

(19) 中华人民共和国国家知识产权局



## (12) 发明专利申请

(10) 申请公布号 CN 103930874 A

(43) 申请公布日 2014. 07. 16

(21) 申请号 201280055561. 0

(74) 专利代理机构 北京律盟知识产权代理有限公司 11287

(22) 申请日 2012. 10. 23

代理人 宋献涛

(30) 优先权数据

61/558, 743 2011. 11. 11 US

(51) Int. Cl.

13/364, 849 2012. 02. 02 US

G06F 9/48 (2006. 01)

G06F 1/32 (2006. 01)

(85) PCT国际申请进入国家阶段日

2014. 05. 12

(86) PCT国际申请的申请数据

PCT/US2012/061415 2012. 10. 23

(87) PCT国际申请的公布数据

W02013/070425 EN 2013. 05. 16

(71) 申请人 高通股份有限公司

地址 美国加利福尼亚州

(72) 发明人 乔舒亚·H·斯塔布斯

安德鲁·J·弗朗茨

肖恩·D·斯威尼

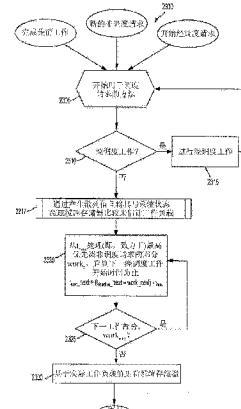
权利要求书4页 说明书22页 附图21页

(54) 发明名称

使用经调度资源组转变通过对便携式计算装置的工作负载估计而节省功率

(57) 摘要

基于用以完成转变资源的所估计处理时间量来调度开始将所述资源转变到第二资源状态组中所指示的状态的开始时间。在经调度开始时间，开始一过程，其中一个或一个以上资源的状态从第一资源状态组所指示的状态切换到所述第二资源状态组所指示的状态。调度转变资源状态的所述过程以在允许刚好及时完成所述过程使得所述资源状态在进入第二应用状态后即刻可由处理器使用的时间开始会有助于使资源等待时间的不利效应最小化。此针对应刚好及时完成所述过程的所述时间的计算可在系统状态和状态之间的转变经准确测量并存储在便携式计算装置的存储器中时得以增强。



1. 一种用于管理便携式计算装置的应用状态的方法,所述便携式计算装置具有至少一个处理器和多个处理器资源,所述方法包括:

将第一资源状态组和第二资源状态组维持在存储器中;

发布在对应于所述第一资源状态组的第一应用状态中操作的处理器从所述第一应用状态转变到对应于所述第二资源状态组的第二应用状态的请求;

通过从散列函数产生散列值且通过比较存储在存储器中的系统状态估计用以完成资源到各个状态的转变的处理时间量;

基于所述所估计的处理时间量调度用以开始将资源转变到所述第二资源状态组中所指示的状态的开始时间;以及

在经调度的开始时间,开始将一个或一个以上资源的状态从所述第一资源状态组所指示的状态切换到所述第二资源状态组所指示的状态的过程。

2. 根据权利要求 1 所述的方法,其中:

所述第一资源状态组是对应于所述处理器的休眠应用状态的休眠资源状态组;且所述第二资源状态组是对应于所述处理器的现用应用状态的现用资源状态组。

3. 根据权利要求 1 所述的方法,其中所述调度开始时间以及开始所述切换状态的过程的步骤由控制器执行。

4. 根据权利要求 1 所述的方法,其中调度开始时间包括:

确定在切换与针对第一处理器发布的第一请求相关联的状态的第一过程与切换与针对第二处理器发布的第二请求相关联的状态的第二过程之间是否存在冲突条件;以及

如果确定存在冲突条件,那么通过修改开始时间来缓解所述冲突条件。

5. 根据权利要求 1 所述的方法,其中所述散列函数使用组合函数和混合函数的至少一者。

6. 根据权利要求 1 所述的方法,其中存储在存储器中的所述系统状态包括依据所述散列函数计算的散列值和实际工作负载完成时间。

7. 根据权利要求 1 所述的方法,其进一步包括:

依据所述散列函数产生表示系统状态的散列值;以及

将所述散列值与存储在存储器中的系统状态比较。

8. 根据权利要求 1 所述的方法,其进一步包括:

将系统状态存储在作为高速缓冲存储器的存储器中。

9. 根据权利要求 1 所述的方法,其进一步包括:

基于一个或一个以上处理器资源所执行的实际工作更新存储在存储器中的系统状态。

10. 根据权利要求 1 所述的方法,其中所述便携式计算装置包括以下中的至少一者:蜂窝式电话、卫星电话、寻呼机、个人数字助理 PDA、智能电话、导航装置、智能本或阅读器、媒体播放器和具有无线连接的膝上型计算机。

11. 一种用于管理便携式计算装置的应用状态的计算机系统,所述便携式计算装置具有至少一个处理器和多个处理器资源,所述系统包括:

处理器,其可操作以:

将第一资源状态组和第二资源状态组维持在存储器中;

发布在对应于所述第一资源状态组的第一应用状态中操作的处理器从所述第一应用

状态转变到对应于所述第二资源状态组的第二应用状态的请求；

通过从散列函数产生散列值且通过比较存储在存储器中的系统状态来估计用以完成资源到各个状态的转变的处理时间量；

基于所述所估计的处理时间量调度用以开始将资源转变到所述第二资源状态组中所指示的状态的开始时间；以及

在经调度的开始时间，开始将一个或一个以上资源的状态从所述第一资源状态组所指示的状态切换到所述第二资源状态组所指示的状态的过程。

12. 根据权利要求 11 所述的系统，其中：

所述第一资源状态组是对应于所述处理器的休眠应用状态的休眠资源状态组；且所述第二资源状态组是对应于所述处理器的现用应用状态的现用资源状态组。

13. 根据权利要求 11 所述的系统，其中所述调度开始时间以及开始所述切换状态的过程的步骤由控制器执行。

14. 根据权利要求 11 所述的系统，其中所述处理器可操作以用于调度开始时间包括：

所述处理器可操作以用于确定在切换与针对第一处理器发布的第一请求相关联的状态的第一过程与切换与针对第二处理器发布的第二请求相关联的状态的第二过程之间是否存在冲突条件；以及

如果确定存在冲突条件，那么通过修改开始时间来缓解所述冲突条件。

15. 根据权利要求 11 所述的系统，其中所述散列函数使用组合函数和混合函数的至少一者。

16. 根据权利要求 11 所述的系统，其中存储在存储器中的所述系统状态包括依据所述散列函数计算的散列值和实际工作负载完成时间。

17. 根据权利要求 11 所述的系统，其中所述处理器进一步可操作以用于：

依据所述散列函数产生表示系统状态的散列值；以及

将所述散列值与存储在存储器中的系统状态比较。

18. 根据权利要求 11 所述的系统，其中所述处理器进一步可操作以用于：

将系统状态存储在作为高速缓冲存储器的存储器中。

19. 根据权利要求 11 所述的系统，其中所述处理器进一步可操作以用于：

基于一个或一个以上处理器资源所执行的实际工作更新存储在存储器中的系统状态。

20. 根据权利要求 11 所述的系统，其中所述便携式计算装置包括以下中的至少一者：蜂窝式电话、卫星电话、寻呼机、个人数字助理 PDA、智能电话、导航装置、智能本或阅读器、媒体播放器和具有无线连接的膝上型计算机。

21. 一种用于管理便携式计算装置的应用状态的计算机系统，所述便携式计算装置具有至少一个处理器和多个处理器资源，所述系统包括：

用于将第一资源状态组和第二资源状态组维持在存储器中的装置；

用于发布在对应于所述第一资源状态组的第一应用状态中操作的处理器从所述第一应用状态转变到对应于所述第二资源状态组的第二应用状态的请求的装置；

用于通过从散列函数产生散列值且通过比较存储在存储器中的系统状态来估计用以完成资源到各个状态的转变的处理时间量的装置；

用于基于所述所估计的处理时间量调度用以开始将资源转变到所述第二资源状态组

中所指示的状态的开始时间的装置；以及

用于在经调度的开始时间开始将一个或一个以上资源的状态从所述第一资源状态组所指示的状态切换到所述第二资源状态组所指示的状态的过程的装置。

22. 根据权利要求 21 所述的系统，其中：

所述第一资源状态组是对应于所述处理器的休眠应用状态的休眠资源状态组；且所述第二资源状态组是对应于所述处理器的现用应用状态的现用资源状态组。

23. 根据权利要求 21 所述的系统，其中调度开始时间以及开始所述切换状态的过程由控制器执行。

24. 根据权利要求 21 所述的系统，其中所述用于调度开始时间的装置包括：

用于确定在切换与针对第一处理器发布的第一请求相关联的状态的第一过程与切换与针对第二处理器发布的第二请求相关联的状态的第二过程之间是否存在冲突条件的装置；以及

用于在确定存在冲突条件下通过修改开始时间来缓解所述冲突条件的装置。

25. 根据权利要求 21 所述的系统，其中所述散列函数使用组合函数和混合函数的至少一者。

26. 根据权利要求 21 所述的系统，其中存储在存储器中的所述系统状态包括依据所述散列函数计算的散列值和实际工作负载完成时间。

27. 根据权利要求 21 所述的系统，其进一步包括：

用于依据所述散列函数产生表示系统状态的散列值的装置；以及

用于将所述散列值与存储在存储器中的系统状态比较的装置。

28. 根据权利要求 21 所述的系统，其进一步包括：

用于将系统状态存储在作为高速缓冲存储器的存储器中的装置。

29. 根据权利要求 21 所述的系统，其进一步包括：

用于基于一个或一个以上处理器资源所执行的实际工作更新存储在存储器中的系统状态的装置。

30. 根据权利要求 21 所述的系统，其中所述便携式计算装置包括以下中的至少一者：蜂窝式电话、卫星电话、寻呼机、个人数字助理 PDA、智能电话、导航装置、智能本或阅读器、媒体播放器和具有无线连接的膝上型计算机。

31. 一种计算机程序产品，其包括其中体现有计算机可读程序代码的计算机可用媒体，所述计算机可读程序代码适于经执行以实施一种用于管理便携式计算装置的应用状态的方法，所述便携式计算装置具有至少一个处理器和多个处理器资源，所述方法包括：

将第一资源状态组和第二资源状态组维持在存储器中；

发布在对应于所述第一资源状态组的第一应用状态中操作的处理器从所述第一应用状态转变到对应于所述第二资源状态组的第二应用状态的请求；

通过从散列函数产生散列值且通过比较存储在存储器中的系统状态来估计用以完成资源到各个状态的转变的处理时间量；

基于所述所估计的处理时间量调度用以开始将资源转变到所述第二资源状态组中所指示的状态的开始时间；以及

在经调度的开始时间，开始将一个或一个以上资源的状态从所述第一资源状态组所指

示的状态切换到所述第二资源状态组所指示的状态的过程。

32. 根据权利要求 31 所述的计算机程序产品，其中：

所述第一资源状态组是对应于所述处理器的休眠应用状态的休眠资源状态组；且  
所述第二资源状态组是对应于所述处理器的现用应用状态的现用资源状态组。

33. 根据权利要求 31 所述的计算机程序产品，其中所述调度开始时间以及开始所述切换状态的过程的步骤由控制器执行。

34. 根据权利要求 31 所述的计算机程序产品，其中调度开始时间包括：

确定在切换与针对第一处理器发布的第一请求相关联的状态的第一过程与切换与针对第二处理器发布的第二请求相关联的状态的第二过程之间是否存在冲突条件；以及  
如果确定存在冲突条件，那么通过修改开始时间来缓解所述冲突条件。

35. 根据权利要求 31 所述的计算机程序产品，其中所述散列函数使用组合函数和混合函数的至少一者。

36. 根据权利要求 31 所述的计算机程序产品，其中存储在存储器中的所述系统状态包括依据所述散列函数计算的散列值和实际工作负载完成时间。

37. 根据权利要求 31 所述的计算机程序产品，其中实施所述方法的所述程序代码进一步包括：

依据所述散列函数产生表示系统状态的散列值；以及  
将所述散列值与存储在存储器中的系统状态比较。

38. 根据权利要求 31 所述的计算机程序产品，其中实施所述方法的所述程序代码进一步包括：

将系统状态存储在作为高速缓冲存储器的存储器中。

39. 根据权利要求 31 所述的计算机程序产品，其中实施所述方法的所述程序代码进一步包括：

基于一个或一个以上处理器资源所执行的实际工作更新存储在存储器中的系统状态。

40. 根据权利要求 31 所述的计算机程序产品，其中所述便携式计算装置包括以下中的至少一者：蜂窝式电话、卫星电话、寻呼机、个人数字助理 PDA、智能电话、导航装置、智能本或阅读器、媒体播放器和具有无线连接的膝上型计算机。

## 使用经调度资源组转变通过对便携式计算装置的工作负载 估计而节省功率

[0001] 优先权和相关申请案声明

[0002] 本申请案依据 35U. S. C. § 119(e) 主张 2011 年 11 月 11 日申请的标题为“使用经调度资源组转变 (USING SCHEDULED RESOURCE SET TRANSITIONS)”的第 61/558,743 号美国临时专利申请案的优先权。其全部内容以引用的方式并入本文中。本申请案还与 2011 年 11 月 8 日申请的标题为“通过调度资源组转变使便携式计算装置中的处理器应用状态之间的资源等待时间最小化 (MINIMIZING RESOURCE LATENCY BETWEEN PROCESSOR APPLICATION STATES IN A PORTABLE COMPUTING DEVICE BY SCHEDULING RESOURCE SET TRANSITIONS)”的第 13/291,767 号美国非临时专利申请案相关。

### 背景技术

[0003] 便携式计算装置 (“PCD”) 正成为人们的个人和专业级必需品。这些装置可包含蜂窝式电话、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装置。

[0004] PCD 通常具有复杂且紧凑的电子封装，其通常由包含中央处理单元、数字信号处理器等的多个处理单元组成。此硬件的大部分可为芯片上系统 (“SOC”) 设计的一部分，如所属领域的一般技术人员所理解。

[0005] 常规 PCD 在不同 SOC 的相应处理器试图进入低功率状态时通常经历显著的滞后时间。低功率状态（其中处理器或类似子系统不在执行任何应用程序或以其它方式实际上闲置）也称为休眠状态，如所属领域的一般技术人员所理解。

[0006] 常规处理器面临的一个问题是，若干通信通常在软件中发生以使处理器进入休眠状态。此问题因为一些资源是共享资源且其状态需要在多个 SOC 子系统之间协调的事实而进一步复杂化。

[0007] 在 SOC 的给定子系统内，本地资源的管理通常较容易且可通过相应操作系统的闲置上下文来进行。然而，为管理共享资源状态的停闭通常需要与所述资源的控制器协调。常规解决方案已通过在允许子系统进入休眠状态之前在软件中使用同步握手来解决此停闭麻烦。此方法是不利的，出于以下若干原因：软件握手缓慢。软件握手容易遭受各种延迟；尤其是中断服务和上下文切换问题。

[0008] 软件握手会延迟功率节省。因为握手是在软件中进行，所以指令处理核心需要保持开启直到完整的握手完成为止。处理器核心较大且复杂，因此这在功率节省方面是需要付出的相当大的牺牲。

[0009] 因此，此项技术中需要一种用于允许 PCD 的处理器在无软件握手的情况下进入休眠状态的方法和系统。

### 发明内容

[0010] 本发明描述一种用于管理便携式计算装置的例如休眠状态和现用状态等应用状

态的方法和系统。对应于应用状态的资源状态组维持在存储器中。可发布在对应于第一资源状态组的第一应用状态中操作的处理器从第一应用状态转变到对应于第二资源状态组的第二应用状态的请求。基于用以完成转变资源的所估计处理时间量调度开始将资源转变到第二资源状态组中指示的状态的开始时间。在经调度开始时间，开始一过程，借助所述过程将一个或一个以上资源的状态从第一资源状态组指示的状态切换到第二资源状态组指示的状态。调度转变资源状态的过程以在允许所述过程刚好在使得进入第二应用状态后资源状态可立即由处理器使用的时间完成的时间开始，有助于使资源等待时间的负面效应最小化。当准确估计资源之间的工作负载时，对转变资源状态的过程进行此调度以在允许所述过程刚好及时完成的时间开始可显著节省功率。描述如何准确估计工作负载的示范性实施例。

## 附图说明

[0011] 在图中，除非另有指示，否则各图中相同参考数字始终指代相同部分。对于具有例如“102A”或“102B”等字母符号表示的参考数字，所述字母符号表示可区分同一图式中存在的两个相同零件或元件。当希望参考数字涵盖在所有图式中具有相同参考数字的所有零件时，参考数字的字母符号表示可省略。

[0012] 图 1 是说明便携式计算装置 (PCD) 的实施例的功能框图；

[0013] 图 2 是说明控制器、系统功率管理器、主处理器、低级驱动器、共享资源与本地资源之间的关系的功能框图；

[0014] 图 3 是说明关于控制器和触发组的细节的功能框图；

[0015] 图 4 说明处理器的示范性现用 - 休眠触发组；

[0016] 图 5 是说明用于管理触发组且以其它方式将处理器从第一应用状态（例如，醒觉状态）转变到第二应用状态（例如，休眠状态）的方法的逻辑流程图；

[0017] 图 6 是说明用于管理触发组且以其它方式将处理器从第二应用状态（例如，休眠状态）转变到第三应用状态（例如，醒觉状态）的方法的逻辑流程图；

[0018] 图 7 是控制器缓冲存储器的功能框图；

[0019] 图 8 是说明用于将处理器从第一应用状态（例如，醒觉状态）转变到第二应用状态（例如，休眠状态）的替代方法的逻辑流程图；

[0020] 图 9 是替代控制器缓冲存储器的功能框图；

[0021] 图 10 是说明用于将处理器从第一应用状态（例如，醒觉状态）转变到第二应用状态（例如，休眠状态）的另一替代方法的逻辑流程图；

[0022] 图 11 是指示与两个请求相关联的处理之间的冲突条件的时间线；

[0023] 图 12 是指示用于缓解图 11 的冲突条件的示范性方法的结果的时间线；

[0024] 图 13 是说明用于将处理器从休眠应用状态转变到醒觉应用状态（包含调度改变资源状态的过程）的方法的类似于图 6 的逻辑流程图。

[0025] 图 14 是说明用于在调度改变资源状态的过程中缓解冲突条件的方法的逻辑流程图。

[0026] 图 15 是指示与经调度请求和非经调度请求相关联的处理之间的冲突条件的时间线；

- [0027] 图 16 是指示用于缓解图 15 的冲突条件的示范性方法的结果的时间线；
  - [0028] 图 17 是指示用于缓解图 15 的冲突条件的第二示范性方法的结果的时间线；
  - [0029] 图 18 是指示用于缓解图 16 的冲突条件的另一示范性方法的结果的时间线；
  - [0030] 图 19 是说明与转变到资源状态组相关联的处理或工作的部分的时间线；
  - [0031] 图 20 是指示当实际工作比预期更快完成时所浪费功率条件的时间线；
  - [0032] 图 21 是指示用于缓解图 20 的所浪费功率条件的示范性方法的结果的时间线；
  - [0033] 图 22 是展示工作的部分的类似于图 17 的时间线；以及
  - [0034] 图 23 是说明用于调度与处理针对资源状态组转变的多个请求相关联的过程的方法的逻辑流程图。
- [0035] 图 24 是说明可由便携式计算装置中的一个或一个以上资源处理的各种工作负载的图形。
  - [0036] 图 25 是说明根据一个示范性实施例状态拓扑内的分组的图表。
  - [0037] 图 26 是说明控制器使用的散列函数的一个实施例的示范性特征的图表。
  - [0038] 图 27A 是说明可在系统状态高速缓冲存储器中使用的元件的图表。
  - [0039] 图 27B 是系统状态高速缓冲存储器的一个示范性实施例。
  - [0040] 图 28 是可在图 27B 的系统状态高速缓冲存储器内找到的系统状态转变的图形表示。
  - [0041] 图 29 说明包括 3 个示范性工作负载的同时工作负载情境。
  - [0042] 图 30 说明在某些状态转变已与系统状态高速缓冲存储器匹配之后图 29 的同时工作负载情境。
  - [0043] 图 31 是用于通过产生散列值且将散列值与系统状态高速缓冲存储器比较而估计工作负载的图 14 和图 23 的子方法或例程的流程图。

## 具体实施方式

- [0044] 本文使用词语“示范性”来表示“充当实例、例子或说明”。本文描述为“示范性”的任何方面不一定解释为与其它方面相比为优选或有利的。
- [0045] 在此描述中，术语“应用”还可包含具有可执行内容的文件，例如：目标代码、脚本、字节代码、标识语言文件和补丁。另外，本文中所提及的“应用”还可包含本质上不可执行的文件，例如可能需要打开的文档或其它需要存取的数据文件。
- [0046] 术语“内容”还可包含具有可执行内容的文件，例如：目标代码、脚本、字节代码、标识语言文件和补丁。另外，本文中所提及的“内容”还可包含本质上不可执行的文件，例如可能需要打开的文档或其它需要存取的数据文件。
- [0047] 如此描述中使用，术语“组件”、“数据库”、“模块”、“系统”等希望指代计算机相关实体，其为硬件、固件、硬件与软件的组合、软件，或执行中的软件。举例来说，组件可为（但不限于为）在处理器上运行的过程、处理器、对象、可执行程序、执行线程、程序和 / 或计算机。借助说明，在计算装置上运行的应用和所述计算装置两者均可为组件。一个或一个以上组件可驻留在过程和 / 或执行线程内，且组件可位于一个计算机上且 / 或分布在两个或两个以上计算机之间。另外，这些组件可从上面存储有各种数据结构的各种计算机可读媒体执行。所述组件可例如根据具有一个或一个以上数据包的信号（例如，来自借助于所述

信号与本地系统、分布式系统中的另一组件交互和 / 或跨越例如因特网等网络而与其它系统交互的一个组件的数据) 借助于本地和 / 或远程过程而通信。

[0048] 在此描述中,术语“通信装置”、“无线装置”、“无线电话”、“无线通信装置”和“无线手持机”可互换使用。随着第三代(“3G”)和第四代(“4G”)无线技术的出现,较大带宽可用性已实现具有更多种类的无线能力的更多便携式计算装置。

[0049] 在此描述中,术语“便携式计算装置”(“PCD”)用于描述基于有限容量电源(例如,电池)操作的任何装置。尽管电池操作的 PCD 已使用了数十年,但与第三代(“3G”)和第四代(“4G”)无线技术的出现关联的可再充电电池的技术进步已实现具有多种能力的许多 PCD。因此,PCD 可为(尤其)蜂窝式电话、卫星电话、寻呼机、PDA、智能电话、导航装置、智能本或阅读器、媒体播放器、上文提及的装置的组合,以及具有无线连接的膝上型计算机。

[0050] 参看图 1,此图是呈无线电话的形式的 PCD100 的示范性、非限制性方面的功能框图,所述 PCD100 用于实施用于管理 PCD100 内的处理器 110、126 的快速休眠状态的方法和系统。如图所示,PCD100 包含芯片上系统 102,其包含多核第一中央处理单元(“CPU”)110A、单一核类型的第二 CPU110B,以及模拟信号处理器 126。

[0051] 这三个处理器 110A、110B 和 126 可耦合在一起。第一 CPU110A 可包括第零核 222、第一核 224 和第 N 核 230,如所属领域的一般技术人员所理解。在替代实施例中,代替于使用两个 CPU110,还可采用两个数字信号处理器(“DSP”),如所属领域的一般技术人员所理解。在另一示范性实施例中,上文提及的任一者可组合使用,如所属领域的一般技术人员所理解。

[0052] 图 1 包含一个或一个以上控制器模块 101。对于此描述的其余部分,控制器模块 101 将以单数形式指代为控制器 101,而非复数形式。所属领域的一般技术人员将认识到,控制器 101 可划分为各个部分且由不同处理器 110、126 执行,而不脱离本发明。或者,控制器 101 可组织为单一元件且由单一处理器 110 或 126 执行。

[0053] 图 1 还说明系统功率管理器 157。系统功率管理器(“SPM”)157 耦合到 CPU110A 和控制器 101。SPM157 通常包括硬件,例如处理器。然而,可针对 SPM157 采用软件和 / 或固件,如所属领域的一般技术人员所理解。SPM157 可负责监视处理器 110、126 和功率轨的状态。SPM157 可检测何时处理器 110、126 即将进入休眠状态或即将离开休眠状态。SPM157 可将处理器 110、126 的这些状态传送到控制器 101。更一般来说,SPM157 可检测何时处理器 110、126 即将从一个应用状态转变到另一应用状态。处理器 110、126 的应用状态可不仅包含其中处理器 110、126 实际上闲置或不在执行任何应用程序的休眠状态和其中其正在执行一个或一个以上应用程序的醒觉或现用状态,而且还包含(或作为替代)以下中的任一者:其中处理器 110、126 正在比其在另一状态中操作高或低的速度下操作的状态;由处理器 110、126 执行应用程序界定的状态不同于由处理器 110、126 执行另一应用程序界定的另一状态;以及由处理器 110、126 同时执行若干应用程序界定的状态不同于由处理器 110、126 同时执行不同数目的应用程序界定的另一状态。

[0054] 控制器 101 可包括由 CPU110 执行的软件。然而,控制器 101 也可由硬件和 / 或固件形成,如所属领域的一般技术人员所理解。

[0055] 一般来说,控制器 101 可负责促进使处理器 110、126 快速进入休眠状态以及快速退出休眠状态。控制器 101 还可负责维持用于跟踪系统状态(如下文将描述)的一个或一

个以上系统状态高速缓冲存储器 2705(见图 27B)。

[0056] 控制器 101 还可包含包括资源组和触发组的一个或一个以上表,如下文将结合图 3 进一步详细描述。控制器 101 还可具有其自身的中断控制器(未说明),用于当 PCD100 中的所有其它硬件元件置于低功率状态且不在起作用时使用。

[0057] 控制器 101 还管理一个或一个以上主处理器 110、126 之间的资源请求。资源请求可由主处理器 110 发布以请求来自资源 105 的动作或功能(参见图 2)。

[0058] 资源 105 在下文更一般描述但可包含(例如)时钟和支持由一个或一个以上主处理器 110、126 执行的软件应用的任务、命令和特征的其它低级处理器。控制器 101 可经设计以防止多个主处理器 110、126 之间的资源请求冲突。

[0059] 图 1 展示 PCD100 可包含存储器 112。在 CPU110 上运行的控制器 101 可存取存储器 112 以促进快速休眠状态且促进快速退出休眠状态,如下文将进一步详细描述。

[0060] 在特定方面中,本文描述的方法步骤的一者或一者以上可由存储在存储器 112 中的形成控制器 101 的可执行指令和参数实施。形成控制器 101 的这些指令可由 CPU110、模拟信号处理器 126 或另一处理器执行。此外,处理器 110、126、存储器 112、存储于其中的指令或其组合可充当用于执行本文中描述的方法步骤中的一者或一者以上的装置。

[0061] 如图 1 中说明,显示控制器 128 和触摸屏控制器 130 高认合到数字信号处理器 110。芯片上系统 102 外部的触摸屏显示器 132 高认合到显示控制器 128 和触摸屏控制器 130。

[0062] 图 1 是说明包含视频译码器 / 解码器(“编解码器”)134 的便携式计算装置(“PCD”)的实施例的示意图,所述编解码器 134 例如逐行倒相制式(“PAL”)编码器、顺序与存储彩色电视系统(“SECAM”)编码器、国家电视系统委员会(“NTSC”)编码器或任何其它类型的视频编码器 134。视频编解码器 134 高认合到多核中央处理单元(“CPU”)110。视频放大器 136 高认合到视频编码器 134 和触摸屏显示器 132。视频端口 138 高认合到视频放大器 136。如图 1 中描绘,通用串行总线(“USB”)控制器 140 高认合到 CPU110。并且,USB 端口 142 高认合到 USB 控制器 140。订户身份模块(SIM)卡 146 也可认合到 CPU110。此外,如图 1 所示,数码相机 148 可认合到 CPU110。在示范性方面中,数码相机 148 是电荷耦合装置(“CCD”)相机或互补金属氧化物半导体(“CMOS”)相机。

[0063] 如图 1 中进一步说明,立体声音频 CODEC150 可认合到模拟信号处理器 126。此外,音频放大器 152 可认合到立体声音频 CODEC150。在示范性方面中,第一立体声扬声器 154 和第二立体声扬声器 156 可认合到音频放大器 152。图 1 展示麦克风放大器 158 也可认合到立体声音频 CODEC150。此外,麦克风 160 可认合到麦克风放大器 158。在特定方面中,调频(“FM”)无线电调谐器 162 可认合到立体声音频 CODEC150。并且,FM 天线 164 高认合到 FM 无线电调谐器 162。此外,立体声头戴式耳机 166 可认合到立体声音频 CODEC150。

[0064] 图 1 进一步指示射频(“RF”)收发器 168 可认合到模拟信号处理器 126。RF 开关 170 可认合到 RF 收发器 168 和 RF 天线 172。如图 1 中所示,小键盘 174 可认合到模拟信号处理器 126。此外,具有麦克风的单声道耳机 176 可认合到模拟信号处理器 126。此外,振动器装置 178 可认合到模拟信号处理器 126。图 1 还展示电源 180(例如,电池)认合到芯片上系统 102。在特定方面中,电源 180 包含可再充电 DC 电池或从连接到 AC 电源的交流(“AC”)到 DC 变压器导出的 DC 电源。

[0065] 如图 1 中描绘,触摸屏显示器 132、视频端口 138、USB 端口 142、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、立体声头戴式耳机 166、RF 开关 170、RF 天线 172、小键盘 174、单声道耳机 176、振动器 178、热传感器 157B 和电源 180 在芯片上系统 322 外部。

[0066] PCD100 的上述元件中的一些元件可包括硬件,而其它元件可包括软件,且另外其它元件可包括硬件与软件的组合。本文使用术语“资源”来指代任何此类元件,不论是硬件、软件还是其组合,其可由处理器控制。资源可在一个方面中界定为此元件的功能性的囊封。除非可能另外指示,否则本文使用术语“处理器”或“主处理器”来指代例如第一 CPU110A、第二 CPU110B 等处理器、模拟信号处理器 126,或指代在软件、固件或类似控制逻辑的控制下操作的任何其它处理器、控制器或类似元件。如下文进一步详细描述,资源的实例是在处理器上执行的软件元件。处理器上的执行线程(例如,与执行应用程序有关的线程)可通过致使发布关于资源的“请求”而存取所述资源。

[0067] 在不同应用状态中,可能必须或需要处理器请求资源的不同配置或状态。举例来说,总线资源可控制总线时钟的速度。在一个应用状态中,处理器可请求允许处理器在例如 100 百万指令 / 秒 (MIPS) 的速率下操作的总线时钟,而在另一应用状态中,处理器可请求允许其在例如 150MIPS 的速率下操作的总线时钟。在处理器准备好进入为休眠状态的应用状态的情况下,处理器可请求零 MIPS 的总线时钟。类似地,在由处理器执行第一应用程序界定的一个应用状态中,处理器可请求 100MIPS,而在由处理器执行第二应用程序界定的另一应用状态中,处理器可请求 150MIPS。同样,在由处理器同时执行特定数目的应用程序界定的一个应用状态中,处理器可请求 100MIPS,而在由处理器同时执行不同数目的应用程序界定的另一应用状态中,处理器可请求 150MIPS。应理解,上文参考的总线时钟仅希望作为可通过处理器发布资源请求配置的资源的实例,而且数字“100”和“150”希望作为处理速度的任意实例。

[0068] 资源配置或状态可分组为资源状态组。资源状态组界定由处理器在特定处理器应用状态中一起使用的一个或一个以上资源的配置或状态。举例来说,特定资源状态组可包含供总线时钟资源向处理器提供特定数目 MIPS 的处理速度的配置或状态信息,以及供解码器(即,资源的另一实例)向处理器提供解码功能的配置或状态信息。

[0069] 图 2 是说明形成系统 103 的控制器 101、系统功率管理器 157、主处理器 110、126、低级驱动器 133、共享资源 105A-C 与本地资源 105D-H 之间的关系的功能框图。图 2 还说明触摸屏 132 可如何耦合到触摸屏驱动器 / 控制器 130。触摸屏驱动器 / 控制器 130 可耦合到第一主处理器 110A 的时钟代码 113A。

[0070] 系统 103 可以使资源等待时间最小化的方式在处理器 110 需要的资源状态组之间切换。术语“资源等待时间”指代在主处理器 110、126 开始使控制器 101 和系统功率管理器 157 准备好转变到另一资源状态组的时间与所述组的资源变为配置为指定状态且准备就绪供处理器使用的时间之间发生的延迟或等待时间。如下文描述,资源状态组可广义上分类为:现用资源状态组,其中处理器具备经配置以辅助处理器执行应用程序以及以其它方式提供处理功率的资源;以及休眠资源状态,其中处理器仅具备辅助处理器维持休眠状态(即,其中处理器不在执行应用程序或以其它方式提供处理功率的状态)的资源。尽管休眠状态中的处理器维持低级功能,但处理器不执行将由所属领域的一般技术人员理解为应用程序的软件。应理解,下文描述的“下一现用状态”特征可应用于任何资源状态组之间

的转变,而不管其为现用组还是休眠组。

[0071] 在图 2 所示的示范性实施例中,第一主处理器 110A 可耦合到系统功率管理器 157 和控制器 101。控制器 101 可耦合到第一主处理器 110A 的时钟代码 113A。控制器 101 可包括一个或一个以上低级驱动器 133。所述一个或一个以上低级驱动器 133 可负责与一个或一个以上共享资源 105A-C 通信。共享资源 105A-C 可包括支持主处理器 110 的任务或功能的任何类型的装置。共享资源 105A-C 可包含例如其它处理器的时钟等装置,以及比如图形处理器、解码器等单一功能元件。

[0072] 共享资源 105A-C 可耦合到一个或一个以上本地资源 105D-H。所述一个或一个以上本地资源 105D-H 可类似于共享资源 105A-C,因为其可包括支持或辅助主处理器 110 的任务或功能的任何类型的装置。本地资源 105D-H 可包含例如其它处理器的时钟等装置,以及比如图形处理器、解码器等单一功能元件。本地资源 105D-H 可包括叶节点。叶节点由所属领域的一般技术人员理解为通常不指代或包含其它相依资源 105 的本地资源 105D-H。

[0073] 控制器 101 可负责管理从所述一个或一个以上主处理器 110、126 发布的请求。举例来说,控制器 101 可管理源自第一主处理器 110A 的请求。第一主处理器 110A 可响应于操作者操纵触摸屏 132 而发布此请求。触摸屏 132 可将信号发布给触摸屏驱动器 / 控制器 130。触摸屏驱动器 / 控制器 130 又可将信号发布给第一主处理器 110A 的时钟代码 113A。

[0074] 控制器 101 还可负责管理针对特定处理器 110 的休眠状态。在进入休眠状态之前。处理器 110 将提供用于管理休眠状态的信息。用于管理休眠状态的信息包含进入和退出休眠状态。此用于管理休眠状态的信息将在下文称为触发和资源状态。资源状态组可包含用于以支持处理器的休眠状态的方式配置一个或一个以上资源的资源信息。

[0075] 触发可界定致使处理器 110 进入休眠状态或离开休眠状态的事件。触发将通常参考包含在控制器 101 内或可由控制器 101 存取的资源状态。资源状态界定特定处理器 110 需要的资源 105 的所要状态。在示范性实施例中,每一处理器 110 可将至少两个资源状态资组提供到控制器 101 :一组现用资源状态和一组休眠资源状态。然而,在其它实施例中,处理器可除单一现用组和单一休眠组或不同于单一现用组和单一休眠组的资源状态组外还提供其它资源状态组。此类其它资源状态组可对应于上文描述的处理器应用状态的一者或一者以上。即,对于任何应用状态,处理器可提供对应的资源状态组。

[0076] 在示范性实施例中,所述组现用资源状态可界定用于当处理器 110 正现用地执行处理功能且需要来自其资源 105 的动作 / 功能时的资源 105 的状态。所述组休眠资源状态可界定当处理器 110 处于休眠或闲置状态的资源 105 的状态。关于触发和资源状态的更多细节将在下文结合图 3 描述。

[0077] 图 3 是说明关于控制器 101、资源组 304 和触发组 314 的细节的功能框图。如先前所述,控制器 101 可包括由 PCD100 的处理器 110、126 的一者或一者以上执行的软件。控制器 101 可将信息存储在存储器 112 中或控制器 101 内的区域(例如,本地存储装置)内,如所属领域的一般技术人员所理解。此信息可包括资源表 302,其包含指派到控制器 101 所服务的每一主处理器 110 的资源组 304。此信息还可包括也指派到每一主处理器 110 且对于每一主处理器 110 可为唯一的触发组 314。

[0078] 每一资源组 304 通常包括关于特定主处理器 110 需要的资源 105 的状态的信息。指派到特定主处理器 110 的每一资源组 304 可包括现用资源组 306 和休眠资源组 308。现

用资源组 306 可界定或描述当特定主处理器 110 为现用或正常起作用时资源 105 的状态。休眠资源组 308 可界定或描述当特定主处理器处于休眠或静止状态时资源 105 的状态,如所属领域的一般技术人员所理解。在图 3 中说明的示范性实施例中,每一资源组 304 还可包括指派到第一主处理器 110 的例如“组 1”和“组 2”等额外组。

[0079] 作为一实例,如图 3 中说明的第一主处理器 (A) 110A 的现用资源组 306 已针对其资源 105 的每一者指派以下值:针对第一共享资源 (SR#1) 105A,值为 1;针对第二共享资源 (SR#2) 105B 的值为 1;针对第 N 共享资源 (SR#N) 105C 的值为 1;同时针对第一本地资源 (LR#1) 105D 的四个值为 1、0、1 和 1。

[0080] 如先前所述,资源 105 的状态不限于单一值且可包含多个值。此外,资源的状态可包含若干不同类型的参数的任一者。举例来说,状态可为可充当资源 105 的特定时钟的时钟速度指定几百兆赫的量。

[0081] 作为另一实例,如图 3 中说明的第一主处理器 (A) 110A 的休眠资源组 308A 已针对其资源 105 的每一者指派以下值:针对第一共享资源 (SR#1) 105A,已向此资源指派值 0;已向第二共享资源 (SR#2) 105B 指派值 0;同时已向第 N 共享资源 (SR#N) 105C 指派值 0。已向第一本地资源 (LR#1) 105D 指派值 0、1、0 和 0。

[0082] 指派给特定主处理器 110 的每一触发组 314 可包括至少三个字段:中断字段 316、“来自组”318 和“去往组”320。触发组 314 的这三个字段的每一者还可包含一组对应的三列:触发开始列 322、清除列 324 和计时器列 326。

[0083] 中断字段 316 描述可由系统功率管理器 157 产生和 / 或检测的动作或活动。中断字段 316 可通常表征为可允许控制器 101 基于 SPM157 检测到的触发事件选择特定处理器 110 需要的特定资源组 304 的“触发事件”。控制器 101 对资源组 304 的选择可避免上文在背景技术部分中描述的耗时的软件握手。

[0084] 复查针对第一主处理器 (A) 110A 的图 3 的第一触发组 (触发组 #1),所述组的字段依据列按序论述。以触发组 314A 的第一列开始,触发开始列 322 具有列举为对应于中断字段 316 的其第一行中的“解码中断”的动作。

[0085] 如先前所述,中断字段 316 可界定致使控制器 101 响应于触发开始字段 322 的检测激活资源组 304 的状态的参数。在图 3 中说明的示范性实施例中,中断字段 316A 已界定或描述为“解码中断”,这意味着当系统功率管理器 110 检测到“解码中断”时(例如,当 PCD100 正解码视频时),此事件可警告控制器 101 复查“触发开始”列下方的第一列 322A1 中的“来自组”字段 318。

[0086] “来自组”字段 318 可包括表示当前资源组 304 对于正由控制器 101 复查的特定主处理器 110 应为什么的值。此字段 318 可依据其识别符列举资源组 304,例如“现用组”、“休眠组”,或比如“组 1”或“组 2”等组号。字段 320 还可包括比如星号等“通配符”。

[0087] “来自组”字段 318 中的通配符表示可致使控制器 101 检索正由特定主处理器 101 使用的最后已知的现用资源组 304。在图 3 中说明的示范性实施例中,“来自组”行 318A 和触发开始列 322A1 具有星号或通配符的值。

[0088] 与“来自组”字段 318 一样,“去往组”320 可包括资源组 304 依据其识别符的列表,例如“现用组”、“休眠组”,或比如“组 1”或“组 2”等组号。字段 320 还可包括表示由处理器 110 利用的最后的资源组 304 的比如星号等“通配符”。在图 3 中说明的示范性实施例

中，“去往组”字段 320A 和触发开始字段列 322A1 具有作为第一资源组 304A 的列 310A 中列举的资源组 1 的“组 1”的值。

[0089] 对于图 3 中说明的实例，当 SPM157 检测到解码中断事件时，其警告控制器 101。控制器 101 复查针对第一主处理器 110 的第一触发组。由于触发开始列 322A1 列举匹配值（解码中断），所以控制器 101 复查“来自组”字段 318A 且确定所述值是通配符或星号。控制器 101 接着复查“去往”字段 320A，其具有表示特定资源组 304A 的“组 1”的值。基于控制器 101 复查到的此信息，控制器 101 将把针对第一主处理器 110A 的当前资源组 304A 从其当前组切换到资源组“组 1”。资源组 1 在指派给第一主处理器 110A 的资源组 304A 的列 310A 中列举。

[0090] 此外，当 SPM157 或控制器 101 检测到例如第一触发组的清除列 324A1 中说明的“不解码”事件时，控制器 101 将接着复查“来自组”字段 318A 且确定此值包括“组 1”。控制器 101 将接着复查“去往组”字段 320，其在此实例中具有通配符或星号的值。这意味着控制器 101 将把第一主处理器 110A 的资源组 304A 从“组 1”资源组切换到处理器 110A 使用的最后现用资源组。

[0091] 触发组的定时器字段 326 可表示特定资源组 304 可由控制器 101 使用的时间量。因此对于图 3 说明的示范性实施例，针对第一触发组的定时器字段 326A1，此字段具有三毫秒的值。这意味着当解码中断事件与第一触发组的触发开始字段 322A1 匹配时，控制器 101 利用“去往组”字段 320A 中指定的资源组 304 持续仅三毫秒的周期。在其它示范性实施例中，可发生或存在以下情形：定时器字段 326 中无信息，或所述值界定为对应于指示不存在针对此转变的定时器触发 326 且所述转变仅应用于不解码字段的值。在界定定时器字段（例如图 3 中说明 – 定时器字段 326A1 和 326A2）的情形中，定时器字段 326 与清除字段 324 之间首先发生无论哪个事件均将通常起始所述转变。

[0092] 图 4 说明处理器 110 的示范性现用 – 休眠触发组 314。在此示范性实施例中，第一列 322 中的中断字段 316 将“停闭”事件界定为针对特定处理器 110 起始休眠组 308（图 3）的动作。“停闭”事件可包含比如操作者选择用于停闭 PCD100 的开 / 关按钮的动作。

[0093] 在图 4 中的示范性实施例中，当检测到“停闭”事件时，控制器 101 将当前现用资源组 306 转变到休眠组 308。休眠组 308 在图 3 中的表 302 的主资源组 304 中列举。

[0094] 当控制器 101 接收到来自 SPM157 的已发生“调用”事件（例如，PCD100 的操作者起始的加电事件）的消息时，控制器将基于触发组 314 的“去往组”字段 320 中列举的通配符或星号值使处理器 110 从其休眠组 308 转变到最后现用资源组 304。

[0095] 如上文描述，系统 103 不限于现用和休眠组 306、308。系统 103 可用于针对除进入或退出如图 3 中说明的休眠状态以外的事件在资源组 304 之间切换。

[0096] 图 5 是说明用于管理触发组 314 以将处理器 110 置于休眠状态的方法 500 的逻辑流程图。框 505 是方法 500 的第一步骤。在框 505 中，每一处理器 110 可基于来自 PCD100 的先前使用情况的数据视需要在控制器 101（图 1-2）中更新其资源组 304 以及其触发组 314。

[0097] 在框 510 中，处理器 110 可请求 SPM157（图 2）产生到控制器 101 的停闭信号。在框 515 中，SPM157 可将停闭信号发送到控制器 101。

[0098] 控制器 101 可在框 520 中接收停闭信号且激活可指派给停闭事件（如图 4 中说

明)的触发组 314。在图 4 说明的示范性实施例中,使停闭信号与触发组 314 的中断字段 316 匹配。触发组 314 引导控制器 101 存取休眠组 308,如“去往组”字段 320 中所指示。在框 525 中,控制器 101 可立即将确认信号发送到 SPM157,同时控制器 101 继续激活与停闭信号事件匹配的触发组 314 所参考的资源组 304。

[0099] 在框 530 中,针对每一匹配的触发组 314,例如在图 4 中说明的对应中断字段 316 中列举“停闭”事件的匹配的触发组 314,控制器 101 可针对图 3 的主处理器 110A 将当前资源组 304 切换到休眠组 308,例如第一资源组 305A 的休眠组 308A。

[0100] 接下来,在框 535 中,控制器 101 可向例如图 2 中说明的低级驱动器 133 发布休眠请求状态。低级驱动器 133 可将所请求的状态传递到对应的资源 105。

[0101] 在框 540 中,每一资源 105 可向控制器 101 和 SPM157 发布停闭信号确认。方法 500 可接着结束。

[0102] 图 6 是说明用于管理触发组 314 以将处理器 110 从休眠状态置于现用状态的方法 600 的逻辑流程图。框 605 是方法 600 中的第一步骤。在框 605 中,以 SPM157 检测唤醒条件或唤醒事件,或直接由可具有其自身的中断控制器(未说明)的控制器 101 检测唤醒事件。示范性实施例可经设计使得唤醒中断不可由 SPM157 检测。在此类示范性实施例中,控制器 101 可使用其中断控制器来对其进行检测且将这些“映射”到针对主处理器 110 的休眠组要求。

[0103] 接下来,在框 610 中,SPM157 可将唤醒信号发送到控制器 101。在框 615 中,控制器 101 可接收来自 SPM157 的唤醒信号且激活与唤醒信号匹配的一个或一个以上触发组 314。举例来说,控制器 101 可使唤醒信号与图 4 的触发组 314 的“现用”列中的中断字段 316 中列举的“调用”事件匹配。在图 4 的示范性实施例中,现用列 324 中的“去往字段”320 将控制器引导到当前处理器 110 使用的最后资源组 304。

[0104] 因此在框 620 中,控制器 101 将基于匹配的触发组 314 改变针对处理器 110 的当前资源组 304。所属领域的一般技术人员认识到,控制器 101 将循环通过如图 3 中说明的其所维持的其所有触发组。

[0105] 接下来,在框 625 中,控制器 101 可将识别哪些主处理器 110 已从休眠状态唤醒的唤醒确认发送到 SPM157。接下来,在框 630 中,具有匹配的唤醒触发组 314 的每一处理器 110 从休眠状态释放且以 SPM157 供应的功率恢复到其现用状态。方法 600 接着结束。

[0106] 图 7-10 说明另一特征,其在本描述中称为“下一现用资源状态组”或“下一现用组”。下一现用组的一个实例为下一醒觉组。下一醒觉组或其它下一现用组可以与上文参看图 6 以及控制器 101 在唤醒事件后即刻切换到的资源组 304 描述相同的方式使用。

[0107] 图 7 类似于图 3 之处在于,其表示存储在控制器 101 中的信息。在示范性实施例中,控制器 101 可包含三个存储器缓冲器,在此描述中为方便起见称为“A”存储器缓冲器 702、“B”存储器缓冲器 704 和“C”存储器缓冲器 706。

[0108] 图 8 是类似于图 5 的逻辑流程图,类似之处在于,其说明用于将处理器置于休眠状态的方法 800。框 805 是方法 800 的第一步骤且类似于上文参看图 5 描述的框 505。框 805 指示处理器 110 可不仅更新现用或醒觉资源状态组和休眠资源状态组而且还可更新下一醒觉资源状态组。如图 8 所示,处理器可致使现用组存储在控制器 101 的“A”缓冲器 702(图 7)中,休眠组存储在控制器 101 的“B”缓冲器 704(图 7)中,且下一醒觉组存储在

控制器 101 的“C”缓冲器 706(图 7) 中。框 505 的其它方面与上文参看框 505(图 5) 所描述相同,且因此此处不描述。

[0109] 框 810、815、820、825、830、835 和 840 分别与图 5 的框 510、515、520、525、530、535 和 540 相同,且因此此处不描述。注意,当处理器开始停闭时,其处于对应于“A”缓冲器 702(图 7) 中存储的醒觉组的醒觉应用状态中。处理器接着以与上文参看图 5 描述相同的方式进入对应于“B”缓冲器 704(图 7) 中存储的休眠组的休眠应用状态中。处理器在对应于“C”缓冲器 706(图 7) 中存储的下一醒觉组的下一醒觉应用状态中从休眠应用状态醒觉(图 6)。通过将下一醒觉组更新预先存储在“C”缓冲器 706(图 7) 中且尽可能早地应用所述更新,控制器 101 可在唤醒事件后立即开始配置所述下一醒觉组指定的资源,借此帮助使资源等待时间最小化。

[0110] 图 9 涉及另一示范性实施例,其中控制器 101 不具有足够的存储器来同时存储所有三个上述资源状态组。在此实施例中,控制器 101' 仅具有“A”缓冲器 902 和“B”缓冲器 904,且不存在可用于“C”缓冲器的存储器空间。在此例子中,再使用“A”缓冲器 902 使得在不同时间其存储(当时)醒觉组以及下一醒觉组。

[0111] 图 10 是类似于图 5 和 9 的逻辑流程图,其类似之处在于,其说明用于将处理器置于休眠状态的方法 1000。框 1005 是方法 800 的第一步骤且类似于上文参看图 8 描述的框 805,但不包含将下一醒觉组存储在“C”缓冲器中。而是,处理器可致使将现用组存储在控制器 101' 的“A”缓冲器 902(图 9) 中且将休眠组存储在控制器 101' 的“B”缓冲器 904(图 9) 中,但处理器等到其已到达休眠应用状态的转变过程中的“无返回点”(该术语是所属领域的一般技术人员所理解的)之后才再使用“A”缓冲器存储下一醒觉组。框 1005 的其它方面与上文参看框 505(图 5) 所描述相同,且因此此处不描述。

[0112] 在框 1008 中,处理器执行称为下一醒觉组的伪更新或虚拟更新的过程。注意,在上文描述的框 1005 中,处理器可通过将资源状态组写入到控制器 101' 中的“A”缓冲器 902 和“B”缓冲器 904 而执行资源状态组的实际更新。所述更新是实际的,因为控制器 101' 从处理器接收中断以向其告知缓冲器内容应更新,从而致使控制器 101' 对更新起作用或应用更新。控制器 101' 通过执行对于准备经更新资源状态组信息以供使用可能是必需的各种任务来应用更新。如果更新缓冲器“B”中的休眠组,那么控制器 101' 可准备经更新的休眠组信息以供在需要切换资源状态组的停闭事件或类似事件随后发生的情况下使用。如果更新“A”缓冲器 902 中的现用组,那么控制器 101' 可致使相应地调整资源。处理器在框 1008 中执行的伪更新包含将针对下一醒觉组的更新存储在“A”缓冲器 902(图 9) 中,而不将中断发送到控制器 101'。因为控制器 101' 未接收中断,所以其尚未应用“A”缓冲器 902(图 9) 中发生的更新。此伪更新在无返回点之后发生,其中处理器 110 将请求 SPM157(图 2) 向控制器 101' 信令停闭且确保不对“A”缓冲器 902 中的当时现用资源组状态信息作出任何进一步更新。

[0113] 框 1010、1015、1020 和 1025 分别与上文参看图 5 的框 510、515、520 和 525 所描述相同,且因此此处不描述。

[0114] 接着,在框 1027 中,控制器 101' 通过检查“A”缓冲器 902(图 9) 以查找更新而响应于其与处理器之间发生的握手(框 1020、1025),且存储所述更新以在图 6 的唤醒方法中使用。(可注意到,归因于使用中断告知接收方控制器 101' 处理器已写入到缓冲器的“消

息”的方式,存储器缓冲器也称为“消息 RAM”。)因此,通过将下一醒觉组预先存储在“A”缓冲器 902(图 9)中,控制器 101' 能够在唤醒事件后立即开始配置所述下一醒觉组指定的资源,借此帮助使资源等待时间最小化。

[0115] 框 1030、1035 和 1040 分别与图 5 的框 530、535 和 540 相同,且因此此处不描述。处理器接着以与上文参看图 5 描述相同的方式相应地进入对应于“B”缓冲器 904(图 9)中存储的休眠组的休眠应用状态。处理器在对应于“B”缓冲器 904(图 9)中存储的下一醒觉组的下一醒觉应用状态中从休眠应用状态醒觉(图 6)。通过预先存储下一醒觉组且尽可能早地应用所述下一醒觉组,控制器 101' 能够在唤醒事件后立即开始配置所述下一醒觉组指定的资源,借此帮助使资源等待时间最小化。

[0116] 图 11-23 说明另一特征,其涉及调度上文描述的资源组转变。所属领域的一般技术人员理解,在许多例子中,处理器应用程序状态的上述改变可以相对可预测周期性发生。举例来说,在 PCD100(图 1)中,处理器可能必须执行视频播放器应用程序以在其中处理器可周期性(例如,每隔 X 毫秒)解码视频数据帧的状态中唤醒或以其它方式转变到所述状态。类似地,处理器可能必须控制 PCD100 的蜂窝式电话功能以例如在其中处理器可周期性(例如,每隔 X 毫秒)检查 RF 通信信号的状态中唤醒或以其它方式转变到所述状态。由于应用程序状态的周期性改变将发生的时间可预测,且由于资源完成到对应于下一应用程序状态的状态的转变所必需的时间量大体固定或恒定,所以必须开始切换资源状态组的过程的时间可预测。举例来说,可预测处理器在时间  $t_{\text{deadline}}$  需要具有处于示范性资源状态组(“R”)指示的状态的一组资源。此示范性资源状态组(“R”)可指定总线时钟资源将改变到例如 100MHz 且电源资源将改变到例如 3V。可确定控制器 101 确保总线时钟资源和电源资源已完成这些转变将花费的时间量(“work\_time”)。(术语“工作”指代控制器 101 必须执行以便实行资源状态转变的处理、配置和硬件控制。)因此,为在时间  $t_{\text{deadline}}$  使资源处于资源状态组“R”指示的状态,在此实例中,控制器 101 需要在  $t_{\text{deadline}}$  至少等于 work\_time 之前的一时间量开始转变总线时钟和电源资源(图 5 中的步骤 530 和 535、图 8 中的步骤 830 和 835 等)的过程。

[0117] 在 PCD100 中,两个或两个以上处理器(例如,主处理器 110A、110B、110C 等,图 2 中)可请求在彼此非常接近的时间进行资源状态组转变,使得控制器 101 将需要致力于针对一个处理器转变资源同时致力于针对另一处理器转变资源。类似地,例如 SPM157 等另一元件可在控制器 101 正致力于转变资源或经调度以致力于转变资源的同时请求资源状态组转变。此类“冲突”条件是不合需要的,因为在示范性实施例中,控制器 101 不能同时执行这些任务。

[0118] 图 11 是说明上述冲突条件的实例的时间线。控制器 100 开始下文描述的调度方法且检测冲突条件的近似时间标记为“ $t_{\text{now}}$ ”。在图 11 所示的实例中,控制器 101 确定为在时间  $t_{\text{deadline}_0}$  使资源处于第一处理器所要求的状态中,控制器 101 需要在时间  $t_{\text{start}_0}$  开始将这些资源转变到所要求的状态的过程或工作(“work\_0”)。类似地,控制器 101 确定为在时间  $t_{\text{deadline}_1}$  使资源处于第二处理器所要求的状态中,控制器 101 需要在时间  $t_{\text{start}_1}$  开始将这些资源转变到所要求的状态的过程或工作(“work\_1”)。可注意,work\_0 与 work\_1 之间的重叠表示冲突条件。

[0119] 除了管理此冲突条件外,控制器 101 还可计算工作(“work\_0”和“work\_1”)的

准确估计值 - 表示图 11 中说明的工作的箭头的实际长度 / 持续时间 / 时间。关于控制器 101 如何计算工作的准确估计值的更多细节将在下文结合图 24-31 描述。

[0120] 图 12 以时间线形式说明用于缓解图 11 所示的冲突条件的方法。为缓解所述冲突, 控制器可调度在开始 work\_1 之前完成工作 work\_0。控制器 101 因此计算其将开始将这些资源转变到所要求状态以便在  $t_{start\_1}$  (即, 经修改截止时间  $t_{deadline\_0'}$ ) 之前完成工作 work\_0 的经修改时间  $t_{start\_0'}$  :

$$[0121] \quad t_{start\_0'} = t_{deadline\_0} - (t_{deadline\_1} - work\_1).$$

[0122] 可注意, 以上计算中的  $t_{start\_0'}$  相对于  $t_{now}$ 。控制器 101 将开始将这些资源转变到所要求状态以便在  $t_{start\_1}$  (即, 经修改截止时间  $t_{deadline\_0'}$ ) 之前完成工作 work\_0 的截止时间  $t_{start\_0'}$  的此计算由控制器 101 作出的准确工作负载估计进一步增强, 如下文结合图 24-31 描述。图 24-31 展示控制器 101 可如何计算表示图 11-12 中说明的工作的箭头的实际长度 / 持续时间 / 时间。

[0123] 图 13 是说明用于将处理器 110 从对应于休眠资源状态组的休眠应用状态转变到对应于现用资源状态组的现用应用状态的方法 1300 的逻辑流程图。方法 1300 类似于图 6 的上述方法 600, 只是方法 1300 包含调度控制器 101 执行以改变或转变资源状态的处理或工作。因为框 1305、1310 和 1315 分别与图 6 的框 605、610 和 615 相同, 所以此处不对其进行描述。在框 1318 中, 控制器 101 调度控制器 101 确定的针对一个或一个以上处理器的资源状态组转变将周期性改变应用状态。

[0124] 如上文描述, 应用状态的所预测改变具有相关联截止时间, 到所述截止时间时对应于下一应用状态的资源组的资源将完全转变。此调度步骤可包含计算资源状态组转变完成将花费的时间量 (“工作”), 以及控制器 101 必须开始转变过程或“工作”以便到截止时间时完成转变的时间。此调度步骤还可包含以上文描述的方式或使用替代方法缓解任何调度冲突。因为框 1320、1325 和 1330 分别与框 620、625 和 630 相同, 所以此处不对其进行描述。然而, 框 1322 提供在框 1320 与 1325 之间, 其中由控制器 101 基于实际工作负载值更新系统状态高速缓冲存储器 2705。框 1322 的更多细节将在下文结合图 27B 描述。

[0125] 图 14 是说明可包含在图 13 的框 1318 中以调度资源状态组转变的方法 1400 的逻辑流程图。例程或子方法框 1402 可包括通过利用控制器 101 (如图 27B 中说明) 产生散列值且将其与系统状态高速缓冲存储器 2705 比较而估计工作负载。例程或子方法框 1402 的更多细节将结合下文描述的图 27 和 31 进一步详细描述。框 1405 指示控制器 101 可评估以下表达式 :

$$[0126] \quad t_{deadline\_X} - work\_X < t_{deadline\_y},$$

[0127] 其中 x 和 y 是针对资源状态转变的两个请求 (例如, 来自第一处理器 X 和第二处理器 y), 且其中  $x > y$ 。如果所述表达式评估为假, 那么两个请求之间不存在冲突条件, 且方法结束。如果所述表达式评估为真, 那么存在上文参看图 11 描述的类型的冲突条件。如果确定存在冲突条件, 那么控制器 101 可在框 1410 中计算经修改开始时间以缓解冲突 :

$$[0128] \quad t_{start\_X'} = t_{deadline\_X} - (t_{deadline\_y} - work\_y).$$

[0129] 控制器 101 可用经修改开始时间替换原始调度的资源状态组转变开始时间。

[0130] 用于缓解调度冲突的方法还可考虑非调度资源状态组转变请求。如上文描述, 经调度资源状态组转变请求包含周期性发生或以其它方式可预测的请求。非调度资源状态组

转变请求可由于不可预测事件（例如，用户使用触摸屏 132（图 2）执行致使 PCD100 唤醒一个或一个以上处理器的动作）而发生。非调度请求不具有资源状态组转变必须完成的相关联截止时间（“ $t_{\text{deadline}}^*$ ”）。而是，其仅适当指代资源状态组转变如果在特定时间开始则将完成的时间（“ $t_{\text{done}}^*$ ”）。

[0131] 图 15 是说明在一旦非调度资源状态组转变请求在  $t_{\text{non-scheduled}_1}$  发生控制器 101 就开始处理（即，致力于）所述请求且继续致力于所述请求直到资源状态组转变在  $t_{\text{done}_1}$  完成为止的情况下可能发生的冲突条件的时间线。注意，在  $t_{\text{start}_0}$  开始且在  $t_{\text{deadline}_0}$  结束的经调度请求的处理（“work\_0”）与非调度请求的处理（“work\_1”）重叠。

[0132] 图 16 是说明用于缓解图 15 的冲突条件的简单示范性方法的时间线。为缓解冲突条件，控制器 101 可首先转变与经调度请求相关联的资源且接着转变与非调度请求相关联的资源。

[0133] 图 17 是说明用于缓解图 15 的冲突条件的第二简单示范性方法的时间线。为缓解冲突条件，控制器 101 可首先转变与经调度请求相关联的资源且接着转变与非调度请求相关联的资源。然而，不同于图 16 所示的方法，work\_0 的开始  $t_{\text{start}_0}$  较早移动到  $t_{\text{start}_0}'$  以允许较早完成 work\_1 以避免对非调度工作的延迟。

[0134] 图 18 是用于缓解图 15 的冲突条件的另一示范性方法的时间线。为缓解冲突条件，控制器 101 可首先计算经修改开始时间：

$$[0135] t_{\text{start}_1} = (t_{\text{deadline}_0} - \text{work}_0) - t_{\text{now}}.$$

[0136] 控制器 101 可在经修改开始时间  $t_{\text{start}_1}$  开始转变与非调度请求相关联的资源的工作的子集或一部分。接着，在  $t_{\text{start}_0}$  处，控制器 101 停止致力于转变与非调度请求相关联的资源，而是切换到转变与经调度请求相关联的资源。在控制器 101 在  $t_{\text{deadline}_0}$  处完成转变与经调度请求相关联的资源之后，控制器 101 可返回到转变与非调度请求相关联的资源的工作。

[0137] 图 19 说明转变与资源状态组改变请求相关联的资源的过程中涉及的工作或处理可在许多例子中划分为子集或部分“ $\text{work}_0$ ”到“ $\text{work}_N$ ”。转变与资源状态组改变相关联的资源的过程中涉及的工作或处理可涉及许多离散任务。因此，控制器 101 可容易地能够在此类离散任务之间临时中止转变到另一资源状态组的过程。举例来说，图 18 中  $t_{\text{start}_1}$  与  $t_{\text{start}_0}$  之间发生的处理或工作的部分可包括一个或一个以上此类离散任务。

[0138] 图 20 是说明工作的子集或一部分可比预期早完成从而导致比截止时间  $t_{\text{deadline}}$  早完成工作  $t_{\text{done}}$  的时间线。这可导致浪费功率，因为工作中涉及的资源比满足截止时间所需更早地消耗功率（如所属领域的一般技术人员所理解）。

[0139] 图 21 说明用于缓解图 20 的浪费功率条件的示范性方法。为缓解所述条件，可延迟或“拖延”在较早完成的工作的子集或部分之后的工作的后续子集或部分。可延迟“ $\text{work}_{N+1}$ ”直到“ $\text{work}_N$ ”预期完成为止，以避免由于在“ $\text{work}_N$ ”之后改变工作中的资源而产生的功率影响。

[0140] 图 22 更充分说明离散任务概念且展示例如可在  $t_{\text{start}_1}$  与  $t_{\text{start}_0}$  之间执行部分  $\text{work}_2$ 。应注意，由于在转变资源状态组的资源的过程中所涉及的一些离散任务不取决于其它任务，所以此类任务可以任何适宜的次序执行。因此，举例来说，尽管工作可在图 19 中展示为涉及循序任务，但在不按顺序执行任务（例如，在  $\text{work}_0$  之前执行  $\text{work}_2$ ）的一些

例子中可能无不利后果。还应注意，离散任务或部分可不具有彼此相等长度。因此，如果离散任务或部分的一者（例如，work<sub>2</sub>\_1）比资源状态组转变的其它部分更好地配合图 22 所示的实例中的 t<sub>start\_1</sub> 与 t<sub>start\_0</sub> 之间的时间间隔，那么控制器 101 可通过以此次序执行所述部分而优化方法。一般来说，可能需要尽可能早地对资源状态组转变执行尽可能最多的动作。因此，相比于在图 22 所示的实例中的 t<sub>start\_1</sub> 与 t<sub>start\_0</sub> 之间的时间间隔中执行较短部分且因此刚好在 t<sub>start\_0</sub> 之前留下不执行工作的间隙，可能更需要执行刚好配合所述时间间隔的较长部分。

[0141] 图 23 是说明用于调度资源状态转变的处理的方法 2300 的逻辑流程图。方法 2300 更一般地传达可为经调度或非调度的两个以上请求可需要同时处理的概念。（为清楚起见，上文参看图 11-22 描述的方法涉及仅处理一个或两个请求以及其间的冲突条件的概率。）

[0142] 方法 2300 在状态 2305 中开始，状态 2305 可由于以下条件的任一者已发生而达到：控制器 101 响应于请求而完成转变资源状态的过程中涉及的处理或工作；控制器 101 接收对于资源状态组转变的非调度请求；或控制器 101 确定用于处理资源状态转变的经调度开始时间（“t<sub>start</sub>”）即将来临。在框 2310（其表示方法 2300 的开始）中，控制器 101 确定任何处理或工作是否已调度，如上所述，此类处理或工作可经调度以按周期性间隔开始，但经调度开始时间可经修改以缓解冲突条件。

[0143] 如果控制器 101 确定该是执行此类经调度处理或工作的时间，那么控制器 101 执行如框 2315 指示的处理或工作。如果控制器 101 确定不是执行任何经调度处理或工作的时间，那么类似于图 14 的框 1402，在框 2317 处，控制器可通过产生散列值且将其与系统状态高速缓冲存储器 2705 比较来估计工作负载。将结合下文描述的图 27 和 31 进一步详细描述例程或子方法框 2317 的更多细节。

[0144] 在框 2320 中，控制器 101 可处理待决的任何非调度请求。可存在一个以上待决的非调度请求。并且，非调度请求可具有与其相关联的优先级。如果一个以上非调度请求待决，那么控制器 101 致力于最高优先级待决非调度请求的部分，从所述时间开始直到下一经调度工作开始时间（t<sub>start</sub>）为止。下一开始时间 t<sub>start\_next</sub> 为：

[0145]  $t_{start\_next} = (t_{deadline\_next}-work\_next)-t_{now}$ 。

[0146] 注意，以上计算中的 t<sub>start\_next</sub> 相对于 t<sub>now</sub>。

[0147] 当控制器 101 完成对与非调度请求相关联的工作的一部分（见图 19）的处理或工作时，控制器 101 确定所述处理或工作是否包含更多部分，如框 2325 指示。如果存在更多部分，那么控制器 101 以与上文参看框 2020 描述相同的方式致力于下一部分。上文的术语“最高优先级”指代可包含在一些实施例中的优先排序方案。举例来说，可向源自用户“切断”PCD100（即，经由触摸屏 132（图 1）起始低功率状态）的非调度请求指派比其它非调度请求低的优先级。

[0148] 当工作完成时，在框 2330 中，类似于图 13 的框 1322，系统状态高速缓冲存储器 2705 由控制器 101 基于实际工作负载值更新。下文将结合图 27B 描述框 1322 的更多细节。

[0149] 图 24 是说明可由便携式计算装置 100 中的一个或一个以上资源 105 处理的各种工作负载的图形。在图 24 中说明的示范性实施例中，描绘四个工作负载：工作负载 A、工作负载 B、工作负载 C 和工作负载 D。

[0150] 所估计截止时间 2405A 可对应于工作负载 D 必须在此截止时间 2405 内完成的时

间（可影响其它工作负载的开始和结束时间）。在一些情况下，截止时间 2405 可致使其它工作负载比不存在工作负载 D 的情况下原本的时间更早开始。

[0151] 如果系统 103（且特定来说，如果控制器 101）不具有总工作负载（总计的工作负载 A-D）或完成所述各个工作负载的时间的准确估计，那么控制器 101 可产生次佳调度，这可导致额外功率被系统 103 不必要地消耗。举例来说，假定截止时间 2405A 是系统产生的所估计截止时间。

[0152] 且假设在第二图形中，说明所记录的完成工作负载的实际时间，其中第二截止时间 2405B 描绘在所有工作负载完成之后在系统 103 内发生实际截止时间。系统 103 将继续为与第四工作负载 D 相关联的资源加电直到所估计截止时间 2405A 为止，这将产生系统的所浪费功率（由框 2410 表示）。

[0153] 在最佳工作负载情境中，资源 105 可由系统 103（且特定来说，控制器 101）接通得越迟，则将有越少功率被系统 103 消耗，如所属领域的一般技术人员所理解。在此最佳情境中，功率成本与具有工作负载的不太准确估计值的系统相比将较低（为关于完成工作负载时的时间的估计值）。

[0154] 在同时工作情形中，准确的工作负载估计致使系统 103 更准确地确定是否存在工作负载的实际同时性或工作负载的重叠使得系统需要将内容较早调度或将开始时间较早推送到时间线以便确保满足所有截止时间。工作负载估计值越准确，则工作负载重叠将越乐观且更准确，且因此，控制器 101 将通常不会早于其需要完成同时工作负载的时间接通资源 105。

[0155] 在系统 103 的示范性实施例中，控制器 101 使用散列值来表示系统状态以及转变。控制器 101 对当前系统状态以及转变执行散列操作。转变可包含如上文参看图 3-4 描述的休眠和现用组。控制器 101 还对客户端产生的去往一个或一个以上资源 105 的传入请求执行散列操作。

[0156] 如图 1-3 中说明的控制器 101 计算上文描述的散列值。控制器 101 正取通常由在示范性实施例中具有约 800 字节长度的数据结构表示系统的状态的元素，且将其缩减为具有约 4 字节或约 32 位长度的数据结构。在一些示范性实施例中，这些数据结构或散列值可具有 64 位长度。这些数据结构的其它尺寸在本发明的范围内，如所属领域的一般技术人员所了解。这些缩减的数据结构可存储在系统状态高速缓冲存储器 2705（见图 27B）中，其中每一较小数据结构具有指派到其自身唯一识别符的约 4 字节。

[0157] 整个系统 103 的状态可容易由具有约 800 字节长度的数据结构表示。同时，从客户端发起的请求可包括具有达 800 字节的数据结构，但此类请求可大大少于约 800 字节。

[0158] 控制器 101 可使用混合函数和组合函数作为其散列 2600 的一部分（见图 26）。散列 2600 的混合函数强调单一位差异，如所属领域的一般技术人员所了解。举例来说，在两个不同系统状态之间（其中每一者可用具有约 800 字节大小的数据结构表示），两个状态之间的差异可非常微小且可归结为大小方面约 2 或 3 字节的差异。

[0159] 同时，散列 2600 的混合函数试图强调系统状态之间的那些单一位差异，其在其表征方面彼此非常接近，具有拥有约 800 字节大小的数据结构。混合函数产生唯一散列值，这通常改进系统散列值的准确性。

[0160] 关于组合函数，此函数使用一状态拓扑，所述状态拓扑致使系统状态之间的分组

在散列值中产生更大唯一性。组合函数的一个目标是取 N 状态且将其归结到单一 1 状态表示不。

[0161] 图 25 是说明根据系统 103 的一个示范性实施例状态拓扑内的分组的图表 2500。图 25 例示系统内的元素之间的差异可如何非常微小。举例来说, 第一钟“a”可具有 200MHz 的频率, 而第二时钟“b”具有 100MHz 的频率。类似地, 第一电压调整器“a”可具有 1000mV 的设定, 而第二电压调整器“b”可具有 1015mV 的设定。

[0162] 图 26 是说明控制器 101 使用的散列函数 2600 的一个实施例的示范性特征的图表。如先前所述, 散列函数 2600 的示范性实施例使用混合和组合函数。组合函数向将系统状态表示为旋转函数的种子的数据结构中使用指派到每一资源 105 的唯一指数, 这在图 26 中说明的散列 2600 的示范性实施例中包括将所述指数乘以 13 模数乘以 32。特定来说, 在图 26 的散列 2600 的示范性实施例中, 选择数字 13 和 32, 因为其相对具有代表性且可产生近似 32 个资源上一组唯一旋转。

[0163] 已发现, 便携式计算装置 100 中通常存在类似且在一拓扑中分组的少于 32 个资源。组合函数内的旋转函数允许关于每一资源影响散列 2600 内的哪些位的一定程度的随机化。这允许例如通过在散列 2600 内的不同位周围散布时钟读数而非集中散列 2600 的前四个或前五个位内的所有时钟读数差异, 来从散列 2600 实现更大唯一性。

[0164] 组合函数正使用每一资源唯一指数的旋转来扭曲系统元素的每一者使得其处于不同旋转角。换句话说, 通过使用每一唯一资源识别符 (其也可称为状态拓扑指数) 用于旋转, 可在状态元素的分组上实现更大唯一性。旋转允许将元素与类似状态组合, 同时使此组合将抵消重要信息的任何几率最小化。

[0165] 通过以此方式进行旋转, 当系统元素压缩时, 其差异 (尽管如上文论述是微小的) 不会重叠那么多, 如所属领域的一般技术人员所理解。在不旋转的情况下, 当状态之间的微小差异组合时, 状态信息可在散列 2600 中丢失。

[0166] 每当系统的当前状态改变时, 可更新散列 2600。在下文描述的图 27-31 中说明的示范性实施例中, 当存在更新时, 整个系统 103 不再次执行散列操作。仅系统 103 的已改变的部分用散列 2600 更新。所属领域的一般技术人员认识到, 在其它示范性实施例中, 如果此实施例对系统 103 提供了一些优点, 那么整个系统 103 可再次执行散列操作。

[0167] 在大多数常规散列中, 使用反馈。所述反馈通常包括散列更新之前散列的结果。此反馈实施例包含所属领域的一般技术人员所理解的比如 MD5 和 SHA。

[0168] 在其它散列系统中, 可不使用反馈, 其中使用 XOR (异或) 技术在旧信息上组合新信息, 如所属领域的一般技术人员所理解。然而, 在不使用反馈的此类散列系统中, 可发生新信息与旧信息之间的显著量的冲突。

[0169] 在上文描述的示范性散列 2600 中, 系统 103 (且特定来说, 控制器 101) 正考虑字段如何具备唯一识别符 (但字段的内容可彼此相似)。控制器 101 可通过使用其唯一指数旋转这些相似点而减轻这些相似点。虽然字段的状态可具有类似值, 但利用唯一识别符, 控制器 101 辨识每一字段表示单独且不同资源 105 的状态。以此方式, 提供可更新散列 2600, 其类似于如上文描述具有反馈回路的散列操作。

[0170] 散列 2600 允许系统快速识别当前存在什么系统状态。控制器 101 维持系统状态高速缓冲存储器 (见图 27B), 其记录针对实现结果和系统状态的各个工作负载所花费的实

际时间。系统状态高速缓冲存储器 2705 中针对各个工作负载所花费和记录的实际时间与上文提及的图 13 的框 1322 和图 23 的框 2330 对应。系统 103 的示范性实施例中的散列允许控制器 101 识别系统 103 处于的当前状态。

[0171] 散列允许当发生状态改变时跟踪系统状态。利用本文描述的可更新散列 2600，所述操作相对于当发生系统状态的改变时更新散列所需的计算能力并非非常昂贵的操作。

[0172] 举例来说，对于相对于作为现用组的当前组将在将来发生的例如休眠组等一组状态，可针对将来将发生的休眠组产生散列 2600。休眠组的散列 2600 可在休眠组实际上实施之前发生对休眠组的改变时更新。散列 2600 可设定为仅当将针对可包含对资源 105 的改变的给定状态发生即刻改变时更新。此针对即刻请求对于散列 2600 的更新可经执行以确定即刻请求是否可影响当前调度的任何其它将来请求的状态（例如，上文描述的休眠组实例）。

[0173] 通过以递归方式计算散列值以解决系统 103 内的多个经调度转变，来自每一转变的所计算系统状态可用于确定将来转变的初始系统状态使得较准确的基于高速缓冲存储器 / 散列的工作负载估计用于每一将来转变。工作负载估计越准确，则可通过较接近其被系统所需要接通资源来作出越高效调度，这可减少系统的总体功率消耗。此外，系统状态的估计越快，则越少开销可考虑到调度中。

[0174] 图 27A 是说明可在图 27B 的系统状态高速缓冲存储器 2705 中使用的元件的图表 2702。大写字母“S”可表示系统状态散列 2600，而小写字母“a”可表示子状态或动作（请求）的散列 2600 的结果。控制器 101 使用系统状态和子状态来寻找系统状态高速缓冲存储器 2705 中的匹配使得可找到工作负载（以时间 ( $t_1$ ) 为单位）。

[0175] 图 27B 说明跟踪当前系统状态（由系统状态列 2710 中的大写字母“S”表示）以及子状态或动作（例如，系统状态列 2715 中的小写字母“a”表示的传入请求或休眠组、现用组）用于计算所得系统状态（由结果系统状态列 2725 中的大写字母“S”表示）的系统状态高速缓冲存储器 2705。系统状态高速缓冲存储器 2705 中的子状态或动作值“a”是散列 2600 及其由上文描述的控制器 101 作出的计算的结果。

[0176] 针对散列 2600 的计算由控制器 101 从希望将来完成的请求组（例如，现用或休眠组）或希望几乎即刻完成的请求组导出。系统考虑希望几乎即刻完成的请求如何可影响将在不远的将来发生的将来或经调度请求。

[0177] 控制器 101 还跟踪针对所述转变的最差情况工作负载估计（以时间为单位）（由估计列 2720 中的小写字母“t”表示）。系统状态高速缓冲存储器 2705 由控制器 101 通过使用散列 2600 及其计算而持续更新。当接收到传入请求时，计算列 2715 中的子状态或“a”值。也可当控制器 101 接收到请求时更新列 2715 中的子状态或“a”值。

[0178] 系统状态高速缓冲存储器 2705 基本上是针对系统记录的过去已发生的历史。因此举例来说，取系统状态高速缓冲存储器 2725 的第一行，在过去，接收子状态或动作 “ $a_1$ ” 的系统状态  $S_1$  产生系统状态  $S_2$ ，且达到此新的系统状态  $S_2$  所花费的时间为时间  $t_1$ 。

[0179] 系统通过将当前系统状态与系统状态列 2710 中的每一值比较且通过将当前子状态或动作与系统状态列 2715 中的每一值比较以便找到匹配行而使用系统状态高速缓冲存储器 2705。一旦系统识别对应的系统状态值 2710 和子状态值 2715，系统就可使用来自时

间估计列 2720 的时间估计“ $t$ ”。系统还可使用来自列 2725 的所得系统状态的估计。

[0180] 对于新的系统状态,例如可能从未在系统状态高速缓冲存储器 2705 中记录的系统状态  $S_5$ ,或如果其是太旧且由系统状态高速缓冲存储器 2705 丢弃或替换的系统状态,那么控制器可执行充分计算以确定对于列 2720 以及列 2725 中记录的所得系统状态的时间估计。

[0181] 插入在系统状态高速缓冲存储器 2705 中的每一时间估计“ $t$ ”在列 2715 的实际动作或子状态应用于列 2710 中列举的当前系统状态之前使用由控制器 101 计算的最差情况情境。一旦列 2715 的实际动作或子状态应用于当前系统状态,就记录完成所述实际动作或子状态所花费的实际时间且将其放置到系统状态高速缓冲存储器 2705 中(在其长于系统确定的先前估计的情况下,其为实际时间)。这允许对系统状态高速缓冲存储器 2705 中记录的时间估计“ $t$ ”的持续改进。

[0182] 所属领域的一般技术人员将认识到,代替于用针对当前动作花费的实际时间更新系统状态高速缓冲存储器 2705 中在列 2720 中记录的时间估计“ $t$ ”,所保存的时间估计可包括滑动窗平均值、加权系统平均值、系统观察到的最差情况时间,和其它类似值。在图 27 中说明的示范性实施例中,系统观察到的最差情况时间记录在时间估计列 2720 中。

[0183] 此最差情况时间值可确保时间估计决不小于系统 103 基于所发布的请求执行的实际工作负载。已发现,此最差情况时间值不允许系统过高估计工作负载超过近似 1%使得工作负载准时调度且准时完成。

[0184] 图 28 是可在图 27B 的系统状态高速缓冲存储器中找到的系统状态转变的图形表示。举例来说,图 28 说明系统状态  $S_1$  可如何接收散列函数计算的动作或子状态  $a_1$ 。参看图 27B,系统状态  $S_1$  与子状态  $a_1$  之间存在匹配。系统状态高速缓冲存储器 2705 规定,将此动作  $a_1$  应用于当前系统状态  $S_1$  将产生具有工作负载估计  $t_1$  的第二系统状态  $S_2$ 。类似地,依据接收动作  $a_2$  的第二系统状态  $S_2$ (如图 28 中说明),系统状态高速缓冲存储器 2705 中也存在匹配,其展示来自此动作  $a_1$  的所得系统状态是具有工作负载估计  $t_2$  的第一系统状态  $S_1$ ,等等。

[0185] 图 29 说明包括三个示范性工作负载的同时工作负载情境:第一经调度工作 A2905A、第二经调度工作 B2905B 和第三经调度工作 C2905C。每一相应工作负载可具有由虚线表示的其自身唯一完成截止时间 A、B 和 C。

[0186] 可由控制器 101 针对每一工作负载计算最差情况时间  $t_w$ 。由于第三经调度工作 C2905C 的最差情况时间  $t_{cw}$  具有特定长度或持续时间,所以控制器 101 的此计算的最差情况时间  $t_{cw}$  致使经调度工作 B2905B 大大早于其经调度截止时间 B 开始和完成(如图 29 中说明)。

[0187] 类似地,由于第二经调度工作 B2905B 的最差情况时间  $t_{bw}$  具有特定长度或持续时间,所以此计算的最差情况时间  $t_{bw}$  致使经调度工作 A2905A 大大早于其经调度截止时间 A 开始和完成(如图 29 中说明)。此表示第三经调度工作 C2905C 的最差情况时间  $t_{cw}$  可如何影响第一经调度工作 A2905A 的开始和完成时间以及第二经调度工作 B2905B 的开始和完成时间。

[0188] 图 30 说明在特定状态转变已与系统状态高速缓冲存储器 2705 匹配之后图 29 的同时工作负载情境。在图 30 的示范性实施例中,假定系统状态高速缓冲存储器 2705 具有

针对第一工作 A2905A 和第二工作 B2905B 的值。

[0189] 系统状态高速缓冲存储器 2705 中的值可包括当前系统状态  $S_1$ , 其中第一工作 A2905A 由子状态  $a_1$  表示, 所述子状态  $a_1$  取时间  $t_1$  的所估计工作负载以达到所得第二系统状态  $S_2$ 。依据第二系统状态  $S_2$  且应用子状态  $a_3$  表示的第二工作 B2905B, 在时间  $t_2$  的所估计工作负载上达到所得第四系统状态  $S_4$ 。

[0190] 同时, 对于子状态  $a_5$  表示的第三工作负载 C2905C 的值,  $t_{cw}$  表示的所估计工作负载和所得系统状态是未知的。换句话说, 子状态  $a_5$  及其对应工作负载  $t_{cw}$  的值和所得系统状态不填充在图 27B 的系统状态高速缓冲存储器 2705 中。

[0191] 系统通常通过将第二工作负载 B2905B 的完成时间与第三工作负载 C2905C 的开始时间比较而向后工作。然而, 第二工作负载 B2905B 受如图 29 中说明的第一工作负载 A2905A 的完成时间影响。

[0192] 由于第一工作负载 A2905A 和第二工作负载 B2905B 的值是已知的且填充在系统状态高速缓冲存储器 2705 中, 所以控制器 101 辨识第一工作负载 A2905A 将在所估计完成时间 3010A(其也是在此实例中第一工作负载 A2905A 必须完成的实际截止时间 3015A)准时完成而无与第二工作负载 B2905B 的任何重叠。类似地, 控制器 101 辨识第二工作负载 B2905B 将在所估计完成时间 3010B 且实际上在其经调度截止时间 3015B 之前准时完成。

[0193] 因此, 控制器 101 可基于第二工作负载 B2905B 的所估计完成时间 3010B 估计第三工作负载 C2905C 的最差情况工作负载  $t_{cw}$ 。控制器 101 递归经过或检查如上文描述的状态以实现工作完成时间的较好估计, 使得资源 105 较接近其实际上被需要且仅针对实际上需要其来服务于一个或一个以上请求的时间(持续时间)而接通或利用。

[0194] 在图 30 的实例中, 发现基于时间  $t_2$  的第二工作负载 2905B 的所计算所估计开始时间 3005B 不与第一工作负载 2905A 的所估计完成时间和截止时间 3010A、3015A 重叠或对其干扰。在此示范性实施例中, 因而控制器 101 可接着聚焦于第三工作负载 C2905C 的计算。

[0195] 图 30 对应于上文描述的图 12 之处在于, 控制器 101 现具有来自系统状态高速缓冲存储器 2705 的允许控制器 101 计算图 12 中描绘的工作负载的实际长度或时间(箭头的长度)的信息。换句话说, 先前, 控制器 101 不具有与上文描述的图 12 一起使用的工作负载和所估计完成时间的准确估计。仅最差情况情境计算已先前结合图 12 使用。利用系统状态高速缓冲存储器 2705, 控制器 101 关于所估计完成时间的较准确计算是可能的, 因为图 27B 的系统状态高速缓冲存储器 2705 持续改善, 如所属领域的一般技术人员所理解。

[0196] 依据图 12, 这意味着 work\_0 和 work\_1 的实际长度(以时间单位计算的箭头的长度, 比如秒或毫秒)可提供在图 27B 的系统状态高速缓冲存储器 2705 中。此外, 系统状态高速缓冲存储器 2705 提供在开始时间  $tstart_0'$  以及在  $tdeadline_0'$  和  $tstart_1$  及  $tdeadline_1$  系统 103 的状态的信息。

[0197] 图 30 反映添加不同工作负载可如何影响系统 103 的状态。控制器 101 必须考虑经添加以供完成的每一工作负载。以及对于工作负载的同时性, 可在稍后时间添加的每一额外工作负载可需要重新计算将与新添加的工作负载组合完成的所有工作负载的开始和完成时间, 如所属领域的一般技术人员所理解。利用较准确且基于系统状态的工作负载估计, 这允许控制器 101 准确估计所有经调度任务的开始时间和完成时间。

[0198] 如先前所述, 控制器 101 可通过复查并评估不论何时将新任务和 / 或请求添加到

调度表时将被调度的所有请求和 / 或任务而计算散列 2600。或者,且在上文描述的示范性实施例中,控制器 101 仅基于所接收的新任务和 / 或请求且通过确定每一个别任务和 / 或请求可如何具有关于经调度的其它任务和 / 或请求的影响(特定来说,新任务和 / 或请求可如何影响其它任务和 / 或请求的开始时间和 / 或完成时间)而计算对散列 2600 的更新。换句话说,在上文描述的示范性实施例中,仅评估针对系统散列 2600 的变量(改变)(产生部分散列),而非调度表中的整个任务和 / 或请求(并非系统 103 的完整散列 2600)。在上文描述的示范性实施例中,转变的变量或差异显著小于可具有 800 字节的存储器大小的总系统状态。

[0199] 控制器 101 使用散列值表示系统状态且跟踪状态之间的转变。系统正使用散列 2600 和系统状态高速缓冲存储器 2705 一起来获得对于时间上(依据时间)共同定位的事件的准确估计以便加速便携式计算装置 100 的资源 105 之间任务和 / 或请求的处理。

[0200] 图 31 是用于通过产生散列值且将散列值与系统状态高速缓冲存储器 2705 比较而估计工作负载的图 14 的子方法或例程 1402 以及图 23 的子方法 2317 的流程图。框 3105 是子方法 1402、2317 的第一框。在框 3105 中,控制器 101 基于如图 26 中说明的散列 2600 产生整个系统 103 的散列值。通常,此框 3105 仅由控制器 101 执行一次。

[0201] 接下来,在框 3110 中,控制器 101 基于传入请求和 / 或任务使用散列 2600 更新散列值。在框 3115 中,控制器 101 将已经计算的散列值与存储在系统状态高速缓冲存储器 2705 中(如图 27B 中说明)的散列值比较。在框 3120 中,控制器 101 确定图 27B 的系统状态高速缓冲存储器 2705 内是否存在值之间的匹配。

[0202] 在框 3125 中,控制器 101 基于高速缓冲存储器 2705 内找到的任何匹配且基于控制器 101 可能需要执行的针对非匹配的任何计算估计工作负载。如先前所述,子方法 1402、2317 可通过在任何工作需要由资源 105 的一者或一者以上执行之前返回到框 3110 而执行部分更新。

[0203] 本说明书中描述的过程或过程流程中的某些步骤自然先于其它步骤以使得本发明如所描述发挥作用。然而,本发明不限于所描述的步骤的次序,只要此类次序或序列不更改本发明的功能性。即,应认识到,一些步骤可在其它步骤之前、之后或与其它步骤并行(大体同时)执行,而不脱离所揭示的系统和方法。在一些例子中,某些步骤可省略或不执行而不脱离如所属领域的一般技术人员所理解的方法。此外,例如“随后”、“接着”、“接下来”等词语不希望限制步骤的次序。这些词语仅用于引导读者通读示范性方法的描述。

[0204] 鉴于以上揭示内容,编程领域的一般技术人员能够编写计算机代码或识别适当硬件和 / 或电路来基于例如本说明书中的流程图和相关联描述容易地实施所揭示的发明。因此,对一组特定程序代码指令或详细硬件装置的揭示不应认为对于适当理解如何制造和使用本发明是必需的。在以上描述中且结合图式(其可说明各种过程流程)更详细阐释所主张的计算机实施的过程的创新性功能性。

[0205] 在一个或一个以上示范性方面中,可以硬件、软件、固件或其任何组合来实施所描述的功能。如果实施于软件中,那么可将所述功能作为一个或一个以上指令或代码存储在计算机可读媒体上。计算机可读媒体可包含可由计算机存取的任何可用非暂时性媒体。借助实例而非限制,此类计算机可读媒体可包括 RAM、ROM、EEPROM、CD-ROM 或其它光盘存储装置、磁盘存储装置或其它磁性存储装置,或可用以携载或存储呈指令或数据结构形式的所

要程序代码且可由计算机存取的任何其它媒体。

[0206] 如本文使用的磁盘和光盘包含压缩光盘（“CD”）、激光光盘、光学光盘、数字多功能光盘（“DVD”）、软性磁盘和蓝光光盘，其中磁盘通常以磁性方式再现数据，而光盘利用激光以光学方式再现数据。以上的组合也应包含在计算机可读媒体的范围内。

[0207] 因此，尽管已详细说明和描述选定方面，但将理解，可在不脱离如所附权利要求书界定的本发明的精神和范围的情况下在其中作出各种替换和更改。

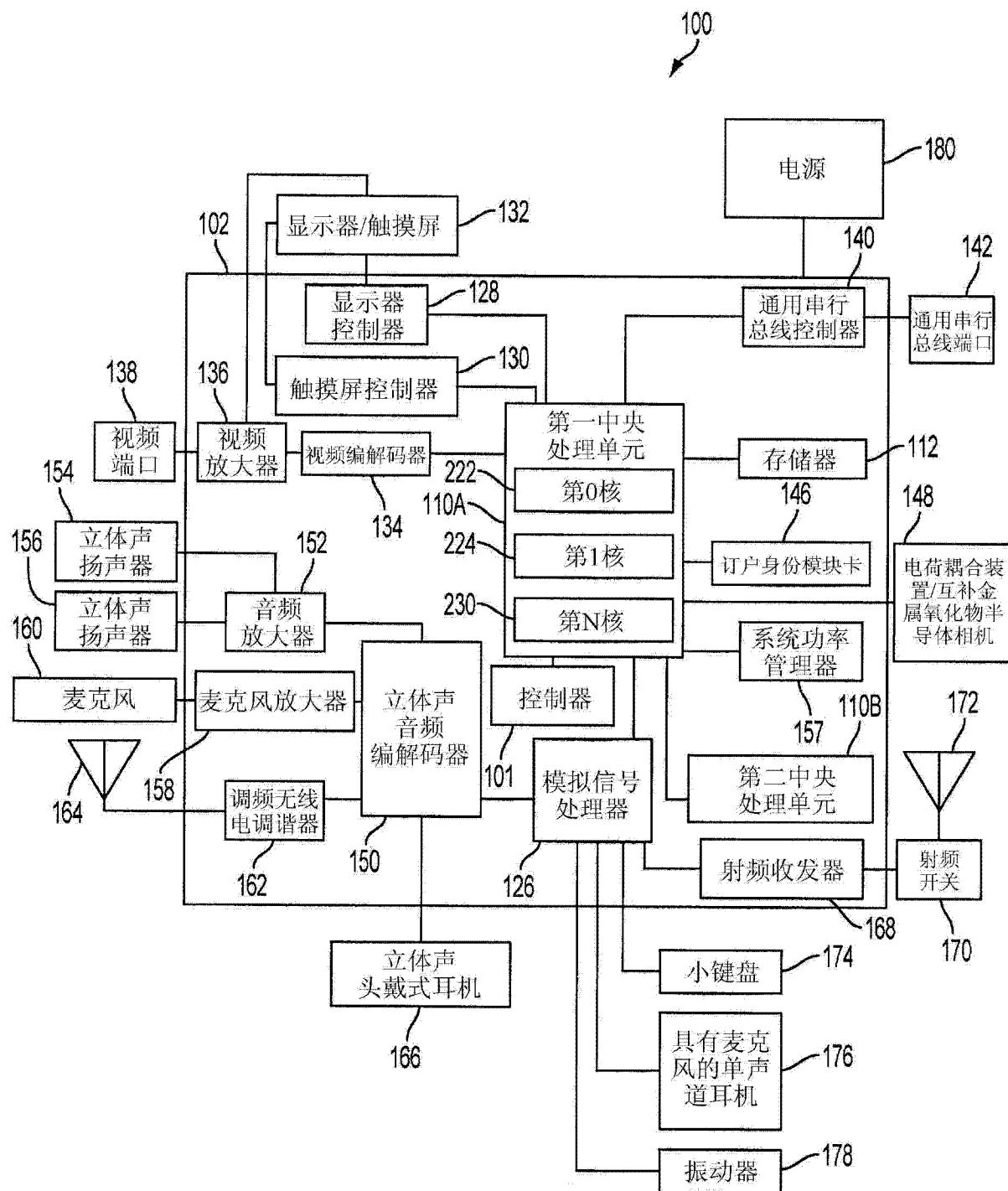


图 1

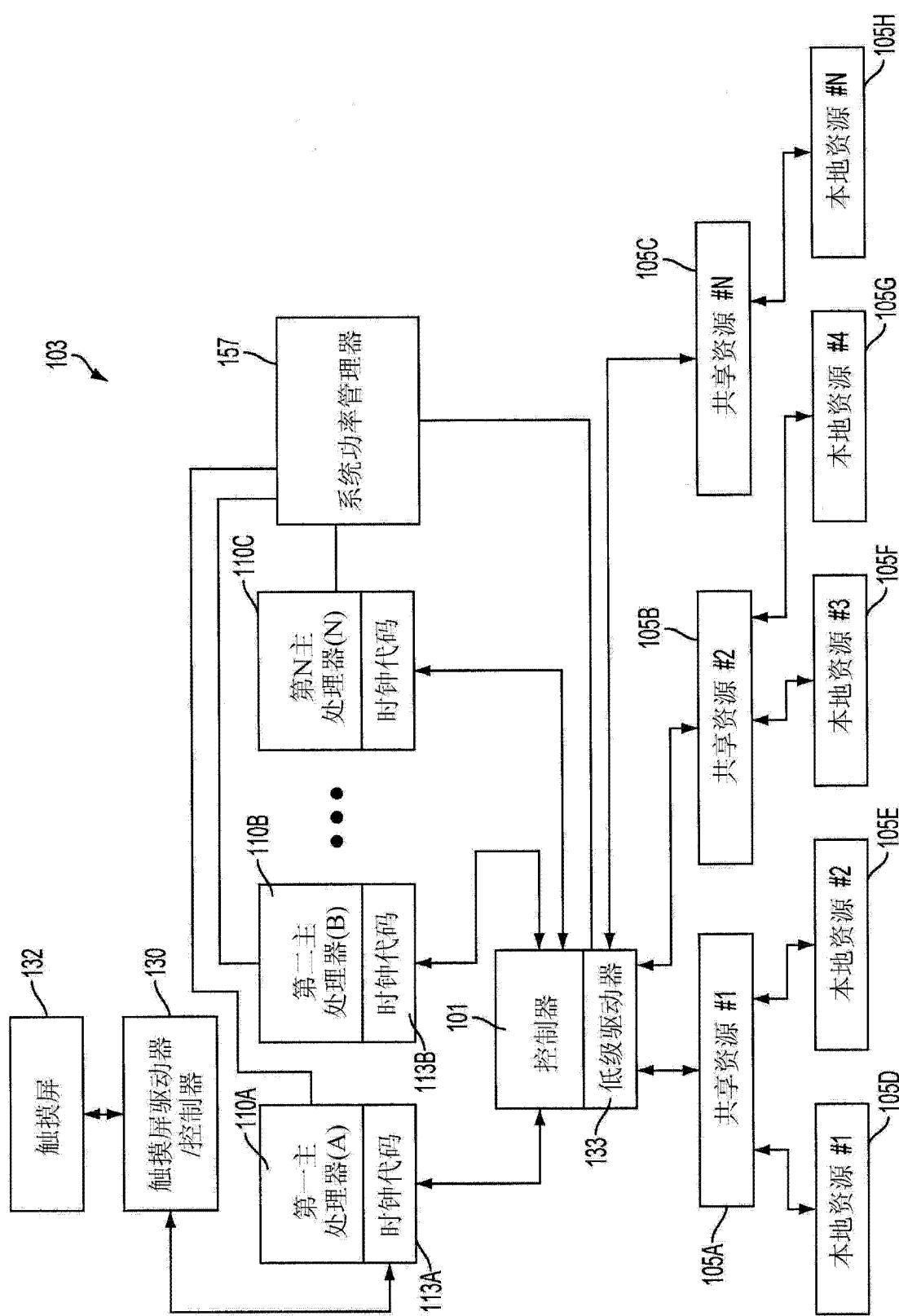


图 2

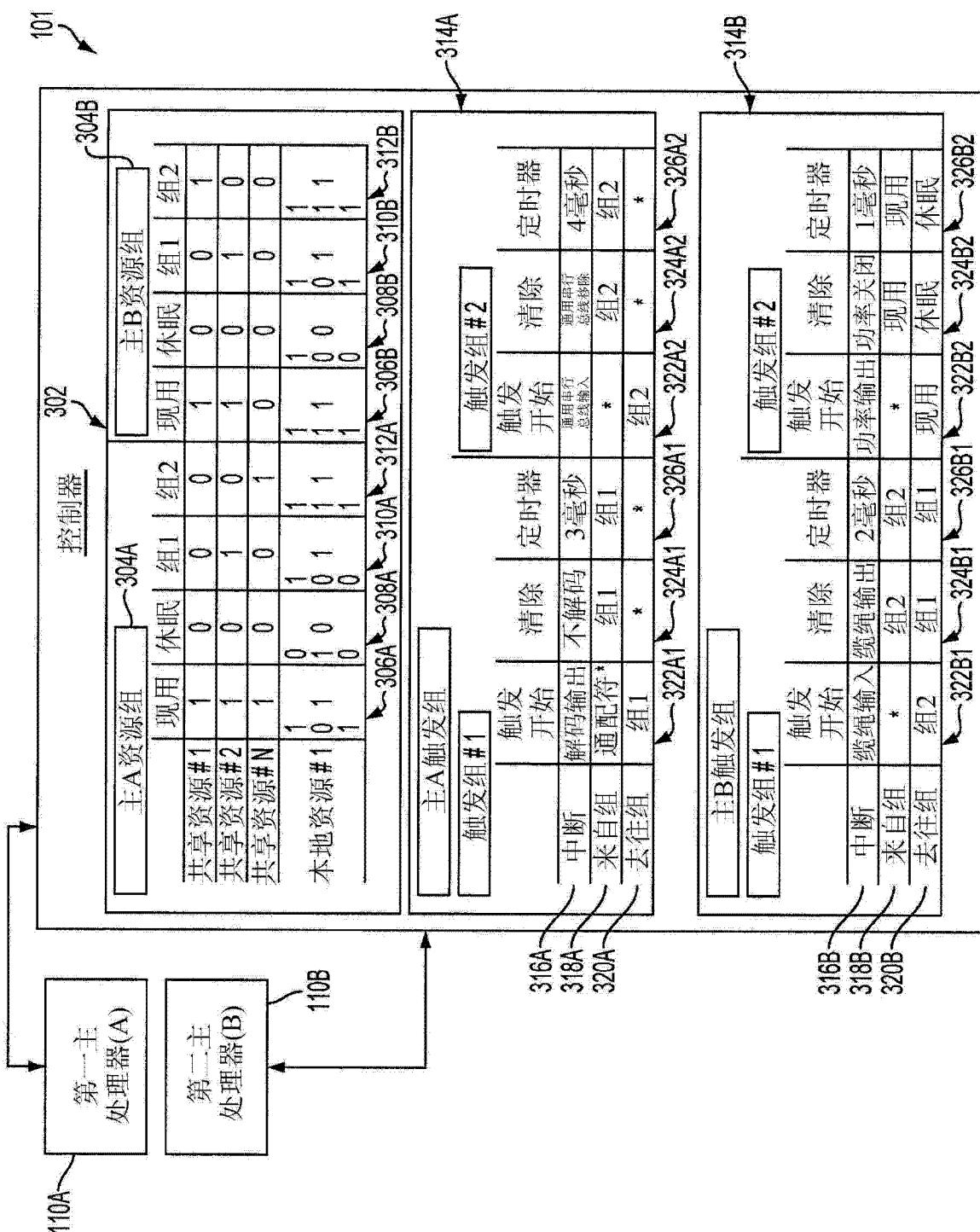


图 3

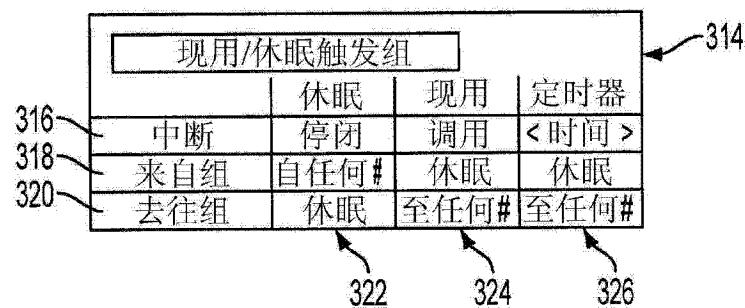


图 4

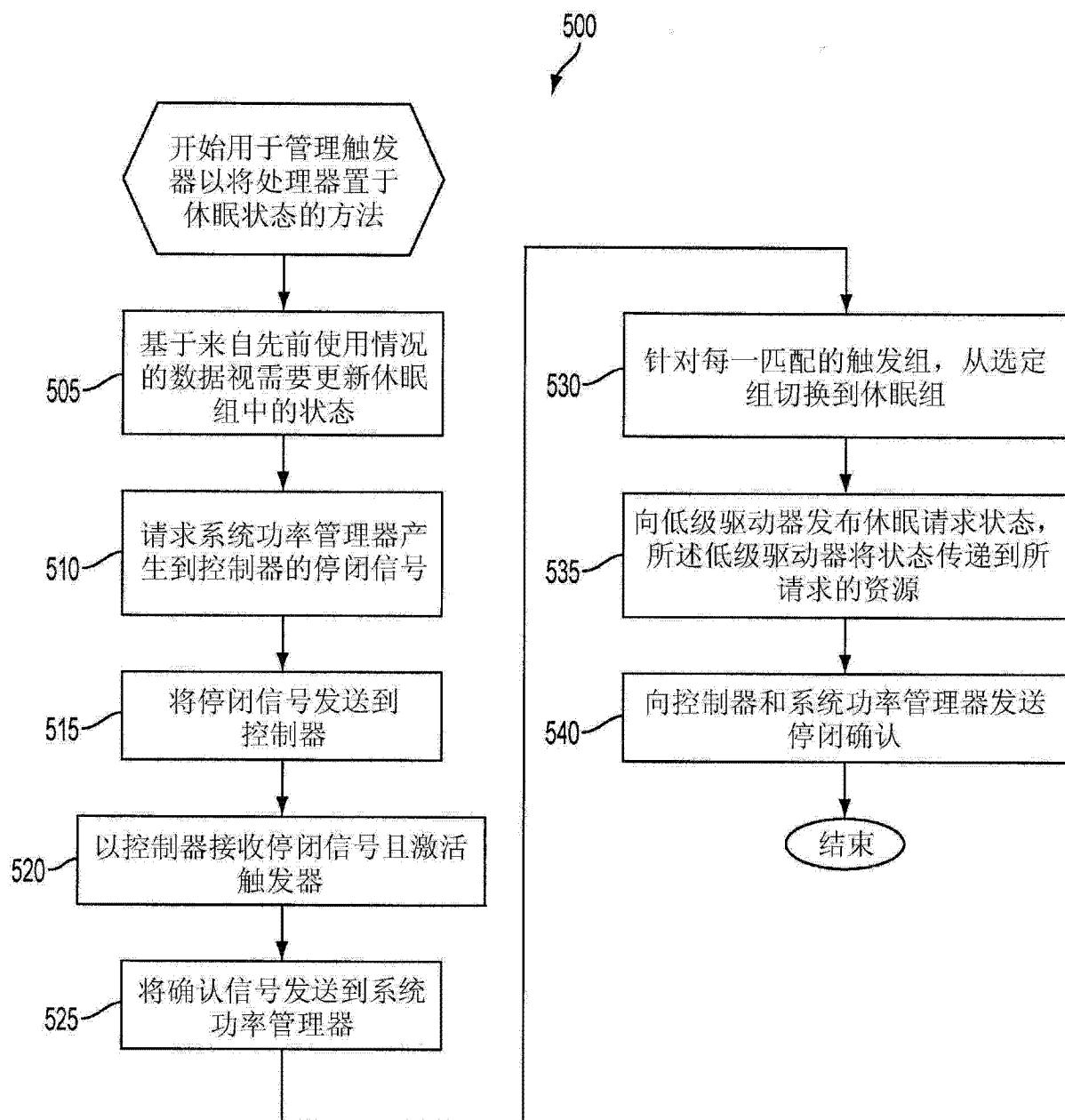


图 5

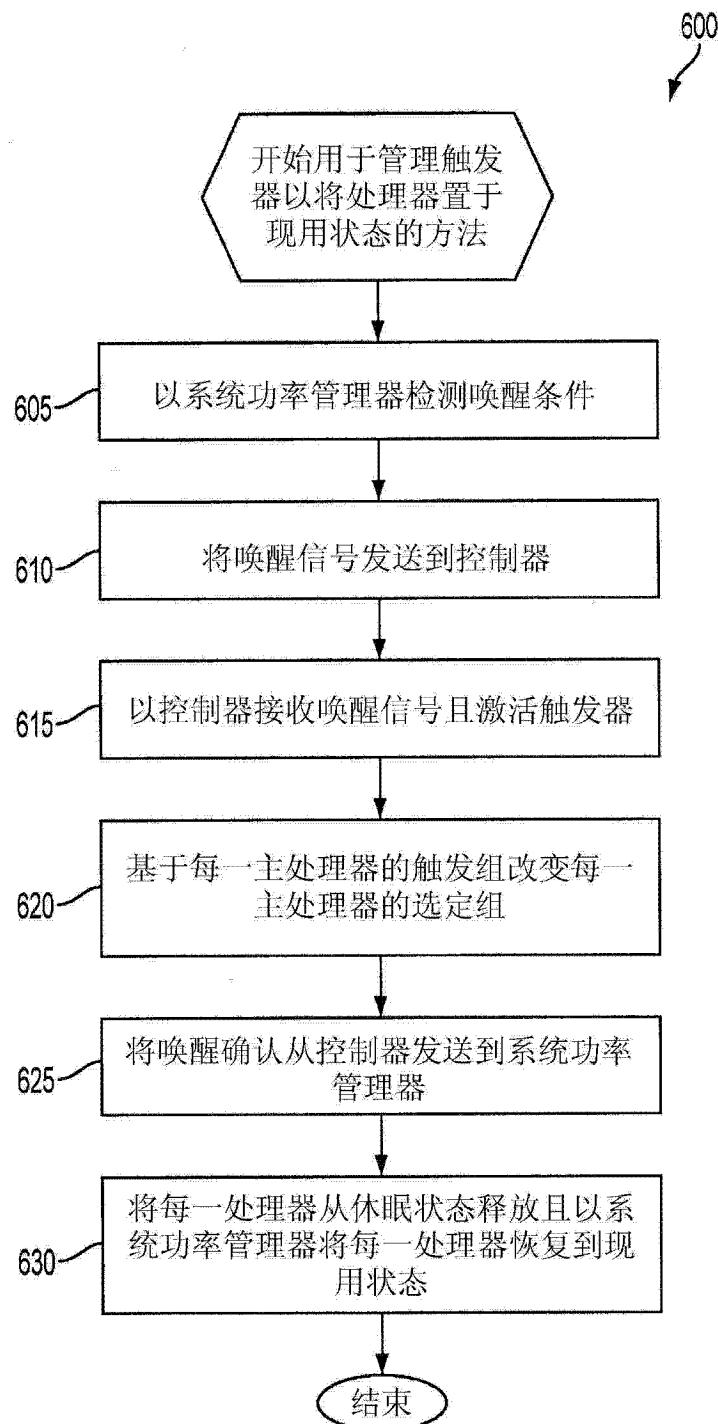


图 6

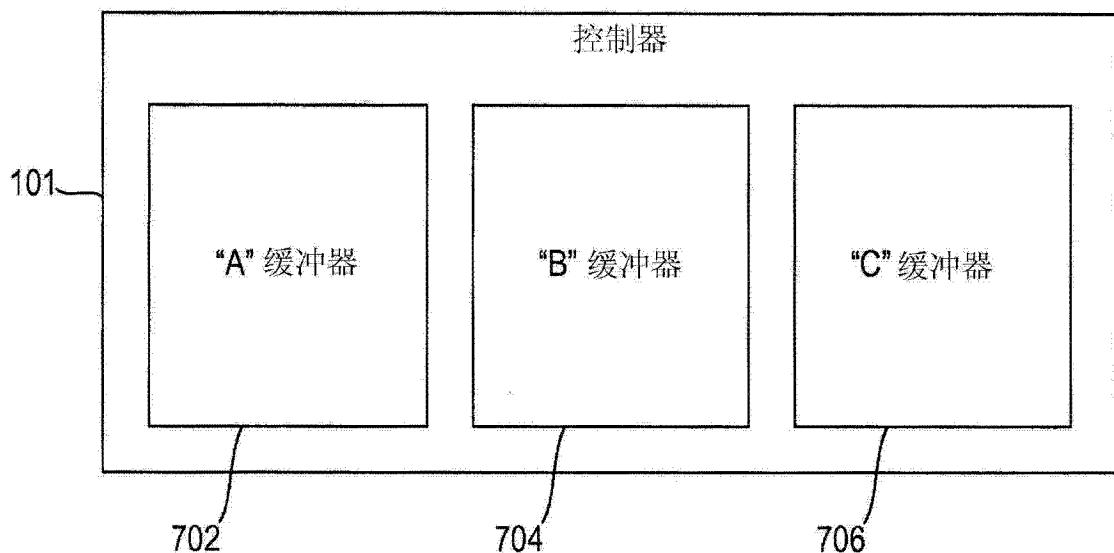


图 7

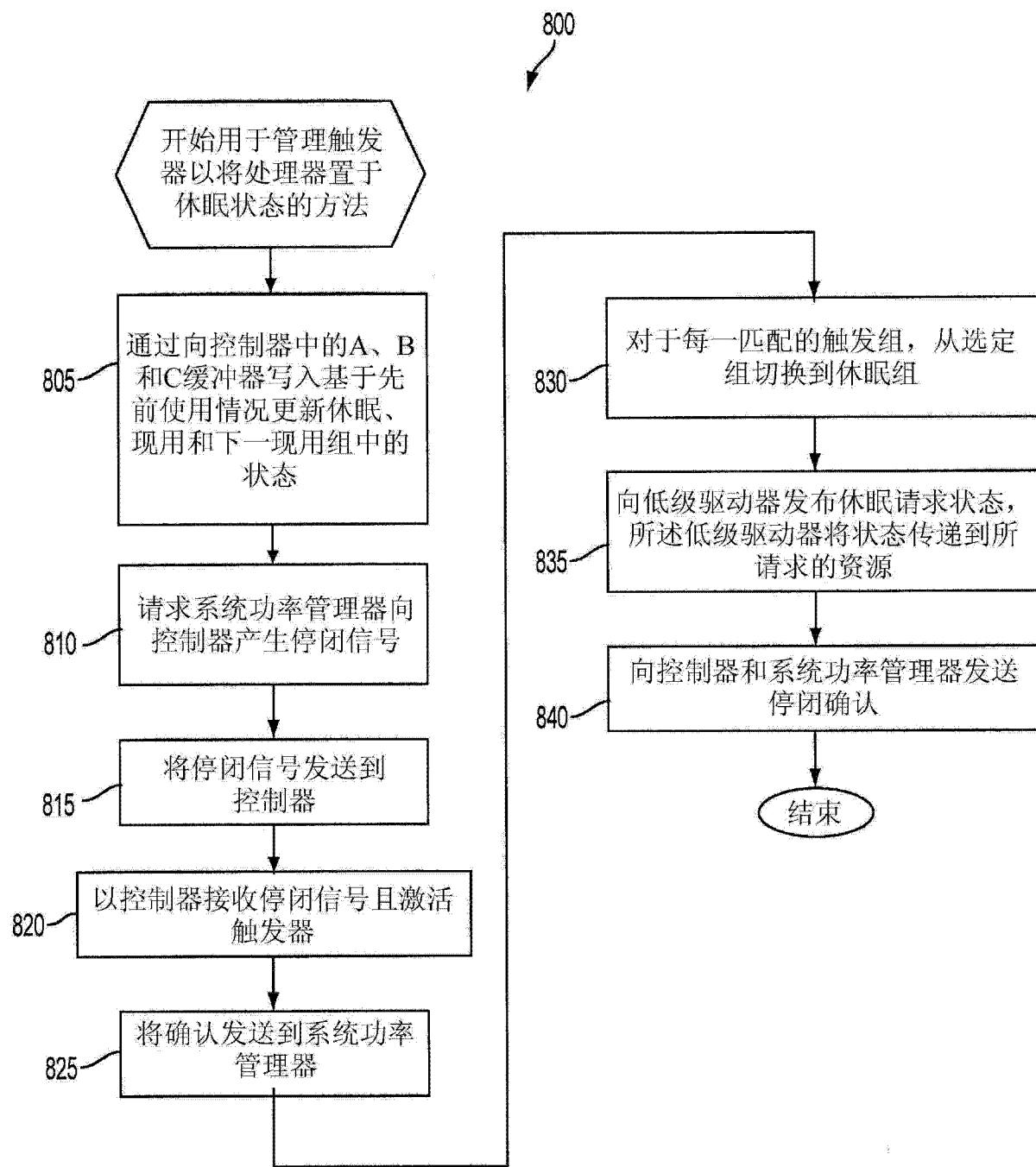


图 8

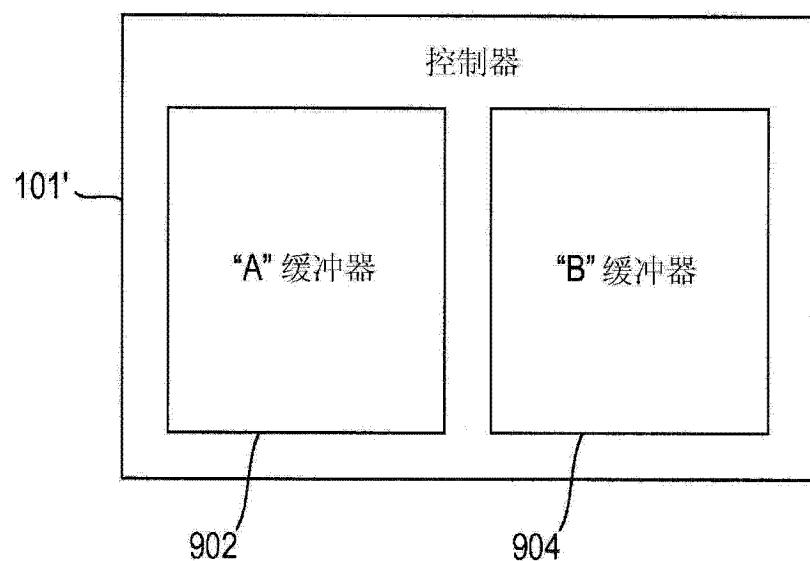


图 9

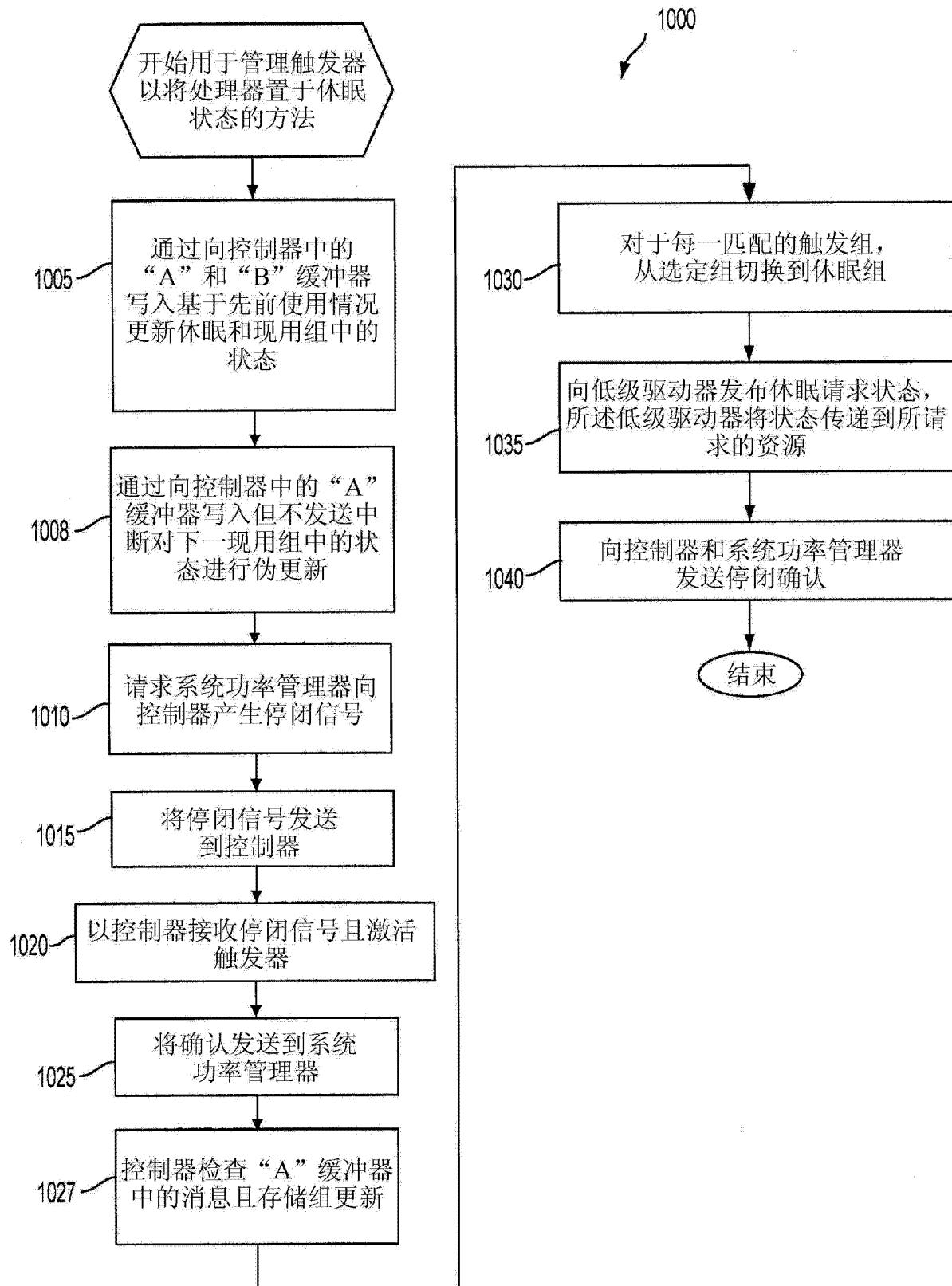


图 10

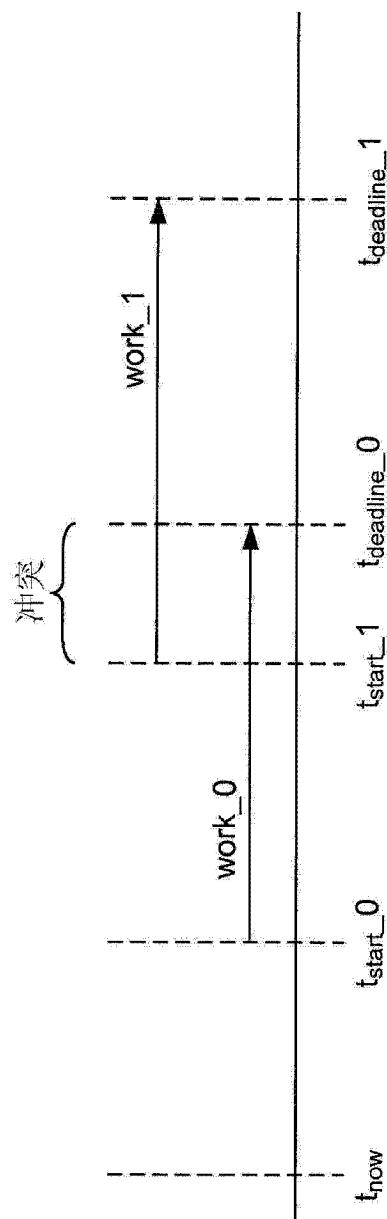


图 11

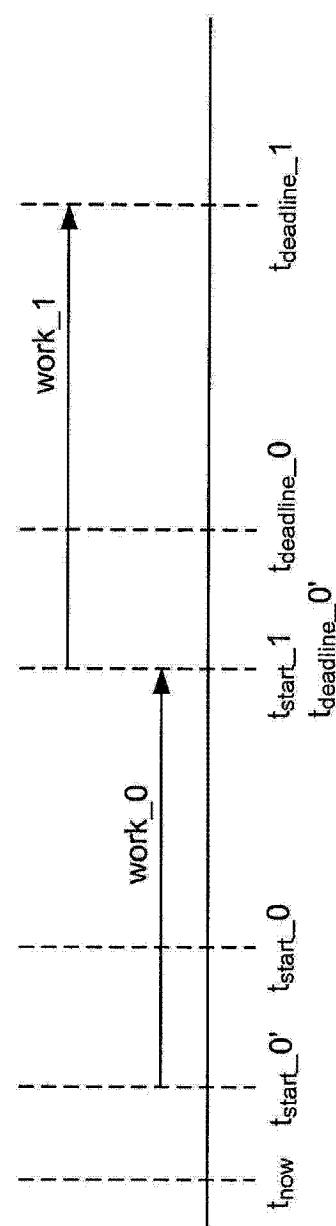


图 12

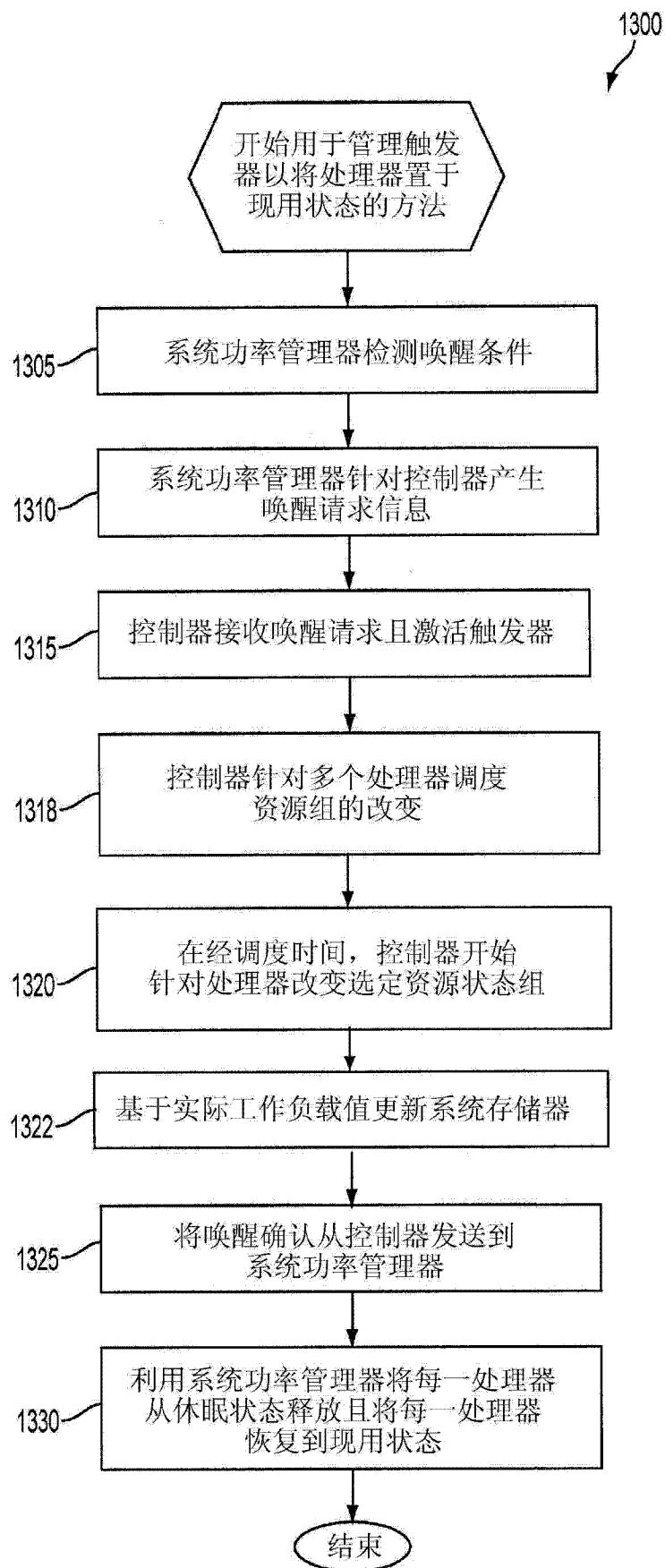


图 13

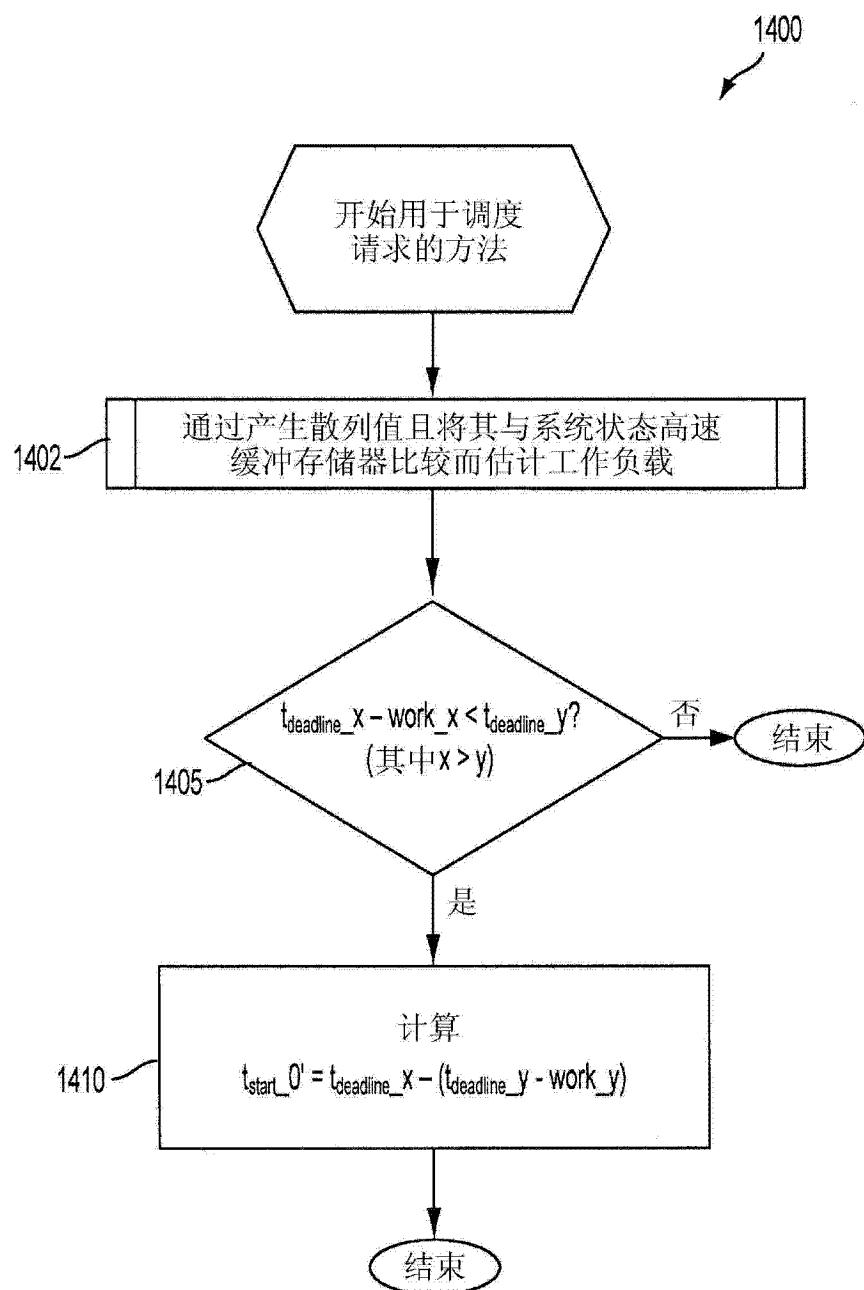


图 14

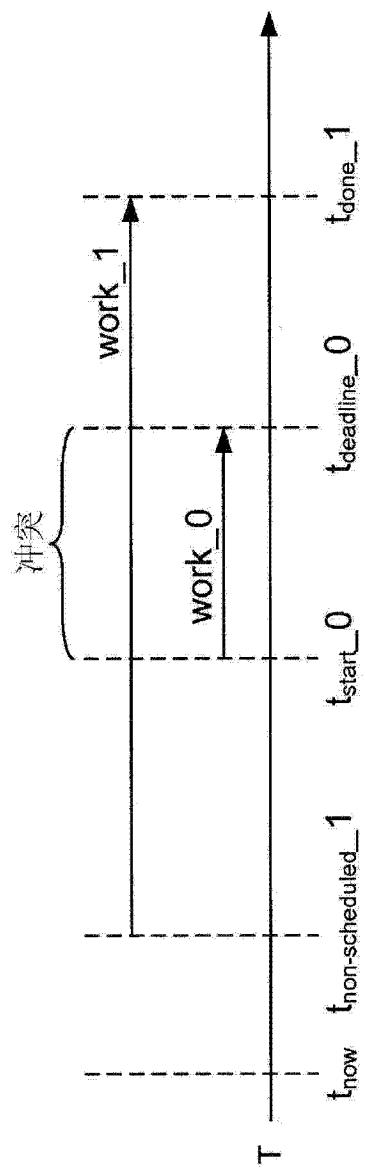


图 15

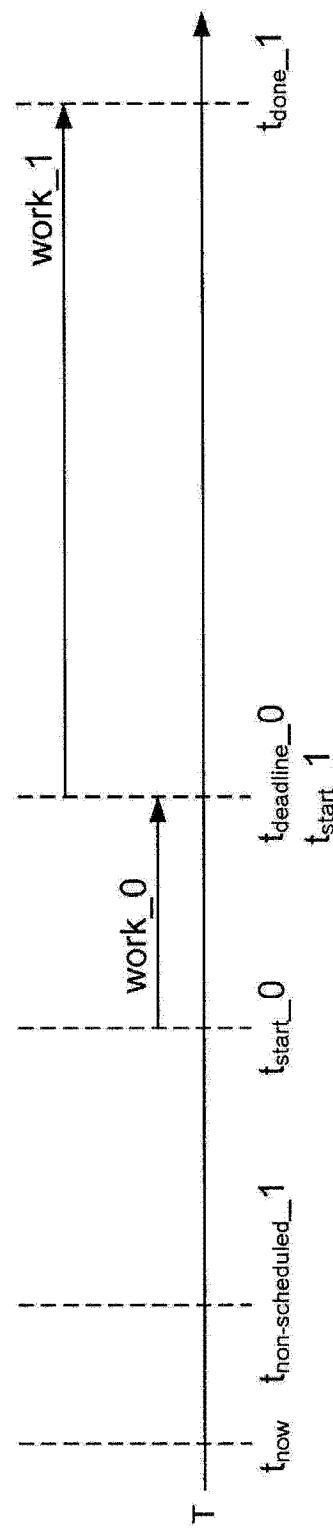


图 16

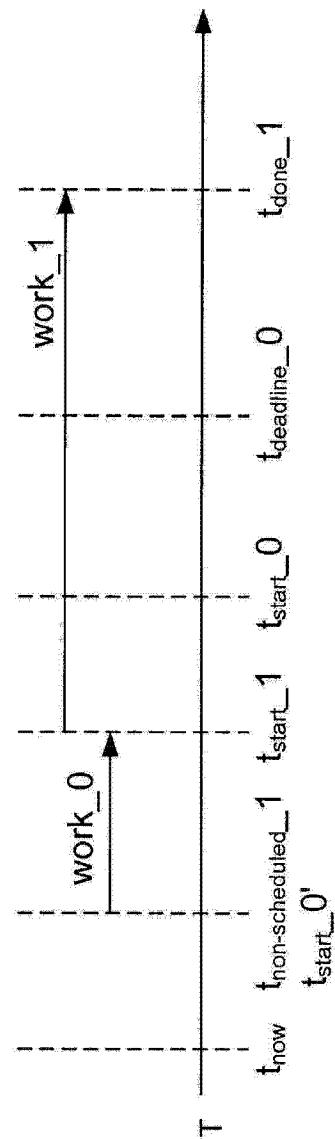


图 17

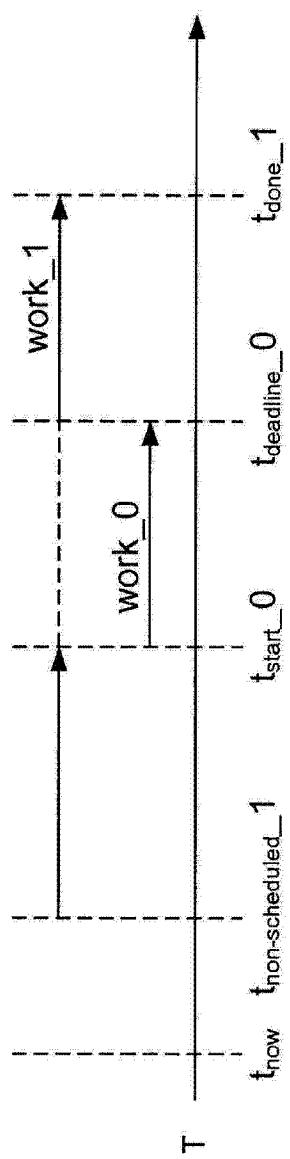


图 18

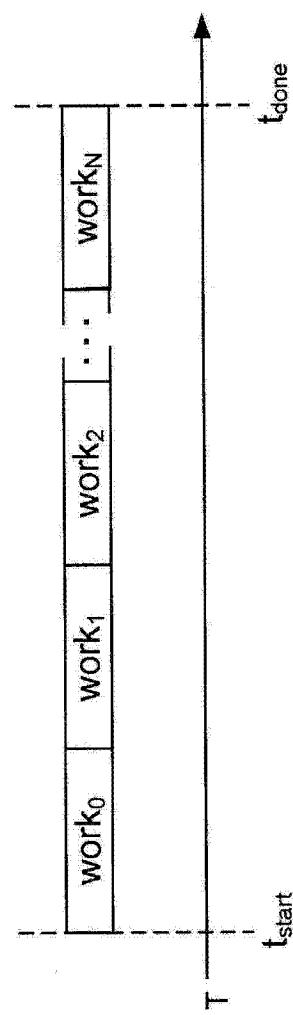


图 19

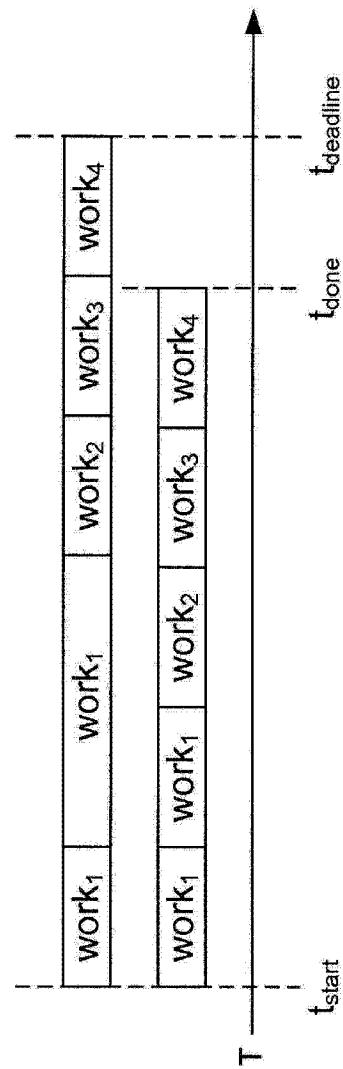


图 20

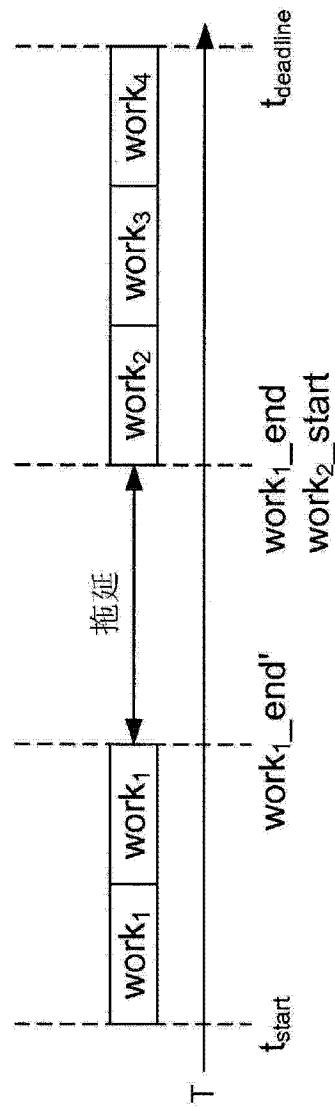


图 21

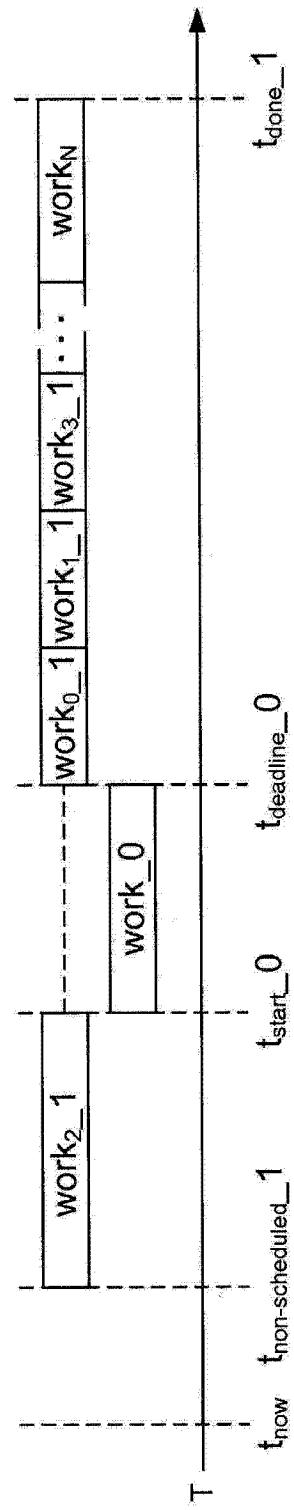


图 22

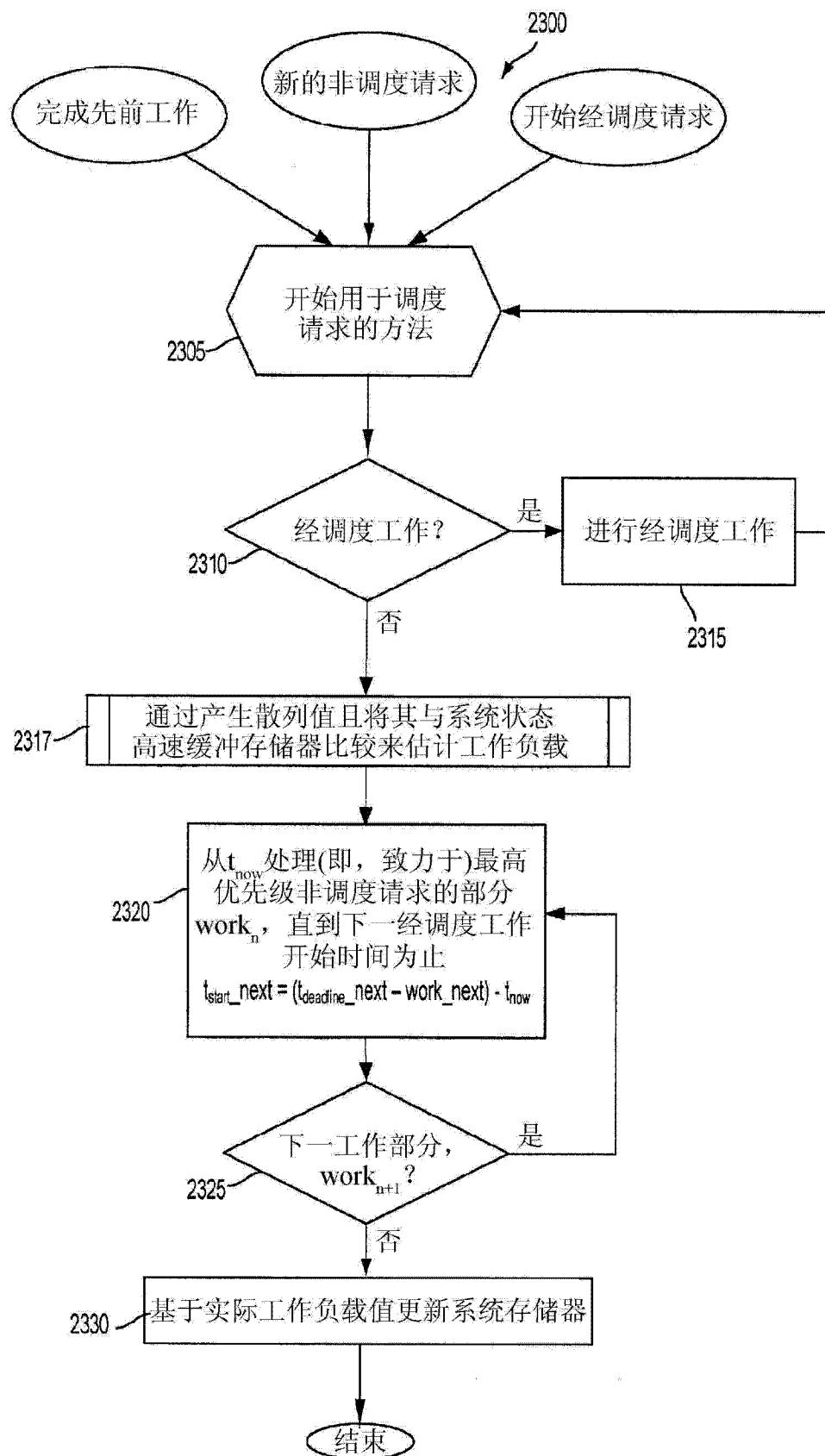


图 23

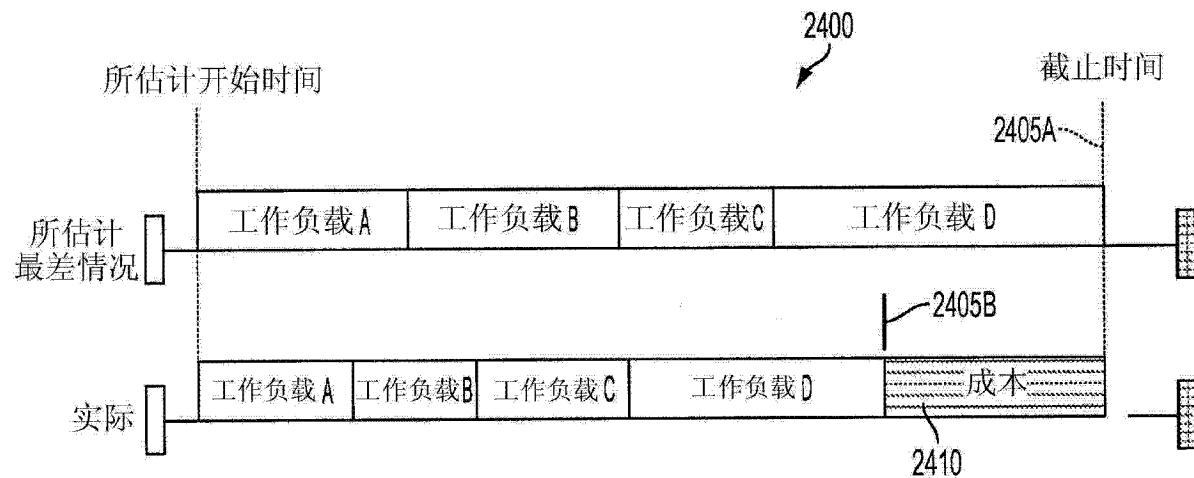


图 24

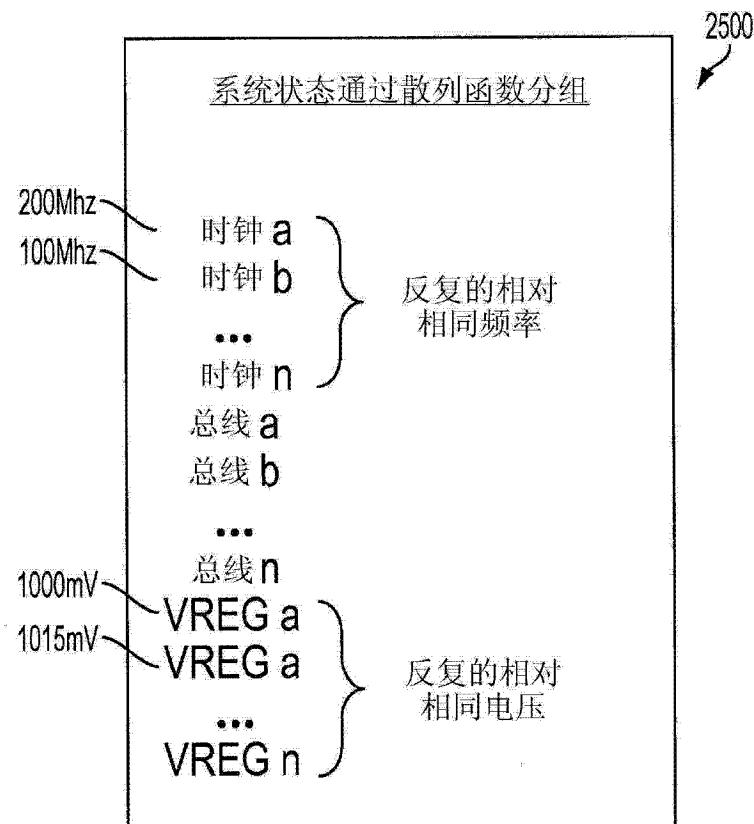


图 25

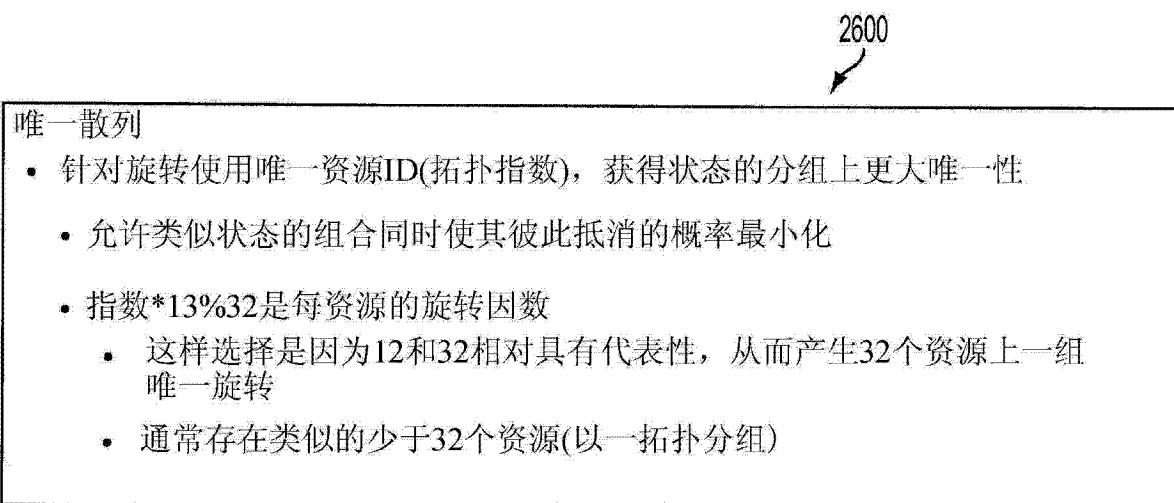


图 26

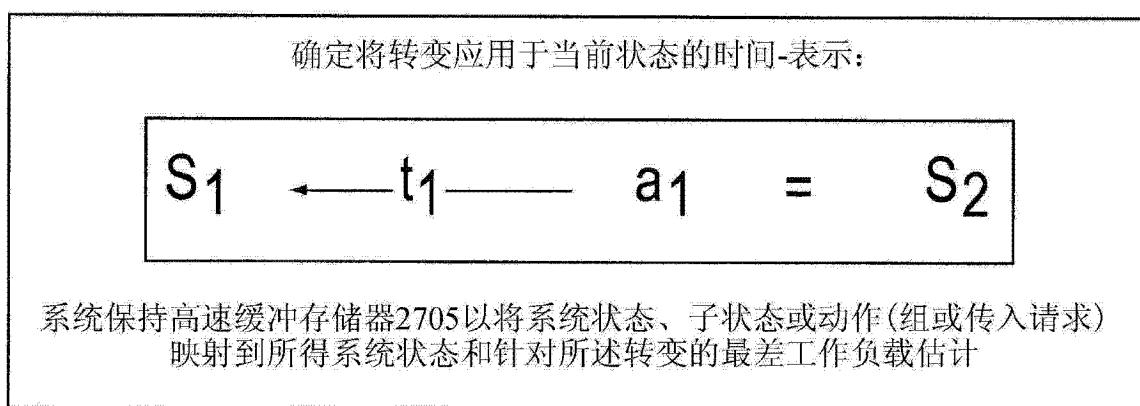


图 27A

2710                  2715                  2720                  2725

系统状态	子状态(动作)	估计	所得系统状态
$S_1$	$a_1$	$t_1$	$S_2$
$S_2$	$a_2$	$t_2$	$S_1$
$S_1$	$a_3$	$t_3$	$S_3$
$S_2$	$a_4$	$t_4$	$S_4$
$S_3$	$a_5$	$t_5$	$S_1$

图 27B

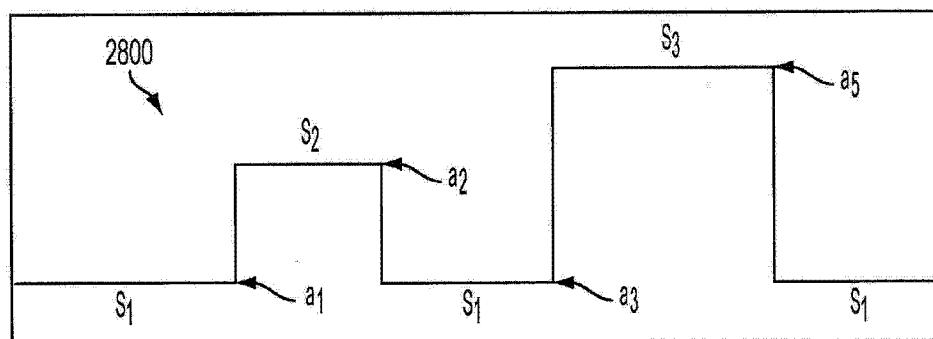


图 28

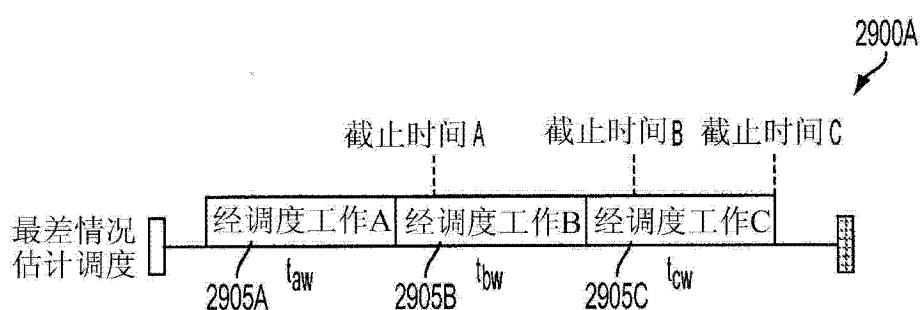


图 29

通过知晓转变的所得系统状态，系统可使用所述所得系统状态用于稍后经调度工作的估计：

知晓工作A如下： $s_1 \leftarrow t_1 — a_1 = s_2$

因此工作B可由以下表示： $s_2 \leftarrow t_2 — a_3 = s_4$

如果高速缓冲存储器中存在匹配的状态，那么系统具有较准确的估计( $t_2$ )

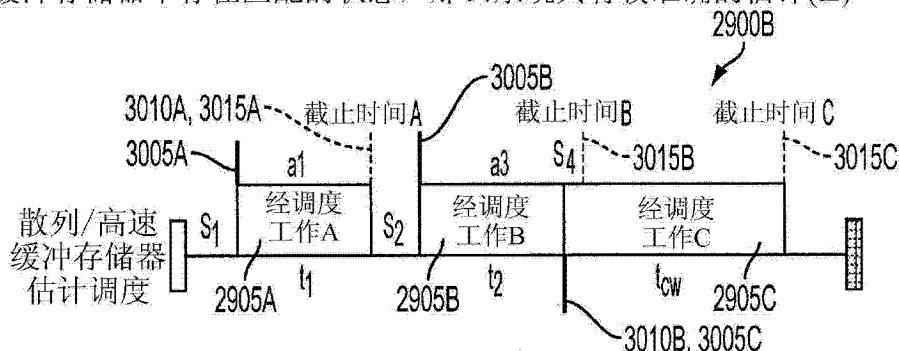


图 30

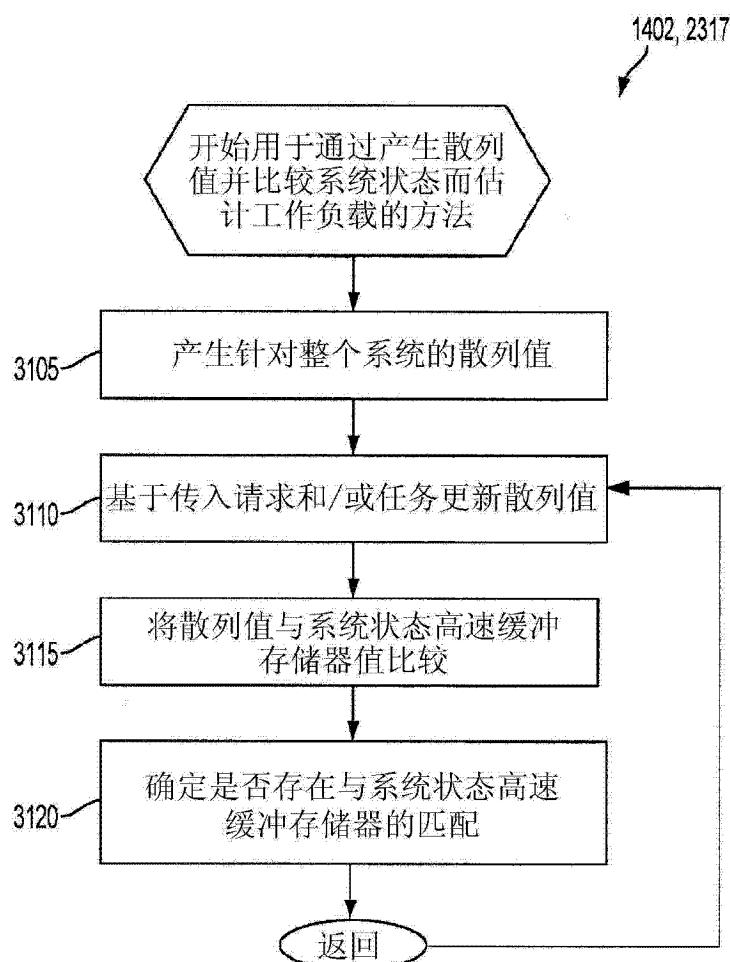


图 31