量的处理部件。

26. 根据权利要求24所述的计算机可读存储介质,所述方法还包括:

确定所述报警还没有被清除;

对所监测的与所述多个独立处理部件中的每一个处理部件唯一关联的温度读数进行重新采样;

基于所重新采样的温度读数,重新查询每一个处理部件的性能数据;

对每一个处理部件的所重新查询的性能数据进行比较,以识别新的最低能效处理部件;以及

将第二工作负载从所述最低能效处理部件重新分配给更高能效处理部件,其中,重新分配所述第二工作负载起到减少所述新的最低能效处理部件的所述功耗的作用。

27. 根据权利要求24所述的计算机可读存储介质,所述方法还包括:

确定所述报警已被清除;以及

准许排队的工作负载被调度给所述最低能效处理部件。

28. 根据权利要求24所述的计算机可读存储介质,还包括:
将所述最低能效处理部件的功率模式从活动模式转换成空闲模式。
29. 根据权利要求28所述的计算机可读存储介质,所述方法还包括:
确定所述报警已被清除;以及
准许所述最低能效处理部件返回到活动功率模式。
30. 根据权利要求24所述的计算机可读存储介质,其中,所述热参数与以下各项中的一项相关联:层叠封装存储器温度、结点温度和电池容量。

多处理器片上系统中的能效感知热管理的方法和系统

背景技术

[0001] 便携式计算设备(“PCD”)正在个人和专业级别上成为人们的必需品。这些设备可以包括蜂窝电话、便携式数字助理(“PDA”)、便携式游戏控制台、掌上型计算机和其它便携式电子设备。

[0002] 典型地,PCD在尺寸方面受到限制,并由此用于PCD中的部件的空间通常极为重要。因此,在典型的PCD规格中通常没有足够的空间可被工程师和设计师利用,使得不能通过采用无源散热部件的巧妙空间布局或者布置来减轻部件的热劣化或者热故障。因此,通常通过各种热管理技术的应用,对PCD中的热能产生进行管理,其中这些技术包括以性能为代价来使电子产品衰弱或关闭电子产品。

[0003] 在PCD中使用热管理技术,以尽力在缓解热能产生和影响PCD所提供的服务质量(“QoS”)之间寻求平衡。在具有异构处理部件的PCD中,由于PCD中的各个处理部件不是同等地生产的,因此平衡折衷的结果可能是难以管理的。因此,本领域已知的热缓解措施(其响应于热触发,统一地对所有处理部件同等地限制功率频率,或者简单地针对最热处理部件来限制电源电压和时钟发生器频率),通常不能以热能产生的速率降低为交换,来优化QoS水平。由于片上系统(“SoC”)中的各个处理部件(无论在设计上是同质的还是异构的)都不可避免地会在性能能力上会发生变化,因此当针对关于QoS的影响进行测量时,并不始终都是最热的处理部件提供最大的潜在热能减少。

[0004] 因此,本领域需要用于能效感知热缓解的方法和系统。此外,本领域还需要用于对处理部件进行对比,识别最低效处理部件和最高效处理部件的系统和方法。

发明内容

[0005] 本文公开了用于在包含同步多处理器片上系统(“SoC”)的便携式计算设备中的能效感知热管理的方法和系统的各种实施例。由于各个处理部件可能具有不同的特性(无论是故意设计的,还是源自于制造过程不同),多处理器SoC可能在给定的温度,呈现不同的处理效率,因此可以利用能效感知热管理技术(其对各个处理部件在它们测量的操作温度时的性能数据进行比较),以便通过将工作负载分配和/或重新分配离开最低效处理部件,在功率和热约束的条件下将性能最大化。实施例并不是简单地寻求防止工作负载运行在不太能量高效的内核上。也就是说,在一些实施例中,当工作负载启动时,系统可以考虑每一个部件的能效,并将该工作负载置于该工作负载适合的最高效的CPU上。例如,如果最高效内核已经被过重地使用,那么可以选择下一个最高效内核。用此方式,该解决方案的实施例对跨SoC用于处理已知工作负载所消耗的平均功率量进行优化。

[0006] 一种这样的方法涉及:监测与异构、同质多处理器SoC中的多个独立处理部件中的每一个处理部件唯一关联的温度读数。由于SoC具有同步架构,因此多个处理部件共享公共的电源和时钟发生器相关联。对热参数进行监测,并且接收到用于指示已超过与该热参数相关联的门限的报警。随后,对所监测的与每一个处理部件唯一关联的温度读数进行采样。基于所采样的温度读数,查询每一个处理部件的性能数据。该性能数据表示当给定的独立

处理部件在给定的温度操作时,其功耗和工作负载处理能力之间的关系。随后,对这些处理部件的性能数据进行比较,以识别最低能效处理部件。一旦识别出最低能效处理部件,可以将其工作负载重新分配给更高效处理部件。有利的是,将工作负载从不太能量高效的处理部件转移到更高能效处理部件减少该最低能效处理部件的功耗,从而优化SoC的整体处理效率。

附图说明

[0007] 在附图中,除非另外指出,否则贯穿各个视图的相同附图标记指代相同的部分。对于利用诸如“102A”或“102B”之类的字母字符进行命名的附图标记而言,这些字母字符命名可以区分在同一附图中出现的两个相同的部分或者元件。当一个附图标记旨在涵盖所有附图之中具有相同附图标记的所有部分时,可以省略用于附图标记的字母字符命名。

[0008] 图1A是示出了在不同的热条件下进行操作的示例性处理部件的一对性能曲线的图;

[0009] 图1B是示出了在不同的热条件下进行操作的两个示例性处理部件 (“低性能”CPU处理部件和“高性能”GPU处理部件)中的每一个处理部件的一对性能曲线的图;

[0010] 图1C是示出了一对示例性内核的一对性能曲线的图;

[0011] 图1D是示出了在图1C示图中描绘的一对示例性内核的一对不同的性能曲线的图;

[0012] 图2A是描绘了包括多个处理部件的片上系统中的异步架构的方面的功能框图;

[0013] 图2B是描绘了包括多个处理部件的片上系统中的同步架构的方面的功能框图;

[0014] 图3是描绘了用于便携式计算设备 (“PCD”)中的能效感知热管理的片上系统的实施例的功能框图;

[0015] 图4是具有无线电话的形式的PCD的示例性、非限制性方面的功能框图,其用于实现对热状况进行监测,对性能数据进行比较,设置最佳功率频率,以及将工作负载调度到最适合于进行高效处理的处理部件的方法和系统;

[0016] 图5A是描绘了用于图4中所示出的芯片的硬件的示例性空间布局的功能框图;

[0017] 图5B是示出了图4和图5A的PCD的示例性软件体系结构的示意图,其用于支持热状况的识别和能效感知热管理算法的应用;

[0018] 图6是描绘了用于异步片上系统中的能效感知热管理的方法的实施例的逻辑流程图;

[0019] 图7是描绘了用于经由工作负载重新分配的同步片上系统中的能效感知热管理的方法700的实施例的逻辑流程图;

[0020] 图8是描绘了用于经由分配排队的工作负载的同步片上系统中的能效感知热管理的方法的实施例的逻辑流程图;

[0021] 图9是描绘了用于经由功率模式调整的同步片上系统中的能效感知热管理的方法的实施例的逻辑流程图;

[0022] 图10是描绘了用于经由功率模式工作循环控制的同步片上系统中的能效感知热管理的方法的实施例的逻辑流程图;

[0023] 图11是描绘了用于处理部件能效评级的运行时验证的方法的实施例的逻辑流程图。

具体实施方式

[0024] 本文所使用的“示例性”一词意味着“用作例子、例证或说明”。本文中描述为“示例性”的任何方面不一定被解释为是排他性的、比其它方面更优选或更具优势。

[0025] 在本说明书中,术语“应用”还可以包括具有可执行内容的文件,例如:目标代码、脚本、字节码、标记语言文件和补丁。此外,本文所引用的“应用”还可以包括:在性质上不可执行的文件,例如,需要被打开的文档或者需要进行访问的其它数据文件。

[0026] 如本说明书中所使用的,术语“部件”、“数据库”、“模块”、“系统”、“热能产生部件”、“处理部件”、“热侵害方”、“处理引擎”等等旨在指代与计算机相关的实体,无论是硬件、固件、硬件和软件的结合、软件或运行中的软件。例如,部件可以是,但不限于是:在处理器上运行的过程、处理器、对象、可执行文件、执行的线程、程序和/或计算机。举例而言,在计算设备上运行的应用和该计算设备都可以是部件。一个或多个部件可以存在于过程和/或执行线程中,并且部件可以位于一个计算机上和/或分布在两个或更多计算机之间。此外,这些部件能够根据其上存储有各种数据结构的各种计算机可读介质来执行。这些部件可以通过本地和/或远程过程的方式进行通信,诸如根据具有一个或多个数据分组的信号(例如,来自一个部件的数据,该部件与本地系统、分布式系统中的另一个部件进行交互和/或以信号的方式跨诸如互联网之类的网络与其它系统进行交互)。

[0027] 在本说明书中,术语“中央处理单元(“CPU”)”、“数字信号处理器(“DSP”)”和“芯片”是可以位于PCD之内的处理部件的非限制性示例,它们可互换地使用,除非另外指出。此外,如本说明书中所区分的,CPU、DSP或芯片可以包括本文通常称为“内核”和“子内核”的一个或多个不同的处理部件。

[0028] 在本说明书中,“异构部件”包括其设计意图不同的部件,以及具有同质的设计方案(相同方式进行设计),但由于生产变化、操作期间的温度和该部件在硅管芯上的位置而具有不同的电特性的部件。本领域普通技术人员应当理解,即使在处理部件的设计方案是同质的情况下,SOC上的每一个处理部件的电特性也将由于以下因素中的一种或多种而发生变化(彼此之间不同):硅泄漏生产变化、切换速率生产变化、各个部件在操作期间的动态温度改变、以及该部件在硅管芯上的位置。因此,本领域普通技术人员应当认识到,从功率和性能的角度来看,SOC上的部件可能不是完全同质的和相同的。

[0029] 在本说明书中,应当理解的是,可以结合能够产生或者消耗能量的设备或部件来使用术语“热量的”和“热能”,其中该能量可以以“温度”为单位进行测量。因此,还应当理解的是,参照某种标准值,术语“温度”预期任何测量,其可以指示产生“热能”的设备或者部件的相对温度或者没有热度的情况。例如,当两个部件处于“热”平衡时,这两个部件的“温度”是相同的。

[0030] 在本说明书中,术语“工作负载”、“过程负载”、“过程工作负载”和“代码块”可互换地使用,并通常针对与给定的实施例中的给定处理部件相关联的处理负担或者处理负担百分比,或者可以分配给给定实施例中的给定处理部件的处理负担或者处理负担百分比。除了上面所定义的之外,“处理部件”或“热能产生部件”或“热侵害方”可以是,但不限于是:中央处理单元、图形处理单元、内核、主内核、子内核、处理区域、硬件引擎等等、或者位于便携式计算设备中的集成电路之内或者之外的任何部件。此外,在某种程度上术语“热负载”、“热分布”、“热特征”、“热处理负载”等等指示在处理部件上运行的工作负担,本领域普通技

术人员应当认识到,本公开内容中对于这些“热”术语的使用与过程负载分布、工作负载负担和功耗有关。

[0031] 在本说明书中,术语“热缓解技术”、“热策略”、“热管理”和“热缓解措施”可互换地使用。

[0032] 本领域普通技术人员应当认识到,术语“DMIPS”表示为了每秒处理给定数量的数以百万计的指令,所需要的Dhrystone迭代的数量。在本说明书中,将该术语使用成测量的通用单位,以指示该示例性实施例中的处理器性能的相对水平,并且其不应被解释为用于建议:落入本公开内容的保护范围之内的任何给定实施例必须包括或者必须不包括:具有任何特定的 Dhrystone评级的处理器。

[0033] 在本说明书中,术语“便携式计算设备”(“PCD”)用于描述在有限容量的电源电压和时钟发生器频率(例如,电池)上操作的任何设备。尽管电池供电的PCD已经使用了数十年,但可再充电电池的技术进步加上第三代(“3G”)和第四代(“4G”)无线技术的出现,已经实现了具有多种能力的众多PCD。因此,PCD可以是蜂窝电话、卫星电话、寻呼机、PDA、智能电话、导航设备、智能本或阅读器、媒体播放器、前述设备的组合、具有无线连接的膝上型计算机等等。

[0034] 对于在具有异构处理部件的PCD中为了QoS优化来管理处理性能而言,可以通过充分利用各个处理引擎的不同性能特性来完成。即使在处理部件在设计上是同质的情况下,SOC上的每一个处理部件的电特性也可能由于任意数量的因素而发生变化(彼此之间不同),其中这些因素包括但不限于:硅泄漏生产变化、切换速率生产变化、各个部件在操作期间的动态温度改变、以及该部件在硅管芯上的位置。因此,本领域普通技术人员应当认识到,从功率和性能的角度来看,SOC上的部件可能不是完全同质的和相同的。因此,在本公开内容中,应当理解的是,对于“异构部件”的引用还意味着:具有同质的设计方案(相同方式进行设计),但由于生产变化、操作期间的温度和该部件在硅管芯上的位置而具有不同的电特性的部件。关于可以在异构处理部件中包括的各种处理引擎的不同性能特性而言,本领域普通技术人员应当认识到,性能差异可能归因于任意数量的原因,其包括但不限于:不同水平的硅、设计方案变化等等。此外,本领域普通技术人员应当认识到,与任何给定的处理部件相关联的性能特性,可以与以下内容相关地发生变化:该处理部件的操作温度、提供给该处理部件的功率等等。

[0035] 例如,考虑包括多个不同的处理内核的示例性异构多核处理器,通常的性能能力范围是从低到高(值得注意的是,本领域普通技术人员应当认识到,还可以考虑包括多个不同的处理部件的示例性异构多处理器片上系统(“SoC”),其中每一个处理部件包含一个或多个内核)。如本领域普通技术人员所应当理解的,与具有相对较高性能能力的处理内核相比,异构处理器中的低性能到中等性能处理内核将在给定的工作负载能力时呈现更低的功率泄漏率,并由此更低速率的热能产生。与更低能力内核相比,更高能力内核能够在更短的时间量内处理给定的工作负载。类似地,处理速度已经被衰弱的高能力内核,与其在全速的未校验的能力时进行的处理相比,可以在给定的工作负载能力时呈现更低的功率泄漏率,并由此更低速率的热能产生。

[0036] 即使如此,根据这些内核进行操作时的热状况,与高性能内核相比,在对给定的工作负载进行处理时,更低性能内核可能是更高效或更低效(就功耗而言)。此外,“最热”内核

还可以是在任何给定时间都是最低能效内核,但并不一定所有场景都是这种情形。因此,为了牺牲处理能力来实现最大的热能缓解回报,与相对温度相对地,能效感知热管理解决方案的实施例鉴于各个处理部件当中的相对处理效率,来做出热管理决定。

[0037] 值得注意的是,可以将给定处理部件的处理效率视作为通过操作频率/ 功耗来表示的功率效率比。或者,可以通过已知的工作负载(例如,DMIPS) 来表示功率效率,可以通过已知的工作负载/功耗来表示该效率比率。一旦确定了各个处理部件的处理效率,则如本领域所通常理解的,可以利用动态控制和电压调节(“DCVS”) 算法来调整向最低能效处理部件提供的功率频率,使得在对于由该SoC可以处理的整体工作负载量具有最小影响的情况下,缓解热能产生。

[0038] 通过考虑异构处理器中的不同内核的各个性能特性(或者性能特性(例如,但不限于电流消耗)的指示符),其中,可以使用这些性能特性来推断由给定内核在给定的操作温度为了处理给定的工作负载而消耗的功率,能效感知热管理算法可以指示将最低能效内核(并不一定是“最热”内核)“调低(dialed down)”,以便在对于整体QoS具有最小影响的情况下,减少热能产生。类似地,响应于减少热能产生的需求,能效感知热管理算法可以使得将活动工作负载从不太高效的内核重新分配给更高效的内核,或者指示将排队的工作负载分配给具有可用能力的更高效内核。值得注意的是,该解决方案的实施例并不是简单地寻求:防止工作负载运行在不太能量高效的内核上。也就是说,在一些实施例中,当工作负载启动时,系统可以考虑每一个部件的能效,并将该工作负载置于该工作负载适合的最高效的CPU上。例如,如果最高效内核已经被过重地使用,那么可以选择下一个最高效内核。用这些方式和其它方式,能效感知热管理解决方案的实施例可以对PCD中的热能产生进行管理,同时优化由用户所体验的整体QoS水平。

[0039] 举一个非限制性例子,可能超过PCD中的监测的热门限,从而触发热报警。热门限可以与以下内容相关联,但并不限于与以下内容相关联:PCD的“皮肤”温度、层叠封装(“PoP”)存储器设备的温度、内核的结点温度、电源和时钟发生器能力、使用情形场景等等。认识到已超过热门限,用于促进能效感知热管理策略的效率管理器模块可以寻求减少一个或多个处理部件的功耗。有利的是,通过减少功耗,可以缓解热能产生,并清除热报警。在已经清除热报警之后,能效感知热管理解决方案的某些实施例可以准许增加不太高效的处理部件的电源电压和时钟发生器频率,其中该不太高效的处理部件是先前减少电源电压和时钟发生器频率的接收者。类似地,在已经清除热报警之后,能效感知热管理解决方案的某些实施例可以准许不太高效的处理部件返回到活动功率模式,其中该不太高效的处理部件是先前转换功率模式的接收者。

[0040] 效率管理器模块可以查询与各个处理部件相关联的性能数据,或者接收指示处理器性能的测量,并且确定活动的和热侵害处理部件中的哪一个或多个在对工作负载进行处理时是最低能效的。也就是说,效率管理器模块可以确定哪些处理部件消耗要被处理的已知工作负载的大部分功率。基于该确定,效率管理器模块可以随后使提供给该最低能效处理部件的功率被减小,从而缓解该多个处理部件的整体热能产生,而不会不必要地牺牲在消耗每一毫瓦(“mW”)功率时所处理的工作负载的平均量。用此方式,效率管理器模块可以优化QoS,与此同时满足减少热能产生的需求。

[0041] 再举一个非限制性例子,一个特定的代码块可以由示例性PCD中的中央处理单元

（“CPU”）或者图形处理单元（“GPU”）中的任意一个来处理。例如，可以分配该特定的代码块由CPU进行处理。但是，该解决方案的示例性实施例的效率管理器模块可能确定GPU处于对该代码块进行更高效处理的地位，响应于该确定，使得该代码块从CPU被重新分配给GPU。用此方式，可以使处理该代码块所需的能量的量最小化，结果，使该SoC的整体热能产生最小化。

[0042] 再举一个非限制性例子，一个特定的代码块可以由示例性PCD中的中央处理单元（“CPU”）或者图形处理单元（“GPU”）中的任意一个来处理。有利的是，替代预先地确定该特定的代码块将由CPU或者GPU中的一个进行处理，一种示例性实施例可以在出现对该代码块进行处理的需要时，选择向这些处理部件中的哪一个处理部件分配对该代码块进行处理的任任务。也就是说，可以对CPU和GPU的性能曲线的“快照”进行比较，以便向最适合对该代码块进行高效处理的处理器分配该工作负载。值得注意的是，应当理解的是，随着该代码块退出调度队列，可以以实时地或者近乎实时的方式，进行用于后续工作负载的分配的后续处理器选择。用此方式，效率管理模块可以充分利用与异构处理器中的各个内核相关联的操作温度，以便通过在工作负载分配之前才选择处理内核，来优化QoS。

[0043] 图1A是示出在不同的热条件下进行操作的示例性处理部件的一对性能曲线（内核85°C、内核50°C）的图300。该处理部件可以是异构多核处理器中的内核，其可以是高效率、中等效率或者低效率内核。具体而言，如本领域普通技术人员所应当确认的，该处理部件可以是能够对给定的代码块进行处理的任何处理引擎，其包括但不限于：CPU、GPU、DSP、可编程阵列、视频编码器/解码器、系统总线、照相机子系统（图像处理）、MDP等等。此外，如上所述，该示例性处理引擎可以是CPU、GPU等等中的内核或者子内核。值得注意的是，可以对能效进行定义，以指示处理部件在特定的功耗水平时的处理性能或者速度。例如，能效可以通过 MIPS/mW（每mW功耗每秒多少百万指令）或者MHz/mW（每mW功耗多少兆赫兹操作时钟频率）来表示。

[0044] 如通过图1A示图可以观察到的，在3500MIPS的工作负载处，在50°C环境下进行操作的示例性内核消耗近似620mW的功率（点315），但在相同的3500MIPS工作负载处，当操作环境达到85°C时，内核的功耗增加到几乎1000mW的功率（点310）。因此，当该示例性处理部件操作在50°C温度时，该示例性处理部件的效率更佳，这是由于与操作在85°C时的3.5 MIPS/mW相比，其可以近似5.6MIPS/mW地进行处理。此外，对于给定的操作温度而言，随着工作负载的增加，内核的处理效率降低。例如，参见内核50°C曲线，当工作负载从3500MIPS增加到近似4300MIPS时，功耗增加到几乎1000mW（点305）。

[0045] 可以从图1A示图中观察到，对于给定的处理部件而言，从功耗的角度来说，随着操作温度上升，该处理部件的效率下降（即，随着处理部件的操作温度增加，其在给定的操作频率处能够处理的MIPS的数量将下降）。值得注意的是，本领域普通技术人员应当认识到，示例性处理部件的操作温度的上升可能是由于任意数量的因素或者因素组合所造成的，其中这些因素包括但不限于：与更高的时钟速度相关联的该处理部件中的功率泄漏增加、与该处理部件相邻的热侵害方、与该处理部件相邻的故障部件、周围环境的改变等等。此外，本领域普通技术人员应当认识到，作为与功耗的增加相关联的功率泄漏率增加的结果，处理部件的工作负载增加可能使得在工作负载分配的时候导致与该处理部件相关联的操作温度上升。不管处理部件的操作温度上升或者下降的原因是什么，重要的是应当注意，根据

图1A示图,通常,随着操作温度的增加,给定处理部件的处理效率相反地发生下降。

[0046] 现转到图1B,描绘了示出了在不同的热条件下进行操作的两个示例性处理部件(“低性能”CPU处理部件和“高性能”GPU处理部件)中的每一个处理部件的一对性能曲线(GPU 105°C、GPU 95°C;CPU 105°C、CPU 95°C)的图400。本质上,在图1B中,图400描绘了用于两个不同的示例性处理部件的性能曲线,其每一个通过图1A示图进行了表示。此外,本领域普通技术人员应当认识到,通过图2中的该性能曲线对来表示的两个示例性处理器GPU、CPU,可以包含在共同的异构多处理器片上系统(“SoC”)上。

[0047] 值得注意的是,通过叠加示例性引擎GPU、CPU的性能曲线,可以观察到,在各个曲线的相交处,定义了各种转换或者交叉点405、410、415。这些交叉点表示门限,当高于或者低于这些门限时,不同的引擎最高效。

[0048] 例如,当处理器GPU、CPU中的每一个都操作在95°C时,示例性GPU、CPU处理器性能曲线的对比分析,可以确定这两个处理器GPU、CPU在近似3700DMIPS的工作负载处(点410)的处理效率是基本等同的。但是,还可以通过该对比分析观察到,在低于点410时,CPU处理部件是更高效的,即,当工作负载低于3700DMIPS时,针对每一DMIPS的工作负载,CPU处理部件消耗更少的功率。相反,在高于点410时,GPU内核是更高效的,即,当工作负载超过3700DMIPS时,针对每一DMIPS的工作负载,GPU内核消耗更少的功率。

[0049] 因此,依赖于CPU运行于105°C和GPU运行于更低的95°C时的示例性对比分析,应用能效感知热管理策略的效率管理器模块,响应于对减少整体热能产生的触发,可以针对工作负载低于点405,指示减少针对不太高效的GPU的功率(即使CPU的温度更高)。

[0050] 此外,应当理解的是,由于任意数量的因素,异构多处理器SoC中的不同处理器和/或内核可以在不同的热状况下进行操作。例如,在图1B示图中,转换点405表示:操作在105°C的示例性CPU处理部件和操作在95°C的GPU处理部件的性能曲线的相交。因此,通过认识到这些示例性处理器操作在不同的温度,一个实施例可以在工作负载分配之前,才利用对比分析来确定这些处理器中的哪个是最适合的,从而对准备好用于处理的给定代码块进行高效地处理(其类似于上面所描述的示例性场景)。例如,可以将低于2400DMIPS的工作负载分配给CPU处理部件,将高于2400 DMIPS的工作负载分配给GPU处理部件,以便确保在最高效状况下,对工作负载进行处理。此外,可以预期的是,能效感知热管理解决方案的实施例可以在处理部件当中和之间重新分配工作负载,使得对集体处理引擎的整体平均效率进行优化。为此,某些实施例并不是简单地寻求:防止工作负载运行在不太能量高效的内核上。也就是说,在一些实施例中,当工作负载启动时,系统可以考虑每一个部件的能效,并将该工作负载置于该工作负载适合的最高效的CPU上。例如,如果最高效的内核已经被过重地使用,那么可以选择下一个最高效内核。

[0051] 值得注意的是,可以预期的是,在分配下一个代码块的事件中,可以执行能效感知热管理算法的某些实施例来优化整体工作负载处理效率。例如,返回参照图1B示图的示例性曲线GPU 95°C和CPU 105°C,假定与这些曲线相关联的处理器GPU、CPU中的每一个当前正在按照2000DMIPS的速率进行处理,效率管理模块正在寻求确定这两个示例性处理器中的哪一个最适合用于对额外的1000DMIPS的工作负载进行高效处理。与当前正在处理的2000DMIPS工作负载相比,能效感知热管理算法可以用于基于假定的每一处理部件GPU、CPU的3000MIPS的总工作负载(先前分配的每一引擎的2000MIPS加上要分配给这些引擎之一的

额外的1000MIPS), 来比较这些曲线。对于该非限制性例子而言, 基于图2示图的示例性曲线 GPU 95°C和CPU 105°C, 热感知调度模块可以选择更高效的GPU, 与按照3000 DMIPS进行处理将消耗超过500mW的功率的CPU相比, GPU在处于相同的工作负载时, 将消耗低于400mW的功率。

[0052] 对上面的例子进行扩展, 在向GPU分配额外的1000DMIPS之后, 效率管理模块可以移动到将运行在CPU上的2000DMIPS的工作负载重新分配给GPU, 从而进一步将GPU的工作负载从3000DMIPS增加到5000 DMIPS。有利的是, 与没有重新分配CPU的2000DMIPS工作负载时其消耗的大约8mW/DMIPS相比, 在5000DMIPS时, GPU将消耗1000mW的功率或者5mW/DMIPS来处理该工作负载。此外, 随着在该例子中将工作负载从CPU中完全移走, 可以预期的是, 效率管理器模块可以将CPU转换到保持状态或者甚至功率下跌 (collapsed) 状态, 从而进一步节省能量和缓解热能产生。

[0053] 能效感知热管理算法的其它实施例可以用于基于如果分配额外的工作负载的曲线的预测偏移, 来比较性能曲线。例如, 返回参见分别在95°C和 105°C的操作温度、均按照2000DMIPS的速率进行处理的处理器GPU和 CPU的例子, 效率管理模块的实施例可以预测源自于分配额外的1000 DMIPS工作负载而造成的性能曲线的偏移。值得注意的是, 由于额外的1000 DMIPS工作负载可能造成其被分配到的处理部件消耗更多的功率, 因此效率管理模块可以认为: 由于该额外的工作负载的结果, 目前与该处理部件相关联的操作温度将上升, 因此, 寻求将与所预测的温度上升相关联的性能曲线进行比较。

[0054] 返回到该例子, 额外的1000DMIPS的工作负载可能将造成GPU的操作温度从95°C增加到100°C并且, 类似地, CPU的操作温度从105°C增加到110°C。因此, 效率管理模块的实施例可以对与分别在100°C和110°C的预测温度处进行操作的内核GPU和CPU相关联的性能数据进行查询和比较 (图1B中没有示出GPU 100°C和CPU 110°C性能曲线)。

[0055] 图1C是示出了示例性内核对 (内核1和内核2) 的一对性能曲线的图500。相对于可被视作为“更慢速”内核的内核1, 可以将内核2视作为“更快速”内核。值得注意的是, 本领域普通技术人员应当认识到, 内核2作为该示例性对中的更快速内核, 这是由于内核2能够按照与内核1 (近似 2100MHz) 相比更高的最大频率 (近似2500MHz) 进行处理。因此, 由于操作频率与MIPS相关, 因此本领域普通技术人员还应当认识到, 内核2 能够比内核1处理更多的MIPS。

[0056] 图1C示图中的点510表示一个频率 (~1600MHz), 其中高于该频率时, 与内核1相比, 内核2在对一个MIPS的工作负载进行处理方面更为高效 [在2000MHz时, 内核2只消耗 ~800mW的功率 (点515), 而在相同的 2000MHz操作频率的情况下, 内核1消耗 ~1100mW的功率 (点520)]。但是, 值得注意的是, 在低于点510时, 示例性内核1是二者中的更高效处理器。因此, 如果内核1和内核2均在低于1600MHz的频率处运行时, 效率管理模块识别用于减少热能产生的触发 (例如, 诸如用于超过皮肤温度门限的热报警), 效率管理模块可以寻求减少向内核2提供的频率, 而不管在该触发的时候, 内核2是否比内核1“更热”。用此方式, 可以缓解热能产生, 同时对处理给定MIPS的工作负载的整体效率进行优化。

[0057] 图1D是示出了在图1C示图中描绘的示例性内核对 (内核1和内核2) 的一对不同的性能曲线的图600。在图1D中, 这些性能曲线中的每一个描绘: 该内核的能效对比向该内核提供的频率。值得注意的是, 图1D示图中的点610和615分别与图1C示图中的点510和515/

520相关。类似地,在图1D示图中描绘的最大操作频率与图1C示图中描绘的最大操作频率相关。

[0058] 使用由图1D示图所表示的性能数据,如果两个内核均运行在它们的最大操作频率或者附近,则能效感知热管理解决方案的实施例可以通过首先减少提供给内核1的频率,对热触发进行响应。值得注意的是,在2000MHz 的最大操作频率处,与内核2相比,消耗的每一mW的功率,内核1处理更少MIPS的工作负载。因此,示例性能能效感知热管理解决方案可以将内核 1的频率减少一个步进,以尝试缓解热能产生和清除热报警。如果不太高效的内核1的频率减少并没有清除该报警,则该示例性能能效感知热管理解决方案可以在第一步进减少之后,重新评估这些内核中的哪一个内核是不太高效的,并随后向最低能效内核应用频率的第二步进减小。可以以这种一步一步的方式来继续该过程,在该步进减少的时间处,减少向最低能效内核提供的频率,直到清除热报警或者以其它方式实现热缓解目标为止。

[0059] 图2A是用于描绘包括异构处理部件的片上系统102A中的异步架构的方面的功能框图。能效感知热管理解决方案的某些实施例可以适用于在不会不必要地使工作负载处理效率劣化的情况下,对这些处理部件的热能产生进行管理。

[0060] 将片上系统102A描述为示出了一系列的处理部件PC 0、PC 1、PC 2 等等。值得注意的是,如本领域普通技术人员所理解的,由于片上系统102A 的架构是异步的,因此这些处理部件中的每一个处理部件与用于控制电源电压和时钟发生器频率的专用时钟源(例如,锁相环(“PLL”))相关联。在该示图中,时钟0唯一地与针对PC 0的电源和时钟发生器相关联。时钟 1唯一地与针对PC 1的电源和时钟发生器相关联。时钟2唯一地与针对PC 2的电源和时钟发生器相关联等等。

[0061] 有利的是,由于异步片上系统中的每一个处理部件都具有专用的时钟源,因此当热能产生已经超过门限时,能效感知热管理解决方案的实施例可以使用DCVS模块,做出关于最低能效处理部件的目标功率减少。

[0062] 图2B是用于描绘包括异构处理部件的片上系统102B中的同步架构的方面的功能框图。能效感知热管理解决方案的某些实施例可以适用于在不会不必要地使工作负载处理效率劣化的情况下,对这些处理部件的热能产生进行管理。

[0063] 将片上系统102B描述为示出了一系列的处理部件PC 0、PC 1、PC 2 等等。值得注意的是,由于片上系统102B的架构是同步的,因此这些处理部件中的每一个处理部件与用于所有处理部件的单个的、共同的时钟源和电源相关联。有利的是,由于同步片上系统中的每一个处理部件共享单个时钟源,因此当热能产生已经超过门限时,能效感知热管理解决方案的实施例可以通过将工作负载从不太高效的处理部件分配或者重新分配到更高效的处理部件,来优化处理效率。

[0064] 值得注意的是,由于不太高效的处理部件不具有工作负载,能效感知热管理解决方案的实施例可以例如指示该不太高效的处理部件的功率状态从活动状态转换到保持状态,或者从保持状态转换到功率下跌(collapsed) 状态。有利的是,通过向更高效的处理器分配新的工作负载,和/或将活动工作负载从不太高效的处理器重新分配给更高效的处理器,这些解决方案的实施例可以优化用于处理给定的工作负载所需要的功率量。此外,通过将同步SoC 102B中的不太高效处理部件从活动状态转换到空闲状态,鉴于在更高效的处理部件不能够维持可接受的QoS的情况下用于将不太高效的处理部件带回在线的时延参数,

能效感知热管理解决方案的实施例还可以对整体功耗效率进行优化。

[0065] 图3是描绘了用于便携式计算设备(“PCD”)100中的能效感知热管理的片上系统102的实施例的功能框图。值得注意的是,可以预期的是,片上系统102在体系结构上可以是同步的,也可以是异步的。如上面结合图1 示图所解释的,电源电压和时钟发生器频率的目标减少和/或跨处理部件的工作负载分配,可以是基于与各个内核或处理器222、224、226、228唯一关联的性能数据的对比分析。值得注意的是,如本领域普通技术人员所应当认识的,仅仅出于说明目的,将处理部件110描述成一组异构处理引擎,其可以表示具有多个、异构的内核222、224、226、228的单个处理部件或者多个异构的处理器222、224、226、228,这些处理器中的每一个处理器可以包括多个内核和/或子内核,也可以不包括多个内核和/或子内核。因此,本文将处理引擎222、224、226和228引用成“内核”,在性质上应被理解成示例性的,而不是限制本公开内容的保护范围。

[0066] 片上系统102可以通过监测模块114对分别与内核222、224、226、228 相关联的温度传感器157进行监测,监测模块114与效率管理器(“EM”) 模块101、DCVS模块26和调度器模块207进行通信。此外,监测模块114 还可以监测任意数量的超过门限的热能指示符,例如但不限于:皮肤温度传感器、PoP存储器温度传感器、结点温度传感器、针对处理部件的电源轨的电流传感器、与电源相关联的电流传感器、电源容量传感器等等。

[0067] 在超过了热门限并且被监测模块114识别到的情况下,则可以触发EM 模块101进行测量,以便以能效感知方式来缓解热能产生。EM模块101可以从监测模块114接收与这些处理部件的能效相关联的一个或多个监测参数的指示,并随后使用这些指示来确定这些处理部件中的哪一个是最能效的。在一些实施例中,EM模块101可以从监测模块114接收温度测量,并使用这些测量来从内核性能数据存贮24中查询性能数据。基于该性能数据,EM模块101可以根据工作负载处理效率,来确定内核222、224、226、228的排序。

[0068] 随后,在异步系统102A中,EM模块101可以确定减少针对不太高效内核的电源电压和时钟发生器频率,或者在同步系统102B中,EM模块101 可以使得将工作负载从不太高效的内核重新分配到更高效的内核,或者将排队的工作负载调度到更高效的内核。由EM模块101所指示的动态DCVS 调整策略可以将关于不太高效的处理部件的处理器时钟速率设置在减少的水平,将某些不太高效处理器的功率状态从活动状态转换成空闲状态等等。在一些实施例中,可以经由针对调度器207的指令来实现由EM模块101 所指示的工作负载分配和/或重新分配。值得注意的是,通过对能效感知热管理策略的应用,EM模块101可以以QoS为代价,减少或者缓解过多的功耗。

[0069] 如本领域普通技术人员所应当认识的,处理内核222、224、226、228 中的一个或多个的操作温度,可能随着对工作负载进行处理,周围状况的改变,相邻热能发生器散发能量等等而发生波动。因此,随着各个处理内核222、224、226、228的操作温度发生波动,与这些引擎222、224、226、228相关联的有关性能数据也会发生波动。随着与内核222、224、226、228 中的每一个相关联的操作温度发生改变,监测模块114认识到该改变,并可向EM模块101发送用于指示该改变的温度数据。测量的操作温度的改变可能触发EM模块101来参考内核性能(“CP”)数据存贮24,以基于测量的操作温度来查询针对这些内核222、224、226、228 中的一个或多个的性能曲线。随后,EM模块101可以将不同的内核222、224、226、228识别成最低能效内核,并调整向其提供的功率频率(经由DCVS模块26),以便对产生的热能进行缓

解,同时维护每消耗一毫瓦功率的工作负载的最高效的处理。EM模块101还可以对所识别的性能曲线进行比较,并选择在该比较的时候,最适合的内核222、224、226、228,以高效处理排队的代码块,或者需要从不太高效内核进行重新分配的代码块。

[0070] 示例性EM模块101被配置为利用与各个不同的处理部件222、224、226、228相关联的一个或多个性能曲线的对比分析,来指示DCVS模块26 调整电源,和/或指示调度器模块207将工作负载分配或者重新分配给最适合于对该工作负载进行高效处理的某个处理部件。值得注意的是,本领域普通技术人员应当认识到,由于处理部件222、224、226、228的操作温度发生改变,由EM模块101所查询和比较的性能曲线也将发生改变。因此,在不同的时间,EM模块101可以选择不同的处理引擎222、224、226、228 来应用能效感知热管理策略。用此方式,某些实施例的优点在于,通过确保将工作负载分配分配给在该分配时间可用的最高效的处理部件,和/或减少最低能效处理部件的功耗、支持允许更高效的处理来处理活动的工作负载,EM模块101在管理热能产生时对QoS进行优化。

[0071] 图4是具有无线电话的形式的PCD 100的示例性、非限制性方面的功能框图,其用于实现对热状况进行监测,对性能数据进行比较,设置最佳功率频率,以及将工作负载调度给最适合于进行高效处理的处理部件的方法和系统。如图所示,PCD 100包括片上系统102,后者包括耦合在一起的异构多核中央处理单元(“CPU”)110和模拟信号处理器126。CPU 110可以包括第零内核222、第一内核224和第N内核230,如本领域普通技术人员所理解的。此外,还可以使用数字信号处理器(“DSP”)来替代CPU 110,如本领域普通技术人员所理解的。此外,如异构多核处理器领域技术人员所理解的,内核222、224、230中的每一个可以在类似的操作条件下,按照不同的效率来处理工作负载。

[0072] 通常,效率管理器模块101可以从监测模块114接收温度数据,并使用该温度数据来查询性能数据,或者推导与内核222、224、230相关联的性能数据,确定内核222、224、230的相对处理效率,并与DCVS模块26 和/或调度器207一起工作,以调整电源、转换功率状态和/或将代码块调度给内核222、224、230。

[0073] 监测模块114与分布在片上系统102各处的多个操作传感器(例如,热传感器157)进行通信,并与PCD 100的CPU 110以及与EM模块101 进行通信。EM模块101可以与监测模块114一起工作,以查询与由监测模块114所监测的温度有关的处理器性能曲线,对这些曲线进行比较,将功率频率设置成最高效水平,以及选择可用的并能够处理代码块的最高效处理器。

[0074] 如图4中所示,显示控制器128和触摸屏控制器130耦合到数字信号处理器110。在片上系统102之外的触摸屏显示器132,耦合到显示控制器 128和触摸屏控制器130。

[0075] 此外,PCD 100还可以包括视频解码器134,例如,逐行倒相(“PAL”) 解码器、顺序与存储彩色电视系统(“SECAM”)解码器、国家电视制式委员会(“NTSC”)解码器或者任何其它类型的视频解码器134。视频解码器 134耦合到多核中央处理单元(“CPU”)110。视频放大器136耦合到视频解码器134和触摸屏显示器132。视频端口138耦合到视频放大器136。如图4中所示,通用串行总线(“USB”)控制器140耦合到CPU 110。此外,USB端口142耦合到USB控制器140。存储器112和用户识别模块(SIM)卡146还可以耦合到CPU 110。此外,如图4中所示,数码相机148可以耦合到CPU 110。在一个示例性方面,数码相机148是电荷耦合器件(“CCD”)相机或者互补金属氧化半导体(“CMOS”)相机。

[0076] 如图4中所进一步示出的,立体声音频CODEC 150可以耦合到模拟信号处理器126。此外,音频放大器152可以耦合到立体声音频CODEC 150。在一个示例性方面,第一立体声扬声器154和第二立体声扬声器156耦合到音频放大器152。图4示出了还可以耦合到立体声音频CODEC 150的麦克风放大器158。另外,麦克风160可以耦合到麦克风放大器158。在一个特定的方面,调频(“FM”)无线调谐器162可以耦合到立体声音频CODEC 150。此外,FM天线164耦合到FM无线调谐器162。此外,立体声耳机166可以耦合到立体声音频CODEC 150。

[0077] 此外,图4还指示射频(“RF”)收发机168可以耦合到模拟信号处理器126。RF开关170可以耦合到RF收发机168和RF天线172。如图4中所示,键盘174可以耦合到模拟信号处理器126。此外,具有麦克风的单声道耳机176可以耦合到模拟信号处理器126。此外,振动器设备178可以耦合到模拟信号处理器126。此外,图4还示出了电源180(例如,电池)耦合到片上系统102。在一个特定的方面,该电源包括可再充电DC电池或者DC电源,后者是从连接到AC电源的交流电(“AC”)到DC变换器导出的。

[0078] 此外,CPU 110还可以耦合到一个或多个内部、片上热传感器157A,以及一个或多个外部、片外热传感器157B。片上热传感器157A可以包括一个或多个正比于绝对温度(“PTAT”)的温度传感器,后者是基于纵向PNP结构,并通常专用于互补金属氧化物半导体(“CMOS”)甚大规模集成(“VLSI”)电路。片外热传感器157B可以包括一个或多个热敏电阻。热传感器157可以产生电压下降,利用模数转换器(“ADC”)控制器103将该电压下降转换成数字信号。但是,在不脱离本发明的保护范围的基础上,也可以使用其它类型的热传感器157。

[0079] 除了受到ADC控制器103的控制和监测之外,热传感器157还可以被一个或多个EM模块101来控制 and 监测。EM模块101可以包括由CPU 110进行执行的软件。但是,EM模块101还可以利用硬件和/或固件来形成,而不脱离本发明的保护范围。EM模块101可以负责查询处理器性能数据和/或接收对处理器性能的指示,并基于对该数据的分析,调整最低能效处理器的功率频率,和/或将代码块分配或者重新分配给在工作负载分配时,最能够高效地处理该代码的处理器。

[0080] 返回到图4,触摸屏显示器132、视频端口138、USB端口142、照相机148、第一立体声扬声器154、第二立体声扬声器156、麦克风160、FM天线164、立体声耳机166、RF开关170、RF天线172、键盘174、单声道耳机176、振动器178、热传感器157B和电源180,在片上系统102之外。但是,应当理解的是,监测模块114还可以通过模拟信号处理器126和CPU 110的方式,从这些外部设备中的一个或多个接收一个或多个指示或信号,以帮助实时管理可在PCD 100上操作的资源。

[0081] 在特定的方面,本文所描述的方法步骤中的一个或多个,可以由存储器112中所存储的可执行指令和参数(其形成一个或多个EM模块101)来实现。除了ADC控制器103之外,用于形成EM模块101的这些指令可以由CPU 110、模拟信号处理器126或者其它处理器来执行,以便执行本文所描述的方法。此外,处理器110、126、存储器112、在其中所存储的指令、或者它们的组合,可以用作用于执行本文所描述的方法步骤中的一个或多个的单元。

[0082] 图5A是描绘了用于图4中所示出的芯片102的硬件的示例性空间布局的功能框图。根据该示例性实施例,应用CPU 110位于芯片102的最左侧区域,而调制解调器CPU 168、126位于该芯片102的最右侧区域。应用CPU 110可以包括异构多核处理器,其包括有第零内核

222、第一内核224 和第N内核230。应用CPU 110可以执行EM模块101A(当体现为软件时),或者其可以包括EM模块101A(当体现为硬件时)。此外,还将应用CPU 110 示出为包括操作系统(“O/S”)模块208和监测模块114。

[0083] 应用CPU 110可以耦合到一个或多个锁相环(“PLL”)209A、209B,它们与应用CPU 110在位置上相邻,并位于芯片102的左侧区域。与PLL 209A、209B相邻并在应用CPU 110之下,可以包括模数(“ADC”)控制器103,其中ADC控制器103可以包括与应用CPU 110的主模块101A进行协作操作的其自己的EM模块101B。

[0084] ADC控制器103的EM模块101B可以结合监测模块114,来负责对可以在“片上”102和“片外”102提供的多个热传感器157进行监测和跟踪。片上或内部热传感器157A可以位于各个位置处。

[0085] 举一个非限制性例子,第一内部热传感器157A1可以位于应用CPU 110 和调制解调器CPU 168、126之间的芯片102的顶部中间区域,并与内部存储器112相邻。第二内部热传感器157A2可以位于芯片102的右侧区域上、调制解调器CPU 168、126之下。该第二内部热传感器157A2还可以位于高级精简指令集计算机(“RISC”)指令集机器(“ARM”)177和第一图形处理器135A之间。数模控制器(“DAC”)173可以位于第二内部热传感器 157A2和调制解调器CPU 168、126之间。

[0086] 第三内部热传感器157A3可以位于芯片102的最右区域中的第二图形处理器135B和第三图形处理器135C之间。第四内部热传感器157A4可以位于芯片102的最右区域中,并在第四图形处理器135D之下。此外,第五内部热传感器157A5可以位于芯片102的最左区域中,并与PLL 209和ADC 控制器103相邻。

[0087] 一个或多个外部热传感器157B也可以耦合到ADC控制器103。第一外部热传感器157B1可以位于片外,并与可包括调制解调器CPU 168、126、ARM 177和DAC 173的芯片102的右上四分之一部分相邻。第二外部热传感器157B2可以位于片外,并与可包括第三和第四图形处理器135C、135D 的芯片102的右下四分之一部分相邻。

[0088] 本领域普通技术人员应当认识到,可以在不脱离本发明的保护范围的基础上,提供图5A中所示出的硬件的各种其它空间布局。图5A示出了一种示例性空间布局,以及主EM模块101A和具有其EM模块101B的ADC 控制器103如何与监测模块114一起工作,以认识取决于图5A中所示出的示例性空间布局的热状况,对处理效率数据进行比较,并分配工作负载或者调整电源以管理热状况,而不会不必要地影响QoS。

[0089] 图5B是示出图4和图5A的PCD 100的示例性软件体系结构200的示意图,其用于支持对热状况的识别和能效感知热管理算法的应用。任意数量的算法可以形成在满足某些热状况时,由EM模块101可以应用的至少一个能效感知热管理技术,或者这些算法可以是该至少一个能效感知热管理技术的一部分。

[0090] 如图5B中所示,CPU或数字信号处理器110经由总线211耦合到存储器112。如上所述,CPU 110是具有N个内核处理器的多核异构处理器。也就是说,CPU 110包括第一内核222、第二内核224和第N内核230。如本领域普通技术人员所知道的,第一内核222、第二内核224和第N内核230 中的每一个都可用于支持专用的应用或程序,并且作为异构处理器的一部分,可以在类似的热操作条件下,提供不同水平的性能。替代地,一个或多个应用或程序可以是分布式的,用于跨这些可用的异构内核中的两个或更多内核来进行处理。

[0091] CPU 110可以从EM模块101(其可以包括软件和/或硬件)接收命令。如果将EM模块101实现成软件,则EM模块101包括由CPU 110执行的指令,CPU 110向由CPU 110和其它处理器执行的其它应用程序发出命令。

[0092] 可以将CPU 110的第一内核222、第二内核224到第N内核230集成在单个集成电路芯片上,或者可以将它们集成或耦合在多电路封装中的分别的管芯上。设计人员可以经由一个或多个共享高速缓存,来耦合第一内核222、第二内核224到第N内核230,并且设计人员可以经由诸如总线、环状、网状和交叉拓扑之类的网络拓扑来实现消息或指令的传送。

[0093] 总线211可以包括经由一个或多个有线或无线连接的多个通信路径,如本领域所已知的。总线211可以具有用于实现通信的额外元件(例如,控制器、缓存器(高速缓存)、驱动器、中继器和接收机),但为了简单起见,省略了这些元件。此外,总线211可以包括地址、控制和/或数据连接,以便在所述的部件当中实现适当的通信。

[0094] 当PCD 100使用的逻辑利用软件来实现时,如图5B中所示,则应当注意的是,可以将以下各项中的一项或多项存储在任何计算机可读介质上,以便由任何与计算机相关的系统或方法使用,或者结合任何与计算机相关的系统或方法来使用:启动逻辑250、管理逻辑260、能效感知热管理接口逻辑270、应用存储280中的应用、以及部分文件系统290。

[0095] 在本文档的上下文中,计算机可读介质是能够包含或存储计算机程序和数据,以便由计算机相关系统或方法使用或者结合计算机相关系统或方法来使用的电、磁、光或其它物理器件或单元。各种逻辑单元和数据存储可以用任何计算机可读介质来体现,以便由指令执行系统、装置或设备使用,或者结合该指令执行系统、装置或设备进行使用,该指令执行系统、装置或设备例如是基于计算机的系统、包含处理器的系统、或者是可以从该指令执行系统、装置或设备获取指令并执行这些指令的其它系统。在本文档的上下文中,“计算机可读介质”可以是能够存储、传输、传播或者传送程序,以便由指令执行系统、装置或设备使用或者结合该指令执行系统、装置或设备进行使用的任何单元。

[0096] 计算机可读介质可以是例如,但不限于:电、磁、光、电磁、红外或半导体系统、装置、设备或传播介质。计算机可读介质的更具体示例(非详尽列表)包括下面的各项:具有一个或多个电线的电连接(电)、便携式计算机磁盘(磁)、随机存取存储器(RAM)(电)、只读存储器(ROM)(电)、可擦除可编程只读存储器(EPROM、EEPROM或闪存)(电)、光纤(光)和便携式压缩盘只读存储器(CDROM)(光)。应当注意,计算机可读介质甚至可以是纸质的,或者能在其上打印程序的其它适当介质,这是由于例如经由纸介质或其它介质的光扫描,可以电子地捕获该程序,随后以适当的方式进行编译、解释或者以其它方式进行处理(如果需要的话),并随后存储在计算机存储器中。

[0097] 在替代的实施例中,其中利用硬件来实现启动逻辑250、管理逻辑260以及或许的能效感知热管理接口逻辑270中的一个或多个,可以通过下面技术中的任何一种或者组合来实现各种逻辑,其中这些技术中的每一种都是本领域公知的:具有用于根据数据信号来实现逻辑功能的逻辑门的离散逻辑电路、具有适当组合的逻辑门的专用集成电路(ASIC)、可编程门阵列(PGA)、现场可编程门阵列(FPGA)等等。

[0098] 存储器112是诸如闪存或固态存储器器件之类的非易失性数据存储设备。尽管将存储器112描绘成单个设备,但存储器112可以是具有分别耦合到数字信号处理器(或者额外的处理器内核)的数据存储的分布式存储器设备。

[0099] 启动逻辑250包括用于选择性地识别、加载和执行选择的程序的一个或多个可执行指令,以用于能效感知对比分析,以及识别可用内核中的一个或多个内核以应用能效感知热管理策略。

[0100] 管理逻辑260包括用于终止能效感知热管理程序,以及选择性地识别、加载和执行更适当的替换程序的一个或多个可执行指令,以用于能效感知对比分析、针对调整的电源的选择和/或针对可用的内核中的一个或多个内核的工作负载分配。管理逻辑260被布置为在运行时或者当PCD 100加电并被该设备的操作者使用时执行这些功能。可以在嵌入式文件系统290的程序存储296中找到替换程序。

[0101] 当该替换程序被数字信号处理器中的内核处理器里的一个或多个执行时,其可以根据由EM模块101和监测模块114所提供的一个或多个信号进行操作。在该方面,监测模块114可以响应于源自于EM模块101的控制信号,来提供事件、过程、应用、资源状态状况、流逝的时间、温度等等的一个或多个指示符。

[0102] 接口逻辑270包括用于呈现、管理和与外部输入进行交互,以观测、配置或者以其它方式更新在嵌入式文件系统290中存储的信息的一个或多个可执行指令。在一个实施例中,接口逻辑270可以结合经由USB端口142接收的制造商输入进行操作。这些输入可以包括要从程序存储296中删除或者增加到程序存储296中的一个或多个程序。替代地,这些输入可以包括针对程序存储296中的程序里的一个或多个的编辑或改变。此外,这些输入可以识别针对启动逻辑250和管理逻辑260中的一个或二者的一个或多个改变,或者对启动逻辑250和管理逻辑260中的一个或二者的完全替换。举例而言,这些输入可以包括针对管理逻辑260的改变,用于当与皮肤温度相关联的温度测量超过某个识别的门限时,指示PCD 100将工作负载从不太高效的内核重新分配给更高效的内核。举另外的例子,这些输入可以包括针对管理逻辑260的改变,用于当电池电平达到某个底量时,指示PCD 100将针对最低能效处理内核的功率减少一个增量。

[0103] 接口逻辑270使制造商能根据PCD 100上的定义的操作状况,可控地配置和调整终端用户的体验。当存储器112是闪存时,可以对以下各项中的一项或多项进行编辑、替换或者以其它方式进行修改:启动逻辑250、管理逻辑260、接口逻辑270、应用存储280中的应用程序或者嵌入式文件系统290中的信息。在一些实施例中,接口逻辑270可以准许PCD 100的终端用户或者操作者来搜索、定位、修改或者替换启动逻辑250、管理逻辑260、应用存储280中的应用和嵌入式文件系统290中的信息。操作者可以使用得到的接口来进行改变,其中这些改变将在PCD 100的下一次启动时实现。替代地,操作者可以使用得到的接口来进行改变,其中这些改变在运行时间期间实现。

[0104] 嵌入式文件系统290包括分层布置的内核性能数据存储24。在该方面,文件系统290可以包括其总的文件系统能力的保留部分,用于存储与各个内核222、224、226、228在各种操作温度下的性能曲线相关联的信息。

[0105] 图6是描绘了用于异步片上系统中的能效感知热管理的方法600的实施例的逻辑流程图。在图6的实施例中,可以基于由监测模块114所收集的实际性能数据,来实验性地确定针对各个处理内核222、224、226、228中的每一个处理内核的性能曲线,或者在一些实施例中,这些性能曲线可以是由各个内核的性能规格来驱动的先验曲线。

[0106] 在一些实施例中,为了实验性地确定各个处理内核222、224、226、228的性能曲

线,监测模块114可以与温度传感器157以及用于监测内核222、224、226、228的功耗的各种其它电压或电流传感器进行通信。在该实施例中,本领域任何普通技术人员将认识到,由监测模块114所收集的数据可以与先前工作负载分配相耦合,并被编译为实验性性能曲线。可以将这些实验性性能曲线存储在CP数据存贮24中,并由能效感知热管理算法进行使用。

[0107] 开始于方框605,监测模块114可以认识到热事件(例如,温度读数超过预定的温度门限),以作为减少热能产生的触发。如先前所描述的,监测模块114可以向EM模块101提供该热报警信息,以便应用能效感知热管理解决方案。

[0108] 在方框610处,EM模块101可以查询与SoC中的各个异构处理部件相关联的性能数据。可以基于由监测模块114向EM模块101所提供的操作温度,来查询相关的性能数据。使用该性能数据,EM模块101可以基于处理部件对工作负载进行高效处理的相对能力,来确定它们的排序。

[0109] 在方框615处,EM模块101可以将提供给不太高效的处理内核中的一个或多个处理内核的功率的频率减少预定的增量。值得注意的是,本领域任何普通技术人员应当认识到,频率的减少直接与由该处理部件所处理的工作负载的量的减少相关。

[0110] 接着,在判定框620处,EM模块101可以与监测模块114一起工作,以确定是否已经清除了触发方框610和615的热报警。如果已经清除了该报警(即,在方框615处实现的频率的减小,引起热能产生的降低,使得成功地清除该报警),则沿着“是”分支,并且方法600返回。EM模块101可以准许增加向不太高效的处理内核提供的频率。但是,如果作为在方框615处采取的动作的结果,还没有清除该报警,则沿着“否”分支返回到方框610,以及识别“新的”最低能效处理部件以进行功率频率的增量减小。值得注意的是,应当理解,“新的”最低能效处理部件可能是先前被识别成最低能效处理部件的同一个处理部件。继续从方框610到620的循环,逐步地减少向最低能效处理部件提供的功率的频率,直到热报警清除为止。

[0111] 图7是描绘了用于经由工作负载重新分配的同步片上系统中的能效感知热管理的方法700的实施例的逻辑流程图。在图7实施例中,可以基于由监测模块114所收集的实际性能数据,来实验性地确定针对各个处理内核222、224、226、228中的每一个处理内核的性能曲线,或者在一些实施例中,这些性能曲线可以是由各个内核的性能规格来驱动的先验曲线。

[0112] 在一些实施例中,为了实验性地确定各个处理内核222、224、226、228的性能曲线,监测模块114可以与温度传感器157以及用于监测内核222、224、226、228的功耗的各种其它电压或电流传感器进行通信。在该实施例中,本领域任何普通技术人员将认识到,由监测模块114所收集的数据可以与先前工作负载分配相耦合,并被编译为实验性性能曲线。可以将这些实验性性能曲线存储在CP数据存贮24中,并由能效感知热管理算法进行使用。

[0113] 开始于方框705,监测模块114可以认识到热事件(例如,温度读数超过预定的温度门限),以作为减少热能产生的触发。如先前所描述的,监测模块114可以向EM模块101提供该热报警信息,以便应用能效感知热管理解决方案。

[0114] 在方框710处,EM模块101可以查询与SoC中的各个异构处理部件相关联的性能数据。可以基于由监测模块114向EM模块101所提供的操作温度,来查询相关的性能数据。使用该性能数据,EM模块101可以基于处理部件对工作负载进行高效处理的相对能力,来确定它

们的排序。

[0115] 在方框715处,EM模块101可以将由不太高效的处理器进行处理的¹活动工作负载重新分配给更高效的处理器。值得注意的是,本领域任何普通技术人员应当认识到,将工作负载从不太高效的处理器移动到更高效的处理器,会引起以减少的功率量来处理该工作负载,即使该不太高效的处理器和更高效的处理器共享共同的电源和时钟发生器。

[0116] 接着,在判定框720处,EM模块101可以与监测模块114一起工作,以确定是否已经清除了触发方框710和715的热报警。如果已经清除了该报警(即,将工作负载从不太高效的处理器重新分配给更高效的处理器,引起热能产生的降低,使得成功地清除该报警),则沿着“是”分支,并且方法700返回。EM模块101可以准许向不太高效的²处理内核进行工作负载的未来分配或者重新分配。但是,如果作为在方框715处采取的动作的结果,还没有清除该报警,则沿着“否”分支返回到方框710,以及识别“新的”最低能效处理部件以进行工作负载减少。值得注意的是,应当理解,“新的”最低能效处理部件可能是先前被识别成最低能效处理部件的同一个处理部件。继续从方框710到720的循环,将工作负载从最低能效处理部件中移走,直到热报警清除为止。

[0117] 图8是描绘了用于经由分配排队的工作负载的同步片上系统中的能效感知热管理的方法800的实施例的逻辑流程图。在图8实施例中,可以基于由监测模块114所收集的实际性能数据,来实验性地确定针对各个处理内核222、224、226、228中的每一个处理内核的性能曲线,或者在一些实施例中,这些性能曲线可以是由各个内核的性能规格来驱动的先验曲线。

[0118] 在一些实施例中,为了实验性地确定各个处理内核222、224、226、228 的性能曲线,监测模块114可以与温度传感器157以及用于监测内核222、224、226、228的功耗的各种其它电压或电流传感器进行通信。在该实施例中,本领域任何普通技术人员将认识到,由监测模块114所收集的数据可以与先前工作负载分配相耦合,并被编译为实验性性能曲线。可以将这些实验性性能曲线存储在CP数据存贮24中,并由能效感知热管理算法进行使用。

[0119] 开始于方框805,监测模块114可以认识到热事件(例如,温度读数超过预定的温度门限),以作为减少热能产生的触发。如先前所描述的,监测模块114可以向EM模块101提供该热报警信息,以便应用能效感知热管理解决方案。

[0120] 在方框810处,EM模块101可以查询与SoC中的各个异构处理部件相关联的性能数据。可以基于由监测模块114向EM模块101所提供的操作温度,来查询相关的性能数据。使用该性能数据,EM模块101可以基于处理部件对工作负载进行高效处理的相对能力,来确定它们的排序。

[0121] 在方框815处,EM模块101可以指示调度器207将排队的工作负载分配给最适合于对这些工作负载进行最高效处理的处理器。值得注意的是,本领域任何普通技术人员应当认识到,将工作负载分配给更高效的处理器,会引起以减少的功率量来处理该工作负载。

[0122] 值得注意的是,EM模块101可以继续与监测模块114一起工作,以识别“新的”更高效处理部件来进行工作负载分配。值得注意的是,应当理解,“新的”更高效处理部件可能是先前被识别成更高效处理部件的同一个处理部件。用此方式,方法800的能效感知热管理解决方案可以持续地确保:将新的工作负载调度给在处理时将会消耗最少的功率量的最适合的处理部件。

[0123] 图9是描绘了用于经由功率模式调整的同步片上系统中的能效感知热管理的方法900的实施例的逻辑流程图。在图9实施例中,可以基于由监测模块114所收集的实际性能数据,来实验性地确定针对各个处理内核222、224、226、228中的每一个处理内核的性能曲线,或者在一些实施例中,这些性能曲线可以是由各个内核的性能规格来驱动的先验曲线。

[0124] 在一些实施例中,为了实验性地确定各个处理内核222、224、226、228 的性能曲线,监测模块114可以与温度传感器157以及用于监测内核222、224、226、228的功耗的各种其它电压或电流传感器进行通信。在该实施例中,本领域任何普通技术人员将认识到,由监测模块114所收集的数据可以与先前工作负载分配相耦合,并被编译为实验性性能曲线。可以将这些实验性性能曲线存储在CP数据存贮24中,并由能效感知热管理算法进行使用。

[0125] 开始于方框905,监测模块114可以认识到热事件(例如,温度读数超过预定的温度门限),以作为减少热能产生的触发。如先前所描述的,监测模块114可以向EM模块101提供该热报警信息,以便应用能效感知热管理解决方案。

[0126] 在方框910处,EM模块101可以查询与SoC中的各个异构处理部件相关联的性能数据。可以基于由监测模块114向EM模块101所提供的操作温度,来查询相关的性能数据。使用该性能数据,EM模块101可以基于处理部件对工作负载进行高效处理的相对能力,来确定它们的排序。

[0127] 在方框915处,EM模块101可以调整不太高效的处理器的功率模式,以尽力减少不必要的功耗。可以预期的是,EM模块101可以识别这样的处理器:该处理器最适合基于各种参数使其功率模式进行调整,其中这些参数包括但不限于:与该处理器从给定的空闲状态功率模式转换到活动功率模式相关联的时延。值得注意的是,本领域任何普通技术人员应当认识到,将处理器的功率模式从活动模式调整到保持模式或者功率下跌(collapsed) 模式,会引起跨该SoC的功耗的平均节省。

[0128] 接着,在判定框920处,EM模块101可以与监测模块114一起工作,以确定是否已经清除了触发方框910和915的热报警。如果已经清除了该报警(即,对最低能效处理器的功率模式进行调整,引起热能产生的降低,使得成功地清除该报警),则沿着“是”分支,以及方法900返回。EM模块101可以准许不太高效的处理内核返回到更高的功耗功率模式。但是,如果作为在方框915处采取的动作的结果,还没有清除该报警,则沿着“否”分支返回到方框910,以及识别“新的”最低能效处理部件以进行功率模式转换。值得注意的是,应当理解,“新的”最低能效处理部件可能是先前被识别成最低能效处理部件的同一个处理部件。

[0129] 图10是描绘了用于经由功率模式工作循环控制的同步片上系统中的能效感知热管理的方法1000的实施例的逻辑流程图。在图10实施例中,可以基于由监测模块114所收集的实际性能数据,来实验性地确定针对各个处理内核222、224、226、228中的每一个处理内核的性能曲线,或者在一些实施例中,这些性能曲线可以是由各个内核的性能规格来驱动的先验曲线。

[0130] 在一些实施例中,为了实验性地确定各个处理内核222、224、226、228 的性能曲线,监测模块114可以与温度传感器157以及用于监测内核222、224、226、228的功耗的各种其它电压或电流传感器进行通信。在该实施例中,本领域任何普通技术人员将认识到,由监测模块114所收集的数据可以与先前工作负载分配相耦合,并被编译为实验性性能曲线。可以将这些实验性性能曲线存储在CP数据存贮24中,并由能效感知热管理算法进行使用。

[0131] 开始于方框1005,监测模块114可以认识到热事件(例如,温度读数超过预定的温度门限),以作为减少热能产生的触发。如先前所描述的,监测模块114可以向EM模块101提供该热报警信息,以便应用能效感知热管理解决方案。

[0132] 在方框1010处,EM模块101可以查询与SoC中的各个异构处理部件相关联的性能数据。可以基于由监测模块114向EM模块101所提供的操作温度,来查询相关的性能数据。使用该性能数据,EM模块101可以基于处理部件对工作负载进行高效处理的相对能力,来确定它们的排序。

[0133] 在方框1015处,EM模块101可以对不太高效的处理器的功率模式进行循环,以尽力减少不必要的功耗。通过对功率模式进行循环,该处理器可以转换经过其各种功率模式,例如,诸如在保持状态和活动状态之间进行切换。通过在多种功率模式中的每一种里驻留一段时间,可以对该处理部件的平均功耗进行优化。可以预期的是,EM模块101可以识别这样的处理器:该处理器最适合基于各种参数对其功率模式进行循环,其中这些参数包括但不限于:与该处理器从给定的功率模式转换到另一种功率模式相关联的时延。值得注意的是,本领域任何普通技术人员应当认识到,使处理器循环经过其功率模式,会引起跨该SoC的功耗的平均节省,而不会完全地牺牲该循环处理器的处理能力。

[0134] 接着,在判定框1020处,EM模块101可以与监测模块114一起工作,以确定是否已经清除了触发方框1010和1015的热报警。如果已经清除了该报警(即,对最低能效处理器的功率模式进行循环,引起热能产生的降低,使得成功地清除该报警),则沿着“是”分支,以及方法1000返回。EM模块101可以准许中断针对不太高效的处理内核的功率模式的工作循环。但是,如果作为在方框1015处采取的动作的结果,还没有清除该报警,则沿着“否”分支返回到方框1010,以及识别“新的”最低能效处理部件以进行功率模式循环。值得注意的是,应当理解,“新的”最低能效处理部件可能是先前被识别成最低能效处理部件的同一个处理部件。

[0135] 图11是描绘了用于处理部件能效评级的运行时验证的方法1100的实施例的逻辑流程图。如上面相对于能效感知热管理解决方案的示例性实施例所解释的,可以使用与各个异构处理部件相关联的性能数据,来确定处理部件中的哪一个处理部件在处理工作负载中是最低能效的(假定针对处理部件的功率频率直接与该处理部件的处理能力相关,可以以MIPS/mW或者MHz/mW为单位来测量能效)。可以基于由监测模块114所收集的实际性能数据,来实验性地确定性能数据,或者在一些实施例中,这些性能曲线可以由各个内核的性能规格来驱动的先验曲线。

[0136] 值得注意的是,根据处理部件的性能规格来导出的先验性能数据可能是不准确的,或者随着该处理部件随时间的消耗,而丧失其准确性。因此,方法1100的实施例寻求:在能效感知热管理解决方案依赖于性能数据做出关于能效的确定之前,对于与给定的处理内核相关联的性能数据的有效性进行验证。

[0137] 由于监测模块114和/或EM模块101可以访问芯片102周围的电流传感器和温度传感器,并且由于EM模块101可以查询存储在CP数据存贮 24中的先前刻画的性能数据,因此可以使用方法1100的实施例来验证所存储的与处理部件相关联的性能数据的准确性,并在需要时进行更新。可以对有关的电流测量和温度测量进行采样,并与存储的性能数据进行比较,以确定处理部件或者子系统是否在活动地呈现预期的电流泄漏特性。

[0138] 在方框1105处,监测模块114可以对与特定的处理部件相关联的电流、操作温度和操作点(例如,设置MHz的功率频率)进行监测。在方框1110处,EM模块101可以基于由监测模块114在方框1105处所测量的操作温度和操作点,来对存储的性能数据进行查询。所存储的性能数据可以包括:鉴于该处理部件的性能规格、操作点和操作温度的预期的电流电平。

[0139] 接着,在方框1115处,EM模块101可以将该预期的电流泄漏与测量的电流泄漏进行比较。在判定框1120处,如果该预期的电流泄漏大于实际测量的泄漏,则沿着“是”分支到方框1140处,并且将该处理部件指定成相对于其初始分类的更低泄漏处理器。可以对CP存贮24中的与该处理部件相关联的预期的电流泄漏进行更新,以反映实际测量的电流泄漏。方法1100返回。

[0140] 返回到判定框1120,如果预期的电流泄漏不大于实际测量的电流泄漏,则沿着“否”分支到判定框1125。在判定框1125处,如果预期的电流泄漏基本等同于实际测量的电流泄漏(在某个统计学意义的可接受范围内),则沿着“是”分支到方框1135,并且维持该处理部件的分类。方法1100返回。

[0141] 如果在判定框1125处,预期的电流泄漏基本不等于实际测量的电流泄漏(即,预期的电流泄漏低于实际测量的电流泄漏),则沿着“否”分支到方框1130处。在方框1130处,将该处理部件指定成相对于其初始分类的更高泄漏处理器。可以对CP存贮24中的与该处理部件相关联的预期的电流泄漏进行更新,以反映实际测量的电流泄漏。方法1100返回。

[0142] 有利的是,通过对处理部件的相对泄漏分类进行验证和更新,可以使能效感知热管理解决方案更适合于在应用热缓解措施之前做出关于哪些处理部件比其它处理部件更高效或者更低效的评估。

[0143] 本说明书所描述的过程或者过程中的某些步骤,自然地在本发明的其它步骤之前以如上所述地工作。但是,本发明并不限于这些所描述的步骤的顺序,如果这种顺序或者序列并不改变本发明的功能的话。也就是说,应当认识到,在不脱离本发明的保护范围或者精神的基础上,一些步骤可以在其它步骤之前执行、之后执行或者并行地执行(基本同时地执行)。在一些实例中,在不脱离本发明的基础上,可以省略或者不执行某些步骤。此外,诸如“此后”、“随后”、“接着”等等之类的词语,并不旨在限制这些步骤的顺序。这些词语仅仅只是用于引导读者遍历该示例性方法的描述。

[0144] 另外,编程领域的普通技术人员能够编写计算机代码或者识别适当的硬件和/或电路,以便例如基于本说明书中的流程图和相关联的描述,没有困难地实现所公开的发明。因此,对于充分地理解如何利用和使用本发明来说,并不认为是必需要公开特定的程序代码指令集或者详细的硬件设备。在上面的描述中以及结合描绘各个流程图的附图,更详细地解释了所主张的计算机实现的过程的创造性功能。

[0145] 在一个或多个示例性方面,本文所描述功能可以以硬件、软件、固件或其任意组合的方式来实现。当在软件中实现时,可以将这些功能存储在计算机可读介质上或者作为计算机可读介质上的一个或多个指令或代码进行传输。计算机可读介质包括计算机存储介质和通信介质二者,其中通信介质包括促进从一个地方向另一个地方传送计算机程序的任何介质。存储介质可以是计算机能够存取的任何可用介质。举例而言,但非做出限制,这种计算机可读介质可以包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储器、磁盘存储器或其它磁存储设备、或者能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由

计算机进行存取的任何其它介质。

[0146] 此外,可以将任何连接适当地称作计算机可读介质。举例而言,如果软件是使用同轴电缆、光纤电缆、双绞线、数字用户线路(“DSL”)或者诸如红外线、无线和微波之类的无线技术从网站、服务器或其它远程源传输的,那么同轴电缆、光纤光缆、双绞线、DSL或者诸如红外线、无线和微波之类的无线技术包括在所述介质的定义中。

[0147] 如本文所使用的,磁盘(disk)和光盘(disc)包括压缩盘(“CD”)、激光盘、光盘、数字多用途盘(“DVD”)、软盘和蓝光盘,其中磁盘通常磁性地复制数据,而光盘则用激光来光学地复制数据。上述的组合也应当包括在计算机可读介质的范围之内。

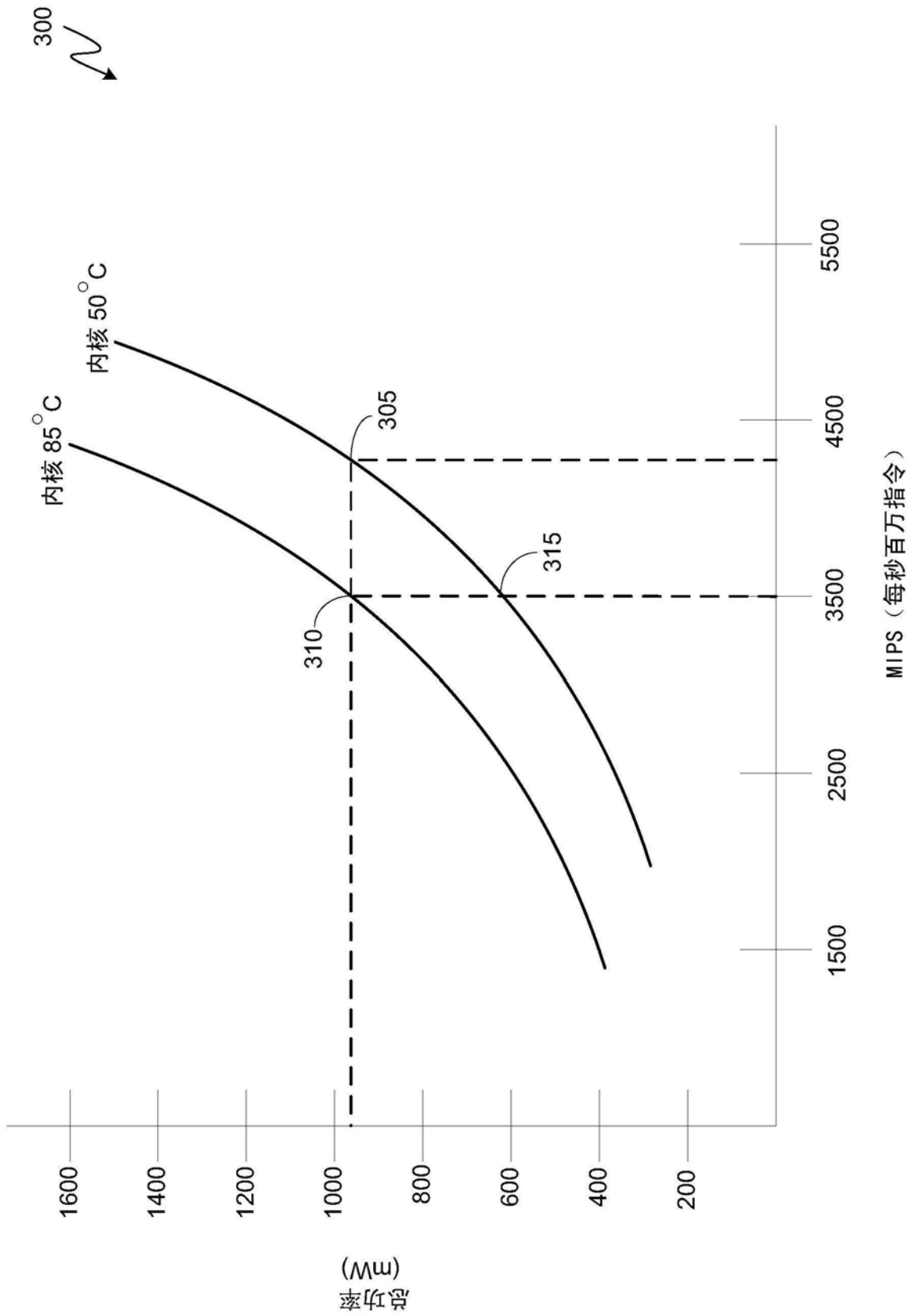


图1A

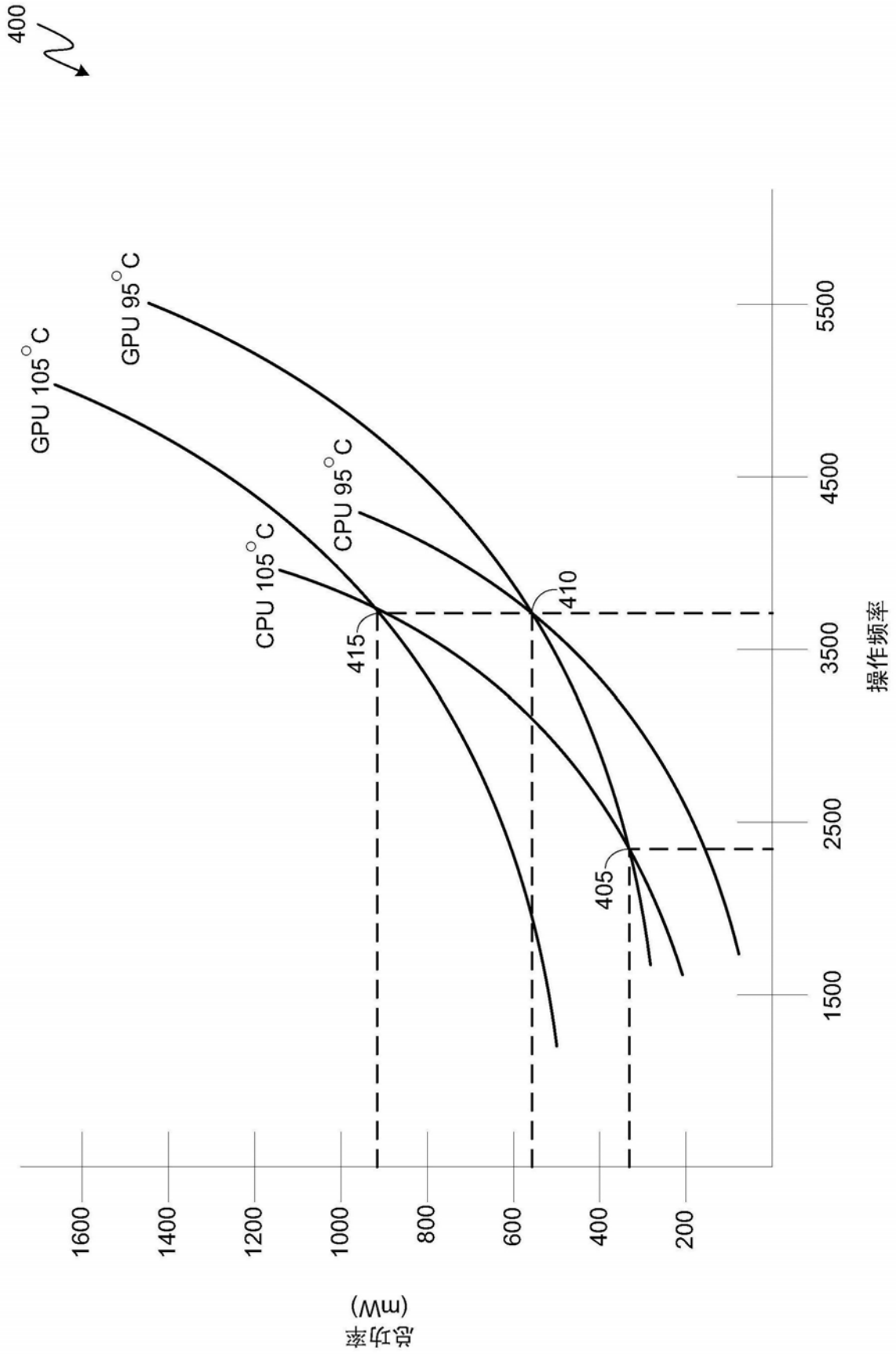


图1B

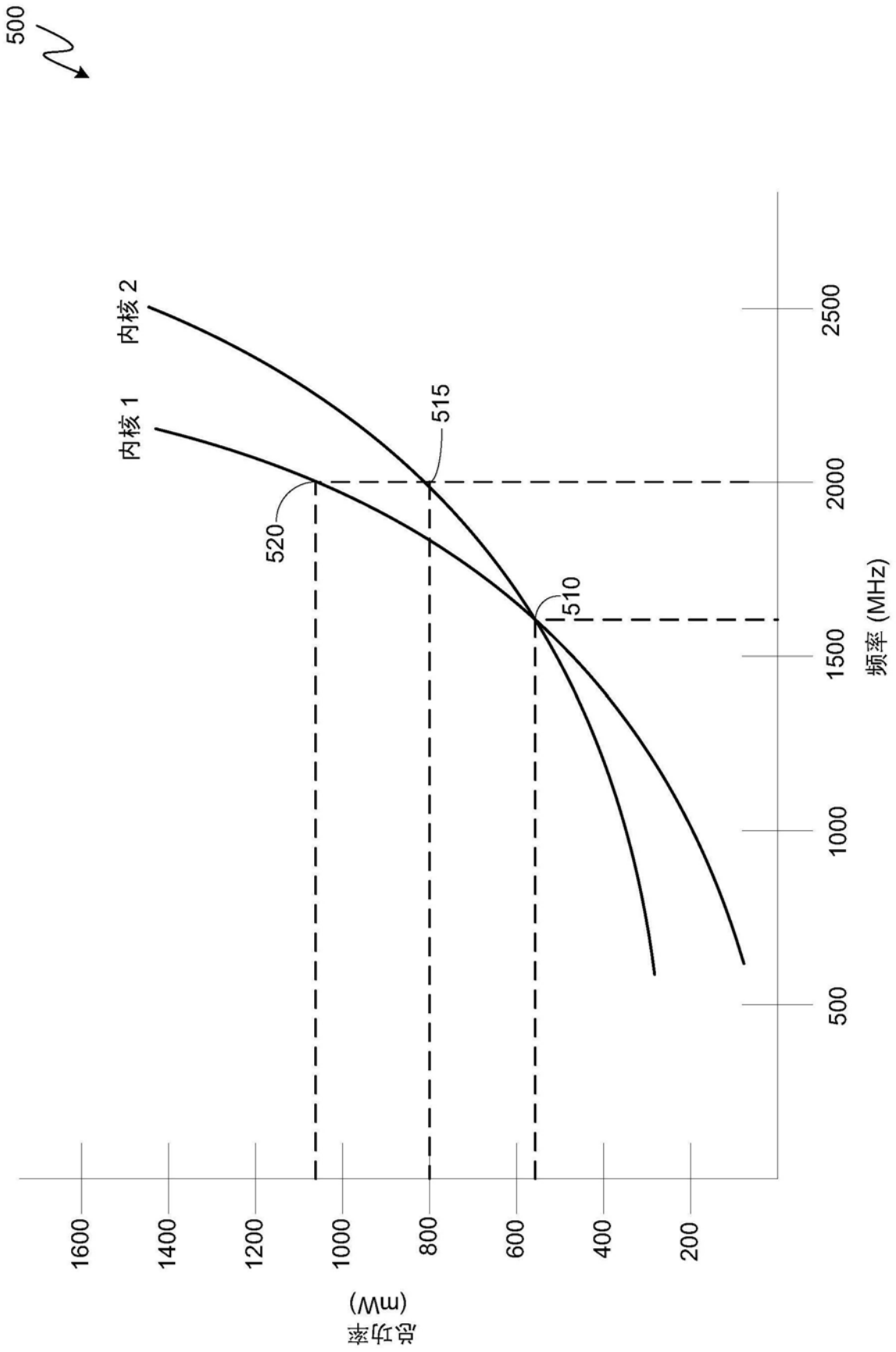


图1C

600
~

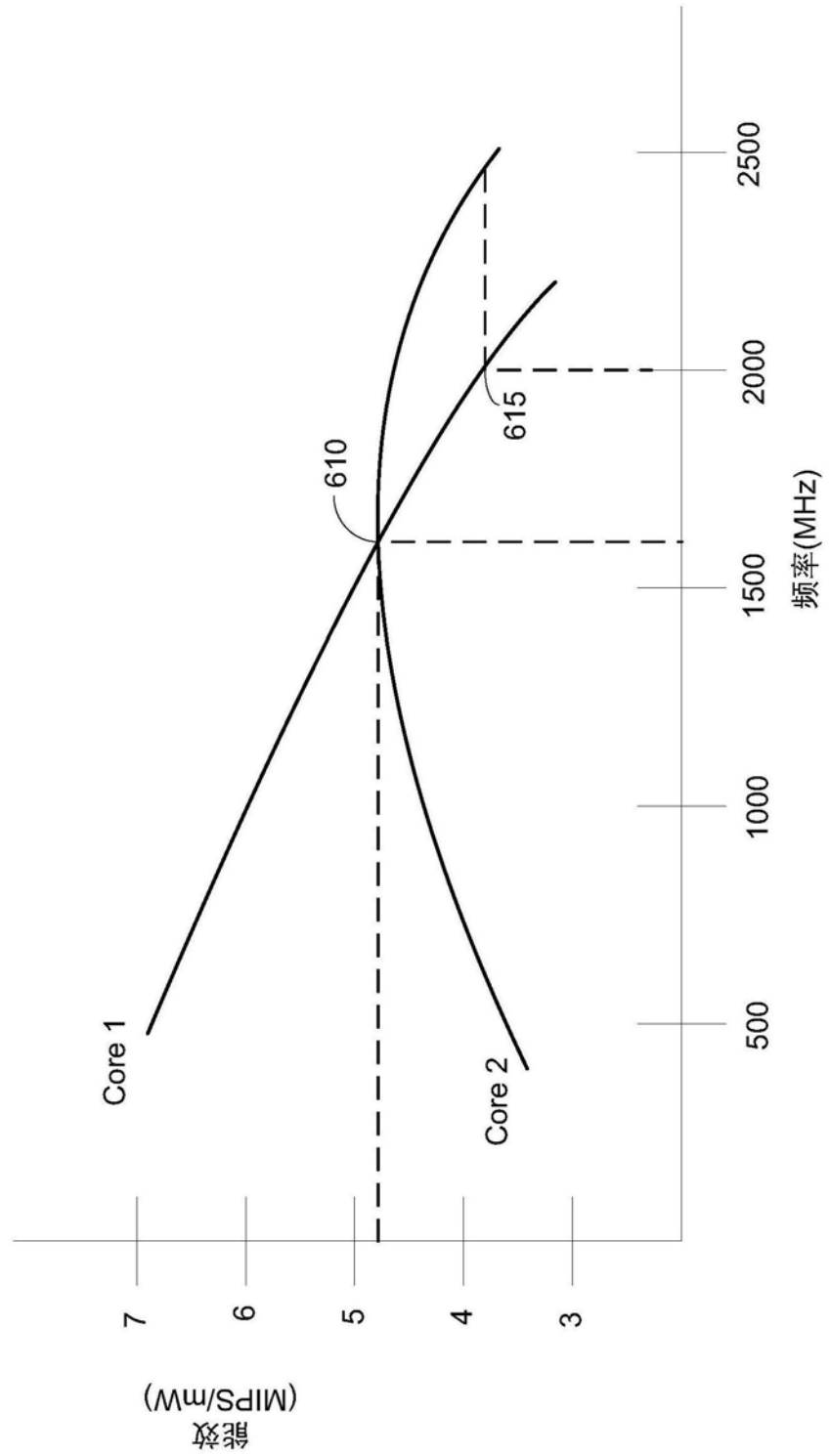


图1D

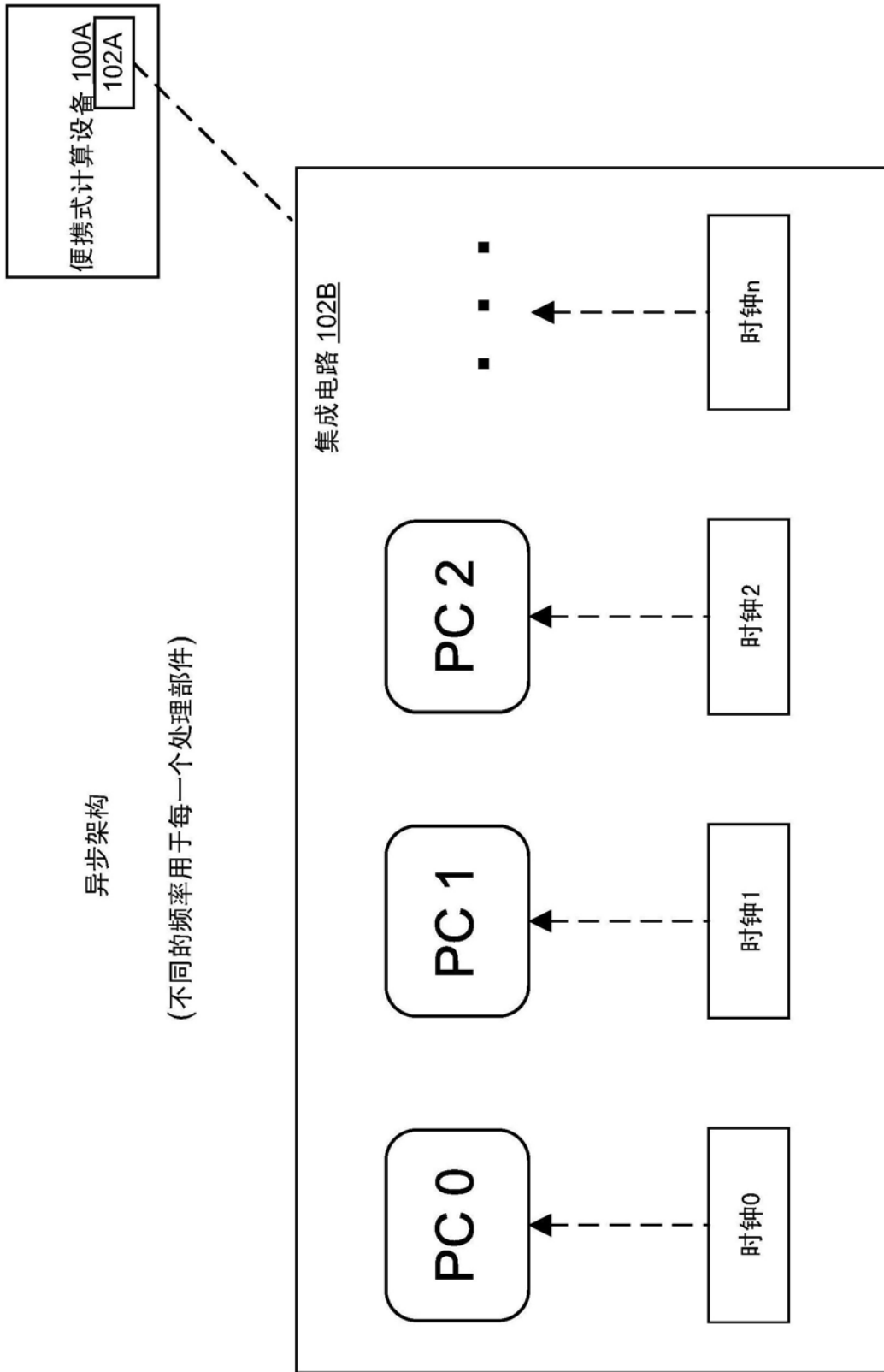


图2A

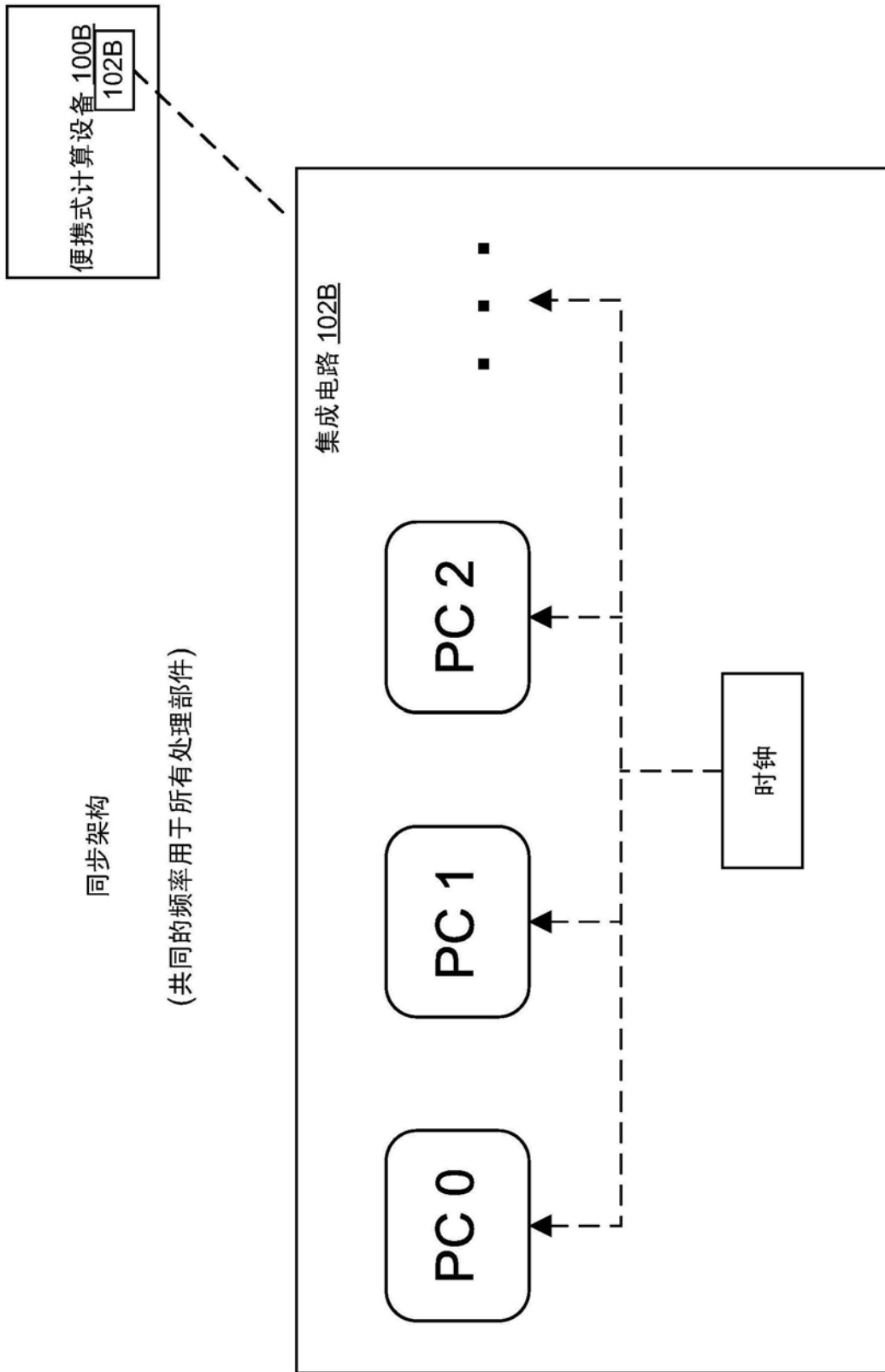


图2B

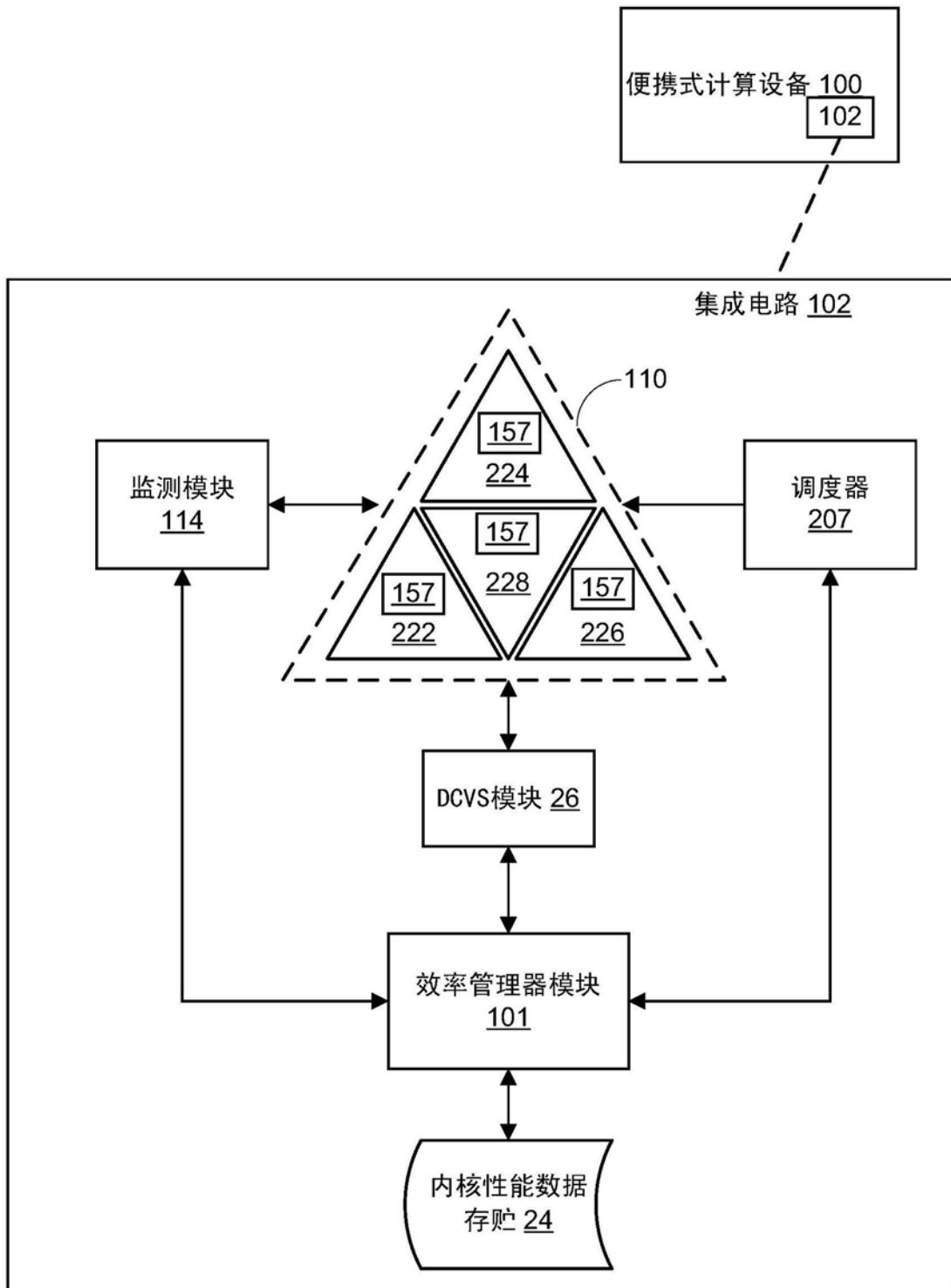


图3

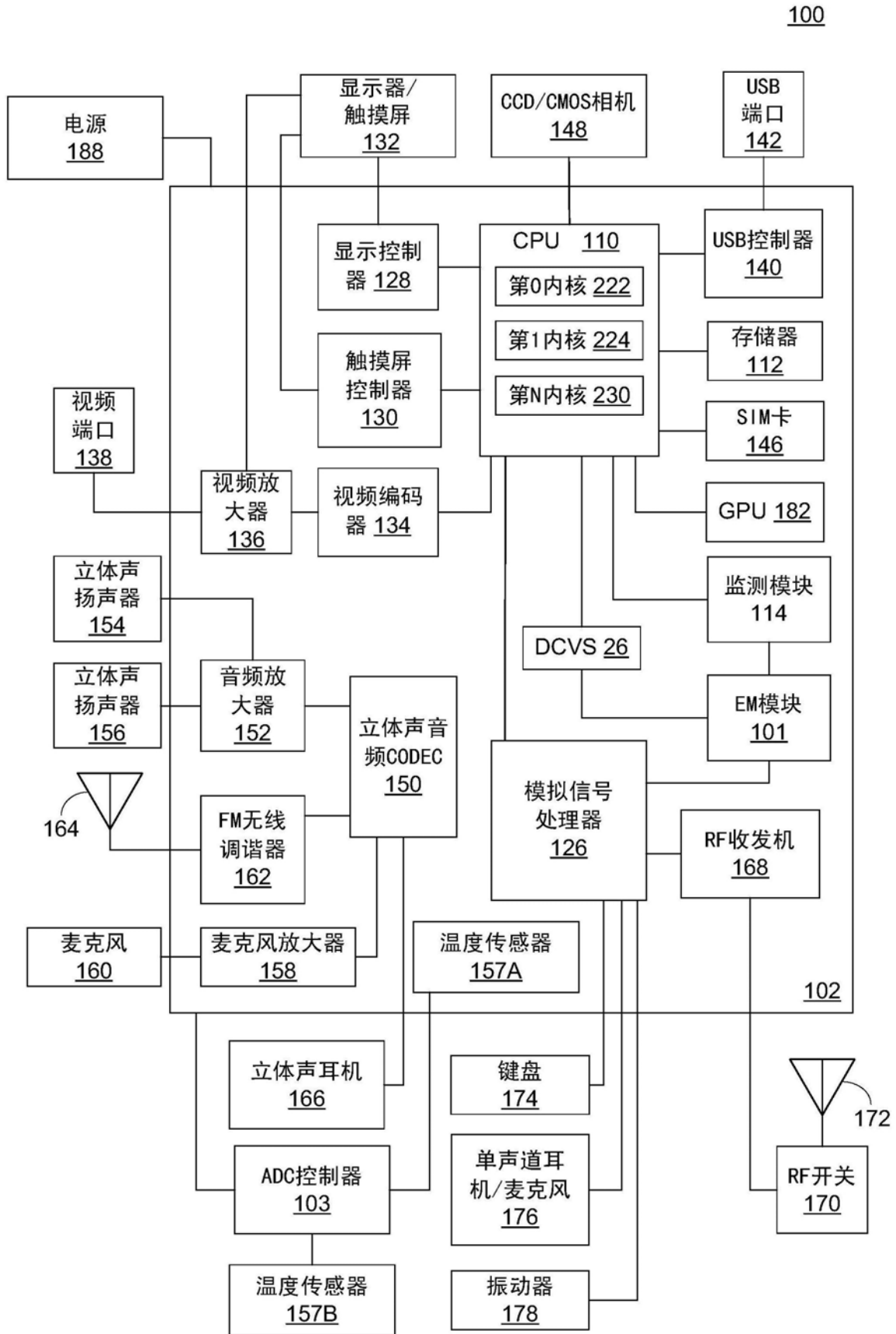


图4

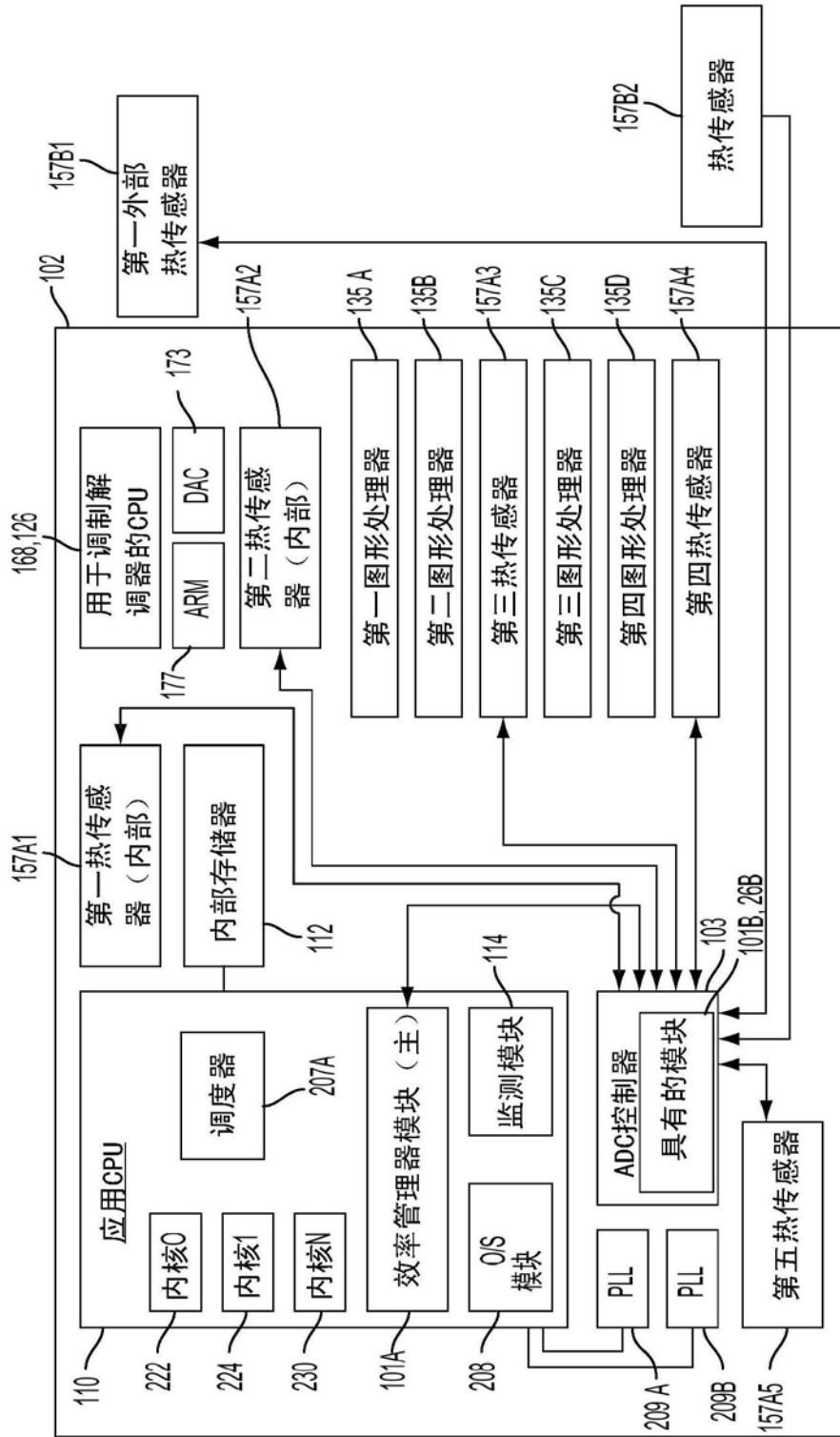


图5A

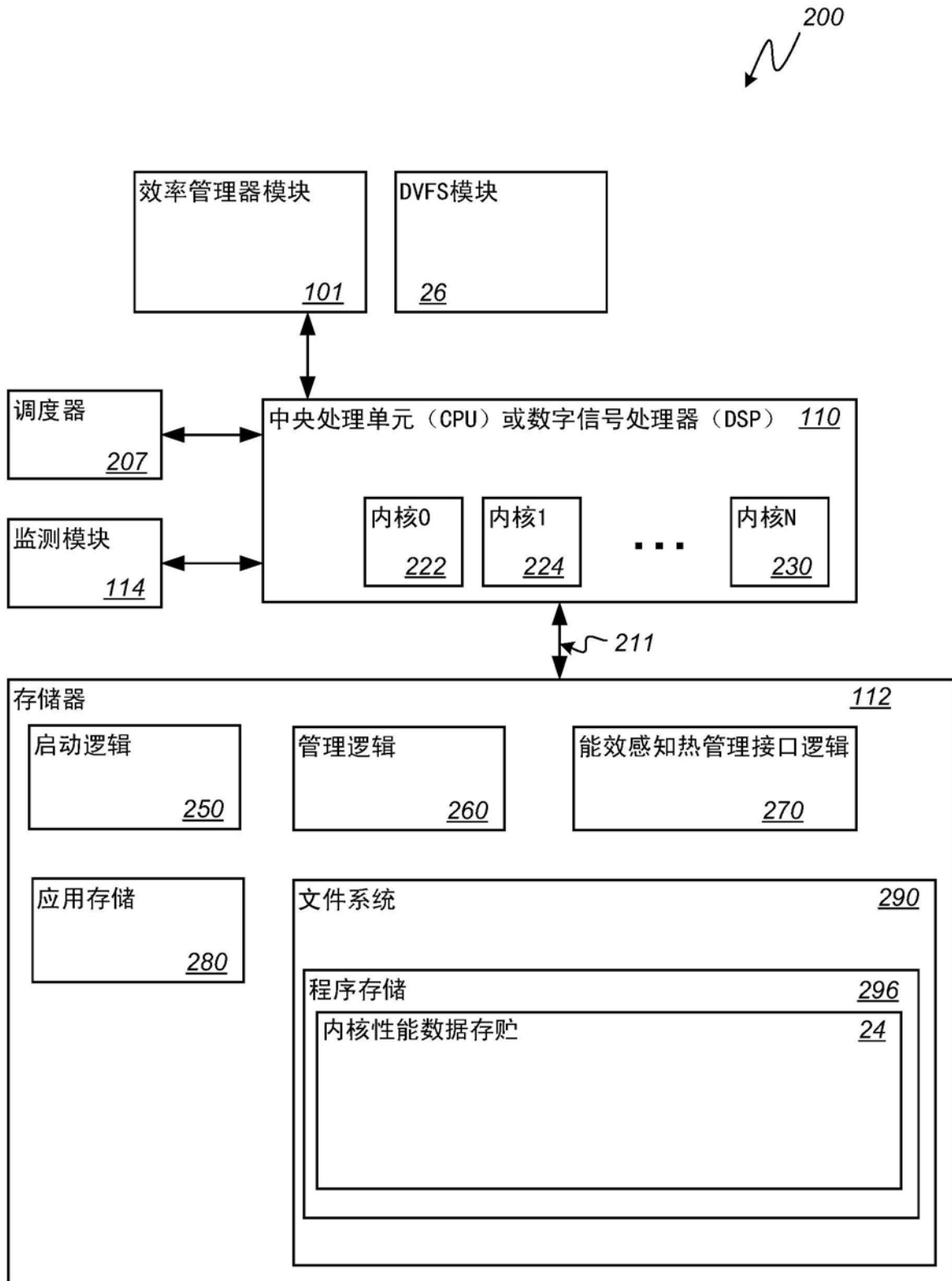


图5B

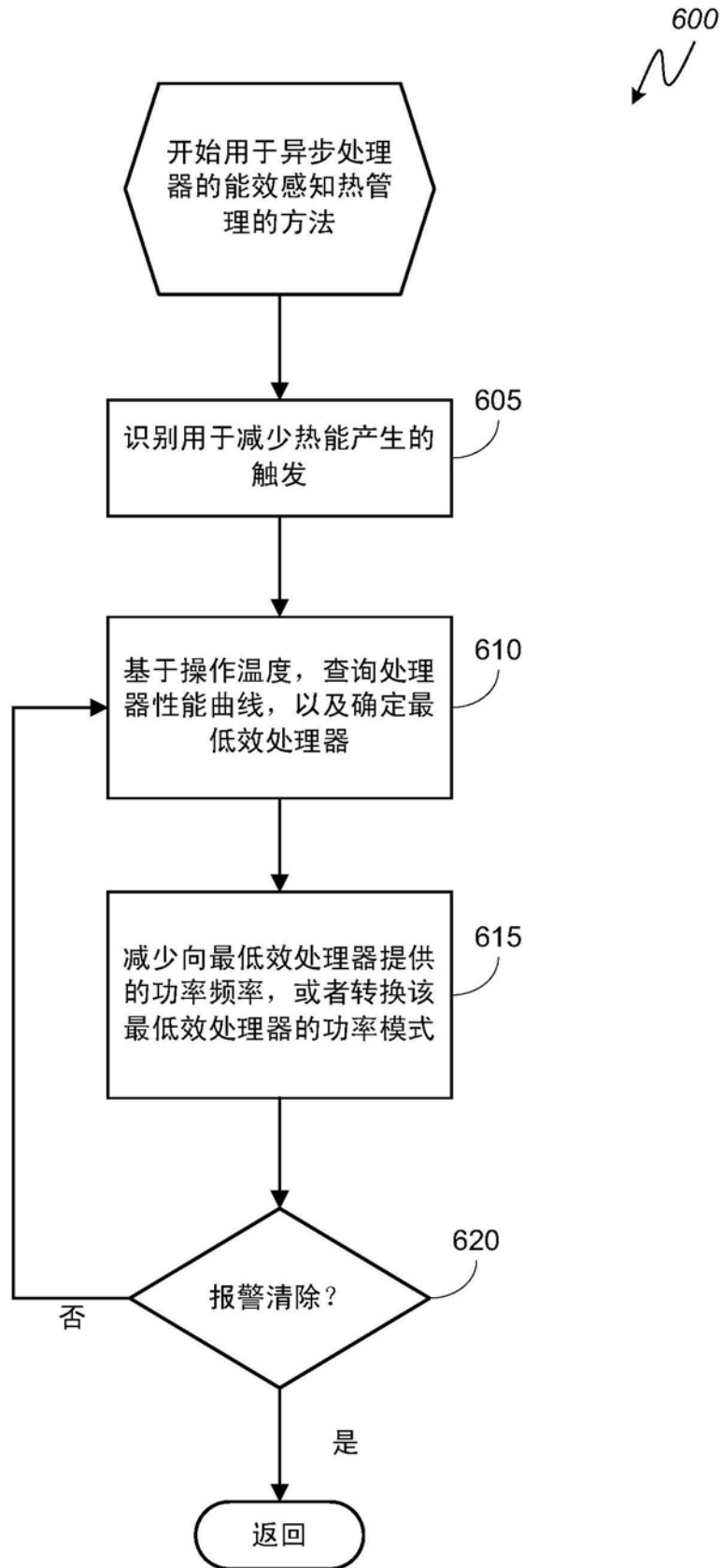


图6

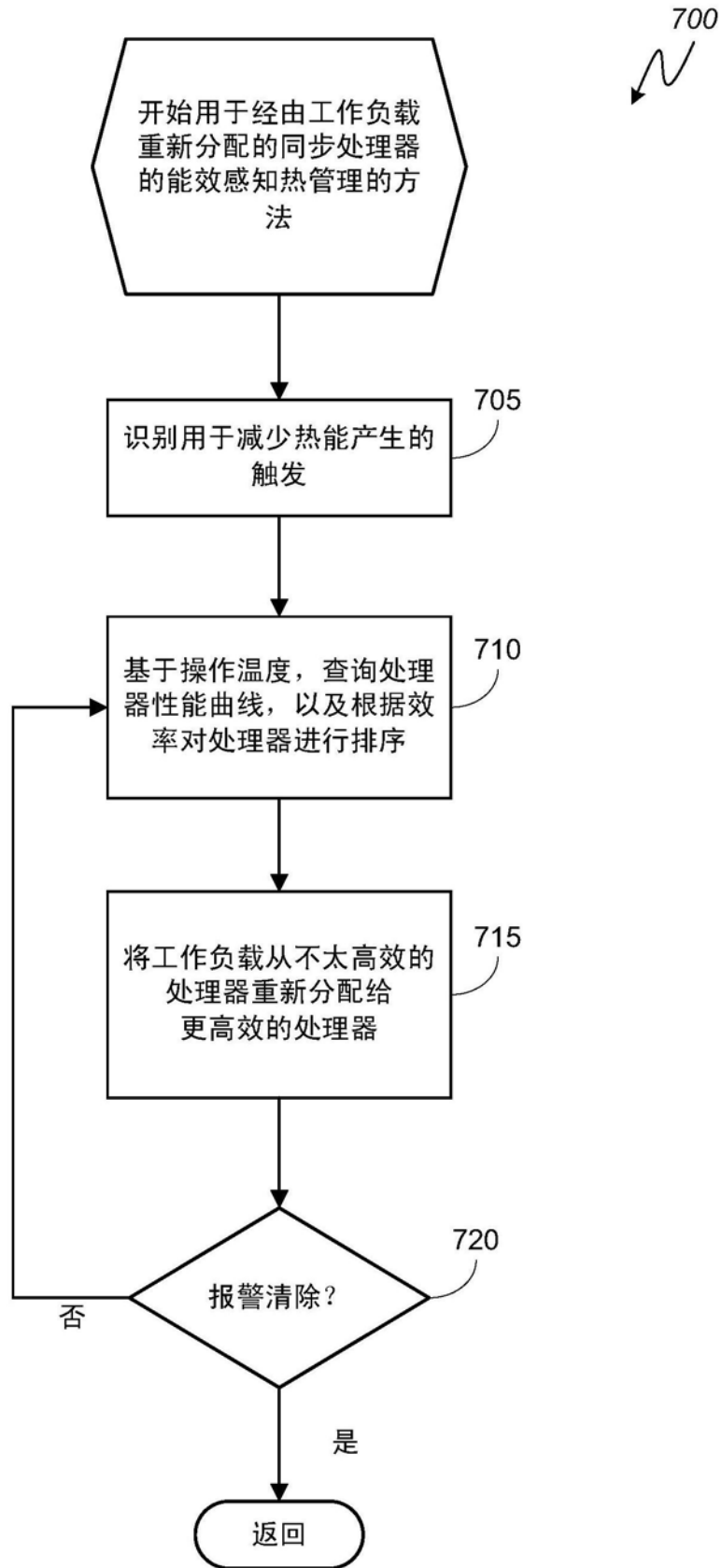


图7

800

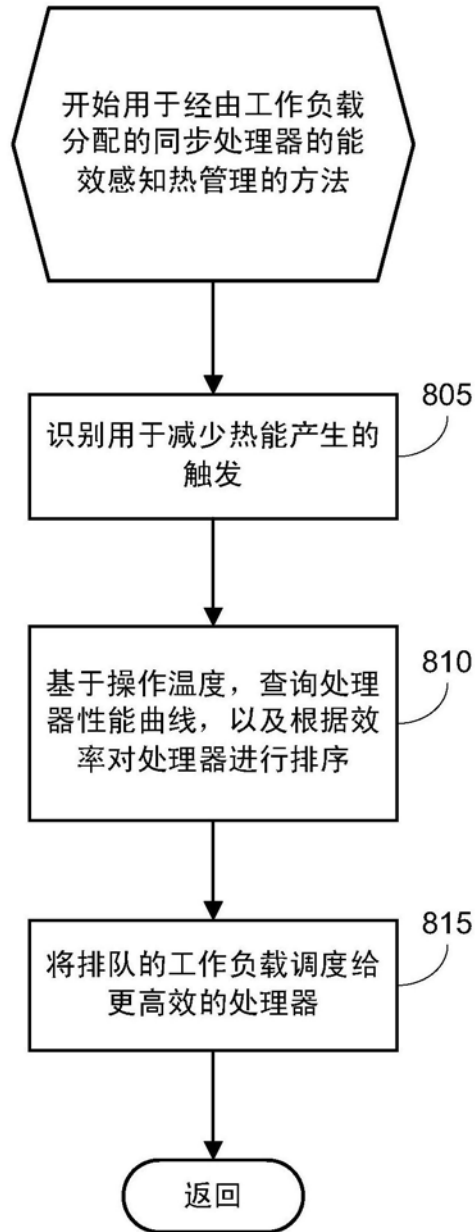


图8

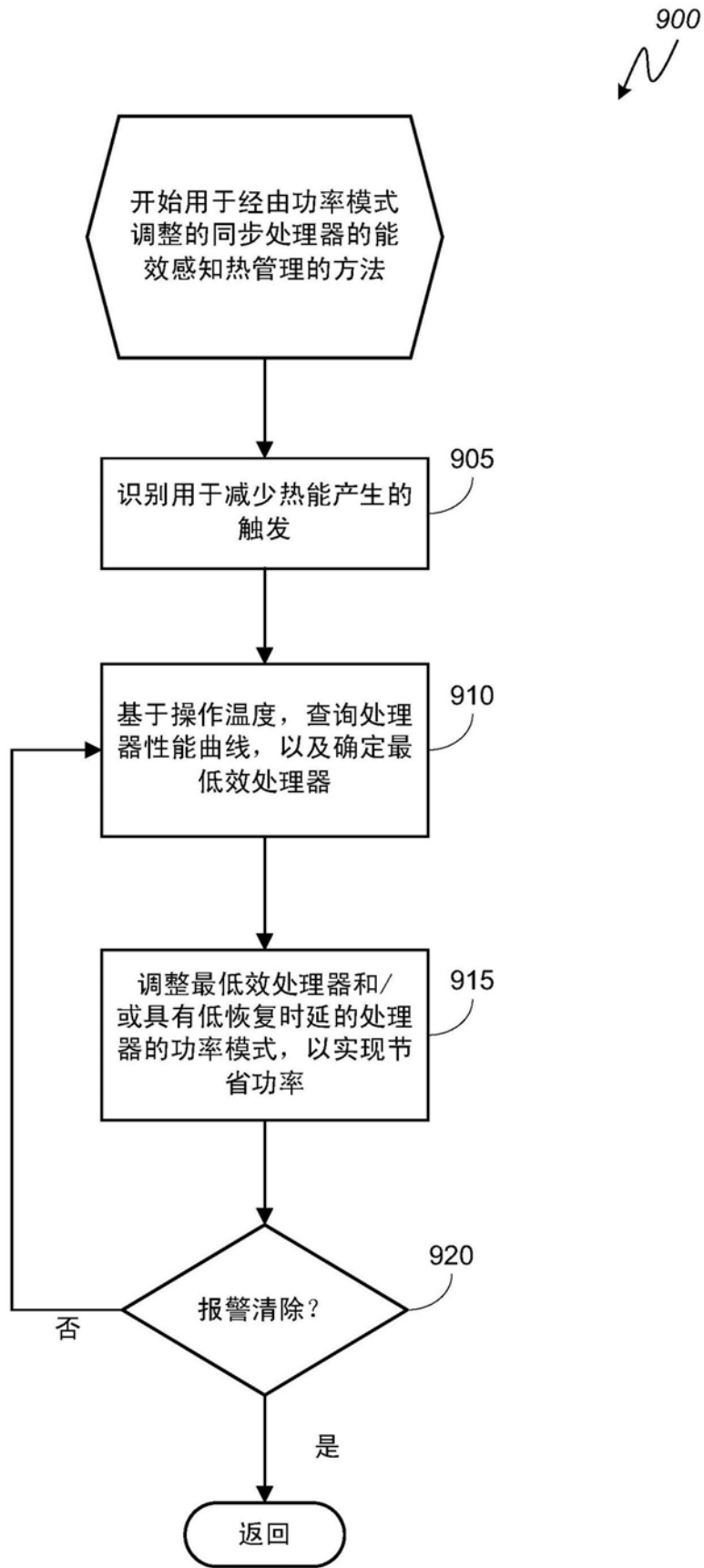


图9

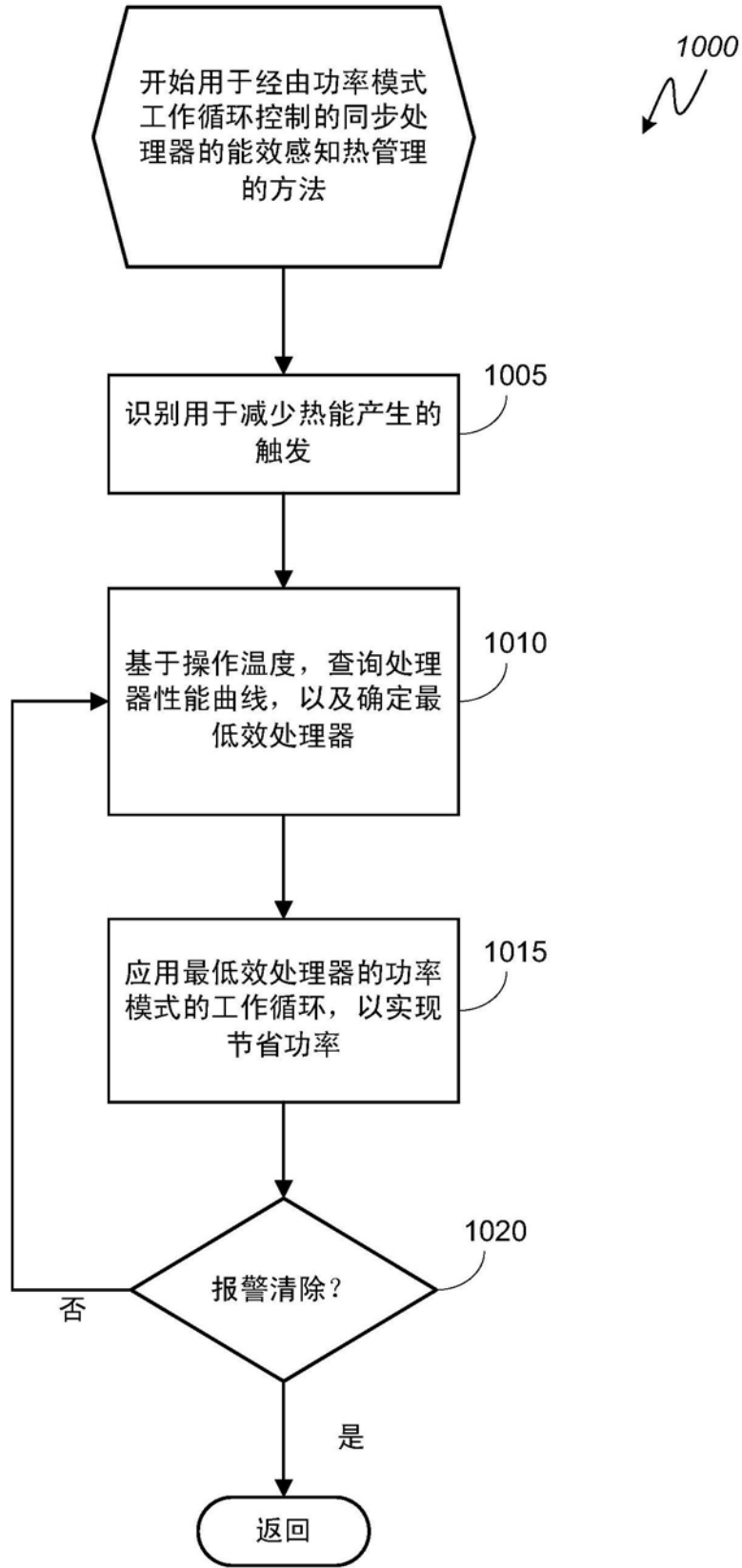


图10

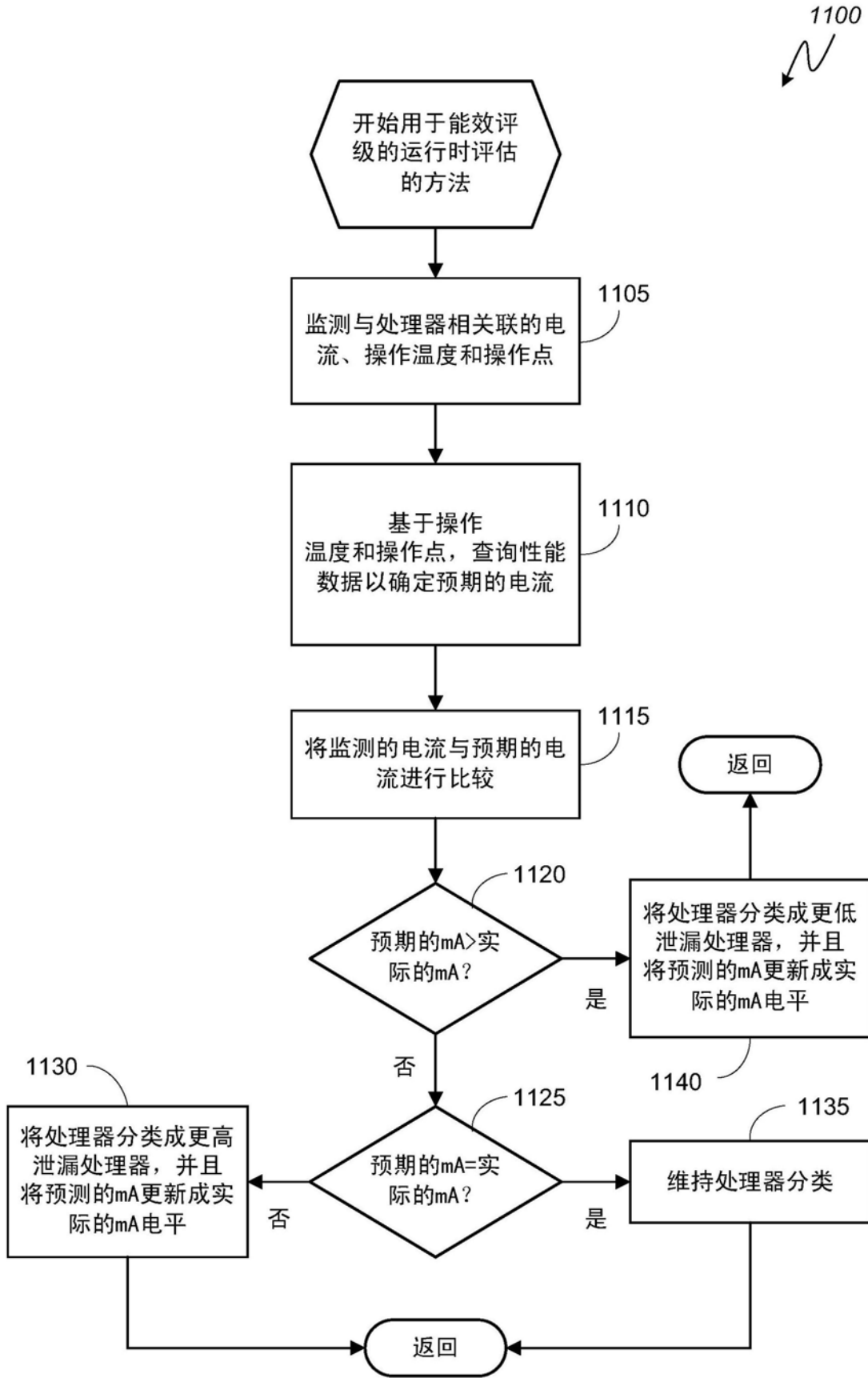


图11