

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 1/32 (2006.01)

G06F 1/20 (2006.01)



[12] 发明专利说明书

专利号 ZL 200680001272.7

[45] 授权公告日 2009年10月28日

[11] 授权公告号 CN 100555174C

[22] 申请日 2006.5.9

[21] 申请号 200680001272.7

[30] 优先权

[32] 2005.5.10 [33] US [31] 11/125,495

[86] 国际申请 PCT/JP2006/309616 2006.5.9

[87] 国际公布 WO2006/121175 英 2006.11.16

[85] 进入国家阶段日期 2007.5.22

[73] 专利权人 索尼计算机娱乐公司

地址 日本东京

[72] 发明人 铃置雅一

[56] 参考文献

JP2005-99984A 2005.4.14

US6247041B1 2001.6.12

US6240521B1 2001.5.29

US2003/0079151A1 2003.4.24

US2002/0002578A1 2002.1.3

US5913068A 1999.6.15

US2005/0050373A1 2005.3.3

审查员 李 锋

[74] 专利代理机构 中国国际贸易促进委员会专利
商标事务所

代理人 鲍 进

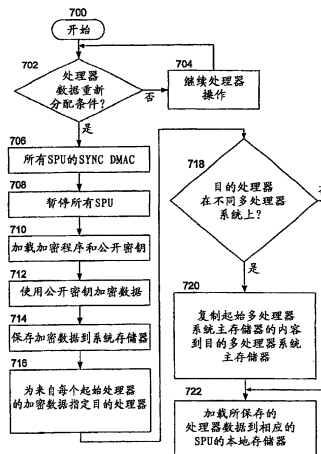
权利要求书4页 说明书27页 附图12页

[54] 发明名称

计算系统中功率管理的方法和设备

[57] 摘要

计算系统内的一或多个处理器和与本地存储器相关的处理器可以实际连接到允许在主存储器和本地存储器之间进行数据传送的主存储器，其中计算系统可以执行的操作包含：在计算系统内生成一或多个处理器的暂停条件；暂停一或多个处理器；把来自一或多个处理器的本地存储器的数据保存到主存储器；休眠一或多个处理器；恢复一或多个处理器的功率；把来自一或多个处理器的所保存的数据传送到一个发起处理器或到计算系统中指定的目的处理器。



1.一种用于管理多处理器计算系统中处理器中间的计算活动分布的方法，包括：

在遇到以下至少之一时，暂停多处理器系统内至少一个处理器的操作：(i)在所述至少一个处理器中执行程序中的检查点指令，以及(ii)预定执行周期的结束；

在所述至少一个处理器的本地存储器中加密存储在所述至少一个处理器的所述本地存储器中的数据，所述数据包括可执行程序代码；

把来自所述至少一个处理器的所述本地存储器的所述加密数据保存到所述多处理器系统的主存储器；

把所述保存数据从所述主存储器传送到所述多处理器系统中另一个处理器的本地存储器；

使所述另一个处理器进入保护模式；

在所述另一个处理器的所述本地存储器中解密所述传送的加密数据；

使所述另一个处理器退出所述保护模式；以及

在所述另一个处理器中恢复程序执行。

2.如权利要求1所述的方法，其中

所述至少一个处理器和所述另一个处理器中的至少之一的所述本地存储器提供可预测的访问时间；并且所述方法还包括：

在所述本地存储器和所述主存储器之间的传送是DMA传送；并且通过使所述DMA传送与数据计算重叠来实现时延隐藏。

3.如权利要求2所述的方法，其中所述本地存储器是SRAM。

4.如权利要求1所述的方法，还包括：

在所述多处理器系统的多个处理器中间重复所述暂停、保存、传送以及恢复步骤，使得功耗被分布在所述多个处理器中。

5.如权利要求1所述的方法，其中进入所述保护模式的所述步骤包括以下至少之一：

进入隔离模式，该模式中禁止数据传入和传出所述另一个处理器；以及

进入安全模式，该模式中不支持来自所述另一个处理器外部的设备、针对进出所述另一个处理器的数据传送的请求。

6.如权利要求1所述的方法，还包括以下至少之一：

使所述至少一个处理器进入休眠；

响应于唤醒条件，使所述至少一个处理器离开休眠；

在所述至少一个处理器离开休眠之后，从所述主存储器向所述至少一个处理器的所述本地存储器传送所述保存数据；以及

使所述至少一个处理器恢复程序执行。

7.如权利要求6所述的方法，其中使所述至少一个处理器离开休眠的所述步骤包含以下至少之一：

中断所述至少一个处理器；

允许预定时间周期结束；以及

在与所述至少一个处理器通信的处理器中执行检查点指令。

8.一种用于管理多处理器计算系统中处理器中间的计算活动分布的设备，包括：

用于在遇到以下至少之一时，暂停至少一个所述处理器的操作的装置：(i)在所述至少一个处理器中执行程序中的检查点指令，以及(ii)预定执行周期的结束；

用于在所述至少一个处理器的本地存储器中加密存储在所述至少一个处理器的所述本地存储器中的数据的装置，所述数据包括可执行程序代码；

用于把来自所述至少一个处理器的所述本地存储器的所述加密数据保存到所述多处理器系统的主存储器的装置；

用于把所述保存数据从所述主存储器传送到所述多处理器系统中另一个处理器的本地存储器的装置；

用于使所述另一个处理器进入保护模式的装置；

用于在所述另一个处理器的所述本地存储器中解密所述传送的

加密数据的装置;

用于使所述另一个处理器退出所述保护模式的装置; 以及
用于在所述另一个处理器中恢复程序执行的装置。

9.如权利要求8所述的设备, 其中

所述至少一个处理器和所述另一个处理器中的至少之一的所述本地存储器提供可预测的访问时间; 并且所述设备还包括:

用于使在所述本地存储器和所述主存储器之间的传送使用 DMA 传送的装置; 以及用于通过使所述 DMA 传送与数据计算重叠来实现时延隐藏的装置。

10.如权利要求9所述的设备, 其中所述本地存储器是SRAM。

11.如权利要求8所述的设备, 还包括:

用于在所述多处理器系统的多个处理器中间重复所述暂停、保存、传送以及恢复装置的操作以使得功耗被分布在所述多个处理器中的装置。

12.如权利要求8所述的设备, 其中所述用于使所述另一个处理器进入保护模式的装置包括以下至少之一:

用于进入隔离模式的装置, 该模式中禁止数据传入和传出所述另一个处理器; 以及

用于进入安全模式的装置, 该模式中不支持来自所述另一个处理器外部的设备、针对进出所述另一个处理器的数据传送的请求。

13.如权利要求8所述的设备, 还包括以下至少之一:

用于使所述至少一个处理器进入休眠的装置;

用于响应于唤醒条件, 使所述至少一个处理器离开休眠的装置;

用于在所述至少一个处理器离开休眠之后, 从所述主存储器向所述至少一个处理器的所述本地存储器传送所述保存数据的装置; 以及

用于使所述至少一个处理器恢复程序执行的装置。

14.如权利要求13所述的设备, 其中所述用于响应于唤醒条件, 使所述至少一个处理器离开休眠的装置包含以下至少之一:

用于中断所述至少一个处理器的装置;

用于允许预定时间周期结束的装置；以及
用于在与所述至少一个处理器通信的处理器中执行检查点指令
的装置。

计算系统中功率管理的方法和设备

技术领域

本发明涉及管理计算系统中的功耗和热量积聚，以及管理多处理器计算系统中处理器中间的计算活动分布的方法和设备。

背景技术

增加时钟频率以及降低现代微处理器的尺寸使计算性能以及在小占用面积内提供这种性能的方便性大大提高。然而，这些改进也导致微处理器中越来越多的功耗以及较大热量积聚。对图形处理器尤其如此。因而，本领域需要一种微处理器热量积聚问题的基于软件的解决方案，其取代例如布置吸热器的硬件解决方案，其作为其补充。

在多处理器计算系统中，有限空间内集中多个高性能处理器以及处理器间处理活动的不均匀分布可使热量积聚的问题更严重。随着在相同计算系统内某些处理器处理更多总计算吞吐量并且相应比不太活跃的处理器经历更大热能积聚，计算活动的这种不均匀分布可以导致局部“热点”的产生。因此，本领域需要多处理器系统中热点问题的软件解决方案。

发明内容

基于一个方面，本发明提供一种方法，包含：当在处理器中执行的程序中遇到检查点指令时，暂停计算系统内处理器的操作；把来自处理器的本地存储器的数据保存到计算系统的主存储器；以及使处理器休眠。优选地，该方法还包括在保存之前加密本地存储器中的数据。优选地，该方法还包括：在检测到唤醒条件时唤醒休眠的处理器。优选地，该方法还包括：把所保存的数据从主存储器传送到处理器的本地存储器。优选地，该方法还包括：在保存之前加密本地存储器中的数据；在唤醒之后使处理器进入保护模式；解密所传送的数据；以及

使处理器退出保护模式。

优选地，进入保护模式包括进入隔离模式，该模式中禁止数据传入和传出处理器。优选地，进入保护模式包括进入安全模式，该模式中不支持来自处理器外部的设备、针对进或出处理器的数据传送的请求。优选地，该方法还包括恢复处理器的操作。优选地，使处理器休眠包括关闭对处理器供电。优选地，使处理器休眠包括大大降低处理器的功耗水平。优选地，使处理器休眠包括至少大大降低处理器的功耗水平。优选地，使处理器休眠包括切断处理器的时钟信号。

优选地，唤醒处理器包括恢复对处理器的满供电；以及初始化处理器。优选地，唤醒处理器包括至少基本恢复对处理器的供电。优选地，该方法还包括在处理器操作恢复过程中向处理器通知其进展的阶段。

优选地，唤醒包括通知休眠的处理器唤醒条件的存在；以及从处理器发出信号以恢复对处理器的满供电。优选地，通知包括向休眠处理器发送中断。优选地，该方法还包括基于预定时间周期的结束产生唤醒条件。优选地，预定时间周期等于处理器的休眠周期。优选地，该方法还包括当与休眠处理器通信的第二处理器中执行检查点指令时产生唤醒条件。优选地，该方法还包括：把所保存的数据从主存储器传送到第二处理器的本地存储器；以及在第二处理器中恢复程序的执行。

基于另一个方面，本发明提供了一种方法，包括：等待第一多处理器系统中至少一个处理器的预定时长的执行周期的结束；在执行周期结束时暂停至少一个处理器中正在进行的程序执行，从而提供至少一个暂停处理器；以及把来自至少一个处理器的至少一个相应本地存储器的数据保存到第一多处理器系统的主存储器。优选地，该方法还包括为每个暂停处理器指定目的处理器，因而提供至少一个目的处理器；把来自主存储器的所保存的数据传送到至少一个目的处理器的至少一个相应本地存储器；以及恢复至少一个目的处理器中正在进行的程序执行，因而开始新的执行周期。优选地，该方法还包括在保存之

前加密至少一个暂停处理器的至少一个本地存储器中的数据。优选地，该方法还包括使至少一个目的处理器进入保护模式；解密所传送的加密数据；以及使至少一个目的处理器退出保护模式。

优选地，进入保护模式包括进入隔离模式，该模式中禁止数据传送进出至少一个目的处理器。优选地，进入保护模式包括进入安全模式，该模式中不支持来自至少一个目的处理器外部的设备、针对进或出至少一个目的处理器的数据传送的请求。优选地，该方法还包括重复等待、暂停、保存、指定、传送以及恢复。优选地，至少一个暂停处理器包括至少两个暂停处理器，并且其中指定包括基于处理器数据重新分配模式为每个暂停处理器指定目的处理器。

优选地，执行周期基本上等于至少一个处理器的帧处理时间。优选地，执行周期为大约16毫秒。优选地，执行周期的时长基本上小于至少一个处理器的临界热上升时间（critical thermal rise time）。优选地，至少一个目的处理器位于第一多处理器系统中。优选地，至少一个目的处理器位于第二多处理器系统上。优选地，该方法还包括复制来自第一多处理器系统的主存储器的所有数据到第二多处理器系统的主存储器。优选地，处理器数据重新分配模式包括模" n "旋转，其中" n "是第一多处理器系统上的处理器的数量。

基于另一个方面，本发明提供了一种计算系统，具有能够用于与主存储器通信的至少一个处理器；以及与至少一个处理器的每个连接的本地存储器，因而提供了至少一个本地存储器，其中计算系统在遇到在至少一个处理器中执行的程序中的检查点指令时，暂停计算系统内至少一个处理器的操作，因而提供至少一个暂停处理器，保存来自至少一个暂停处理器的本地存储器的数据到计算系统的主存储器；以及使至少一个暂停处理器休眠。

基于另一个方面，本发明提供了一种计算系统，具有能够用于与主存储器通信的至少一个处理器的计算系统；以及连接到至少一个处理器的每个的本地存储器，因而提供至少一个本地存储器，其中计算系统用于：等待计算系统的第一多处理器系统中至少一个处理器的预

定时长的执行周期的结束，在执行周期结束时暂停正在进行的至少一个处理器的程序执行，因而提供至少一个暂停处理器，以及把来自至少一个暂停处理器的至少一个本地存储器的数据保存到第一多处理器系统的主存储器。

基于另一个方面，本发明提供了一种包含可执行程序的存储介质，该可执行程序用于使计算系统执行操作，所述操作包含：在遇到处理器中执行的程序中的检查点指令时，暂停计算系统内处理器的操作；把来自处理器的本地存储器的数据保存到计算系统的主存储器；以及使处理器休眠。

基于另一个方面，本发明提供了一种包含可执行程序的存储介质，该可执行程序用于使计算系统执行操作，所述操作包含：等待第一多处理器系统中至少一个处理器的预定时长的执行周期的结束；在执行周期结束时暂停正在进行的至少一个处理器的程序执行，因而提供至少一个暂停处理器；以及把来自至少一个处理器的至少一个相应本地存储器的数据保存到第一多处理器系统的主存储器。

当在这里结合附图对本发明进行描述时，本领域技术人员会明白其它方面、特性、优点等等。

附图说明

为了图解本发明的各个方面，在附图中示出了当前最优的形式，然而可以理解，本发明不局限于所示的具体结构和手段。

图1是图解了适于基于本发明的一或多个方面使用的处理系统的模块图；

图2是图解了具有可以基于本发明的一或多个方面使用的两个或更多子处理器(SPU)的多处理系统的结构的图例；

图3是包含可以基于本发明的一或多个方面使用的图2的多个多处理器系统(或"多处理器系统")的计算系统的图例；

图4A是图解基于本发明的一或多个方面可以结合休眠过程和恢复处理器操作过程而执行的处理步骤的流程图的第一部分；

图4B是图解图4A中示出的基于本发明的一或多个方面执行的处

理步骤的流程图的第二部分;

图5是图解基于本发明的一或多个方面在休眠过程期间的数据处理的模块图;

图6是图解基于本发明的一或多个方面从休眠状态开始在恢复处理器操作过程期间的数据处理的模块图;

图7是图解基于本发明的一或多个方面可以执行以完成处理器数据重新分配过程的处理步骤的流程图;

图8是图解基于本发明的一或多个方面的多个处理器的处理器数据重新分配过程的数据保存和数据恢复部分的模块图;

图9是图解可以用于实现本发明的一或多个其它方面的优选处理器元件(PE)的图例;

图10是图解可以基于本发明的一或多个其它方面调整的图9的系统的示例性子处理单元(SPU)的结构图例; 以及

图11是图解可以基于本发明的一或多个其它方面调整的图9的系统的示例性子处理单元(SPU)的结构图例。

具体实施方式

在此公开中, 在保存处理器的本地存储器中存储的数据的背景下, 术语"数据"所有数据, 即除了可执行程序代码处理的数字和文本信息之外, 还包含可执行程序代码。这里, 术语"宽带引擎"(也可以被称作处理器元件)是多处理器计算系统。这里, 术语"休眠"通常是指这样的处理器条件, 其中处理器使用较少功率和/或比在正常操作时消耗更少功率。因而, 这种休眠包含这样的处理器条件, 所述条件包括一个功耗或消耗水平范围, 该范围为从仅仅降低水平的功耗或消耗直到零或接近零水平的功耗或消耗。

相应地, 术语"使处理器休眠"(以及这个动词的其它结合)是指把处理器设置在这种低功耗和/或低功率消耗条件或零功耗和/或零功率消耗条件下。使处理器休眠可以通过部分地或完全地降低提供给处理器的电压来实现。可选地, 处理器的电流可以被部分地或完全地降低。在另一个实施例中, 处理器的时钟信号可以通过改变程度 (degree)

或完全切断来降低。另外，可以使用降低或消除对处理器的供电的一或多个上述方案的组合。此外，本发明不局限于使用一或多个上述方案使处理器休眠。此外，术语休眠不局限于上述定义。

这里，术语"唤醒"处理器可以是指恢复处理器的供电或时钟信号提供。相应地，"唤醒条件"是指示唤醒处理器的请求的条件。然而，术语"唤醒"不局限于上述定义。

这里，术语"检查点"通常是指指示所期望的处理器状态和/或指示处理器的条件或指示处理器所处的计算系统的程序指令或指令组。在一个实施例中，处理器的程序代码中的检查点命令处理器进入也被称为休眠状态的休眠条件。在一个实施例中，多个可能休眠状态功率水平的一或多个可以在检查点指令中指定。在另一个实施例中，检查点可以命令一或多个处理器把一或多个相应本地存储器的内容转移到相应处理器外部的存储器。注意，检查点可以被用来表示除了上面注解的那些之外的状态和/或指令。此外，术语"检查点"不局限于上述定义。

这里，术语"热点"可以是指装置内处于高于期望的温度的位置。在一个实施例中，如果装置内的位置高于阈值温度，则热点存在，其中阈值可以被预定或可以在操作装置期间被调节。

在一个实施例中，术语"热点"可以是指多处理器计算系统中的处理器，其是多处理器计算系统的任何处理器中最热(处于最高温度)的处理器。可选地，术语"热点"可以是指多处理器计算系统中处于比大多数其它处理器更高的温度的处理器。然而，术语"热点"不局限于上述定义。

这里，术语"计算系统"可以是指用于计算的装置。在一个实施例中，计算系统可以是单处理器。在另一个实施例中，计算系统可以是例如宽带引擎的多处理器设备，其也被称为处理器元件(PE)。在另一个实施例中，术语"计算系统"可以是指包含多个多处理器系统以及可能包含其它设备的装置，所述其它设备包含但不局限于多处理器系统外部的存储器，例如总线系统的数据传输装置以及数据通信控制设备。

然而，术语"计算系统"不局限于上述定义。

这里，术语"正在进行的程序执行"以及"正在进行的执行"可以是指用于实现处理器数据重新分配过程和/或处理器休眠、除了已执行的那些操作之外的操作。然而，术语"正在进行程序执行"不局限于上述定义。这里，"暂停"处理器可以是指停止该正在进行的程序执行。在被暂停之后，处理器优选地仍然能够执行与休眠相关的任务，以及处理器数据重新分配过程。然而，术语"暂停"不局限于上述定义。

参照附图，其中类似数字表示类似单元，其中在图1示出了适于使用本发明的一或多个方面的处理系统100。为了简洁和清晰，图1的模块图涉及和描述这里如图所示的装置100，然而应当理解，该描述可以被轻易地应用于具有等同效果的方法的各个方面。装置100优选地包含处理器102、本地存储器104、优选为动态随机访问存储器(DRAM)的系统存储器106以及总线108。

可以使用能够向系统存储器106请求数据以及操作数据以实现期望结果的任何已知技术来实现处理器102。例如，处理器102可以使用能够执行软件和/或固件的任何已知微处理器来实现，包含标准微处理器、分布式微处理器等等。例如，处理器102可以是图形处理器，其能够请求和操作数据，例如包含灰度信息、色彩信息、纹理数据、多边形信息、视频帧信息等等的像素数据。

尤其是，本地存储器104优选地与处理器102位于相同的芯片上；然而，本地存储器104优选地不是硬件高速缓冲存储器，其原因在于优选地没有片上或片外硬件高速缓存电路、高速缓存寄存器、高速缓冲存储器控制器等等来实现硬件高速缓冲存储器功能。在可选实施例中，本地存储器104可以是高速缓冲存储器和/或可以使用附加高速缓冲存储器。由于片上空间常常有限，本地存储器104的大小可以远小于系统存储器106。处理器102优选地提供数据访问请求以从系统存储器106通过总线108把数据复制到本地存储器104以进行程序执行和数据操作。可以使用例如直接存储器访问(DMA)技术的任何已知技术来实现利于数据访问的机构。

装置100也优选地包含例如通过总线108与处理器102连接的存储介质，例如只读存储器(ROM)。存储介质110优选地包含受信解密程序，其可读到处理器102的本地存储器104并且使用安全解密密钥来解密信息。优选地，存储介质110是永久可编程设备(例如，快擦写ROM)，其中实现某等级的安全性，使得解密程序产生受信功能并且不能由外部软件操作来篡改。存储介质110的安全性是优选，使得解密程序和/或其它信息(例如受信解密密钥)不可被未经授权的实体访问。例如，在装置100的制造期间优选地把解密程序建立和存储在存储介质110中。

优选地，处理器102和本地存储器104被布置在公共集成电路上。因而，这里的这些单元可以被称作"处理器102"。在可选实施例中，存储介质110也可以布置在具有一或多个其它单元的公共集成电路上。

图2是图解具有两个或更多子处理器102的多处理器系统100A(这里也称作多处理系统)的结构图例。这里其它地方的概念可以被应用于多处理器系统100A。系统100A包含多个处理器102A-102D，相关本地存储器104A-104D，和通过总线系统108互连的共享存储器106。这里共享存储器106也可以被称作主存储器106或系统存储器106。尽管通过例子图解了四个处理器102，但是可以在不偏离本发明的实质和范围的前提下使用任何数量的处理器。处理器102可以全部具有相同构造或可以包含不同构造。

本地存储器104优选地与其相应处理器102位于相同芯片(相同半导体衬底)上。然而，本地存储器104优选地不是常规硬件高速缓冲存储器，其原因在于不存在片上或片外硬件高速缓存电路、高速缓存寄存器、高速缓冲存储器控制器等等来实现硬件高速缓冲存储器功能。

处理器102优选地提供数据访问请求以从系统存储器106通过总线系统108把数据复制到相应本地存储器104以进行程序执行和数据操作。优选地使用每个处理器(未示出)的直接存储器访问控制器(DMAC)实现利于数据访问的机构。针对本发明的其它特性，每个处理器的DMAC优选地具有与这里其它地方所述基本相同的功能。

系统存储器106优选地是通过高带宽存储器连接(未示出)连接到处理器102的动态随机访问存储器(DRAM)。另外或可选地, DRAM 106可以通过总线系统108连接到处理器102。尽管系统存储器106优选地是DRAM, 但是存储器106可以使用例如静态随机访问存储器(SRAM)、磁性随机访问存储器(MRAM)、光存储器、全息存储器等等的其它手段来实现。

每个处理器102优选地使用处理流水线来实现, 其中逻辑指令以流水线的方式处理。尽管流水线可以被分成处理指令的任何数量的级段, 但是流水线通常包括获取一或多个指令、解码指令、检查指令间的依赖、发出指令以及执行指令。在这点上, 处理器102可以包含指令缓冲器、指令解码电路、依赖检查电路、指令发送电路以及执行级段。

在一或多个实施例中, 处理器102和本地存储器104可以被布置在公共半导体衬底上。在一或多个其它实施例中, 共享存储器106也可以被布置在公共半导体衬底上, 或它可以被分别布置在例如分立半导体衬底上。

在一或多个可选实施例中, 一或多个处理器102可以充当连接到其它处理器102并且能够通过总线108连接到共享存储器106的主处理器。主处理器可以调度和协调其它处理器102对数据的处理。然而, 不同于其它处理器102, 主处理器可以被连接到硬件高速缓冲存储器, 其用于缓存从共享存储器106以及处理器102的一或多个本地存储器104中的至少一个获得的数据。主处理器可以提供数据访问请求以使用例如DMA技术的任何已知技术, 从系统存储器106通过总线108把数据(其可以包含程序数据)复制到高速缓冲存储器以进行程序执行和数据操作。

在一个实施例中, 多处理系统110A还可以包含直接或通过总线系统108连接到共享存储器106和处理器102A-102D的存储介质(未示出)。这个存储介质可以是随机访问存储器(RAM)、只读存储器(ROM)、DRAM、可编程只读存储器(PROM)或其它常规数据存储实现。此外, 附加数据存储设备(未示出)也可以直接或通过总线系统108被连接到

共享存储器106和/或处理器102A-102D。

图3是包含适于基于本发明的一或多个方面使用的多个多处理器系统的计算系统300的模块图。多处理器系统100A1-100A3可以等价于图2的多处理器系统110A，但不限于这个实现。计算系统300还可以包含主存储器302和引入易失和/或非易失存储器(未示出)，和/或随机访问和/或顺序存储器访问数据存储设备(未示出)的附加存储设备。计算系统300优选地还包含总线304。计算系统300也可以包含其它数据通信以及数据通信控制装置(未示出)。为了简单，图3中省略了每个多处理器系统100A1-100A3的内部结构的图解。

图4A和4B(集中称作图4)一起形成一个流程图，该流程图图解了可以结合基于本发明的一或多个方面的休眠过程和处理器唤醒过程来执行的处理步骤。下面，结合图4的处理步骤序列的描述，对图4-6进行参考。

处理从操作400处开始。在操作402中，可以是单处理器102的计算系统优选地确定暂停条件是否已经建立。如果不存在暂停条件，则在操作404中，正在进行的程序执行优选地继续进行。如果暂停条件已经建立，则在操作406中，处理继续进行(在下面讨论)。

在一个优选实施例中，当遇到在处理器102中执行的程序中的检查点指令时，处理器102产生暂停条件，其中检查点指令向处理器102表明处理器102将要经历休眠过程。然而，在可选实施例中，要休眠的处理器可以从连接到处理器102的另一个设备处接收相关指示。例如，在多处理器系统中，一个处理器可以向自身和/或向多处理器系统内的一或多个其它处理器发送指示暂停条件的建立的信号。

在另一个可选实施例中，连接到要暂停的处理器102的定时器(软件驱动或硬件驱动)可以超时并且向处理器102发送中断信号，该信号指示暂停条件。按照这种方式，暂停条件可以基于实时调度来建立。

在另一个实施例中，暂停条件可以基于计算系统内一或多个指定事件的出现而产生。这些事件可以包含但不限于：要暂停的处理器所位于的计算系统的工作负载水平低于指定阈值，以及计算系统内热条

件的出现，例如计算系统中的测试点超过某个阈值温度。此外，在一个可选实施例中，指定事件可以是与要暂停的处理器102通信的处理器执行检查点指令。

在操作406中，如果建立暂停条件，则优选地暂停处理器102中的程序执行。

参照图5，在图解本地存储器104的第一个块中，示出了堆指针604和栈指针606的位置。从图解本地存储器104的第一个块指向图解本地存储器104的第二个块的箭头通常指向在这2个块之间的切换，并且不具体地映射任何两个存储器位置之间的数据传送。

箭头604和608优选地分别标识本地存储器104中的堆指针和栈指针的位置。优选地，描述指针604和608的位置的信息被分别存储在本地存储器104的块626和628中，以便以后传送到主存储器106以及最终恢复到本地存储器104。

参照图4和5，在操作408中，加密程序和公开密钥618优选地被从主存储器106载入本地存储器104的自由空间606。在操作410中，优选地使用加密程序和公开密钥618加密本地存储器104中的数据602。优选地，堆栈610也在操作410中被加密。加密程序自身也可以被加密。在操作412中，加密的程序和数据(一起被称为"模块")612被优选地保存到系统存储器106。加密堆栈614也被优选地保存到系统存储器106。优选地，本地存储器104中的堆指针604和栈指针608被保存到主存储器106。用附图标记616表示存储在主存储器106中的指针604和608的组合。指针604和608优选地在保存到主存储器106之前不被加密。然而，在可选实施例中，指针604和608可在被保存到主存储器106之前被加密。一旦完成操作412，主存储器106可以具有剩余自由空间620。

在操作414中，处理器102的功耗被降低，因而把处理器102设置在休眠状态。在一个实施例中，在休眠状态下，处理器102的用电量为零或接近零。在其它实施例中，休眠处理器102可以具有一个处理器功耗水平，该水平与处理器102的正常操作功率水平相比要低出各种不同的量。在一个实施例中，指示所选择的处理器进入休眠状态的实体可

以选择所期望的休眠功率水平，并且执行操作以使处理器在所选择的休眠功率水平休眠。相关"实体"可以是处理器102自身，或与要休眠的处理器通信的设备。

在操作416中，进行关于处理器唤醒条件是否已经建立的判定。根据处理器102的功率水平，这个判定可以由休眠处理器102自身或与休眠处理器102通信的另一个设备进行。具体地，如果对休眠处理器102的供电被完全关闭，则唤醒条件优选地由除了休眠处理器102之外的设备检测。可选地，如果休眠处理器102的工作条件例如允许进程某种基本通信活动(例如利用非零电源水平)，例如中断信号的检测或接收串行或并行数据通信，则休眠处理器102自身可以检测唤醒条件。

在某些实施例中，指定时间段的结束、处理器102内发生的事件、或休眠处理器102所处的计算系统内其它地方出现的事件可导致处理器恢复条件的产生。在一个实施例中，当与休眠处理器102通信的另一个处理器遇到指示处理器的唤醒条件的程序指令(例如程序检查点)时，可以建立处理器唤醒条件。在另一个实施例中，当与休眠处理器102通信的一或多个其它处理器，或休眠处理器102所处的计算系统的工作负载水平超过指定阈值时，可以产生处理器唤醒条件。

在对休眠处理器102的供电被完全关闭的实施例中，与休眠处理器102通信的另一个处理器("检测处理器")或其它适当设备优选地检测处理器唤醒条件。此后，对休眠处理器102的供电优选地被重新接通，因而唤醒(或"唤起")休眠处理器102。如果需要，则唤起的处理器优选地经历常规初始化过程。此后，检测处理器优选地通知唤起的处理器102它已经从休眠状态唤起的事实，以指示唤起的处理器102进行其下一个操作。优选地，这个下一个操作是下面讨论的操作420。唤起的处理器102的状态的通信可以通过中断信号传输，总线系统上的数据传输或其它已知通信方法来完成。在某些实施例中，多个不同中断可以被用来触发唤起的处理器102的某个范围的可能唤醒后操作。

在休眠处理器102的功率水平足够高以致允许进行基本处理器操作的实施例中，可以把中断发送到休眠处理器102以表示处理器唤醒条

件。此后，唤起的处理器102可以自身选择要执行的下一个操作。可选地，外部设备可以发送向恢复的处理器指示要执行的下一个操作的第二中断。

返回操作416，如果恢复条件未被建立，则在操作418中，休眠处理器102的低功率或零功率水平被优选地保持。如果唤醒条件被建立，则在操作420中，优选地恢复对处理器102的操作供电。

参照图4和6，在操作422中，处理器102优选地从主存储器106把所加密的模块612、所加密的堆栈614以及指针616(优选地为非加密的)传送到本地存储器104。处理器102也优选地从存储介质110把解密程序和私有密钥602加载到本地存储器104。这里注意，解密程序和私有密钥在这里集中地用附图标记602表示。在一个优选实施例中，存储介质110优选地是易失只读存储器(ROM)。然而，在可选实施例中，其它高度受信的存储器可以被用作存储介质110。

通常，处理器102用于改变状态以便实现各种条件状态。这些状态会被简要介绍并且接着在下面更详细地讨论。在第一状态中，处理器102处于普通模式。处理器102优选地用于切换到第二状态，该状态中处理器102处于保护操作模式。处理器102也优选地用于切换到第三状态，该状态中受信解密代码可以被调用并且在处理器102内执行。

在本公开内容中考虑了两种保护模式，这些模式提供本地存储器104的不同程度的隔离，以避免与处理器102外部的存储器产生潜在破坏性的相互作用。在这个部分中讨论的模式是安全模式和隔离模式。

在安全操作模式中，不支持针对处理器102的本地存储器104(或任何其它存储器设备、寄存器等等)中存储的数据的请求，因而确保执行敏感操作的受信环境。尽管处于安全模式中，然而处理器102可以请求从系统存储器106传送数据到本地存储器104，或可以请求从本地存储器104传送数据到系统存储器106。进一步地，在处理安全操作模式的同时，处理器102可以启动进、出受信环境的数据传送，而不管源或目的地如何。在一个优选实施例中，当处于保护操作模式中的一种时，受信环境限于处理器102及其包含的本地存储器104。

在隔离操作模式中，处理器102和任何外部设备均不可以启动进或出处理器102的本地存储器104的数据传送。

在一个优选实施例中，在进入一种保护模式之前，从主存储器106和/或存储介质110中读取例如加密和/或解密程序的任何程序。然而，在可选实施例中，一旦实现安全操作模式，则处理器102可以从存储介质110把解密程序读到本地存储器104中。优选地，受信解密密钥也存储在存储介质110内，并且也被保存到本地存储器104中以便以后使用。

在操作424中，处理器102优选地进入保护模式。如这里所述，保护模式可以是隔离模式或安全模式。隔离模式是优选的。然而，在可选实施例中，可以使用安全模式。

优选地，一旦处理器处于保护模式，则在操作426中，处理器102使用私有密钥解密加密的模块612并且把解密的信息存储在本地存储器104中。在操作428中，处理器102优选地退出保护模式。在操作430中，正在进行的程序执行优选地在唤起的处理器中恢复。

在可选实施例中，在操作406中暂停的正在进行的程序执行可以在除了操作414中休眠的处理器之外的处理器中恢复。优选地，由其它处理器执行操作422至430，以允许按照这种方式恢复正在进行的程序执行。对在"其它"处理器上执行操作420的需要将取决于其它处理器是否在开始操作422之前启动或休眠。

图7是图解可被执行以进行基于本发明的一或多个方面的处理器数据重新分配过程的流程图。图8是基于本发明的一或多个方面图解多个处理器的处理器数据重新分配过程的数据保存和数据恢复部分的模块图。下面参考图7-8进行图7的处理步骤的讨论。结合图2的多处理器系统实施例讨论图7的方法，但不限于这个实施例。

处理优选地从操作700开始。在操作702中，确定处理器数据重新分配条件是否已经建立。如果未建立这种条件，则处理器操作继续进行操作704。如果建立处理器数据重新分配条件，则处理继续进行操作706。

处理器数据重新分配条件优选地由执行周期的结束来驱动。执行

周期优选地具有预定时长。然而，在可选实施例中，可以由处理器102在正在进行的程序执行期间确定执行周期的时长。优选地，在执行周期期间，各个处理器优选地没有中断地完成正在进行的程序执行。

在一个实施例中，执行周期的时长可以基于SPU的帧速率设置。在一个优选实施例中，帧速率是每16毫秒一个帧，并且因此执行周期时长是16毫秒(msecs)。

在一个实施例中，处理器102准备数据以在屏幕上呈现图像。在这个屏幕上的每个相继图像是一个“帧”，并且帧被处理并且在屏幕上呈现的速率是“帧速率”。由于通常期望无需中断处理器102执行完成单个帧所需的处理，所以处理器102完成单个帧所需的时间对应于有用和方便的执行周期时长。然而，在可选实施例中，可以使用等于帧处理时间的倍数的执行周期时长。在其它可选实施例中，可以使用小于帧处理时间的执行周期时长。

在其它实施例中，当然可以调整分隔相继处理器数据重新分配过程的时间段以匹配不同的帧速率。此外，在其它可选实施例中，执行周期可以具有短于或长于经历处理器数据重新分配的一或多个SPU的帧速率的时长。

在一个优选实施例中，处理器的执行周期的时长基于处理器102的一或多个特性来确定。确定执行周期的最重要的两个特性是温度速率(处理器102温度增加的速率，按度每秒来测量)和作为温度速率的函数的“临界热上升时间”。“临界热上升时间”是处理器102从正常工作温度升温到临界点期间的时段。临界点是处理器温度优选地始终保持低于，并且处理器温度应当从不超过的温度。临界热上升时间是处理器在执行高度需求的应用程序时从正常工作温度上升到临界点所需的时间。由于需求的应用程序的功率需求可能变化，所以临界热上升时间的时长可以在一个范围内变化。

通常，执行周期足够低，使得在参与处理器数据重新分配的处理器间进行功耗平均。在一个实施例中，执行周期是临界热上升的50%或更低。在另一个实施例中，执行周期低于或等于临界热上升时间的

50%但是大于其35%。在另一个实施例中，执行周期小于等于临界热上升时间的35%但是大于其20%。在另一个实施例中，执行周期小于或等于临界热上升时间的20%但是大于其10%。在另一个实施例中，执行周期小于等于处理器102的临界热上升时间的10%。然而，本发明不局限于执行周期的上述时长的具体范围。

在其它可选实施例中，处理器数据重新分配条件可以由除了指定时间段的结束之外的其它事情来驱动，例如在SPU所位于或与一或多个SPU通信的计算系统内事件的出现。例如，超出指定阈值的(处理器的或整个计算系统的)工作负载水平可以建立处理器数据重新分配条件。在另一个实施例中，例如超过指定阈值的测试点(在SPU中或计算系统的其它地方)温度的热条件可以被用来产生处理器数据重新分配条件。

在操作706中，优选地同步SPU 102的DMAC(未示出)。在操作708中，优选地暂停SPU 102的操作。暂停SPU 102优选地包含允许完成全部直接存储器访问存储器操作，以及允许正在进行的程序执行停止。

在操作710中，SPU 102优选地把加密程序和相关公开密钥加载到其相应的本地存储器。优选地，单个多处理器系统110A内的SPU 102使用公共公开密钥和公共私有密钥。然而，在可选实施例中，不同公开密钥/私有密钥对可以被多处理器系统内的不同SPU使用。在具体本地存储器104x的内容被重新分配给不同多处理器系统上的处理器本地存储器的情况下，目的多处理器系统的公开密钥应当被载入该本地存储器104x。

在操作712中，优选地使用所存储的公开密钥加密在相应本地存储器中的数据。在操作714中，加密的数据被保存到系统存储器106。参考图8，来自四个本地存储器104A-104D(LM0-LM3)的数据通过数据传送800，804，808和812被保存到系统存储器106。在数据传送箭头上示出的符号中，"S"是指保存，"L"是指加载，"D"是指数据，字母"D"之后的数字标识数据所来自的本地存储器的编号。因而，"S(D1)"804标识把来自本地存储器1 104B的数据保存到系统存储器106，以及

L(D1)806标识把来自本地存储器1的数据加载到本地存储器2 104C。

在操作716中,为来自相应本地存储器104A-104D的数据指定目的处理器。应当理解,在这个部分中讨论的"指定"操作可以在操作722的"加载"之前任一点处执行。在一个实施例中,用于执行操作716的指定的处理器数据重新分配模式对于大量顺序处理器数据重新分配过程可以是固定的。可选地,这个模式可以针对每个处理器数据重新分配过程独立地选择。

图8的模块图图解了把来自起始(originating)处理器的数据的传送映射到目的处理器的一个可能处理器数据重新分配模式(或"重新分配模式")。然而,在可选实施例中,许多其它处理器数据重新分配模式可以被实现。

在图8的图解实施例中,示出了四个SPU的本地存储器104A-104D。示出四个"保存"操作800, 804, 808以及812,这些操作把来自相应本地存储器的数据移动到系统存储器106。也示出了四个"加载"操作802, 806, 810和814。在这个实施例中,处理器数据重新分配模式简单地涉及把起始本地存储器的标识号增加1,其中明显的例外是数据来自最高编号的本地存储器。来自最高编号的本地存储器的处理器数据最终被存储在本地存储器0 104A,即最低编号的本地存储器中。这个重新分配模式被称作"模4"旋转。容易理解,这个处理器数据重新分配模式可以轻易地被应用于较大数量的处理器,例如在多处处理器系统的一个实施例中发现的8个处理器。因而,在这种情况下,可以使用"模8"旋转。此外,在涉及n-SPU(其中"n"是SPU的数量)多处处理器系统的情况下,处理器数据重新分配过程可以同时在这个多处处理器系统内的所有SPU上或在一或多个SPU的任何子集上执行。在最简单的情况下,如果传送仅针对单处理器的数据,则处理器数据重新分配模式对应于把来自这个单处理器的数据保存到主存储器,以及随后把所保存的数据传送回到这个单处理器。

在另一个实施例中,在每个处理器内的每个SIMD(单指令,多个数据)流水线内的任务可以在每个处理器内被模4旋转,以避免在每个

处理器内产生任何热点。在其它实施例中，在指定处理器内对来自每个流水线的数据的重新分配可以基于除了模4之外的模式来重新分配。按照这种方式，流水线数据重新分配优选地实现各个流水线间的功耗平均。

在操作718中，针对来自本地存储器104A-104D中的一个的数据的每个加载操作，优选地确定目的本地存储器是否在与起始SPU位于不同多处理器系统的SPU上。如果目的SPU与起始SPU在相同多处理器系统上，则处理优选地继续操作722。如果目的SPU和起始SPU在不同多处理器系统上，则在操作720中，起始多处理器系统上的主存储器106的数据优选地被转储或复制到目的多处理器系统上的主存储器。优选地实现这个主存储器数据传送，以保证目的多处理器系统中的目的SPU可根据需要得到最初位于起始多处理器系统主存储器106、支持先前在起始多处理器系统中的起始SPU上正在进行的操作所需的程序和数据。

在操作722中，来自本地存储器104A-104D的所保存的数据优选地被载入相应目的SPU的本地存储器。

在此，将描述适于实现此处所述一个或多个特征的多处理器系统的优选计算机体系结构。根据一个或多个实施例，可将多处理器系统实现为用于富媒体应用（例如，游戏系统、家庭终端、PC系统、服务器系统和工作站）的独立和/或分布式处理的单芯片解决方案。在某些应用（例如游戏系统和家庭终端）中，可能需要进行实时计算。例如，在实时分布式游戏应用中，一个或多个联网图像解压缩、3D计算机图像、音频生成、网络通信、物理模拟和人工智能处理必须执行得足够快，以便能够为用户提供实时体验的假象。从而，在多处理器系统中的每个处理器优选地能够在很短和可预测的时间内完成任务。

为此，根据该计算机体系结构，优选地由公共计算模块（或单元）构建出多处理计算机系统的所有处理器。该公共计算模块优选地具有一致性结构，优选地整个采用同一指令集体系结构。多处理计算机系统可由一个或多个客户机、服务器、PC、移动计算机、游戏机、PDA、

机顶盒、电器、数字电视和使用计算机处理器的其它设备组成。

如果需要，一个或多个计算机系统也可以是网络成员。一致性模块化结构使得能够通过多处理计算机系统对应用和数据进行高效、高速处理，如果采用网络的话，则能够通过网络对应用和数据进行快速传送。该结构还简化了多种大小的网络成员的构建，优选地增强每个成员的处理能力，以及优选地利于由这些成员处理的应用程序的准备。

参照图 9，基本处理模块为处理器元件 (PE) 500。PE 500 包括 I/O 接口 502、处理器单元 (PU) 504 和多个子处理单元 508，即，子处理单元 508A，子处理单元 508B，子处理单元 508C 和子处理单元 508D。本地 (或内部) PE 总线 512 在 PU 504、子处理单元 508 和存储器接口 511 之间传送数据 and 应用程序。本地 PE 总线 512 可具有，例如，传统体系结构，或可将其实现为分组交换网络。如果实现为分组交换网络的话，尽管需要更多硬件，但增大了有效带宽。

能够使用用于实现数字逻辑的各种方法构建 PE 500。优选地将 PE 500 构建成采用在硅衬底上的互补金属氧化物半导体 (CMOS) 的单个集成电路。衬底的可选材料包括砷化镓 (gallium arsenide)、砷化镓铝 (gallium aluminum arsenide) 以及采用多种掺杂的其它所谓 III-B 化合物。PE 500 也可使用超导材料实现，例如，使用快速单磁通量子 (RSFQ) 逻辑。

PE 500 通过高带宽存储器连接 516 与共享 (主) 存储器 514 紧密相关。尽管存储器 514 优选地为动态随机访问存储器 (DRAM)，还可使用其它装置将存储器 514 实现为例如静态随机访问存储器 (SRAM)、磁随机访问存储器 (MRAM)、光存储器、全息存储器等。

PU 504 和子处理单元 508 优选地均与包含直接存储器访问 DMA 功能的存储器流控制器 (MFC) 相连，存储器流控制器 (MFC) 与存储器接口 511 结合使用，以便于在 PE 500 的 DRAM 514 与子处理单元 508 和 PU 504 之间的数据传送。应该注意，DMAC 和/或存储器接口 511 可相对子处理单元 508 和 PU 504 集成或独立布置。确实，

DMAC 功能和/或存储器接口 511 功能可与一个或多个（优选地，所有）子处理单元 508 和 PU 504 集成在一起。还应注意，DRAM 514 可相对 PE 500 集成或独立布置。例如，可如图所示将 DRAM 514 进行片外布置，或者，可以集成方式将 DRAM 514 进行片上布置。

PU 504 可为例如能够对数据和应用进行单独处理的标准处理器。在操作中，PU 504 优选地通过子处理单元对数据和应用程序的处理进行调度（schedule）和协调（orchestrate）。子处理单元优选地为单指令多数据（SIMD）处理器。在 PU 504 的控制下，子处理单元按照并行和独立方式对这些数据和应用程序执行处理。PU 504 优选地使用 PowerPC 核心实现，PowerPC 核心是采用精简指令集计算（RISC）技术的微处理器体系结构。RISC 使用简单指令的组合执行更复杂的指令。从而，处理器的定时可基于更简单和更快的操作，使得微处理器以给定时钟速度执行更多指令。

应该注意，PU 504 可使用承担主处理单元角色的一个子处理单元 508 来实现，此处所述主处理单元通过子处理单元 508 对数据和应用程序的处理进行调度和协调。此外，在处理器元件 500 内还可实现多于一个的 PU。

根据该模块化结构，具体计算机系统所采用 PE 500 的数量取决于该系统所需的处理能力。例如，服务器可采用四个 PE 500，工作站可采用两个 PE 500，PDA 可采用一个 PE 500。指定来处理特定软件单元的 PE 500 的子处理单元的数量取决于在单元内程序和数据的复杂度和量。

图 10 图解了子处理单元(SPU)508 的优选结构和功能。SPU 508 体系结构优选地填补了通用处理器（其用于实现宽范围的应用的高平均性能）与专用处理器（其用于实现单个应用的高性能）之间的空白。SPU 508 用于实现游戏应用、媒体应用、宽带系统等的高性能，并对实时应用的编程人员提供高级别控制。SPU 508 的某些功能包括图形几何流水线、表面细分、快速傅里叶变换、图像处理关键字、流处理、MPEG 编码/解码、加密、解密、设备驱动器扩展、建模、游戏物理技

术、内容创建和音频合成与处理。

子处理单元 508 包括两个基本功能单元，即，SPU 核心 510A 和存储器流控制器 (MFC) 510B。SPU 核心 510A 用于程序执行，数据操作等，而 MFC 510B 负责系统的 SPU 核心 510A 与 DRAM 514 之间数据传送的功能。

SPU 核心 510A 包括本地存储器 550、指令单元 (IU) 552、寄存器 554、一个或多个浮点执行级 556 和一个或多个定点执行级 558。本地存储器 550 优选地使用单端口随机访问存储器，例如，SRAM。然而多数处理器通过采用高速缓存来缩减存储器的时延，SPU 核心 510A 实现相对较小的本地存储器 550，而非高速缓存。当然，为了对实时应用（以及此处所提到的其它应用）的编程人员提供一致和可预知的存储器访问时延，在 SPU 508A 内的高速缓冲存储器体系结构并不是优选的。高速缓冲存储器的高速缓存命中与否 (hit/miss) 特性导致易失性存储器访问时间从数周期到数百周期。这种非易失性削弱了例如在实时应用编程中想得到的访问定时可预测性。通过将 DMA 传送与数据计算重叠，在本地存储器 SRAM 550 中可实现时延隐藏。这为实时应用的编程提供了高级别控制。由于与 DMA 传送相关的时延和指令开销超出了为高速缓存命中失误提供服务的时延和指令开销，当 DMA 传送尺寸足够大且充分可预测（例如，可在需要数据之前发出 DMA 命令）时，SRAM 本地存储器方法表现出优势。

运行在其中给定一个子处理单元 508 上的程序使用本地地址访问相关本地存储器 550，然而，还在总系统的存储器映射内为本地存储器 550 的每个位置指定实地址 (RA)。这允许特权软件 (Privilege Software) 将本地存储器 550 映射到处理进程的有效地址 (EA, Effective Address)，以便进行在一个本地存储器 550 与另一本地存储器 550 之间的 DMA 传送。此外，PU 504 也能够使用有效地址直接访问本地存储器 550。在优选实施例中，本地存储器 550 包含 556 千字节存储空间，寄存器 552 的容量为 128×128 位。

SPU 核心 504A 优选使用处理流水线实现，其中，以流水线方式

对逻辑指令进行处理。尽管可将流水线划分成对指令进行处理的任意数量的级段，不过流水线通常包括，获取一个或多个指令，将指令解码，检查指令之间的依赖，发出指令，和执行指令。在此方面，IU 552 包括指令缓冲器、指令解码电路、依赖检查电路和指令发送电路。

指令缓冲器优选地包含与本地存储器 550 相连接且用于临时性存储它们所获得指令的多个寄存器。指令缓冲器优选地按照所有指令成组，即基本同时离开寄存器的方式工作。尽管指令缓冲器可为任何尺寸，优选地，其尺寸不大于约两个或三个寄存器。

一般而言，解码电路将指令分解 (break down)，并产生执行相应指令功能的逻辑宏操作。例如，逻辑宏操作可指定运算和逻辑操作、对本地存储器 550 的加载和存储操作、寄存器源操作数和/或立即数据操作数。解码电路也可指示指令使用那些资源，例如，目标寄存器地址、结构资源、功能单元和/或总线。解码电路还可提供指示其中需要资源的指令流水线级段的信息。指令解码电路优选地能够基本同时将数量等于指令缓冲器的寄存器数量的若干指令解码。

依赖检查电路包括数字逻辑，用于执行测试以确定给定指令的操作数是否依赖流水线中其它指令的操作数。如果依赖，则不应执行给定指令，直到更新了该其它指令为止（例如，通过允许其它指令完成执行）。优选地，依赖检查电路确定从解码电路 112 同时分配的多个指令的依赖。

指令发送电路用于向浮点执行级 556 和/或定点执行级 558 发送指令。

优选地将寄存器 554 实现为相对较大的统一寄存器堆，例如，128 项寄存器堆。这允许在无需寄存器重命名的条件下得到深流水线化高频实现。重命名硬件通常会消耗处理系统中的较大部分面积和功率。从而，当通过软件循环展开 (software loop unrolling) 或其它交错技术掩盖延时，可实现有益操作。

优选地，SPU 核心 510A 属于超标量体系结构，使得在每个时钟周期中发出多于一个的指令。SPU 核心 510A 优选地以超标量方式工

作，该超标量达到与指令缓冲器的同时指令分配的数量，诸如在 2 和 3 之间（表示在每个时钟周期发出两个或三个指令）的数量相对应的程度。根据所需的处理能力，可能会采用或更多或更少的浮点执行级 556 和定点执行级 558。在优选实施例中，浮点执行级 556 以每秒 32×10^9 次浮点操作 (32GFLOPS) 的速度运行，定点执行级 558 以每秒 32×10^9 次操作 (32GOPS) 的速度运行。

MFC 510B 优选地包括总线接口单元 (BIU) 564、存储器管理单元 (MMU) 562 和直接存储器访问控制器 (DMAC) 560。除 DMAC 560 外，MFC 510B 优选地以与 SPU 核心 510A 和总线 512 相比为一半的频率（半速）运行，以符合低功耗设计目标。MFC 510B 用于处理从总线 512 进入 SPU 508 的数据和指令，为 DMAC 提供地址转换，以及用于保持数据一致性的窥探操作。BIU 564 提供在总线 512 与 MMU 562 以及 DMAC 560 之间的接口。从而，将 SPU 508（包括 SPU 核心 510A 和 MFC 510B）和 DMAC 560 与总线 512 进行物理和/或逻辑连接。

MMU 562 优选地用于将有效地址（由 DMA 命令获得）转换成用于存储器访问的实地址。例如，MMU 562 可将有效地址的高位转换成实地址位。不过，低地址位优选地不可转换的，并被视为逻辑和物理的，以用于形成实地址以及请求对存储器的访问。在一个或多个实施例中，MMU 562 可基于 64 位存储器管理模型实现，并可提供具有 4K-、64K-、1M-和 16M-字节页尺寸和 256MB 段尺寸的 2^{64} 字节有效地址空间。优选地，MMU 562 用于支持高达 2^{65} 字节虚拟存储器和用于 DMA 命令的 2^{42} 字节（4 万亿字节）物理存储器。MMU 562 的硬件可包括 8 项全关联 (fully associative) SLB，256 项 4 路组关联 (4 way set associative) TLB 和 TLB 的 4x4 置换管理表 (RMT) - 用于硬件 TLB 命中失误处理。

DMAC 560 优选地用于对来自 SPU 核心 510A 和诸如 PU 504 和/或其它 SPU 的一个或多个其它设备的 DMA 命令进行管理。存在有三类 DMA 命令：Put 命令，用于将数据从本地存储器 550 移动到共享

存储器 514; Get 命令, 用于将数据从共享存储器 514 移动到本地存储器 550 中; 以及存储控制命令, 包括 SLI 命令和同步命令。同步命令可包括原子命令、发送信号命令和专用障碍命令。根据 DMA 命令, MMU 562 将有效地址转换成实地址, 并将实地址转发到 BIU 564。

SPU 核心 510A 优选地使用通道接口和数据接口与 DMAC 560 内的接口进行通信 (发送 DMA 命令、状态等)。SPU 核心 510A 通过通道接口将 DMA 命令分配到 DMAC 560 中 DMA 队列。一旦 DMA 命令处在 DMA 队列中, 则通过在 DMAC 560 内的发送和完成逻辑对其进行处理。当完成对于 DMA 命令的所有总线事务时, 将完成信号通过通道接口向回发送到 SPU 核心 510A。

图 11 表示 PU 504 的优选结构和功能。PU 504 包括两个基本功能单元, PU 核心 504A 和存储器流控制器 (MFC) 504B。PU 核心 504A 实现程序执行、数据操作、多处理器管理功能等, 而 MFC 504B 执行与 PU 核心 504A 和系统 100 的存储器空间之间的数据传送相关的功能。

PU 核心 504A 可包括一级高速缓存 570、指令单元 572、寄存器 574、一个或多个浮点执行级 576 和一个或多个定点执行及 578。一级高速缓存为通过 MFC 504B 从共享存储器 106、处理器 102, 或存储器空间的其它部分接收的数据提供数据缓存功能。由于优选地将 PU 核心 504A 实现为超流水线, 从而优选地将指令单元 572 实现为具有多级 (包括获取、解码、依赖检查、发送等) 的指令流水线。此外, PU 核心 504A 优选地属于超标量配置, 从而, 每个时钟周期从指令单元 572 发出多于一个的指令。为实现高处理能力, 浮点执行级 576 和定点执行级 578 在流水线配置中包括多个级。根据所需处理能力, 可采用更多或更少数量的浮点执行级 576 和定点执行级 578。

MFC 504B 包括总线接口单元 (BIU) 580、二级高速缓冲存储器、非可高速缓存单元 (NCU) 584、核心接口单元 (CIU) 586 和存储器管理单元 (MMU) 588。多数 MFC 504B 以与 PU 核心 504A 和总线 108 相比为一半的频率 (半速) 运行, 以符合低功耗设计目标。

BIU 580 提供总线 108 与二级高速缓存 582 以及 NCU 584 逻辑块之间的接口。为此，BIU 580 在总线 108 上可起到主和从设备的作用，以执行完全一致的存储器操作。作为主设备，它可向总线 108 发出载入/存储请求，以代表二级高速缓存 582 和 NCU 584 进行服务。BIU 580 还可实现命令的流控制机制，用于限制可向总线 108 发送命令的总数量。可将总线 108 上的数据操作设计成占八拍 (beat)，因此，优选地将 BIU 580 设计成约 128 字节高速缓存线，一致性和同步粒度为 128KB。

优选地将二级高速缓存存储器 582 (和支持硬件逻辑) 设计成高速缓存 512 KB 数据。例如，二级高速缓存 582 可处理可高速缓存载入/存储，数据预取、指令获取、指令预取、高速缓存操作和障碍操作。二级高速缓存 582 优选地为 8 路组关联系统。二级高速缓存 582 可包括与六 (6) 个舍去 (castout) 队列 (例如，六个 RC 机) 相匹配的六个再载入队列，和八个 (64 字节宽) 存储队列。二级高速缓存 582 可用于提供一级高速缓存 570 中某些或全部数据的备份拷贝。这在处理节点进行热切换 (hotswap) 时恢复状态比较有用。该配置还允许一级高速缓存 570 以更少端口更快速地运行，并允许进行更快速的高速缓存到高速缓存的传送 (由于请求可停止在二级高速缓存 582)。该配置还提供将高速缓存一致性管理传递到二级高速缓存存储器 582 的机制。

NCU 584 与 CIU 586、二级高速缓存存储器 582 和 BIU 580 相接口，其通常起到用于 PU 核心 504A 和存储器系统之间的非可高速缓存操作的排队/缓冲电路的作用。NCU 584 优选地负责未由二级高速缓存 582 处理的与 PU 核心 504A 进行的所有通信，例如，禁止高速缓存的载入/存储、障碍操作和高速缓存一致性操作。NCU 584 优选地以半速运行，以符合上述功耗目标。

CIU 586 设置在 MFC 504B 和 PU 核心 504A 的边界，对来自执行级 576, 578、指令单元 572 和 MMU 单元 588 以及去往二级高速缓存 582 和 NCU 584 的请求起到路由、仲裁和流控制点的作用。PU 核

心 504A 和 MMU 588 优选地以全速运行,而二级高速缓存 582 和 NCU 584 以 2:1 速度比运作。从而,在 CIU 586 中存在频率边界,当它在两个频域之间转发请求以及重载入数据时,其一项功能是正确处理频率交叉。

CIU 586 由三个功能块组成:载入单元、存储单元和重载入单元。此外,数据预取功能通过 CIU 586 执行,其优选地是载入单元的功能部分。CIU 586 优选地用于:(i)接受来自 PU 核心 504A 和 MMU 588 的载入和存储请求;(ii)将请求从全速时钟频率转换成半速(2:1 时钟频率转换);(iii)将可高速缓存的请求路由到二级高速缓存 582,并将非可高速缓存的请求路由到 NCU 584;(iv)在对二级高速缓存 582 和 NCU 584 的请求之间进行公平仲裁;和(v)针对对二级高速缓存 582 和 NCU 584 的分配进行流控制,使得在目标窗口中接收请求并且避免溢出;(vi)接受载入返回数据,并将其路由到执行级 576, 578、指令单元 572,或 MMU 588;(vii)将窥探请求路由到执行级 576, 578、指令单元 572,或 MMU 588;以及(viii)将载入返回数据和窥探业务从半速转换成全速。

MMU 588 优选地提供 PU 核心 540A 的地址转换,例如,通过二级地址转换设备。通过可比 MMU 588 小得多和更快的各自指令和数据 ERAT(对实地址转换有效)阵列,优选地在 PU 核心 504A 中设置第一级转换。

在优选实施例中,PU 504 运行在 4-6GHz, 10F04,并具有 64 位实现。寄存器优选地为 64 位长(虽然一个或多个专用寄存器可以更小),有效地址为 64 位长。指令单元 570、寄存器 572 和执行级 574 和 576 优选地使用 PowerPC 技术实施,以实现(RISC)计算技术。

关于该计算机系统的模块化结构的更多细节可参看美国专利 No. 6,526,491,在此其全部内容引作参考。

根据本发明的至少一个其它方面,可使用合适的硬件,例如图中所示的硬件,来实现上述方法和装置。可使用任何已知技术(例如,标准数字电路)、任何可用于执行软件和/或固件程序的已知处理器、

一个或多个可编程数字设备或系统（例如，可编程只读存储器（PROM））、可编程阵列逻辑设备（PAL）等，来实现这样的硬件。此外，尽管将图中所示装置显示成分为特定的功能块，然而这样的功能块也可通过独立电路实现和/或将其组合成一个或多个功能单元，此外，本发明的各个方面可通过可存储在合适存储介质（例如，软盘、存储器芯片等）以便进行运输和/或发行的软件和/或固件程序来实现。

此处尽管参照具体实施例描述了本发明，然而应该理解，这些实施例仅仅是对本发明的原理和应用的示意性描述。因此，在不偏离如所附权利要求所定义的本发明精神和范围的条件，可对示意性实施例进行多种修改，并可设计出其它方案。

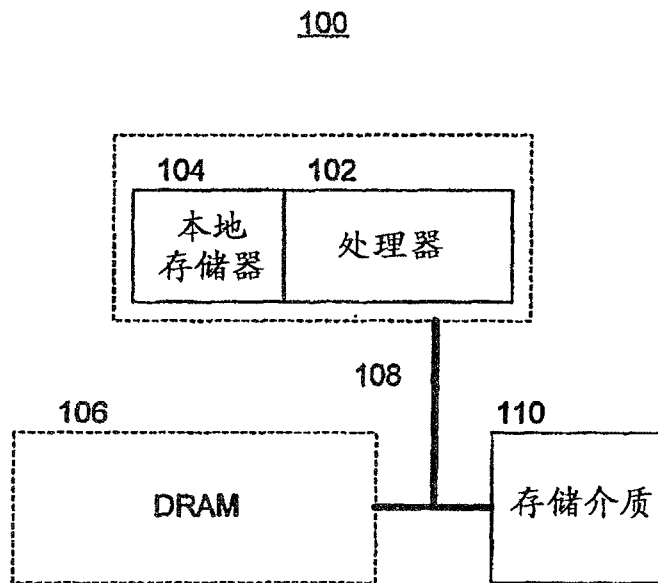


图1

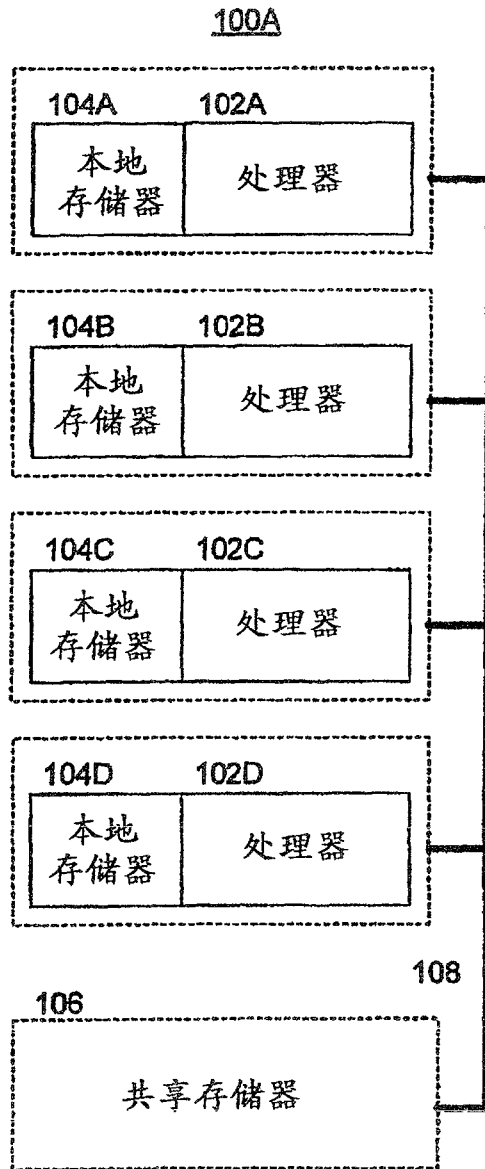


图 2

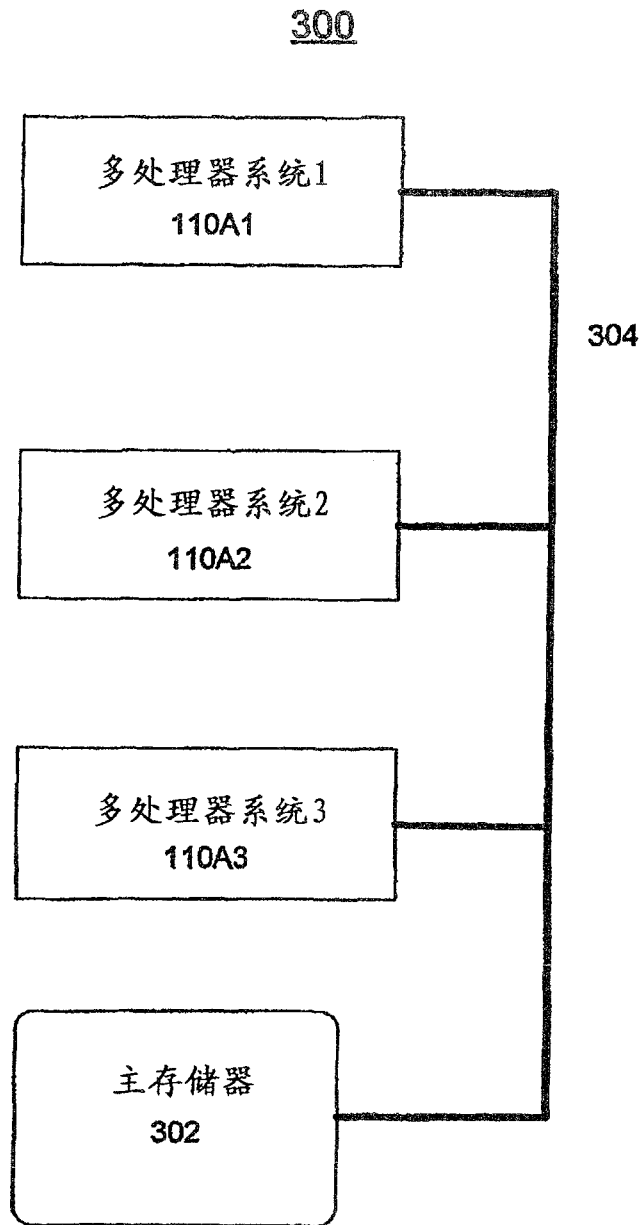


图 3

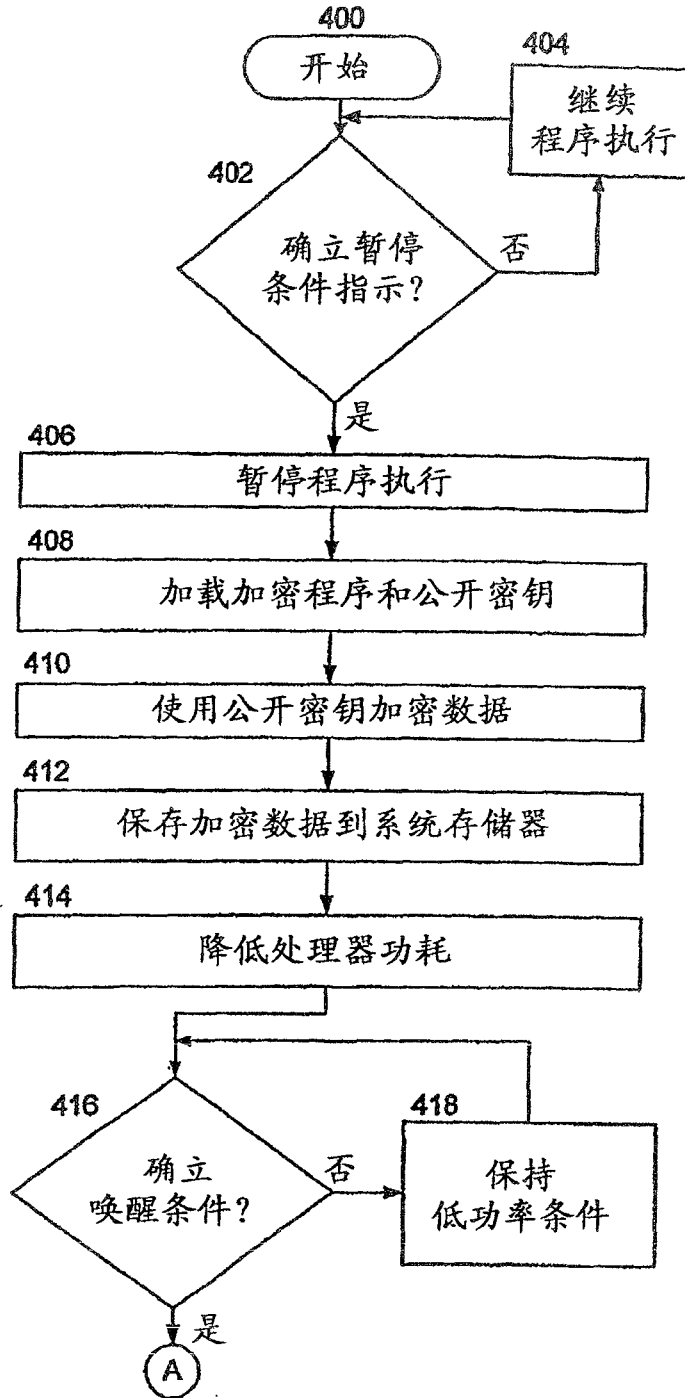


图 4A

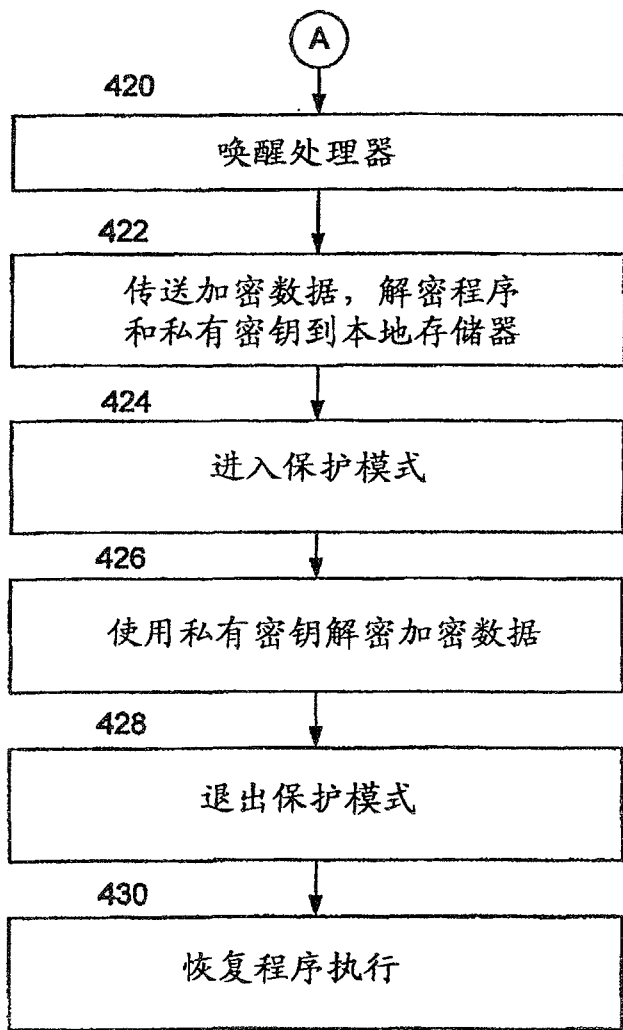


图 4B

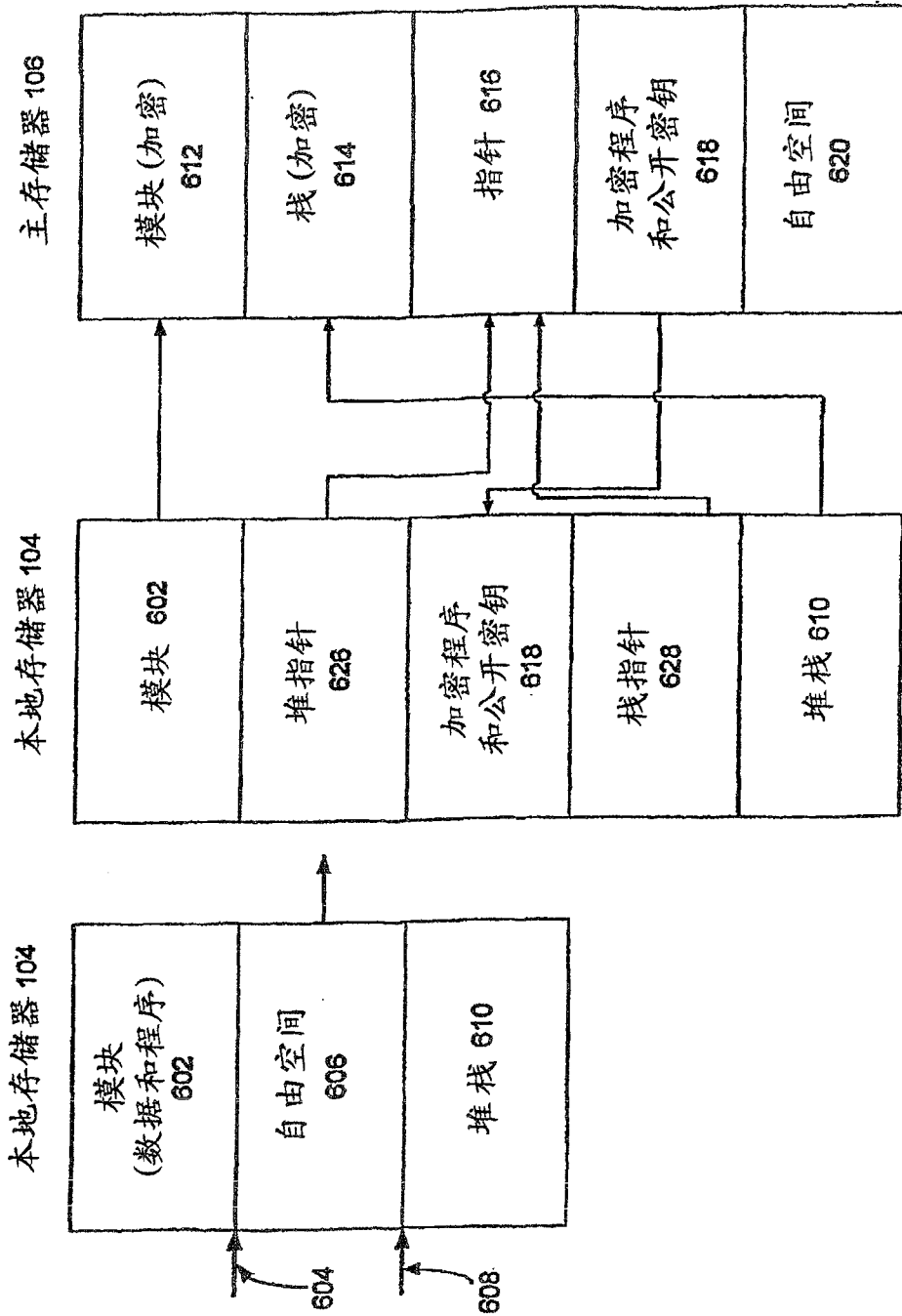


图5

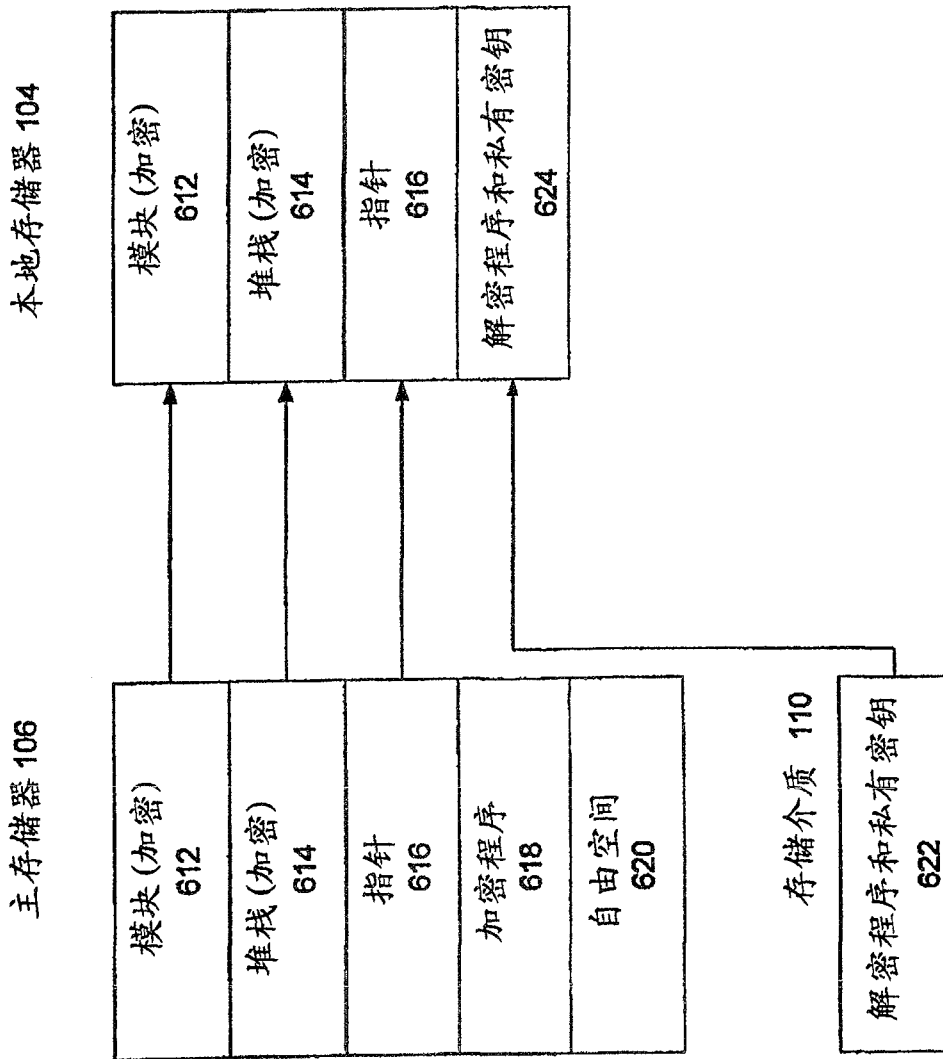


图6

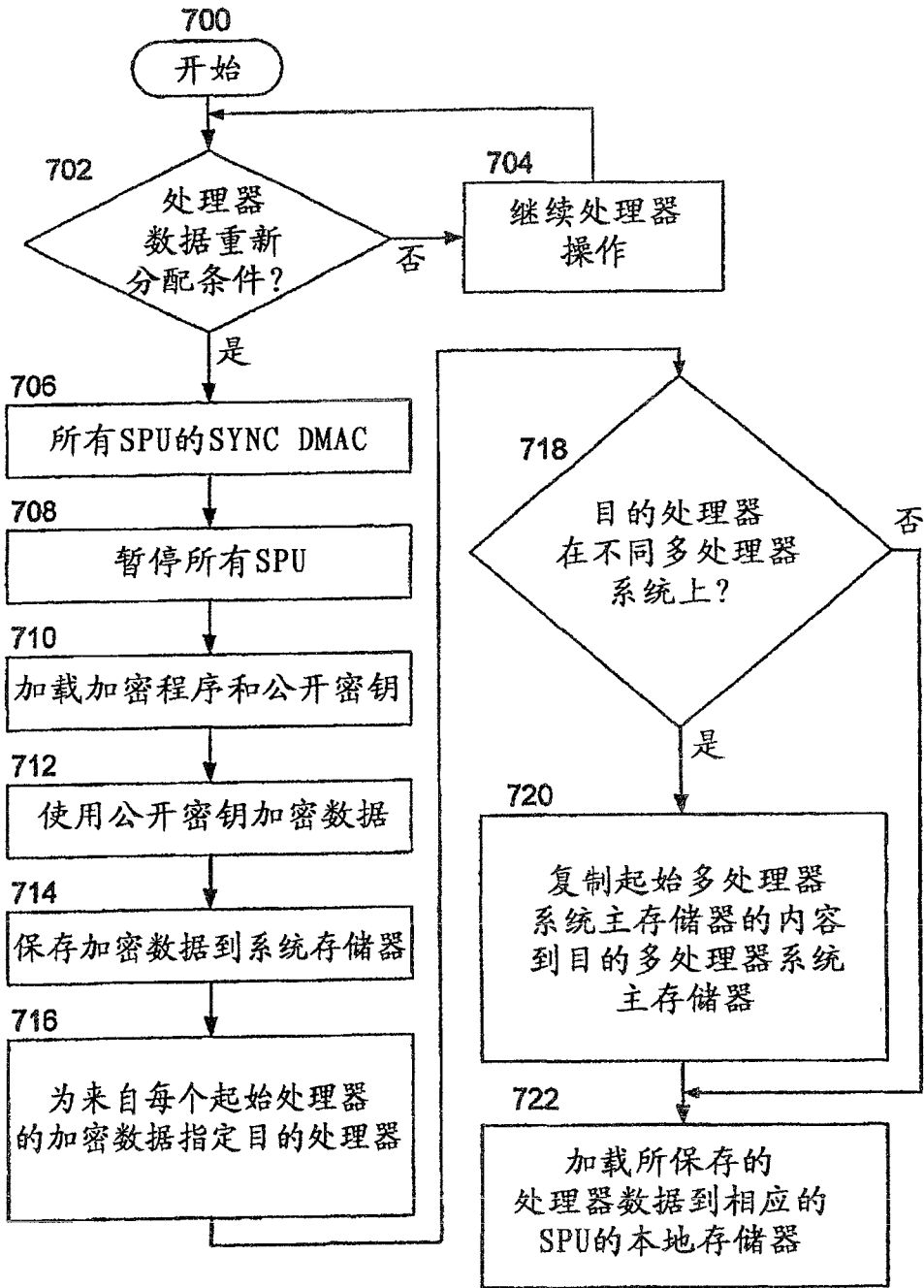


图7

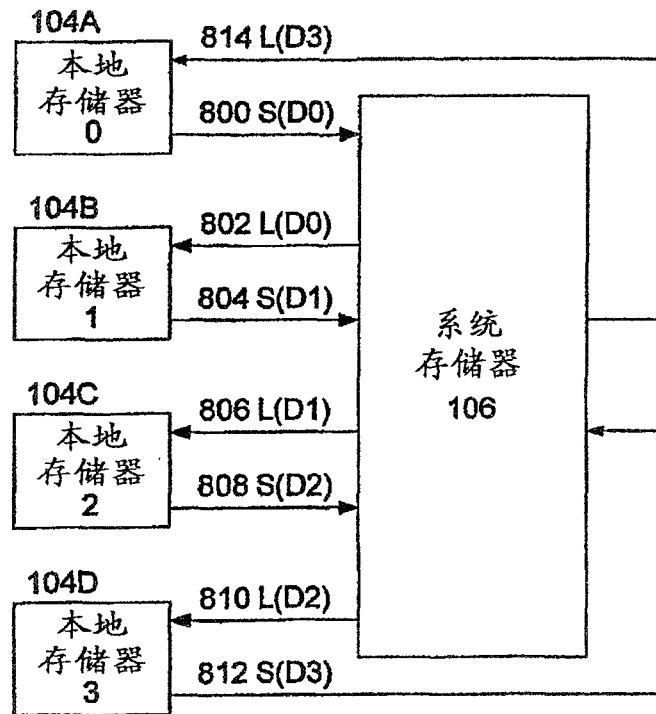


图 8

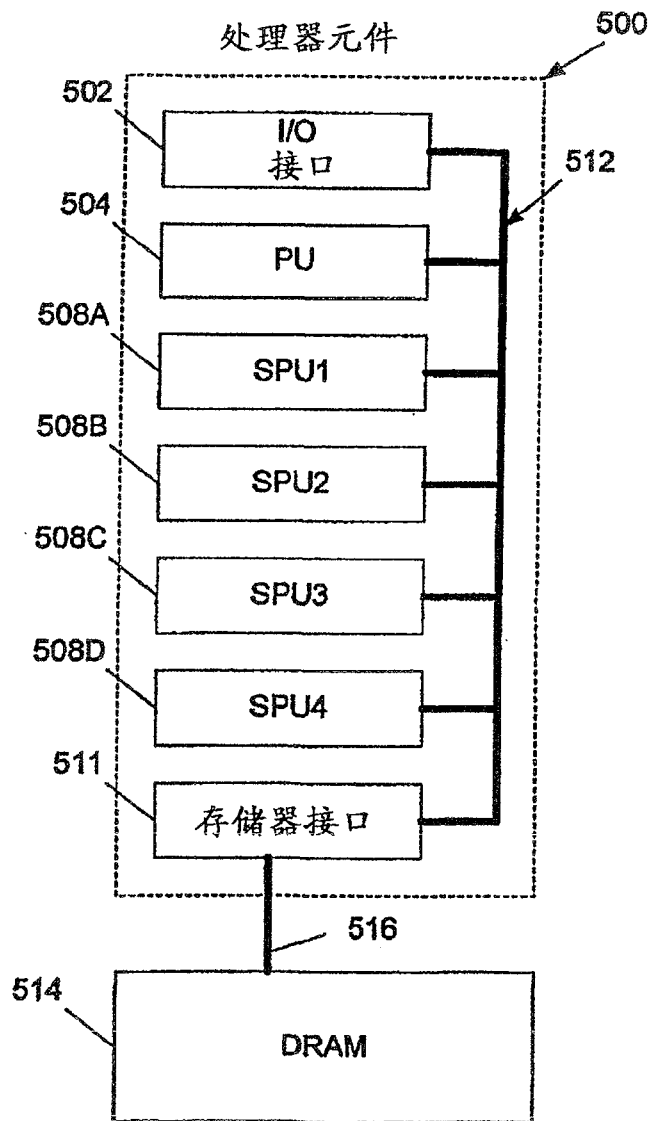


图9

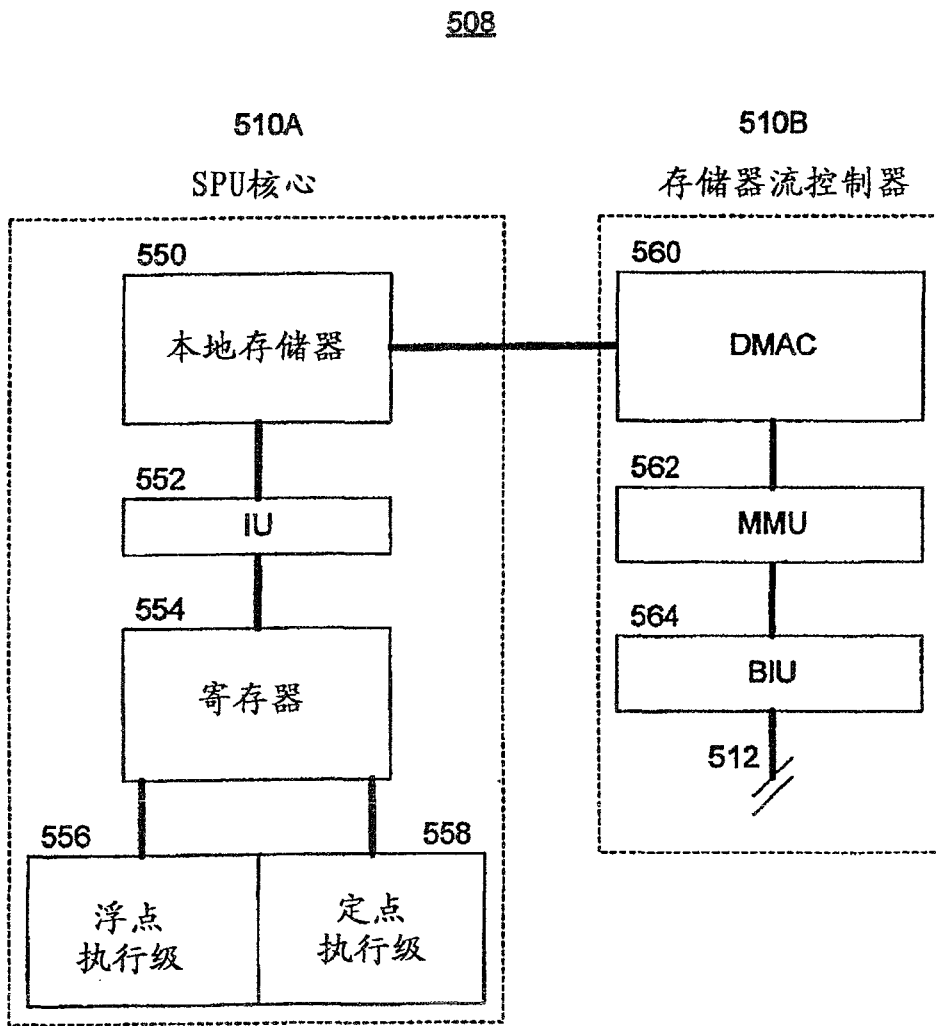


图10

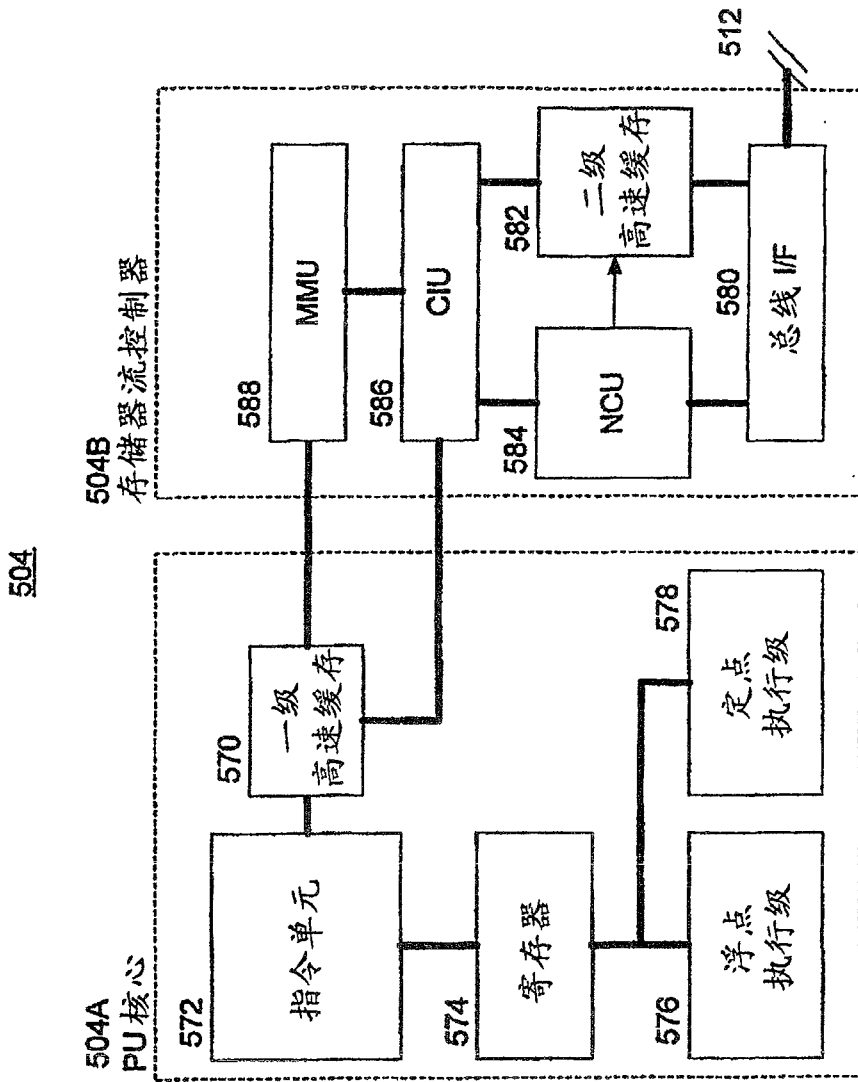


图11