



(12)发明专利

(10)授权公告号 CN 105074666 B

(45)授权公告日 2020.04.21

(21)申请号 201480009390.7

帕维尔·波托普利亚克

(22)申请日 2014.02.14

卢奇安·科德雷斯库 理查德·郭
布赖恩·C·拜尔德费尔

(65)同一申请的已公布的文献号

(74)专利代理机构 北京律盟知识产权代理有限公司 11287

申请公布号 CN 105074666 A

代理人 宋献涛

(43)申请公布日 2015.11.18

(51)Int.CI.

G06F 9/50(2006.01)

(30)优先权数据

G06F 9/52(2006.01)

13/777,314 2013.02.26 US

G06F 9/455(2006.01)

(85)PCT国际申请进入国家阶段日

2015.08.19

(56)对比文件

(86)PCT国际申请的申请数据

CN 102741828 A, 2012.10.17,

PCT/US2014/016391 2014.02.14

CN 102103567 A, 2011.06.22,

(87)PCT国际申请的公布数据

CN 101517536 A, 2009.08.26,

W02014/133784 EN 2014.09.04

CN 101739235 A, 2010.06.16,

(73)专利权人 高通股份有限公司

US 20040139442 A1, 2004.07.15,

地址 美国加利福尼亚州

US 20080244599 A1, 2008.10.02,

(72)发明人 迈克尔·R·麦克唐纳

审查员 董泽华

埃里克·J·普隆德克

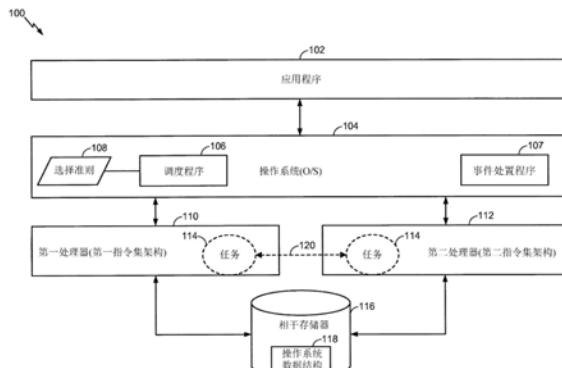
权利要求书4页 说明书11页 附图5页

(54)发明名称

执行在具有不同指令集架构的处理器上的
操作系统

(57)摘要

一种设备包含具有第一指令集的第一处理器和具有不同于所述第一指令集的第二指令集的第二处理器。所述设备还包含存储操作系统的至少一部分的存储器。所述操作系统可同时在所述第一处理器与所述第二处理器上执行。



1.一种执行多个处理器上的代码的方法,包括:

执行第一处理器上的第一代码,所述第一处理器与第一指令集架构相关联且经配置以存取共享数据结构中的存储在虚拟地址处的函数表的第一表项;

在执行所述第一代码期间,执行操作系统事件;

在执行所述操作系统事件后,暂停所述第一代码的执行且起始在第二处理器上的第二代码的执行,所述第二处理器与第二不同指令集架构相关联且经配置以存取所述共享数据结构中的存储在所述虚拟地址处的所述函数表的第二表项,所述第二代码用以处置所述操作系统事件;以及

在完成处置所述操作系统事件的所述第二代码的执行后,暂停所述第二代码且恢复在所述第一处理器上的所述第一代码的执行,其中所述共享数据结构的共同虚拟地址空间包括:

所述函数表的所述第一表项,其对应于经编译以在所述第一处理器上执行的函数的第一版本,所述函数的所述第一版本使用所述第一指令集表示,以及

所述函数表的所述第二表项,其对应于经编译以在所述第二处理器上执行的所述函数的第二版本,所述函数的所述第二版本使用所述第二指令集表示。

2.根据权利要求1所述的方法,其中:

所述第一表项和所述第二表项分别对应于所述共享数据结构中的第一位置识别符和第二位置识别符;且

所述第一位置识别符与所述第二位置识别符不同。

3.根据权利要求1所述的方法,其中所述第一处理器和所述第二处理器各经配置以存取同一相干存储器,所述相干存储器包括所述共享数据结构,且其中所述第一处理器与所述第二处理器集成到移动计算装置内。

4.根据权利要求1所述的方法,其中所述第一处理器和所述第二处理器经配置以经由处理器间消息传递、共享存储器、应用程序编程接口API或其任何组合来相互通信。

5.根据权利要求1所述的方法,其中所述第一处理器包括具有多个硬件线程的数字信号处理器DSP,且其中所述第二处理器包括多核心中央处理单元CPU。

6.根据权利要求1所述的方法,其中所述操作系统事件包括页面故障、陷阱、系统故障、系统请求、硬件错误、地址错误、权限违反或其任何组合。

7.根据权利要求1所述的方法,其中所述操作系统事件使执行的流程从操作系统的用户模式转变到所述操作系统的核心模式。

8.一种设备,其包括:

第一处理器,其具有第一指令集,所述第一处理器与第一指令集架构相关联且经配置以存取共享数据结构中的存储在虚拟地址处的函数表的第一表项;

第二处理器,其具有不同于所述第一指令集的第二指令集,所述第二处理器与第二不同指令集架构相关联且经配置以存取所述共享数据结构中的存储在所述虚拟地址处的所述函数表的第二表项;以及

存储器,其存储可同时在所述第一处理器与所述第二处理器上执行的操作系统的至少一部分,其中所述共享数据结构的共同虚拟地址空间包括:

所述函数表的所述第一表项,其对应于经编译以在所述第一处理器上执行的函数的第

一版本,所述函数的所述第一版本使用第一指令集表示,以及

所述函数表的所述第二表项,其对应于经编译以在所述第二处理器上执行的所述函数的第二版本,所述函数的所述第二版本使用第二指令集表示。

9. 根据权利要求8所述的设备,其中:

所述第一表项与所述第二表项不同;且

所述第一处理器和所述第二处理器集成到移动计算装置内。

10. 根据权利要求8所述的设备,其中所述存储器存储一或多个函数表,且其中所述第一表项对应于所述一或多个函数表中的第一位置识别符,且所述第二表项对应于所述一或多个函数表中的第二位置识别符。

11. 根据权利要求10所述的设备,其中:

所述一或多个函数表中的所述第一位置识别符识别与第一地址相关联的函数的第一版本的位置;且

所述一或多个函数表中的所述第二位置识别符识别与不同于所述第一地址的第二地址相关联的所述函数的第二版本的位置。

12. 根据权利要求10所述的设备,其中所述一或多个函数表包括:

第一函数表,其可由所述第一处理器存取且存储所述第一位置识别符,其中所述第一位置识别符包括到第一地址的第一跳转或分支指令;以及

第二函数表,其可由所述第二处理器存取且存储所述第二位置识别符,其中所述第二位置识别符包括到不同于所述第一地址的第二地址的第二跳转或分支指令,其中所述第一函数表和所述第二函数表与不同物理地址相关联。

13. 根据权利要求10所述的设备,其中所述一或多个函数表包括:

第一函数表,其可由所述第一处理器存取且存储所述第一位置识别符,其中所述第一位置识别符包括第一地址;以及

第二函数表,其可由所述第二处理器存取且存储所述第二位置识别符,其中所述第二位置识别符包括不同于所述第一地址的第二地址,

其中所述第一函数表和所述第二函数表与不同物理地址相关联。

14. 根据权利要求8所述的设备,其中:

所述共享数据结构包括至少部分基于特定指令集索引的一或多个函数表。

15. 根据权利要求8所述的设备,其中所述操作系统包含经配置以调度使用所述第一处理器上的所述第一指令集和所述第二处理器上的所述第二指令集的任务的执行的调度程序。

16. 根据权利要求15所述的设备,其中所述调度程序经配置以调度支持所述第一处理器或所述第二处理器中的任一者上的所述第一指令集和所述第二指令集两者的特定任务的执行。

17. 根据权利要求15所述的设备,其中所述调度程序基于与特定任务相关联的最低功耗的识别选择所述第一处理器或所述第二处理器。

18. 根据权利要求15所述的设备,其中所述调度程序基于包含负荷最小的处理器的识别、特定任务的类型、与所述特定任务相关联的最低功耗的识别或其任何组合的选择准则选择所述第一处理器或所述第二处理器。

19. 根据权利要求8所述的设备,其中所述操作系统经配置以提供系统调用以开始使用指定指令集的一或多个任务。

20. 根据权利要求8所述的设备,其中所述操作系统经配置以提供系统调用以将使用特定指令集的特定任务的进行中的执行切换到不同指令集。

21. 一种方法,其包括:

在可同时在具有第一指令集架构的第一处理器和具有第二不同指令集架构的第二处理器上执行的操作系统的调度程序处接收调度用于执行的任务的请求,所述第一处理器经配置以存取共享数据结构中的存储在虚拟地址处的函数表的第一表项,且所述第二处理器经配置以存取所述共享数据结构中的存储在所述虚拟地址处的所述函数表的第二表项;

响应于确定所述任务可用于使用所述第一处理器上的所述第一指令集架构和使用所述第二处理器上的所述第二不同指令集架构的执行,基于至少一个选择准则调度所述第一处理器上的所述任务;以及

在所述任务的执行期间,将所述任务的执行从所述第一处理器切换到所述第二处理器,其中所述共享数据结构的共同虚拟地址空间包括:

所述函数表的所述第一表项,其对应于经编译以在所述第一处理器上执行的函数的第一版本,所述函数的所述第一版本使用第一代码表示,以及

所述函数表的所述第二表项,其对应于经编译以在所述第二处理器上执行的所述函数的第二版本,所述函数的所述第二版本使用第二代码表示。

22. 根据权利要求21所述的方法,其中所述至少一个选择准则包括与所述任务相关联的最低功率使用。

23. 根据权利要求21所述的方法,其中所述至少一个选择准则包括负荷最小的处理器、所述任务的类型、与所述任务相关联的最低功率使用或其任何组合,且其中切换所述任务的执行包括:

起始从所述第一处理器的系统调用;

从所述第一处理器的调度去除所述任务;

将所述任务添加到所述第二处理器的调度;以及

从所述第二处理器上的所述系统调用返回。

24. 一种设备,其包括:

用于处理与第一指令集相关联的指令的第一装置,所述用于处理指令的第一装置与第一指令集架构相关联且经配置以存取共享数据结构中的存储在虚拟地址处的函数表的第一表项;

用于处理与不同于所述第一指令集的第二指令集相关联的指令的第二装置,所述用于处理指令的第二装置与第二不同指令集架构相关联且经配置以存取所述共享数据结构中的存储在所述虚拟地址处的所述函数表的第二表项;以及

用于存储操作系统的至少一部分的装置,所述操作系统可同时在所述用于处理的第一装置和所述用于处理的第二装置上执行,所述用于存储的装置包括所述共享数据结构,其中所述共享数据结构的共同虚拟地址空间包括:

所述函数表的所述第一表项,其对应于经编译以在所述第一装置上执行的函数的第一版本,所述函数的所述第一版本使用第一指令集表示,以及

所述函数表的所述第二表项，其对应于经编译以在所述第二装置上执行的所述函数的第二版本，所述函数的所述第二版本使用第二指令集表示。

25. 根据权利要求24所述的设备，其进一步包括用于基于与任务相关联的最低功耗的识别调度任务的执行的装置，其中所述任务支持所述用于处理的第一装置或所述用于处理的第二装置上的所述第一指令集和所述第二指令集两者。

26. 根据权利要求24所述的设备，其进一步包括用于调度支持所述用于处理的第一装置或所述用于处理的第二装置上的所述第一指令集和所述第二指令集两者的任务的执行的装置，其中所述用于处理的第一装置和所述用于处理的第二装置经集成到移动计算装置内。

27. 一种非暂时性计算机可读媒体，其包括：

操作系统的单一实例，

其中所述操作系统的所述单一实例包括当在包含第一处理器和第二处理器的电子装置处执行时引起所述操作系统在所述第一处理器与所述第二处理器上的同时执行的指令，所述第一处理器与第一指令集架构相关联，且所述第二处理器与不同于所述第一指令集架构的第二指令集架构相关联，所述第一处理器经配置以存取共享数据结构中的存储在虚拟地址处的函数表的第一表项，且所述第二处理器经配置以存取所述共享数据结构中的存储在所述虚拟地址处的所述函数表的第二表项，其中所述共享数据结构的共同虚拟地址空间包括：

所述函数表的所述第一表项，其对应于经编译以在所述第一处理器上执行的函数的第一版本，所述函数的所述第一版本使用第一代码表示，以及

所述函数表的所述第二表项，其对应于经编译以在所述第二处理器上执行的所述函数的第二版本，所述函数的所述第二版本使用第二代码表示。

执行在具有不同指令集架构的处理器上的操作系统

[0001] 对相关申请案的交叉参考

[0002] 本申请案主张2013年2月26日申请的共同拥有的美国非临时专利申请案第13/777,314号的优先权，其内容明确地被以引用的方式全部并入本文中。

技术领域

[0003] 本发明大体上与多处理器装置处的操作系统的执行有关。

背景技术

[0004] 技术进展已经产生更小且更强大的计算装置。例如，当前存在多种便携式个人计算装置，包含无线计算装置，例如，便携式无线电话、个人数字助理 (PDA)、平板计算机和寻呼装置，其体积小，重量轻，且易于由用户携带。许多此类计算装置包含并入于其中的其它装置。例如，无线电话还可包含数字静态相机、数码摄像机、数字记录器和音频文件播放器。并且，此类计算装置可处理可执行指令，包含软件应用程序，例如，可用以访问互联网的网络浏览器应用程序，和利用静态相机或摄像机并提供多媒体播放功能性的多媒体应用程序。因而，这些装置可执行通常由例如桌上型计算机的较大装置执行的功能。

[0005] 一些电子装置可包含多个处理器。例如，移动电话可包含中央处理单元 (CPU) (有时被称作应用程序处理器) 和数字信号处理器 (DSP)。每一处理器可比其它处理器更好地适合于执行某些计算任务。例如，可充当主要处理器且可执行移动电话操作系统的CPU可在执行“控制码”(例如，网络浏览器和电子表格代码) 中更有效率。另一方面，当执行信号处理和其它数学集中性功能时，DSP可比CPU更有效率。

[0006] 不同处理器可针对不同工作负荷优化且可具有不同架构(例如，指令集)。例如，电话可包含根据第一架构实施的CPU，和根据第二架构实施的DSP。因为不同处理器具有不同架构，所以每一处理器可执行其自身的操作系统。即使在处理器上执行同一操作系统，每一处理器仍充当执行其自身的操作系统实例的单独机器。归因于架构的差异，两个操作系统实例可能不能够共享数据和相互分担处理任务。另外，存储器是从每一环境来分派，而非从共享池。

[0007] 随着移动应用程序变得更复杂，某些应用程序或其线程可能更好地适合于CPU执行或DSP执行。然而，因为CPU与DSP执行不同操作系统，所以可强迫电话执行仅CPU或仅DSP上的应用程序(和其每一线程)。结果，移动应用程序常规地是针对CPU开发，且并不试图充分利用例如DSP的其它可用处理器的处理能力。

[0008] 在个人计算领域，一些软件供应商已尝试通过提供“通用”二进制(也被称作“丰满”二进制)来解决不同架构在不同计算机中的使用。通常，通用二进制包含用于多个架构中的每一者的同一软件的不同版本。在于特定计算机处的通用二进制的安装或执行期间，基于所述特定计算机的架构识别和安装或执行软件的恰当版本。然而，因为仅单一架构存在于每一计算机上，所以通用二进制不提供使用具有不同架构的处理器同时执行同一软件实例的方式。此外，可能难以实施和处理将在处理器之间共享的数据结构。

发明内容

[0009] 揭示同时执行在具有不同架构的多个处理器上的软件(例如,操作系统)的同一实例的系统和方法。例如,基于本发明,可在移动电话的CPU上执行游戏的图形和用户接口(UI)线程,而在移动电话的DSP上执行游戏的多媒体处理线程,即使CPU与DSP具有不同指令集架构。根据描述的技术,CPU代码和DSP代码可显得如同存在于对称性多处理(SMP)机器上。因此,可共享数据结构和通信基元,且可利用跨不同架构的多线程。

[0010] 根据第一实施方案,可在能够存取多个处理器的单一操作系统实例内提供操作系统函数(例如,核心代码)的多个版本。例如,操作系统可能能够存取一或多个函数中的每一者的第一版本和第二版本,其中第一版本是使用CPU的指令集实施且第二版本是使用DSP的指令集实施。应了解,因为函数的两个版本将在运行时间可用,所以可对应用程序开发者提供设计用于在可用架构中的一或者上执行的应用程序(和其个别线程或部分)的机会。

[0011] 此外,操作系统的调度程序可经配置以确定特定代码是否希望(或应)由CPU或由DSP执行。存储器(例如,虚拟和/或物理地址空间)和核心数据结构可由处理器和在其上执行的代码共享。在一个实施方案中,可将对操作系统的DSP请求传递到CPU,且接着可将处理结果发送回到DSP,或反之亦然。此外,同一任务(即,软件线程)的不同部分可由不同处理器执行。

[0012] 根据第二实施方案,DSP代码可运行“远程”操作系统环境。在DSP处发生的操作系统事件(例如,页面故障、例外、系统请求、陷阱等)可由CPU处置。因此,即使任务显得在DSP环境中运行,操作系统请求仍可实际上由CPU环境来处置。

[0013] 在特定实施例中,一种方法包含执行第一处理器(例如,DSP)上的第一代码。所述方法还包含在执行所述第一代码时,执行操作系统事件。所述方法进一步包含在执行所述操作所述方法进一步包含在执行所述操作系统事件后,暂停所述第一代码的执行且起始在第二处理器(例如,CPU)上的第二代码的执行,所述第二代码用以处置所述操作系统事件。所述方法包含在完成处置所述操作系统事件的所述第二代码的执行后,暂停所述第二代码且恢复在所述第一处理器(例如,所述DSP)上的所述第一代码的执行。

[0014] 在另一特定实施例中,一种设备包含具有第一指令集的第一处理器。所述设备还包含具有第二指令集的第二处理器,所述第二指令集不同于所述第一指令集。所述设备进一步包含存储器,其存储可同时在所述第一处理器与所述第二处理器上执行的操作系统的至少一部分。

[0015] 在另一特定实施例中,一种方法包含在操作系统的调度程序处接收调度用于执行的任务的请求。所述操作系统可同时在具有第一架构的第一处理器和具有第二架构的第二处理器上执行。所述方法还包含响应于确定所述任务可用于使用所述第一处理器上的所述第一架构和使用所述第二处理器上的所述第二架构执行,基于至少一个选择准则在所述第一处理器上调度所述任务。所述方法进一步包含在所述任务的执行期间,将所述任务的执行从所述第一处理器切换到所述第二处理器。

[0016] 在另一特定实施例中,一种设备包含用于处理与第一指令集相关联的指令的第一装置。所述设备还包含用于处理与第二指令集相关联的指令的第二装置。所述设备进一步包含用于存储操作系统的至少一部分的装置,所述操作系统可同时在所述用于处理的第一装置与所述用于处理的第二装置上执行。

[0017] 在另一特定实施例中，一种非暂时性计算机可读媒体包含操作系统的单一实例。所述操作系统的所述单一实例包含当在包含第一处理器和第二处理器的电子装置处执行时引起所述操作系统在所述第一处理器与所述第二处理器上的同时执行的指令。所述第一处理器与第一指令集架构相关联，且所述第二处理器与不同于所述第一指令集架构的第二指令集架构相关联。

[0018] 在另一特定实施例中，一种设备包含存储器，其可由与第一指令集架构相关联的第一处理器和由与第二指令集架构相关联的第二处理器存取。所述存储器存储至少一个数据结构。所述数据结构包含经编译以在所述第一指令集架构上执行的第一函数的第一位置识别符。所述数据结构还包含经编译以在所述第二指令集架构上执行的第二函数的第二位置识别符。

[0019] 由所揭示实施例中的至少一者提供的一个特定优势为同时执行在具有不同架构的多个处理器上的操作系统的单一实例和与操作系统相关联的应用程序/线程的能力。例如，操作系统代码和/或应用程序代码可基于选择标准（例如，哪一处理器负荷最少，哪一处理器最适合于执行那个类型的代码等）来调度以用于在多个可用处理器中的特定处理器上执行。由所揭示实施例中的至少一者提供的另一优势为应用程序开发者开发可充分利用传统上不用于第三方应用程序的处理器（例如，DSP）的计算能力的应用程序的能力。在审阅全部申请案后，本发明的其它方面、优点和特征将变得显而易见，所述全部申请案包含以下章节：附图说明、具体实施方式和权利要求书。

附图说明

[0020] 图1为说明可操作以支持在具有不同架构的多个处理器上的操作系统的单一实例的执行的系统的特定实施例的图；

[0021] 图2为说明图1的操作系统数据结构的特定实施例的图；

[0022] 图3为说明执行在具有不同架构的多个处理器上的操作系统的单一实例的方法的特定实施例的流程图；

[0023] 图4为说明执行在具有不同架构的多个处理器上的操作系统的单一实例的方法的另一特定实施例的流程图；和

[0024] 图5为包含可操作以支持在具有不同架构的多个处理器上的操作系统的单一实例的执行的组件的无线装置的框图。

具体实施方式

[0025] 图1为可操作以支持在具有不同架构（替代地被称作“指令集架构（ISA）”或“指令集”）的多个处理器上的操作系统的单一实例的同时执行的系统100的说明性实施例的图。例如，在图1的系统100中，操作系统（O/S）104和与其相关联的组件/应用程序可同时在具有第一ISA的第一处理器110和具有第二ISA的第二处理器112上执行。应注意，虽然在图1中展示两个处理器，但这只为了说明。本文中描述的技术可用于包含两个以上处理器的系统中。

[0026] 在特定实施例中，处理器110、112中的至少一者可为中央处理单元（CPU），例如，经配置以执行与CPU指令集相关联的指令的CPU。例如，CPU可具有多个执行核心（例如，双核CPU、四核CPU等）。在多处理器电子装置（例如，移动计算装置，例如移动电话）中，CPU可负责

和/或较好地适合于执行控制码,例如,输入/输出(I/O)代码、用户接口(U/I)代码、浏览器代码、电子表格代码等。例如,当移动电话执行包含U/I任务(或线程)和多媒体处理任务(或线程)的游戏时,CPU可在执行U/I任务时比在执行多媒体处理任务时更有效率地执行。

[0027] 在特定实施例中,处理器110、112中的至少一者可为数字信号处理器(DSP),例如,经配置以执行与DSP指令集相关联的指令的处理器。例如,DSP可具有多个硬件线程,且可支持动态交错的多线程处理和实时处理。在多处理器电子装置(例如,移动电话)中,DSP可负责和/或较好地适合于执行无线信号处理功能和数学集中性代码。例如,当移动电话执行包含U/I任务(或线程)和多媒体处理任务(或线程)两者的游戏时,DSP可在执行多媒体处理任务时比在执行U/I任务时更有效率地执行。

[0028] 应注意,可结合描述的技术使用各种处理器组合。在一个实例中,系统100可包含属于相同类型但具有不同指令集的多个处理器。为了说明,两个处理器110、112可为CPU,但CPU的ISA可不同。替代地,两个处理器110、112可为DSP,但DSP的ISA可不同。在另一实例中,处理器110、112中的一者可为CPU,且处理器110、112中的另一者可为DSP。不同于CPU和DSP的处理器也可存在于系统100中。其它类型或处理器可包含(但不限于)网络处理单元(NPU)、图形处理单元(GPU)等。也应注意到,虽然本文中的描述提及CPU型架构和DSP型架构,但这只是举例来说。可结合描述的技术使用各种架构/指令集。不同架构可支持不同函数、不同操作码、针对共同函数的不同类型和数目的论断等。

[0029] 系统100可包含数据存储能力。数据存储装置可包含易失性存储器(例如,随机存取存储器(RAM)、基于寄存器的存储器或基于高速缓冲存储的存储器)和/或非易失性存储器(例如,硬盘)。在图1的实施例中,系统100包含存储器116。如本文进一步描述,存储器116可为“相干”存储器,其存储至少一个操作系统数据结构118以支持操作系统104跨处理器110、112的同时执行。例如,操作系统数据结构118可包含锁定、调度队列和函数指针的表,如进一步参看图2所描述。

[0030] 操作系统104可通常可操作以在系统100处执行系统层级(例如,核心层级)函数。例如,操作系统104可执行阻塞/解除阻塞,且可提供针对硬件资源(例如,处理器110、112和存储器116)的抽象化和对所述硬件资源的多路复用存取。除了提供系统层级功能性外,操作系统104可支持一或多个应用程序(例如,用户模式应用程序)的执行。例如,应用程序102可为由用户或管理员安装在系统100处的第三方应用程序。

[0031] 操作系统104可包含一或多个模块或逻辑块来执行操作系统功能。例如,操作系统104可包含调度程序106和事件处置程序107。调度程序106可经配置以调度在处理器110、112处的任务的执行,且事件处置程序107可经配置以处置(例如,响应)操作系统事件。在特定实施例中,操作系统104可包含防止处理器110、112中的任一者修改正使用的操作系统状态(例如,存储器116中的数据)或处理器110、112中的另一者可依赖于的防护或控制。

[0032] 响应于接收到执行特定任务114的请求(例如,来自操作系统104或应用程序102),调度程序106可确定特定任务114支持(例如,可用于使用以下各者来执行)第一处理器110的第一指令集、第二处理器112的第二指令集还是两者。在特定实施例中,确定任务是否支持特定指令集包含确定系统100是否能够存取对应于正使用特定指令集的指令表示的所述任务(和任何相关联的子例程、操作系统调用等)的代码。当调度程序106确定请求的任务114支持指令集中的仅一者时,调度程序106可调度请求的任务114,以用于在对应的支撑的

处理器110或112上执行。

[0033] 当调度程序106确定请求的任务114支持多个指令集/多个处理器时,调度程序106可选择在其上调度任务114以供执行的特定处理器。在选择在其上调度任务114的特定处理器110、112后,调度程序106可提供系统调用以开始使用对应的指令集的任务。

[0034] 在特定实施例中,调度程序106可在选择特定处理器时使用选择准则108。选择准则108可包含负荷最小的处理器的识别。例如,调度程序106可选择处理器110、112中的负荷最小的在其上调度任务114,以达成负荷平衡。替代地或此外,选择准则108可包含任务的类型。例如,调度程序106可调度CPU上的U/I任务,且可调度DSP上的数学集中性任务。替代地,或此外,选择准则108可包含与任务114相关联的最低功耗的识别。例如,调度程序106可调度将在执行任务114时消耗最少量的功率的处理器110或112上的任务114,由此减少系统100处的总功率消耗。其它选择准则也可由调度程序106使用。

[0035] 在特定实施例中,当任务114支持第一处理器110的第一指令集和第二处理器112的第二指令集两者时,调度程序106可经配置以将任务的执行从处理器110、112中的一者切换到另一者。此外,此动态切换可在任务114的执行期间且明显地关于系统的较高阶组件(例如,应用程序102)执行。例如,任务114可从第一处理器110“迁移”(例如,使用系统调用)到第二处理器112,或反之亦然,如在120处所展示。此迁移可至少部分因能够存取同一相干存储器116和存储于其上的操作系统数据结构118的第一处理器110和第二处理器112而使得成为可能。应了解,动态迁移支持多个指令集的任务的能力可释放用于仅支持一个指令集的任务的处理资源。

[0036] 应注意,虽然前文描述任务的调度和迁移,但也可在图1的系统100处执行子任务的调度和迁移。例如,可分别在CPU、DSP和GPU处同时调度和执行I/O任务/线程、多媒体处理任务/线程和游戏的图形任务/线程。调度恰当处理器上的任务可增加效率且减少在图1的系统100处的功率消耗。因此,当将系统100集成到电池供电的装置(例如,移动电话)内时,系统100可导致增加的性能和电池寿命。例如,用户可能能够在其移动电话上比游戏只在CPU上运行的情况更长地打游戏,而不需要对移动电话充电。

[0037] 操作期间,操作系统104、其任务(例如,任务114)和/或与其相关联的应用程序(例如,应用程序102)可在处理器110、112中的一或两者上执行,即使处理器110、112具有不同架构。例如,根据在图1的系统100处的操作的第一实施例,处理器110、112中的一者可为“主”处理器,且另一处理器可为“次”处理器。在此实施例中,可将从次处理器到操作系统的请求传递到主处理器,且可将处理结果发送回到次处理器。

[0038] 为了说明,主处理器可为CPU,且次处理器可为DSP。DSP可执行“远程”操作系统环境。在第一代码(例如,DSP特定代码)的执行期间,DSP可检测和/或执行操作系统事件。操作系统事件可为使执行的流程从操作系统的用户模式切换到操作系统的核心模式的事件。操作系统事件的实例包含(但不限于)页面故障、陷阱、系统故障、系统请求(例如,与例如互斥量或信号量的同步目标相关联)、硬件错误(例如,总线错误)、地址错误、权限违反等。DSP可依赖于CPU处置操作系统事件。例如,在执行操作系统事件后,DSP可暂停第一代码的执行,且可起始CPU上的第二代码的执行,其中第二代码经配置以处置操作系统事件。在特定实施例中,在CPU处的事件处理代码的起始是使用处理器间消息传递、由调度程序106、使用相干存储器116、使用应用程序编程接口(API)或其任何组合来执行。因此,处理器110、112中的

每一者可经配置以在执行代码的同时和/或在暂停的同时与处理器110、112中的另一者通信。在CPU处的事件处理代码可处置操作系统事件,且可将任何执行结果提供到DSP。在完成事件处理后,事件处置程序可在CPU处暂停,且可在DSP处恢复DSP代码。

[0039] 根据在图1的系统100处的操作的第二实施例,操作系统104可包含操作系统例程、模块和函数的多个版本。例如,当第一处理器110为CPU且第二处理器112为DSP时,操作系统104的单一实例可包含一或多个例程、模块和函数中的至少两个版本——在CPU的ISA的指令中表示的第一版本和在DSP的ISA的指令中表示的第二版本。在此实施例中,调度程序106可自由地调度在处理器110或112上的应用程序102(或其组件),这是因为操作系统104具有响应由应用程序102使用处理器进行的操作系统调用的能力。

[0040] 图1的系统100可因此实现在具有不同架构的多个处理器上的操作系统104、其组件和与此相关联的应用程序的同时执行。此可增加效率且减少在图1的系统100处的功率消耗。图1的系统100也可使应用程序开发者能够开发充分利用传统上并不用于第三方应用程序的处理器(例如,DSP)的能力的应用程序。这可使图1的系统100比不支持跨具有不同架构的处理器的动态应用程序调度和数据共享的系统对应用程序开发者更有吸引力。

[0041] 图2为说明可存储在图1的操作系统数据结构118中且大体标明为200的函数表的特定实施例的图。如参照图1的操作系统数据结构118所描述,图2的表可用以识别在多个指令集中可用的函数的“正确”版本(即,匹配用于试图执行函数的处理器的指令集的函数的版本)。

[0042] 在运行具有共享数据结构的共同操作系统时的一个困难可为处理含有函数的数据结构。例如,当具有第一指令集的第一处理器与具有第二指令集的第二处理器具有共同操作系统时,在第一指令集和第二指令集两者中实施的函数可能需要可用。然而,所述函数可出现在不同地址。因此,存储函数的地址(例如,“函数指针”)的数据结构需要能够参考用于特定处理器/指令集(其地址可取决于处理器的指令集而不同)的“正确”函数。

[0043] 根据描述的技术,处理这些数据结构时的困难可因使用间接层级而减小。例如,存储器可存储可为具有不同指令集架构的多个处理器存取的共享数据结构。数据结构可包含经编译以在第一指令集架构上执行的第一函数的第一位置识别符和经编译以在第二指令集架构上执行的第二函数的第二位置识别符。位置识别符可为指令、偏移、虚拟地址等,如本文进一步描述。此外,第一函数与第二函数可为同一操作系统函数的不同版本。通过引入间接层级,所描述的技术可有利地实现在多个架构上的同一高阶(例如,应用程序级)代码的执行。例如,代码可包含对操作系统函数“X”的调用。当在与第一架构相关联的第一处理器上执行时,可将对“X”的调用映射到“X”的第一实施方案,其特定针对第一架构且位于存储器中的第一物理地址处。当在与第二架构相关联的第二处理器上执行时,可将对“X”的同一调用映射到“X”的第二实施方案,其特定针对第二架构且位于存储器中的第二物理地址处。

[0044] 本文中描述了实施此“架构独立”代码执行的三个实施例。在图2中,展示了三个操作系统函数——“OS调度”函数,和“OS读取”函数,和“OS写入”函数。所述函数中的每一者的第一版本(“proc1”)和所述函数中的每一者的第二版本(“proc2”)是在存储器202中的不同位置处可用。

[0045] 在第一实施例(在图2中标明为“选项1”)中,表210、212可包含将执行重定向到“正

确”函数的短(例如,恒定长度)指令序列。可将第一表210映射到第一处理器上的特定虚拟地址,且可将第二表212映射到第二处理器上的同一虚拟地址。然而,取决于处理器架构,表210、212可导致(例如,分支或跳到)不同物理地址。对于每一处理器的虚拟地址到物理地址转译可使“正确”表在同一虚拟地址处映射。对函数的间接参考指向表项,而非实际函数。

[0046] 例如,如图2中所展示,处理器1可支持“分支”指令,且处理器2可支持“跳转”指令。用于处理器1的表210可包含用于“OS调度”函数、“OS读取”函数和“OS写入”函数的第一版本的分支指令序列。用于处理器2的表212可包含用于“OS调度”函数、“OS读取”函数和“OS写入”函数的第二版本的跳转指令序列。对在高级代码中的函数“OS调度”的调用可指向存储在特定虚拟地址处的表的第一项。从高级代码的角度看来,将同一虚拟地址用于函数映射表,而不管哪一处理器正执行高级代码。当在处理器1上执行高级代码时,可存取表210,且可执行到“OS调度”函数的“proc1”版本的分支。即,当在处理器1上执行高级代码时,可“选择”表210,而非表212,这是因为对于处理器1,在高级代码中的用于函数映射表的虚拟地址映射(例如,经由特定针对处理器1的虚拟到物理地址映射表)到表210的物理地址。当在处理器2上执行高级代码时,可存取表212且可执行到“OS调度”函数的“proc2”版本的跳转。即,当在处理器2上执行高级代码时,可“选择”表212,而非表210,这是因为对于处理器2,在高级代码中的用于函数映射表的虚拟地址映射(例如,经由特定针对处理器2的虚拟到物理地址映射表)到表212的物理地址。因此,第一实施例可类似于用于位置独立代码的“程序连接表”。

[0047] 在第二实施例(在图2中标明为“选项2”)中,表220、222可包含用于每一函数的地址。替代将地址直接存储到函数,数据结构(例如,由高级代码所使用)可含有表220、222中的地址或索引。表220、222可分别含有用于第一和第二处理器架构的函数的地址。对于每一处理器的虚拟地址到物理地址转译可使“正确”表220、222在同一虚拟地址处映射。

[0048] 例如,如图2中所展示,用于处理器1的函数表可包含用于“OS调度”函数、“OS读取”函数和“OS写入”函数的第一版本的地址(例如,虚拟地址)。用于处理器2的函数表可包含用于“OS调度”函数、“OS读取”函数和“OS写入”函数的第二版本的地址(例如,虚拟地址)。对在高级代码中的函数“OS调度”的调用可指向存储在特定虚拟地址处的表的第一项。如以上参照第一实施例所描述,从高级代码的角度看来,将同一虚拟地址用于函数映射表,而不管哪一处理器正执行高级代码。当在处理器1上执行高级代码时,可存取表220,且可使用来自表220的第一项的“OS调度”的“proc1”版本的所得地址。即,当在处理器1上执行高级代码时,可“选择”表220,而非表222,这是因为对于处理器1,在高级代码中的用于函数映射表的虚拟地址映射(例如,经由特定针对处理器1的虚拟到物理地址映射表)到表220的物理地址。当在处理器2上执行高级代码时,可存取表222,且可使用来自表220的第一项的“OS调度”函数的“proc2”版本的所得地址。即,当在处理器2上执行高级代码时,可“选择”表222,而非表220,这是因为对于处理器2,在高级代码中的用于函数映射表的虚拟地址映射(例如,经由特定针对处理器2的虚拟到物理地址映射表)到表222的物理地址。因此,第二实施例类似于用于位置独立代码的“全域偏移表”。

[0049] 在第三实施例(在图2中标明为“选项3”)中,单一表230可包含在第一和第二处理器指令集两者中实施的函数的地址。替代直接将地址存储到函数,数据结构(例如,由高级代码所使用)可含有表230中的地址或索引。因为使用单一表230替代两个不同表,所以表

230可进一步由处理器指令集架构来索引。因此,表230中的查询可指定函数/表索引/偏移以及架构(例如,“proc1”或“proc2”)两者。表230可含有用于第一和第二处理器架构两者的函数的地址,如所展示。应了解,不同于以上所描述的第一和第二实施例,因为使用单一表230,所以在第三实施例中,虚拟地址到物理地址映射可跨处理器保持相同。

[0050] 例如,如图2中所展示,函数表可包含用于在处理器1的架构以及处理器2的架构两者中的“OS调度”函数、“OS读取”函数和“OS写入”的版本的地址。对高级代码中的函数“OS调度”的调用可指向存储在特定虚拟地址处的表的第一项。当在处理器1上执行高级代码时,可执行在表230上的查找,其中查找指定对于“OS调度”函数和“proc1”架构的偏移,且可使用“OS调度”的“proc1”版本的所得地址。当在处理器2上执行高级代码时,可执行在表230上的查找,其中查找指定对于“OS调度”函数和“proc2”架构的偏移,且可使用“OS调度”的“proc2”版本的所得地址。

[0051] 应注意,在第二和第三实施例中,从不存储到函数的指针,而取而代之,存储表项指示,且每当在调用会场时查找函数的地址。

[0052] 在操作期间,数据结构中到函数的指针可替代地指对应的表项。在选项1中,表中的项为指令/指令序列,因此可以与在无由图2的表介绍的间接层级的情况下所进行相同的方式调用指针。对于选项2和选项3,表可读取,且接着用于特定处理器的“正确”函数地址可为可用的(例如,用于在随后分支或跳转中使用)。

[0053] 应注意,可组合参看图2描述的选项。例如,可在系统(例如,图1的系统100)中同时实施图2中说明的多个实施例。为了说明,参照表210和212描述的“程序连接表”实施例可结合参照表220和222描述的“全域偏移表”实施例使用。当第一次在处理器1上在高级代码中调用函数时,所述调用可解析到表210中的代码的区段,其中代码的区段分支到函数的恰当版本,并且还将分支的目的地添加到表220。将表220中的地址用作目的地,对函数的随后高级调用导致表210的分支指令。另外,应注意,可使用起到将数据结构中的函数指示转译到用于给定指令集的正确函数的地址的作用的其它结构或程序。

[0054] 图3为说明执行在具有不同架构的多个处理器上的操作系统的单一实例的方法300的特定实施例的流程图。在说明性实施例中,方法300可由图1的系统100执行,且可参照图2的函数和表来描述。

[0055] 方法300可包含在302执行第一处理器上的第一代码。在特定实施例中,第一处理器可为多线程DSP。例如,图1的第一处理器110可执行对应于第一任务(例如,图2的“OS调度”、“OS读取”或“OS写入”函数)的第一代码。

[0056] 方法300还可包含在304,在执行第一代码时执行使执行的流程从用户模式转变到核心模式的操作系统事件。操作系统事件可为页面故障、陷阱、系统故障、系统请求、硬件错误、地址错误、权限违反等。例如,在执行第一代码时,图1的第一处理器110可执行操作系统事件。

[0057] 方法300可进一步包含在306,在执行操作系统事件后,暂停第一代码的执行且起始在第二处理器上的第二代码的执行。第二代码可经配置以处置操作系统事件。第二处理器可为多核心CPU。例如,在执行操作系统事件后,图1的第一处理器110可暂停第一代码的执行,且可引起在图1的第二处理器112上的第二代码的执行。

[0058] 方法300可包含在308,在完成处置操作系统事件的第二代码后,暂停第二代码且

恢复第一处理器上的第一代码的执行。例如,可暂停在图1的第二处理器112处的第二代码的执行,且可恢复在图1的第一处理器110处的第一代码的执行。

[0059] 图3的方法300可因此使“次”处理器执行“远程”操作系统环境,且依赖于“主”处理器处置操作系统事件。例如,即使代码可能已被写入且针对执行特定操作系统的CPU编译,所述代码仍可在与CPU一起运行特定操作系统的DSP上执行,且DSP可依赖于用于事件处理和其它操作系统任务的CPU。从过程在DSP上运行的角度看来,如果所述过程正在CPU上运行那么将已可用的设施/服务可仍为可用的。

[0060] 在特定实施例中,图3的方法300可由DSP、CPU、控制器、现场可编程门阵列(FPGA)装置、专用集成电路(ASIC)、另一硬件装置、固件装置或其任何组合来实施。作为实例,图3的方法300可由执行指令的处理器执行,如关于图5所描述。

[0061] 图4为说明执行在具有不同架构的多个处理器上的操作系统的单一实例的方法400的另一特定实施例的流程图。在说明性实施例中,方法400可由图1的系统100执行,且可参照图2的函数和表来描述。

[0062] 方法400可包含在402,在操作系统的调度程序处接收调度用于执行的任务的请求。操作系统可同时在具有第一架构的第一处理器和具有第二架构的第二处理器上执行。例如,在图1中,调度程序106可接收调度用于执行的任务114的请求。

[0063] 方法400还可包含在404确定所述任务是否支持多个指令集。例如,在图1中,调度程序106可确定任务114是否支持第一处理器110的第一指令集和第二处理器112的第二指令集两者。当所述任务支持单一指令集时,方法400可包含在406调度和开始在具有支持的指令集的处理器上的任务。

[0064] 当任务支持多个指令集时,方法400可包含在408调度在支持的处理器中的一者上的任务,且在410,开始使用调度的处理器的指令集的任务。可基于选择准则(例如,负荷最少的处理器、任务的类型、与任务相关联的最低功耗等)来选择调度的处理器。例如,在图1中,调度程序106可基于选择准则108调度用于在第一处理器110上执行的任务114,且可产生系统调用以开始使用第一指令集的任务114。

[0065] 方法400还可包含在412,在任务的执行期间,将任务的执行切换到其它处理器和到其它指令集。例如,在图1中,在第一处理器110上的任务114的执行期间,可将任务114的执行切换到第二处理器112,如在120处所展示。

[0066] 在说明性实施例中,切换任务的执行可包含在414进行第一处理器上的系统调用以将任务的执行切换到其它指令集。继续进行到416,可从第一处理器(即,在410处使用的调度的处理器)的调度去除任务。前进到418,可将任务添加到第二处理器(即,与其它指令集相关联的另一处理器)的调度。继续到420,方法400可包含从具有第二指令集的第二处理器上的系统调用返回。

[0067] 图4的方法400可因此实现根据可改良在多ISA系统处的执行性能、资源使用和/或功率消耗的选择准则在这些系统处的任务的动态调度。

[0068] 在特定实施例中,图4的方法400可由DSP、CPU、控制器、现场可编程门阵列(FPGA)装置、专用集成电路(ASIC)、另一硬件装置、固件装置或其任何组合来实施。作为实例,图4的方法400可由执行指令的处理器执行,如关于图5所描述。

[0069] 参看图5,展示电子装置500的框图。在特定实施例中,电子装置500或其组件可包

含在机顶盒、音乐播放器、视频播放器、娱乐单元、导航装置、通信装置、个人数字助理(PDA)、固定位置数据单元、计算装置或其任何组合中。

[0070] 电子装置500包含耦合到存储器532的CPU 510和DSP 574。CPU 510可包含多个核心(例如,第一核心570和第二核心572)且DSP 574可包含多个硬件线程(例如,第一硬件线程576和第二硬件线程578)。虽然在图5中展示两个核心和两个硬件线程,但应注意,也可使用具有不同数目个核心和/或硬件线程的其它配置。

[0071] 图5包含耦合到CPU 510和显示器528的显示控制器526。译码器/解码器(编码解码器)534可耦合到CPU 510和DSP 574。扬声器536和麦克风538可耦合到编码解码器534。图5还指示无线控制器540可耦合到DSP 574和天线542(例如,经由射频(RF)接口)。因此,在图5的实施例中,CPU 510可用以执行显示功能,DSP 574可用以执行无线信号处理功能,且CPU 510和DSP 574皆可能能够执行音频I/O功能。

[0072] 存储器532可为包含可执行指令556的有形非暂时性计算机可读或处理器可读存储媒体。例如,存储器532可存储图1的操作系统104和/或其组件(例如,图1的调度程序106、图1的事件处置程序107和图1的选择准则108)的至少一部分。当存储器532(例如,RAM)存储操作系统的一部分时,操作系统的其余部分可存储在另一存储器或存储装置(例如,硬盘)中。存储器532也可存储操作系统数据结构558(例如,图1的操作系统数据结构118和/或图2的表)。指令556可由例如CPU 510或DSP 574的处理器执行,以执行各种功能和方法,包含图3和4的方法300和400。

[0073] 在特定实施例中,CPU 510、DSP 574、显示控制器526、存储器532、编码解码器534和无线控制器540包含于系统级封装或片上系统装置522中。在特定实施例中,输入装置530和电力供应器544耦合到片上系统装置522。此外,在特定实施例中,如图5中所说明,显示器528、输入装置530、扬声器536、麦克风538、天线542和电力供应器544在片上系统装置522外部。然而,显示器528、输入装置530、扬声器536、麦克风538、天线542和电力供应器544中的每一者可耦合到片上系统装置522的组件,例如,接口或控制器。

[0074] 结合所描述实施例,一种设备包含用于处理与第一指令集相关联的指令的第一装置。例如,用于处理的第一装置可包含图1的第一处理器110、图1的第二处理器112、图5的CPU 510、图5的DSP 574、另一处理器(例如,NPU或GPU)、经配置以处理指令的另一装置或模块或其任何组合。

[0075] 所述设备还可包含用于处理与第二指令集相关联的指令的第二装置。例如,用于处理的第二装置可包含图1的第一处理器110、图1的第二处理器112、图5的CPU 510、图5的DSP 574、另一处理器(例如,NPU或GPU)、经配置以处理指令的另一装置或模块或其任何组合。

[0076] 所述设备可进一步包含用于存储操作系统的至少一部分的装置,所述操作系统可同时在所述用于处理的第一装置与所述用于处理的第二装置上执行。例如,用于存储操作系统的至少一部分的装置可包含图1的相干存储器116、图5的存储器532、经配置以存储数据的另一装置或模块或其任何组合。

[0077] 所述设备还可包含用于存储与操作系统相关联的至少一个共享数据结构的装置。例如,用于存储至少一个共享数据结构的装置可包含图1的存储器116、图5的存储器532、经配置以存储数据的另一装置或模块或其任何组合。

[0078] 所述设备可包含用于在用于处理的第一装置或用于处理的第二装置中的任一者上调度支持第一指令集和第二指令集两者的任务的执行的装置。例如，用于调度的装置可包含图1的调度程序106、经配置以调度用于执行的任务的另一装置或模块或其任何组合。

[0079] 所属领域的技术人员将进一步了解，结合本文中揭示的实施例描述的逻辑块、配置、模块、电路和算法步骤可实施为电子硬件、由处理器执行的计算机软件或两者的组合。各种说明性组件、块、配置、模块、电路和步骤已在上文大体按其功能性来描述。此功能性是实施为硬件还是处理器可执行指令取决于特定应用和强加于整个系统的设计约束。熟练的技术人员可针对每一特定应用以不同方式实施所描述的功能性，但此类实施决策不应被译为引起对本发明的范围的偏离。

[0080] 结合本文中所揭示的实施例而描述的方法或算法的步骤可直接体现于硬件中、由处理器执行的软件模块中或两者的组合中。软件模块可驻留在随机存取存储器 (RAM)、快闪存储器、只读存储器 (ROM)、可编程只读存储器 (PROM)、可抹除可编程只读存储器 (EPROM)、电可抹除可编程只读存储器 (EEPROM)、寄存器、硬盘、装卸式磁盘、压缩光盘只读存储器 (CD-ROM)、数字多功能光盘 (DVD) 或此项技术中已知的任何其它形式的非暂时性存储媒体中。示范性存储媒体耦合到处理器，使得处理器可从存储媒体读取信息和将信息写入到存储媒体。在替代方案中，存储媒体可与处理器成一体式。处理器和存储媒体可驻留在专用集成电路 (ASIC) 中。ASIC可驻留在计算装置或用户终端中。在替代方案中，处理器和存储媒体可作为离散组件驻留在计算装置或用户终端中。

[0081] 例如，根据所描述实施例，非暂时性计算机可读存储媒体(例如，CD、DVD、存储器装置等)可存储操作系统的单一实例。操作系统可包含当在包含第一处理器和第二处理器的电子装置处执行时引起在第一处理器与第二处理器上的操作系统的同时执行的指令。第一处理器与第一指令集架构相关联，且第二处理器与不同于第一指令集架构的第二指令集架构相关联。

[0082] 提供对所揭示实施例的先前描述以使所属领域的技术人员能够制作或使用所揭示的实施例。对这些实施例的各种修改将易于对所属领域的技术人员显而易见，并且可在不脱离本发明的范围的情况下将本文中定义的原理应用到其它实施例。因此，本发明并不希望限于本文展示的实施例，而应符合与如由所附权利要求书界定的原理和新颖特征一致的可能最广范围。

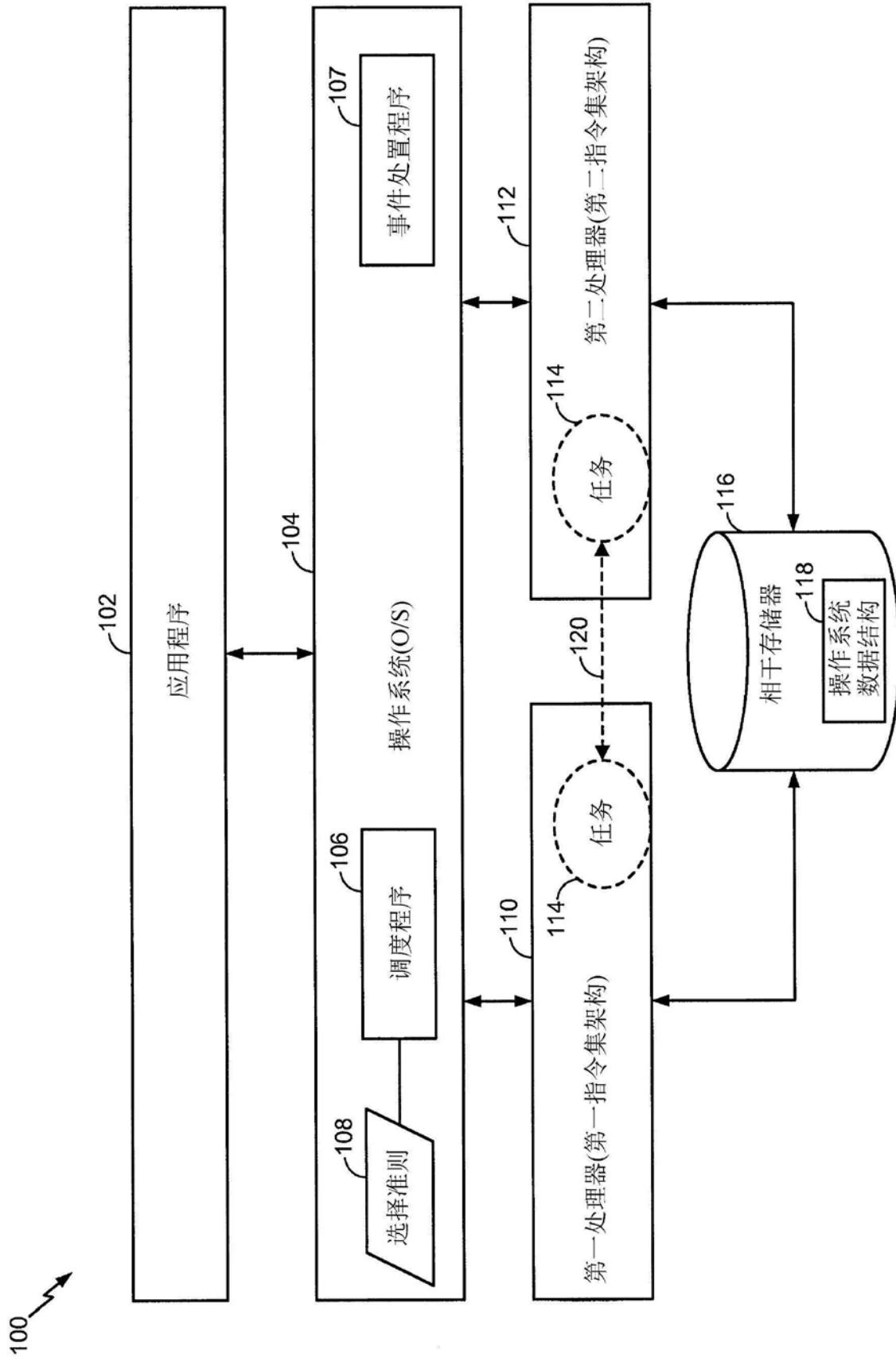


图1

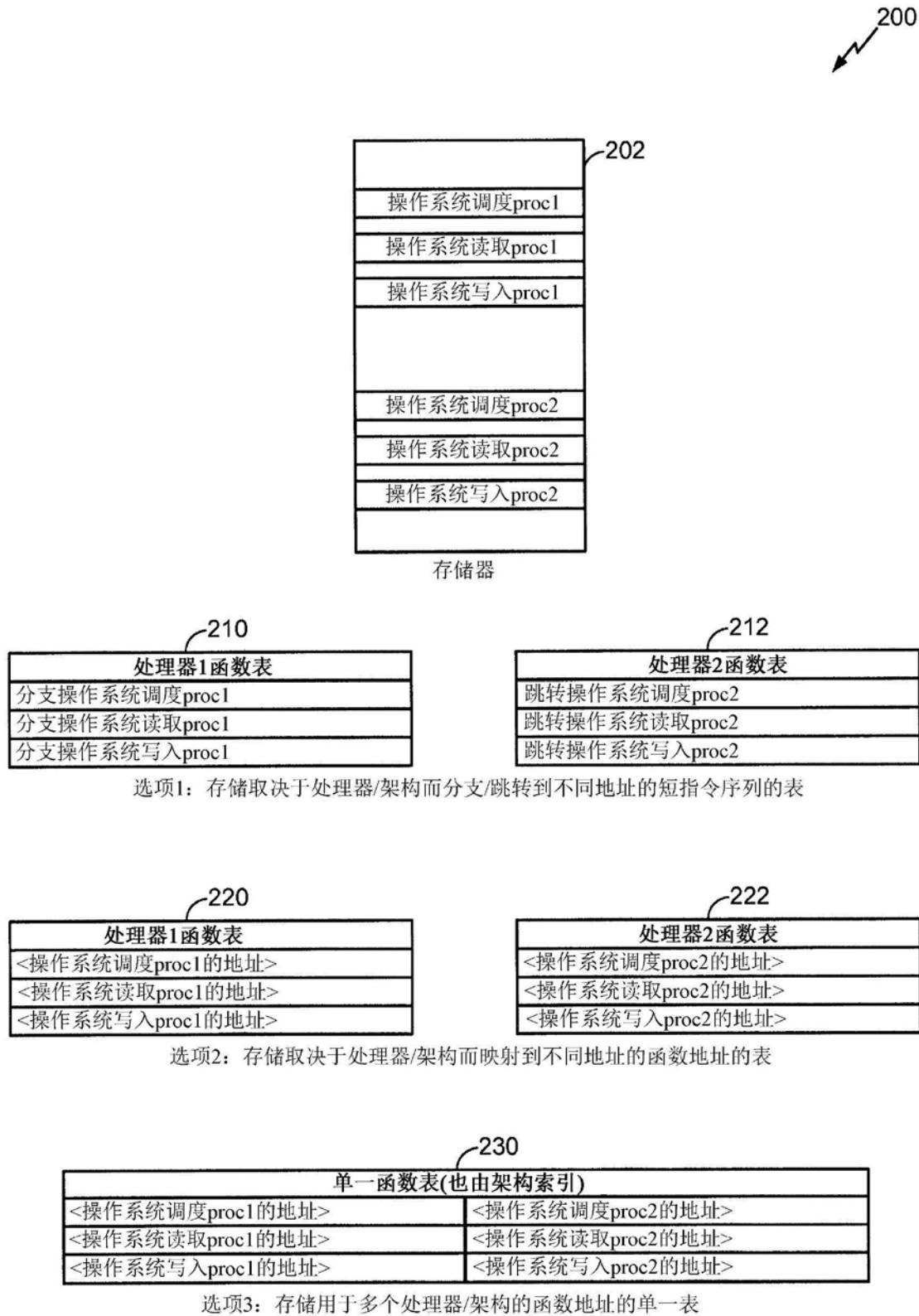


图2

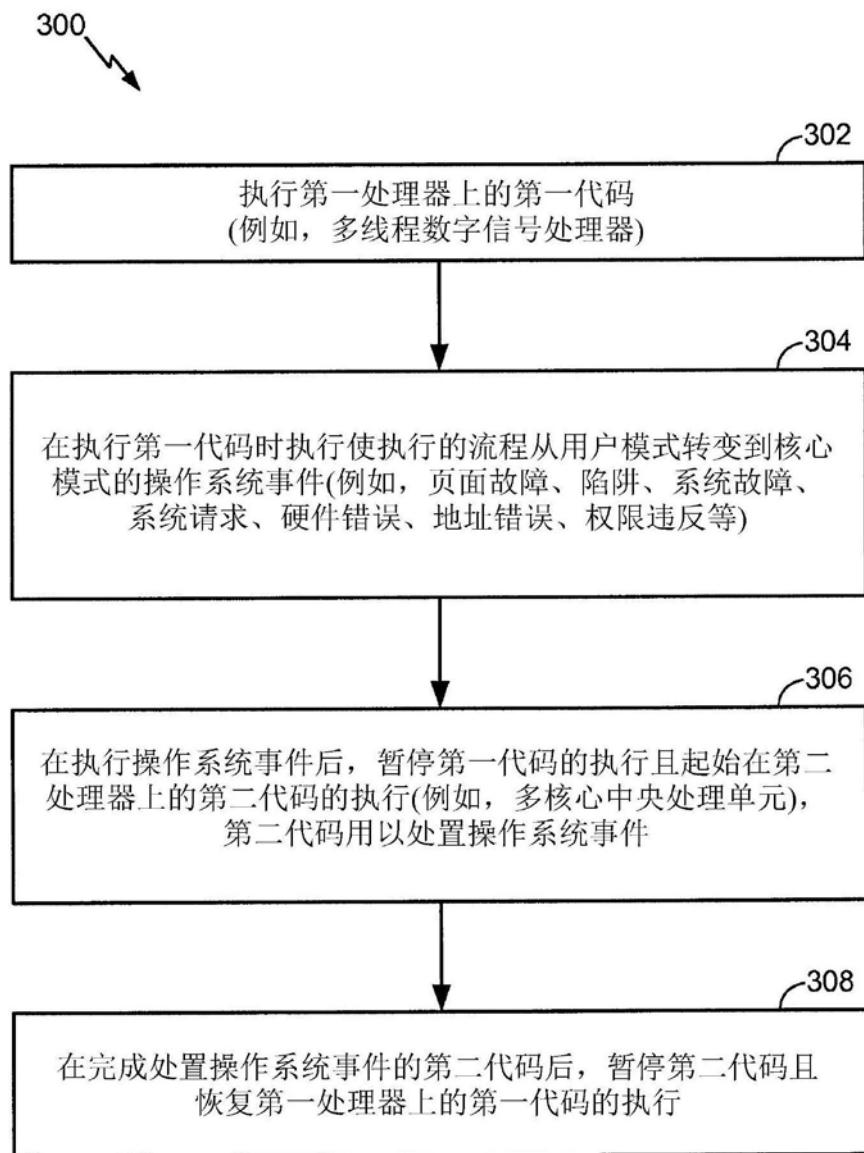


图3

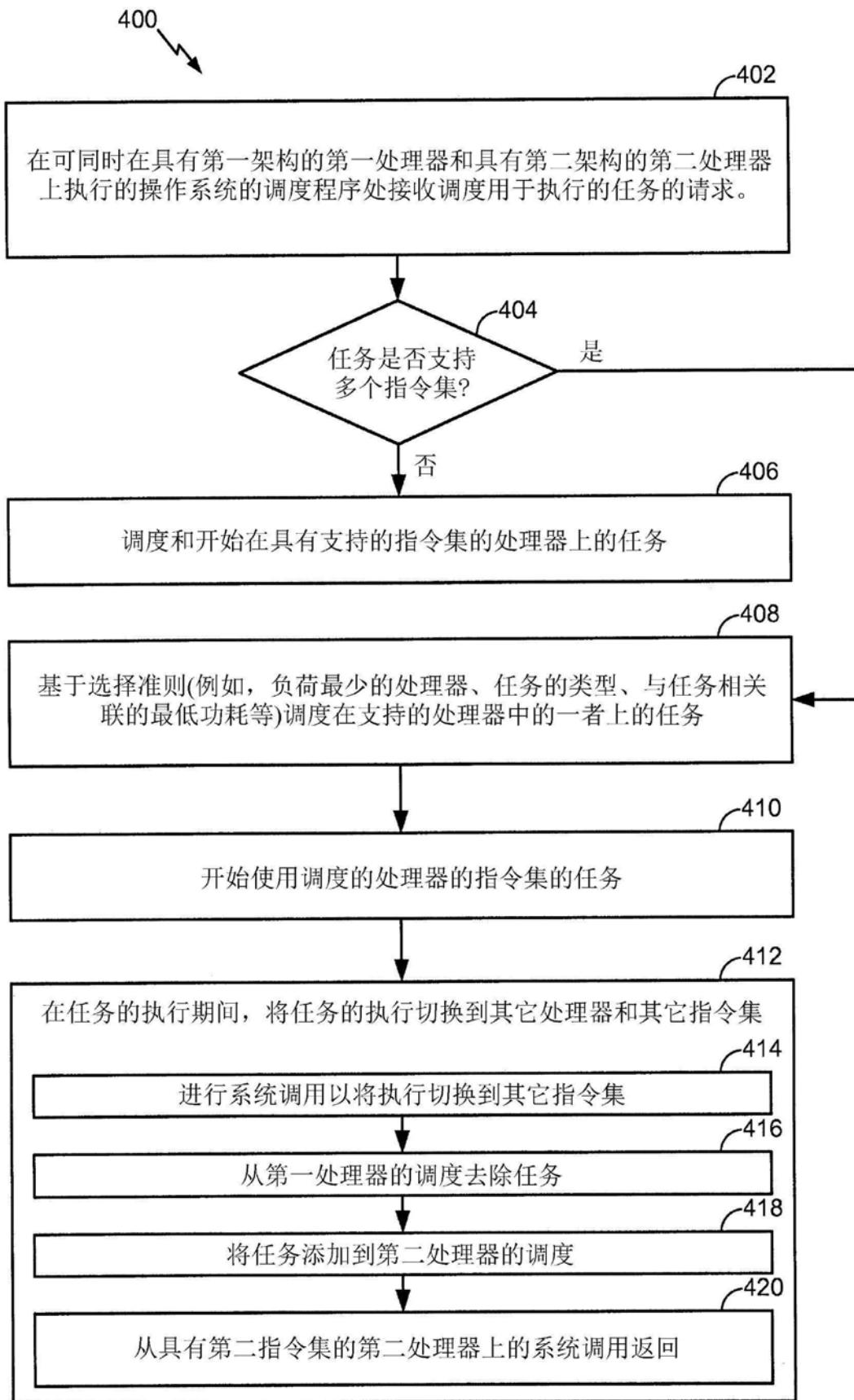


图4

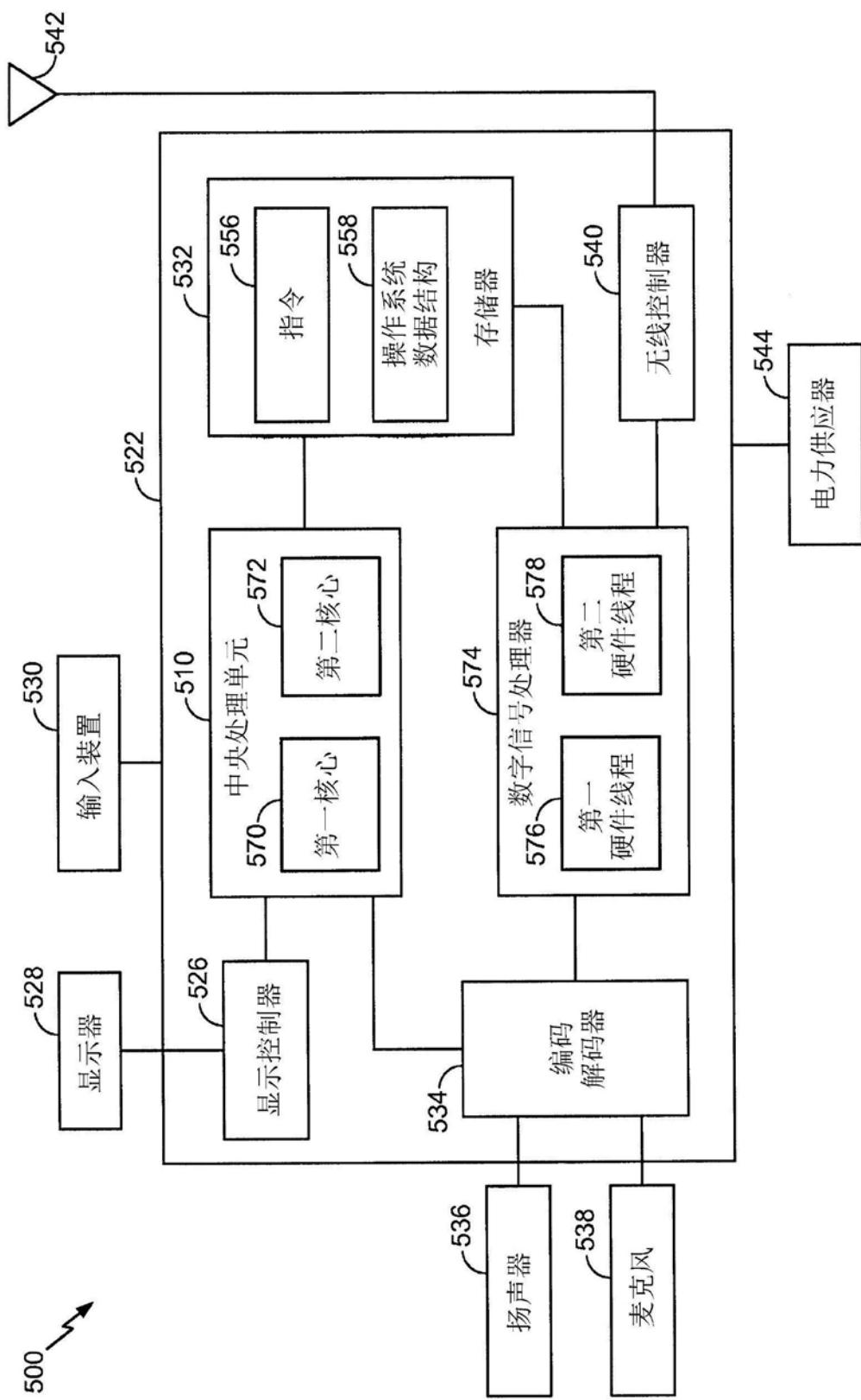


图5