



(12)发明专利

(10)授权公告号 CN 106104488 B

(45)授权公告日 2019.06.07

(21)申请号 201580014125.2

德

(22)申请日 2015.03.05

司瓦浦奈尔·普拉迪普库尔马·沙

(65)同一申请的已公布的文献号

克哈雪特

申请公布号 CN 106104488 A

徐飞

(43)申请公布日 2016.11.09

(74)专利代理机构 北京律盟知识产权代理有限公司 11287

(30)优先权数据

代理人 宋献涛

14/219,696 2014.03.19 US

(51)Int.Cl.

(85)PCT国际申请进入国家阶段日

G06F 9/52(2006.01)

2016.09.14

G06F 9/54(2006.01)

(86)PCT国际申请的申请数据

(56)对比文件

PCT/US2015/018987 2015.03.05

US 2011078249 A1,2011.03.31,

(87)PCT国际申请的公布数据

CN 103109274 A,2013.05.15,

W02015/142538 EN 2015.09.24

US 5742785 A,1998.04.21,

(73)专利权人 高通股份有限公司

US 8359603 B1,2013.01.22,

地址 美国加利福尼亚州

审查员 董雪

(72)发明人 阿列克谢·弗拉狄米罗维奇·布尔

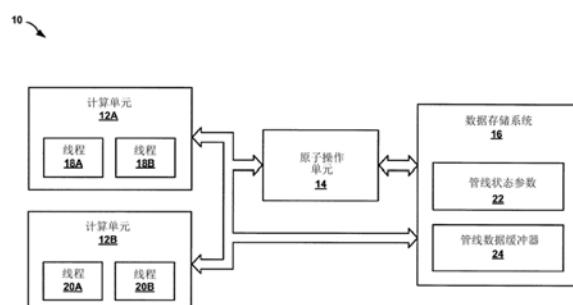
权利要求书5页 说明书42页 附图13页

(54)发明名称

用于支持任务间通信的基于硬件的原子操作

(57)摘要

本发明描述用于支持并行计算系统中的任务间通信的技术。所述用于支持任务间通信的技术可使用基于硬件的原子操作来维持管线的状态。管线可指代先入先出FIFO组织的缓冲器，其允许各种任务作为数据产生者或数据消费者而与缓冲器交互。各种管线实施方案可使用多个状态参数来界定管线的状态。本发明中所描述的所述基于硬件的原子操作可以原子方式修改多个管线状态参数。以原子方式修改多个管线状态参数可避免竞态条件，当多个产生者和/或多个消费者同时尝试修改管线的状态时原本将发生所述竞态条件。以此方式，可在并行计算系统中支持基于管线的任务间通信。



1.一种用于基于硬件的原子操作的方法,其包括:

以一或多个处理器执行基于硬件的原子读取-修改-写入操作,所述基于硬件的原子读取-修改-写入操作在单个存储器事务中修改指示管线的状态的两个或更多个管线状态参数,所述管线对应于用于支持任务间通信的数据缓冲器,其中所述两个或更多个管线状态参数包括以下参数中的至少两个参数:识别多少包当前存储在所述管线中的参数、识别存储所述管线的起始包的包条目的参数、识别多少未提交保留当前对于所述管线待决的参数、或者识别多少包条目当前保留在所述管线中的参数,其中以所述一或多个处理器执行所述基于硬件的原子读取-修改-写入操作包括执行非可中断操作集合,所述非可中断操作集合包括:

从存储器子系统读取对应于所述两个或更多个管线状态参数的值的操作,其中所述从存储器子系统读取对应于所述两个或更多个管线状态参数的值的操作包括在所述单个存储器事务中,在寄存器的四重寄存器群组内的三个连续寄存器中读取和返回三个管线状态参数的数据的操作,所述寄存器的四重寄存器群组包括四个寄存器;

产生所述两个或更多个管线状态参数的经修改值的操作;以及

将所述两个或更多个管线状态参数的所述经修改值写入到所述存储器子系统的操作。

2.根据权利要求1所述的方法,其中所述两个或更多个管线状态参数包括第一管线状态参数和第二管线状态参数,且其中产生所述两个或更多个管线状态参数的所述经修改值的所述操作包括:

基于第一修改功能产生所述第一管线状态参数的经修改值的操作;以及

基于第二修改功能产生所述第二管线状态参数的经修改值的操作,所述第二修改功能不同于所述第一修改功能。

3.根据权利要求1所述的方法,其中所述两个或更多个管线状态参数包括第一管线状态参数和第二管线状态参数,其中从所述存储器子系统读取对应于所述两个或更多个管线状态参数的所述值的所述操作包括读取对应于第一管线状态参数的第一值的操作,且其中产生所述两个或更多个管线状态参数的经修改值的所述操作包括:

基于对应于所述第一管线状态参数的所述第一值产生所述第二管线状态参数的经修改值的操作。

4.根据权利要求1所述的方法,其中产生所述两个或更多个管线状态参数的所述经修改值的所述操作包括:

基于所述两个或更多个管线状态参数中的一或者而确定所述管线是否具有足够的未提交和未保留条目来支持写入保留请求的操作;以及

响应于确定所述管线具有足够的未提交和未保留条目来支持所述写入保留请求而产生所述两个或更多个管线状态参数的所述经修改值以使得所述经修改值反映针对所述管线已做出写入保留的操作。

5.根据权利要求1所述的方法,其中产生所述两个或更多个管线状态参数的所述经修改值的所述操作包括:

基于所述两个或更多个管线状态参数中的一或者而确定所述管线是否具有其中存储包数据的足够的未读取和未保留包条目来支持读取保留请求的操作;以及

响应于确定所述管线具有其中存储包数据的足够的未读取和未保留包条目来支持所

述读取保留请求而产生所述两个或更多个管线状态参数的经修改值以使得所述经修改值反映针对所述管线已做出读取保留的操作。

6. 根据权利要求1所述的方法,其中产生所述两个或更多个管线状态参数的所述经修改值的所述操作包括:

产生所述两个或更多个管线状态参数值的第一管线状态参数的经修改值以使得所述第一管线状态参数的所述经修改值反映所述管线具有相对于在执行所述基于硬件的原子读取-修改-写入操作之前所述管线具有的未提交保留的数目来说少一个的未提交保留的操作;

基于所述两个或更多个管线状态参数中的一或者确定是否已经提交用于所述管线的全部保留的操作;以及

响应于确定已经提交用于所述管线的全部保留而产生所述两个或更多个管线状态参数的所述经修改值以使得所述经修改值反映所提交条目已经放置到所述管线中的操作。

7. 根据权利要求1所述的方法,其中产生所述两个或更多个管线状态参数的所述经修改值的所述操作包括:

产生所述两个或更多个管线状态参数值的第一管线状态参数的经修改值以使得所述第一管线状态参数的所述经修改值反映所述管线具有相对于在执行所述基于硬件的原子读取-修改-写入操作之前所述管线具有的未提交保留的数目来说少一个的未提交保留的操作;

基于所述两个或更多个管线状态参数中的一或者确定是否已经提交用于所述管线的全部保留的操作;以及

响应于确定已经提交用于所述管线的全部保留而产生所述两个或更多个管线状态参数的经修改值以使得所述经修改值反映所提交条目已经从所述管线移除的操作。

8. 根据权利要求1所述的方法,

其中所述方法进一步包括,以包含在所述一或多个处理器中的硬件单元接收来自第一线程的执行原子操作的第一请求,

其中所述基于硬件的原子读取-修改-写入操作是第一基于硬件的原子读取-修改-写入操作,

其中以所述一或多个处理器执行所述第一基于硬件的原子读取-修改-写入操作包括以所述硬件单元响应于接收到所述第一请求而执行所述第一基于硬件的原子读取-修改-写入操作以服务于所述第一请求,其中所述方法进一步包括:

在完成所述第一基于硬件的原子读取-修改-写入操作之前,以所述硬件单元接收来自第二线程的执行原子操作的第二请求;

响应于接收到所述第二请求,以所述硬件单元等待开始用于服务于所述第二请求的第二基于硬件的原子读取-修改-写入操作的执行直到所述第一基于硬件的原子读取-修改-写入操作的执行完成为止;以及

在所述第一基于硬件的原子读取-修改-写入操作的执行完成之后,以所述硬件单元执行修改两个或更多个管线状态参数的所述第二基于硬件的原子读取-修改-写入操作以服务于所述第二请求。

9. 根据权利要求1所述的方法,

其中所述一或多个处理器包括图形处理单元GPU,且

其中以所述一或多个处理器执行所述基于硬件的原子读取-修改-写入操作包括以所述GPU执行修改指示所述管线的所述状态的所述两个或更多个管线状态参数的所述基于硬件的原子读取-修改-写入操作。

10. 根据权利要求1所述的方法,其中所述数据缓冲器包括用于支持任务间通信的先入先出FIFO组织的数据缓冲器。

11. 根据权利要求1所述的方法,其中:所述两个或更多个管线状态参数包含四个管线状态参数,所述四个管线状态参数中的每一者都由位于128位块中的一个不同的32位值所表示,所述四个管线状态参数包括:所述识别多少包当前存储在所述管线中的参数、所述识别存储所述管线的起始包的包条目的参数、所述识别多少未提交保留当前对于所述管线待决的参数、以及所述识别多少包条目当前保留在所述管线中的参数。

12. 一种用于基于硬件的原子操作的装置,其包括:

先入先出FIFO组织的数据缓冲器;

一或多个处理器,其经配置以执行基于硬件的原子读取-修改-写入操作,所述基于硬件的原子读取-修改-写入操作在单个存储器事务中修改指示管线的状态的两个或更多个管线状态参数,所述管线对应于用于支持任务间通信的所述先入先出FIFO组织的数据缓冲器,其中所述两个或更多个管线状态参数包括以下参数中的至少两个参数:识别多少包当前存储在所述管线中的参数、识别存储所述管线的起始包的包条目的参数、识别多少未提交保留当前对于所述管线待决的参数、或者识别多少包条目当前保留在所述管线中的参数,其中所述一或多个处理器经配置以执行所述基于硬件的原子读取-修改-写入操作包括所述一或多个处理器经配置以执行非可中断操作集合,所述非可中断操作集合包括:

从存储器子系统读取对应于所述两个或更多个管线状态参数的值的操作,其中所述从存储器子系统读取对应于所述两个或更多个管线状态参数的值的操作包括在所述单个存储器事务中,在寄存器的四重寄存器群组内的三个连续寄存器中读取和返回三个管线状态参数的数据的操作,所述寄存器的四重寄存器群组包括四个寄存器;

产生所述两个或更多个管线状态参数的经修改值的操作;以及

将所述两个或更多个管线状态参数的所述经修改值写入到所述存储器子系统的操作。

13. 根据权利要求12所述的装置,其中所述两个或更多个管线状态参数包括第一管线状态参数和第二管线状态参数,且其中产生所述两个或更多个管线状态参数的经修改值的所述操作包括:

基于第一修改功能产生所述第一管线状态参数的经修改值的操作;以及

基于第二修改功能产生所述第二管线状态参数的经修改值的操作,所述第二修改功能不同于所述第一修改功能。

14. 根据权利要求12所述的装置,其中所述两个或更多个管线状态参数包括第一管线状态参数和第二管线状态参数,其中从所述存储器子系统读取对应于所述两个或更多个管线状态参数的所述值的所述操作包括读取对应于所述第一管线状态参数的第一值的操作,且其中产生所述两个或更多个管线状态参数的经修改值的所述操作包括:

基于对应于所述第一管线状态参数的所述第一值产生所述第二管线状态参数的经修改值的操作。

15. 根据权利要求12所述的装置，其中产生所述两个或更多个管线状态参数的所述经修改值的所述操作包括：

基于所述两个或更多个管线状态参数中的一或者而确定所述管线是否具有足够的未提交和未保留条目以支持写入保留请求的操作；以及

响应于确定所述管线具有足够的未提交和未保留条目来支持所述写入保留请求而产生所述两个或更多个管线状态参数的所述经修改值以使得所述经修改值反映针对所述管线已做出写入保留的操作。

16. 根据权利要求12所述的装置，其中产生所述两个或更多个管线状态参数的所述经修改值的所述操作包括：

基于所述两个或更多个管线状态参数中的一或者而确定所述管线是否具有其中存储包数据的足够的未读取和未保留包条目来支持读取保留请求的操作；以及

响应于确定所述管线具有其中存储包数据的足够的未读取和未保留包条目来支持所述读取保留请求而产生所述两个或更多个管线状态参数的经修改值以使得所述经修改值反映针对所述管线已做出读取保留的操作。

17. 根据权利要求12所述的装置，其中产生所述两个或更多个管线状态参数的所述经修改值的所述操作包括：

产生所述两个或更多个管线状态参数值的第一管线状态参数的经修改值以使得所述第一管线状态参数的所述经修改值反映所述管线具有相对于在执行所述基于硬件的原子读取-修改-写入操作之前所述管线具有的未提交保留的数目来说少一个的未提交保留的操作；

基于所述两个或更多个管线状态参数中的一或者确定是否已经提交用于所述管线的全部保留的操作；以及

响应于确定已经提交用于所述管线的全部保留而产生所述两个或更多个管线状态参数的所述经修改值以使得所述经修改值反映所提交条目已经放置到所述管线中的操作。

18. 根据权利要求12所述的装置，其中产生所述两个或更多个管线状态参数的所述经修改值的所述操作包括：

产生所述两个或更多个管线状态参数值的第一管线状态参数的经修改值以使得所述第一管线状态参数的所述经修改值反映所述管线具有相对于在执行所述基于硬件的原子读取-修改-写入操作之前所述管线具有的未提交保留的数目来说少一个的未提交保留的操作；

基于所述两个或更多个管线状态参数中的一或者确定是否已经提交用于所述管线的全部保留的操作；以及

响应于确定已经提交用于所述管线的全部保留而产生所述两个或更多个管线状态参数的经修改值以使得所述经修改值反映所提交条目已经从所述管线移除的操作。

19. 根据权利要求12所述的装置，

其中所述一或多个处理器包括图形处理单元GPU，且

其中所述一或多个处理器进一步经配置以执行修改指示所述管线的所述状态的所述两个或更多个管线状态参数的所述基于硬件的原子读取-修改-写入操作。

20. 根据权利要求12所述的装置，其中所述装置包括无线通信装置和移动电话手持机

中的至少一者。

21. 一种用于基于硬件的原子操作的设备,其包括:

用于在单个存储器事务中存储指示管线的状态的两个或更多个管线状态参数的装置,所述管线对应于用于支持任务间通信的先入先出FIFO组织的数据缓冲器,其中所述两个或更多个管线状态参数包括以下参数中的至少两个参数:识别多少包当前存储在所述管线中的参数、识别存储所述管线的起始包的包条目的参数、识别多少未提交保留当前对于所述管线待决的参数、或者识别多少包条目当前保留在所述管线中的参数;以及

用于以非可中断操作集合执行修改所述两个或更多个管线状态参数的基于硬件的原子读取-修改-写入操作的装置,所述非可中断操作集合包括:

从存储器子系统读取对应于所述两个或更多个管线状态参数的值的操作,其中所述从存储器子系统读取对应于所述两个或更多个管线状态参数的值的操作包括在所述单个存储器事务中,在寄存器的四重寄存器群组内的三个连续寄存器中读取和返回三个管线状态参数的数据的操作,所述寄存器的四重寄存器群组包括四个寄存器;

产生所述两个或更多个管线状态参数的经修改值的操作;以及

将所述两个或更多个管线状态参数的所述经修改值写入到所述存储器子系统的操作。

22. 一种非暂时性计算机可读存储媒体,具有存储于其上的指令,所述指令在执行时致使一或多个处理器:

执行基于硬件的原子读取-修改-写入操作,所述基于硬件的原子读取-修改-写入操作在单个存储器事务中修改指示管线的状态的两个或更多个管线状态参数,所述管线对应于用于支持任务间通信的先入先出FIFO组织的数据缓冲器,其中所述两个或更多个管线状态参数包括以下参数中的至少两个参数:识别多少包当前存储在所述管线中的参数、识别存储所述管线的起始包的包条目的参数、识别多少未提交保留当前对于所述管线待决的参数、或者识别多少包条目当前保留在所述管线中的参数,其中所述基于硬件的原子读取-修改-写入操作包括非可中断操作集合,所述非可中断操作集合包括:

从存储器子系统读取对应于所述两个或更多个管线状态参数的值的操作,其中所述从存储器子系统读取对应于所述两个或更多个管线状态参数的值的操作包括在所述单个存储器事务中,在寄存器的四重寄存器群组内的三个连续寄存器中读取和返回三个管线状态参数的数据的操作,所述寄存器的四重寄存器群组包括四个寄存器;

产生所述两个或更多个管线状态参数的经修改值的操作;以及

将所述两个或更多个管线状态参数的所述经修改值写入到所述存储器子系统的操作。

用于支持任务间通信的基于硬件的原子操作

技术领域

[0001] 本发明涉及计算系统，且更具体地说涉及并行计算系统。

背景技术

[0002] 并行编程模型可支持任务并行度和数据并行度中的一者或两者以便解决计算问题。任务并行度可允许将计算问题划分为多个任务。所述任务可循序地、同时地和/或并行地在一或多个处理器核心上执行。数据并行度可允许通过将数据分布到不同处理元件且致使处理元件中的每一者对其经指派数据集合执行相同操作集合而对不同数据集合并行地执行相同操作集合。

[0003] 多核处理器可用以支持任务并行度，其中每一核心经配置以执行特定任务。在某些情况下，多核处理器中的核心中的一或更多者可为单指令多数据 (SIMD) 处理器或单程序多数据 (SPMD) 处理器，其可包含多个处理元件以支持数据并行度。在此些情况下，支持数据层级并行度的任务可能能够在多核处理器上循序或并行地执行。

[0004] 若干不同类型的处理器可支持任务并行度和/或数据并行度，包含多核中央处理单元 (CPU)、图形处理单元 (GPU)、数字信号处理器 (DSP)、Cell宽带引擎 (Cell/B.E.) 处理器等。虽然 GPU 传统地经设计以支持三维 (3D) 图形到显示器的再现，但许多现代的 GPU 中包含的可编程着色器架构可用以有效支持在使用并行编程模型编程的通用非图形特定程序中发现的任务并行度和数据并行度两者。使用 GPU 的并行架构来执行非图形特定程序可被称为图形处理单元上的通用计算 (GPGPU)。

发明内容

[0005] 本发明描述用于支持并行计算系统中的任务间通信的技术。所述用于支持任务间通信的技术可使用基于硬件的原子操作来维持管线的状态。管线可指代允许各种任务作为数据产生者或数据消费者而与缓冲器交互的缓冲器。各种管线实施方案可使用多个状态参数来界定管线的状态，且这些状态参数可能需要在每次数据产生者或消费者与管线交互时进行修改。本发明中所描述的所述基于硬件的原子操作可以原子方式修改多个管线状态参数。以原子方式修改多个管线状态参数可避免竞态条件，当多个产生者和/或多个消费者同时尝试修改管线的状态时原本将发生所述竞态条件。以此方式，可在并行计算系统中支持基于管线的任务间通信。

[0006] 在一个实例中，本发明描述一种方法，其包含：以一或多个处理器执行修改指示管线的状态的两个或更多个管线状态参数的基于硬件的原子读取-修改-写入操作。所述管线对应于用于支持任务间通信的数据缓冲器。

[0007] 在另一实例中，本发明描述一种装置，包含：一或多个处理器，其经配置以执行修改指示管线的状态的两个或更多个管线状态参数的基于硬件的原子读取-修改-写入操作。所述管线对应于用于支持任务间通信的数据缓冲器。

[0008] 在另一实例中，本发明描述一种设备，其包含用于存储指示管线的状态的两个或

更多个管线状态参数的装置。所述管线对应于用于支持任务间通信的数据缓冲器。所述设备进一步包含用于执行修改所述两个或更多个管线状态参数的基于硬件的原子读取-修改-写入操作的装置。

[0009] 在另一实例中，本发明描述一种存储指令的非暂时性计算机可读存储媒体，所述指令在执行时致使一或多个处理器执行修改指示管线的状态的两个或更多个管线状态参数的基于硬件的原子读取-修改-写入操作。所述管线对应于用于支持任务间通信的数据缓冲器。

[0010] 在附图及以下描述中阐述本发明的一或多个实例的细节。本发明的其它特征、目标和优点将从所述描述和图式以及权利要求书显而易见。

附图说明

[0011] 图1是说明实施本发明的基于硬件的原子操作的实例计算系统的框图。

[0012] 图2是说明根据本发明的界定实例原子保留写入命令的接口和功能性的实例伪码的概念图。

[0013] 图3是说明根据本发明的界定实例原子提交写入命令的接口和功能性的实例伪码的概念图。

[0014] 图4是说明根据本发明的界定实例原子保留读取命令的接口和功能性的实例伪码的概念图。

[0015] 图5是说明根据本发明的界定实例原子提交读取命令的接口和功能性的实例伪码的概念图。

[0016] 图6是图1的实例并行计算系统的框图，其中更详细说明图1的原子操作单元的实例。

[0017] 图7是说明根据本发明的与数据存储系统交互的实例原子操作数据路径的概念图。

[0018] 图8是说明可用以实施本发明的基于硬件的原子操作的实例计算装置的框图。

[0019] 图9是说明根据本发明的可在图形处理单元(GPU)或其它计算装置中使用的实例计算单元的框图。

[0020] 图10是说明图1的计算系统的一部分的框图，其中更详细说明实例GPU和存储器配置。

[0021] 图11是说明用于执行根据本发明的基于硬件的原子操作的实例技术的流程图。

[0022] 图12是说明用于执行根据本发明的基于硬件的原子操作的另一实例技术的流程图。

[0023] 图13是说明用于执行根据本发明的原子保留写入操作和原子保留读取操作的实例技术的流程图。

[0024] 图14是说明用于执行根据本发明的原子提交写入操作和原子提交读取操作的实例技术的流程图。

具体实施方式

[0025] 本发明描述用于支持并行计算系统中的任务间通信的技术。用于支持任务间通信

的技术可使用基于硬件的原子操作来维持管线的状态。管线可指代先入先出 (FIFO) 组织的缓冲器,其允许各种任务作为数据产生者或数据消费者而与缓冲器交互。各种管线实施方案可使用多个状态参数来界定管线的状态,且这些状态参数可能需要在每次数据产生者或消费者与管线交互时进行修改。本发明中所描述的基于硬件的原子操作可以原子方式修改多个管线状态参数。以原子方式修改多个管线状态参数可避免竞态条件,当多个产生者和/或多个消费者同时尝试修改管线的状态时原本将发生所述竞态条件。以此方式,可在并行计算系统中支持基于管线的任务间通信。

[0026] 计算问题可划分为在一或多个处理器核心上循序地、同时地和/或并行地执行的多个任务。任务可指代由并行计算系统执行的程序。线程可指代由并行计算系统中的单个处理元件执行的任务的实例。任务可替代地在本文中被称作内核,且线程可替代地在本文中被称作工作项目。在某些情况下,由并行计算系统执行的不同任务可能需要彼此通信。例如,一个任务可基于由另一任务产生的数据而产生数据。

[0027] 管线可用以允许不同任务彼此通信。从编程者的观点,管线可提供FIFO组织的缓冲器,其允许各种任务作为数据产生者或数据消费者而与缓冲器交互。FIFO组织的缓冲器可指代其中写入接口和读取接口根据FIFO原理操作的缓冲器。根据FIFO原理操作的写入接口可指代附加数据以写入到已经存储在缓冲器中的任何数据的末尾的接口。根据FIFO原理操作的读取接口可指代从缓冲器选择且移除最早的未读数据以写入到缓冲器的接口。

[0028] 因为管线可实施FIFO组织的缓冲器,所以管线也可以称为具有根据FIFO原理操作的写入和读取接口。从编程者的观点,对于具有根据FIFO原理操作的写入接口的管线,将数据写入到管线的任务可能不一定指定写入数据的管线中的特定位置。实际上,管线可附加数据以写入到已经存储在管线中的任何数据的末尾。类似地,对于具有根据FIFO原理操作的读取接口的管线,从管线读取数据的任务可能不一定指定读取数据的管线中的特定位置。实际上,管线可选择且返回存储在管线中的作为最早未读数据的数据以写入到管线。通过提供根据FIFO原理操作的读取和写入接口,管线可提供抽象数据结构,借此产生者和消费者任务可有效地彼此通信而不需要编程者关注用以存储数据的存储器的内部组织。

[0029] 本发明中所描述的管线可在一些实例中使用数据包作为用于对管线写入和读取数据的基本数据单元。不是空的管线可含有一或多个包的序列。管线的起始包可指代对应于存储在管线中的包的序列中的第一包的包。管线的结束包可指代对应于存储在管线中的包的序列中的最后包的包。空的管线可不含有任何包。

[0030] 管线可包含用于存储数据包的多个包条目,其中所述包条目中的每一者能够存储单个数据包。非空的管线的包的序列可存储在用于管线的包条目的邻接集合中。在某些情况下,包条目可存储于存储器空间中,且包条目的邻接集合可从存储器空间的末尾绕回到存储器空间的开始。非空的管线的起始包条目可指代存储管线的起始包的包条目。非空的管线的结束包条目可指代存储管线的结束包的包条目。第一包偏移参数可为指示非空的管线的起始包条目的值。对于空的管线,第一包偏移参数可为指示当对管线写入数据时数据将写入到的第一包条目的值。

[0031] 各种管线实施方案可使用多个状态参数来界定管线的状态,且这些状态参数可能需要在每次数据产生者或消费者与管线交互时进行修改。由于并行计算系统中的并行度,管线可能需要支持多个数据产生者和/或多个数据消费者,其各自可潜在地同时尝试与管

线交互且修改管线状态参数。允许多个数据产生者和/或消费者潜在地同时修改多个管线状态参数可导致竞态条件,这可能阻止管线恰当地运作。

[0032] 根据本发明的一些方面,描述用于执行基于硬件的原子操作以维持管线的状态的技术。例如,包含在处理器中的硬件单元可经配置以执行基于硬件的原子读取-修改-写入操作,其修改指示管线的状态的两个或更多个管线状态参数。以原子方式使用基于硬件的原子读取-修改-写入操作来修改两个或更多个管线状态参数可避免竞态条件,当多个数据产生者和/或消费者同时尝试修改管线的状态时原本将发生所述竞态条件。以此方式,针对使用多个状态参数来界定管线的状态的管线可实现恰当功能性。

[0033] 使用多个状态参数来界定管线的状态的管线的一个实例是实施为环形缓冲器(即,循环缓冲器)的管线。实施为环形缓冲器的管线可具有经组织为环形缓冲器的管线条目。环形缓冲器可指代使用单个固定大小缓冲器如同所述缓冲器是端到端连接的数据结构。换句话说,位于所述固定大小缓冲器的相对末端处的包条目可概念地认为彼此邻近以用于实施环形缓冲器。

[0034] 因为环形缓冲器不具有固定起始包条目和固定结束包条目,所以通常需要多个状态参数来完全界定实施为环形缓冲器的管线的状态。例如,第一包偏移参数可指向管线的起始包条目,且包计数参数可指示包含于管线中的包的数目。作为另一实例,第一包偏移参数可指向管线的起始包条目,且最后包偏移参数可指向管线的最后包条目。每次添加到实施为环形缓冲器的管线或从所述管线移除数据时,状态参数中的每一者可能需要更新。因此,当管线被实施为环形缓冲器时多个管线状态参数可能需要更新。

[0035] 使用多个状态参数来界定管线的状态的管线的另一实例是支持保留的管线。管线保留可允许产生者线程保留且对管线中的包条目的邻接集合进行写入,且允许消费者线程保留且从管线中的包条目的邻接集合进行读取。管线保留可用以减轻当多个产生者同时尝试对管线写入一个以上包和/或多个消费者同时尝试从管线读取一个以上包时可产生的交错问题。

[0036] 一般来说,多个产生者和/或多个消费者尝试对管线写入一个以上包和/或从管线读取一个以上包的问题可能在其中管线支持作为产生者或消费者绑定到管线的多个任务的情境中产生。所述问题还可能在其中单个任务作为产生者或消费者绑定到管线且任务的执行分布到并行执行的多个线程(即,任务的实例)的情境中产生。这可例如在支持数据并行度的并行计算系统(例如,包含单指令多数据(SIMD)执行单元和/或单程序多数据(SPMD)执行单元的并行计算系统)中发生。

[0037] 为了使用支持保留的管线,产生者可在用于写入的管线中保留包条目的邻接集合,将包数据写入到保留的包条目,且将保留的包条目提交到管线。类似地,消费者可在用于读取的管线中保留包条目的邻接集合,从保留包条目读取包数据,且将保留的包条目提交到管线。

[0038] 因为多个产生者可能同时保留用于管线的包条目和/或同时对保留包条目进行写入,所以在管线的给定时间多个写入保留可能未解决(即,未提交)。在此些情况下,管线在更新管线状态参数以反映数据包已经写入到管线的保留包条目之前可等待直到全部未解决的写入保留已经提交为止。

[0039] 类似地,因为多个消费者可能同时保留用于管线的包条目和/或同时从保留包条

目进行读取,所以在管线的给定时间多个读取保留可能未解决(即,未提交)。在此些情况下,管线可在更新管线状态参数以反映包含在保留包条目中的数据包已经从管线读取且移除之前等待直到全部未解决的读取保留已经提交为止。

[0040] 为了维持支持保留的管线的状态,可能需要一或多个管线状态参数来界定管线的保留状态。这些管线状态参数可被称为保留状态参数。例如,第一保留状态参数可指示管线存在多少未解决的保留,且第二保留状态参数可指示多少包条目保留在管线中。每次在管线中保留和/或提交包条目,所述保留状态参数可能需要更新。另外,当全部未解决的保留提交到管线时,保留状态参数和其它管线状态参数可能需要更新。因此,当从支持保留的管线保留包条目和/或将包条目提交到所述管线时可能需要更新多个管线状态参数。

[0041] 如上文所论述,当管线经组织为环形缓冲器时和/或当管线支持保留时,可能需要多个管线状态参数来界定管线的状态。由于并行计算系统中的并行度,管线可能需要支持多个数据产生者和/或多个数据消费者,其各自可潜在地同时尝试与管线交互且修改管线状态参数。允许多个数据产生者和/或消费者潜在地同时修改多个管线状态参数可导致竞态条件,这可能阻止管线恰当地运作。

[0042] 根据本发明的一些方面,描述用于执行基于硬件的原子操作以维持管线的状态的技术。例如,包含在处理器中的硬件单元可经配置以执行基于硬件的原子读取-修改-写入操作,其修改指示管线的状态的两个或更多个管线状态参数。以原子方式使用基于硬件的原子读取-修改-写入操作来修改两个或更多个管线状态参数可避免竞态条件,当多个数据产生者和/或消费者同时尝试修改管线的状态时原本将发生所述竞态条件。以此方式,对于使用多个状态参数的管线,例如实施为环形缓冲器的管线和/或支持保留的管线,可实现恰当功能性。

[0043] 用于当更新多个管线状态参数时避免竞态条件的另一解决方案涉及将线程修改管线的状态的尝试置于用于所述线程的经编译代码的关键区段中,且使用互斥锁来强制执行关键代码区段。互斥锁可确保没有两个线程同时在其关键区段中执行,进而确保两个线程无法同时修改管线状态参数。可通过使用例如比较-交换原子操作来实施互斥锁。然而,当使用互斥锁时,许多线程可能竞争所述锁,进而导致做出多个尝试来成功获取所述锁。线程获取锁的重复尝试会降低性能和/或功率功效。

[0044] 然而,本发明中所描述的基于硬件的原子操作可确保两个线程无法同时修改管线的管线状态参数,且可实现此行为而不需要使用互斥锁。例如,当实施本发明的基于硬件的原子操作的硬件单元从多个线程接收多个请求以修改管线的多个管线状态参数时,硬件单元可循序处理所述请求以使得与来自第一线程的第一请求相关联的全部管线状态参数修改是在执行与来自第二线程的第二请求相关联的管线状态参数修改中的任一者之前完成。使用此硬件单元可允许在并行计算系统中实现恰当管线功能性,其中多个同时执行的线程同时尝试修改多个管线状态参数。以此方式,可在并行计算系统中提供基于管线的任务间通信而无与使用基于软件的互斥技术用于修改管线状态参数相关联的性能缺陷和/或电力消耗缺陷。

[0045] 在一些实例中,本发明中所描述的基于硬件的原子操作可包含基于硬件的原子读取-修改-写入操作,其修改指示管线的状态的两个或更多个管线状态参数。读取-修改-写入操作在所述操作从开始到结束执行而无来自可由与请求读取-修改-写入操作的任务或

线程同时执行的其它任务和/或线程请求的任何其它读取-修改-写入操作的中断的意义上可为原子的。换句话说，在原子读取-修改-写入操作的执行期间，没有其它线程和/或任务可能够读取或修改由读取-修改-写入操作修改的管线状态参数。以此方式，在其中两个或更多个任务和/或线程同时尝试修改管线的管线状态参数的情况下可避免竞态条件。

[0046] 原子读取-修改-写入操作在所述原子读取-修改-写入操作可作为单个不可分操作由硬件单元实施的意义上可为基于硬件的，所述单个不可分操作可由以通信方式耦合到所述硬件单元的线程和/或任务调用。所述单个不可分操作在所述硬件单元从起始到结束执行操作而不中断所述操作以执行可由与请求读取-修改-写入操作的任务或线程同时执行的其它任务和/或线程请求的其它读取-修改-写入操作的意义上可为不可分的。例如，如果硬件单元接收到执行原子读取-修改-写入操作的多个请求，那么所述硬件单元可针对所述请求中的每一者循序执行原子读取-修改-写入操作以使得每次执行一个原子读取-修改-写入操作，且使得每一原子读取-修改-写入操作的执行是在开始执行另一原子读取-修改-写入操作之前完成。通过使用基于硬件的原子读取-修改-写入操作，可通过同时执行任务和/或线程而不需要任务和/或线程使用互斥锁来修改多个管线状态参数。

[0047] 在一些实例中，为了执行修改两个或更多个管线状态参数的基于硬件的原子读取-修改-写入操作，硬件单元可经配置以执行非可中断操作集合。例如，所述非可中断操作集合可包含从存储器子系统读取对应于管线的管线状态参数的值的操作、产生管线状态参数的经修改值的操作，以及将管线状态参数的经修改值写入到存储器子系统的操作。在操作是从起始到结束执行而不中断执行以执行可与其它读取-修改-写入操作相关联的其它操作（例如，与请求当前执行的读取-修改-写入操作的任务或线程同时执行的任务和/或线程所请求的其它读取-修改写入操作）的意义上可为非可中断的。致使硬件单元以此方式执行非可中断读取-修改-写入序列允许所述硬件单元针对可请求此些操作的同时执行的线程和/或任务实施原子读取-修改-写入操作。

[0048] 根据本发明的一些方面，根据本发明的实施基于硬件的原子读取-修改-写入操作的硬件单元可作为单个原子读取-修改-写入操作的部分而针对不同管线状态参数执行不同修改操作。例如，硬件单元可执行基于第一修改功能产生第一管线状态参数的经修改值的第一修改操作，以及基于第二修改功能产生第二管线状态参数的经修改值的第二修改操作。所述第二修改功能可不同于所述第一修改功能。作为单个原子读取-修改-写入操作的部分使用不同修改功能修改不同管线状态参数可允许在单个原子操作中以不同方式更新不同管线状态参数。

[0049] 例如，为了支持原子管线保留操作，第一修改功能可用以将保留包计数参数递增针对管线保留所请求的包的数目，且第二修改功能可用以将保留计数递增恒定值一。作为另一实例，为了支持原子管线提交操作，第一修改功能可用以将保留计数递减恒定值一，第二修改功能可用以将包计数参数递增或递减保留计数参数，且第三修改功能可用以将保留包计数参数复位到零。其它实例是可能的且在本发明的范围内。

[0050] 根据本发明的额外方面，根据本发明的实施基于硬件的原子读取-修改-写入操作的硬件单元可基于在原子操作的执行期间从存储器子系统获得的一或多个额外管线状态参数的值而修改一或多个管线状态参数的值。例如，硬件单元可在单个读取-修改-写入操作期间读取分别对应于管线的第一和第二管线状态参数的第一和第二值，且产生第一和第

二管线状态参数的经修改值以使得第二管线状态参数的经修改值是基于对应于第一管线状态参数的第一值而产生。在单个原子读取-修改-写入操作期间基于不同管线状态参数值修改管线状态参数值可允许基于管线的当前状态在单个原子操作中更新管线状态参数。

[0051] 例如,为了支持原子提交操作,取决于所述提交操作是否为读取提交操作或写入提交操作可将包计数参数递减或递增保留包计数参数。作为另一实例,为了支持原子提交读取操作,可基于保留包计数参数调整第一包偏移参数。其它实例是可能的且在本发明的范围内。

[0052] 根据本发明的额外方面,基于不同管线状态参数值修改管线状态参数值可涉及基于不同管线状态参数值有条件地更新管线状态参数值。有条件地更新管线状态参数值可涉及基于不同管线状态参数值确定是否更新或不更新管线状态参数值。基于其它管线状态参数值有条件地更新管线状态参数值可允许基于管线的当前状态而更新或不更新管线状态参数。

[0053] 根据本发明的额外方面,根据本发明的实施基于硬件的原子读取-修改-写入操作的硬件单元可将管线状态参数值返回到请求线程和/或任务。在某些情况下,返回的管线状态参数值可反映在由基于硬件的原子读取-修改-写入操作执行的修改之前的管线的状态。将值返回到请求线程和/或任务可允许所述线程或任务确定哪些包条目已经保留用于保留操作以及在存储用于管线的包条目的存储空间中保留包条目位于何处。

[0054] 在一些实例中,本发明的基于硬件的原子读取-修改-写入操作可包含原子保留写入操作。原子保留写入操作可基于管线状态参数中的一或者者而确定管线是否具有足够的未提交和未保留的条目来支持写入保留请求。响应于确定管线具有足够的未提交和未保留条目来支持写入保留请求,原子保留写入操作可产生管线状态参数的经修改值以使得所述经修改值反映针对所述管线已进行写入保留。

[0055] 在其它实例中,本发明的基于硬件的原子读取-修改-写入操作可包含原子保留读取操作。所述原子保留读取操作可基于管线状态参数中的一或者者而确定管线是否具有足够的未读取和未保留包条目,其中存储包数据以支持读取保留请求。响应于确定管线具有足够的其中存储包数据以支持读取保留请求的未读取和未保留包条目,原子保留读取操作可产生管线状态参数的经修改值以使得所述经修改值反映针对所述管线已进行读取保留。

[0056] 在额外实例中,本发明的基于硬件的原子读取-修改-写入操作可包含原子提交写入操作。原子提交写入操作可产生第一管线状态参数的经修改值以使得所述经修改值反映所述管线相对于在执行原子提交写入操作之前所述管线具有的未提交保留的数目来说少一个的未提交保留。原子提交写入操作可进一步基于管线状态参数中的一或者者确定是否管线的全部保留已经提交。响应于确定管线的全部保留已经提交,原子提交写入操作可产生管线状态参数的一或多个经修改值以使得所述经修改值反映提交条目已经置于管线中。

[0057] 在其它实例中,本发明的基于硬件的原子读取-修改-写入操作可包含原子提交读取操作。原子提交读取操作可产生第一管线状态参数的经修改值以使得所述经修改值反映所述管线相对于在执行原子提交读取操作之前所述管线具有的未提交保留的数目来说少一个的未提交保留。原子提交读取操作可进一步基于管线状态参数中的一或者者确定是否管线的全部保留已经提交。响应于确定管线的全部保留已经提交,原子提交读取操作可产生管线状态参数的一或多个经修改值以使得所述经修改值反映提交条目已经从管线移除。

[0058] 根据本发明的额外方面,用以存储在单个原子读取-修改-写入操作期间读取和写入的多个管线状态参数的位数目可小于或等于包含在在单个存储器事务期间从数据存储单元读取或写入到数据存储单元的数据块中的位数目。例如,用以存储多个管线状态参数的位数目可小于或等于存储器字中的位数目和/或高速缓冲存储器线中的位数目。在某些情况下,存储于存储器子系统中的管线中的每一者的管线状态参数可经对准以使得管线的全部管线状态参数存储在存储器子系统的单个存储器字和/或单个高速缓冲存储器线中。以此方式,原子读取-修改-写入操作可能够修改多个管线状态参数而不需要多于一个存储器读取事务或多于一个存储器写入事务。

[0059] 在一些实例中,界定管线的状态的管线状态参数可包含指示多少包当前存储在管线中的包计数参数,以及指示存储管线的起始包的包条目的第一包偏移参数。这些参数可用以界定实施为环形缓冲器的管线的状态。

[0060] 在其它实例中,界定管线的状态的管线状态参数可包含指示多少未提交保留当前对于管线待决的保留计数参数,以及指示多少包条目当前保留在管线中的保留包计数参数。这些参数可被称为保留状态参数且可用以界定支持保留的管线的状态。

[0061] 在额外实例中,界定管线的状态的管线状态参数可包含包计数参数、第一包偏移参数、保留计数参数以及保留包参数。管线状态参数的此组合可用以界定支持保留且实施为环形缓冲器的管线的状态。

[0062] 虽然本发明的基于硬件的原子读取-修改-写入操作主要相对于管线状态参数的修改而描述,但应理解相同或类似类型的操作可应用于不同类型的数据。一般来说,可例如在其中需要作为单个操作的部分以原子方式修改两个或更多个值的应用中使用本发明的基于硬件的原子读取-修改-写入操作。以原子方式使用基于硬件的原子读取-修改-写入操作修改两个或更多个值可允许同时执行的任务和/或线程修改共享对象而不需要使用锁和/或关键代码区段。

[0063] 图1是说明实施本发明的基于硬件的原子操作的实例计算系统10的框图。计算系统10可经配置以执行一或多个任务。任务可指代由计算系统10执行的计算机程序。在一些实例中,计算系统10可为并行计算系统。例如,计算系统10可在不同计算单元上并行地执行两个或更多个任务。作为另一实例,对于一或多个任务,计算系统10可并行地执行相应任务的多个实例。线程可指代由计算系统10中的单个处理元件执行的任务的实例。

[0064] 计算系统10包含计算单元12A-12B(共同地为“计算单元12”)、原子操作单元14以及数据存储系统16。如图1中所示,计算单元12以通信方式耦合到原子操作单元14,且原子操作单元14以通信方式耦合到数据存储系统16。计算单元12可经由原子操作单元14间接地以通信方式耦合到数据存储系统16。虽然图1中未具体图示,但在一些实例中,计算单元12除间接耦合到数据存储系统16之外还可以通信方式直接耦合到数据存储系统16。

[0065] 计算单元12经配置以执行一或多个任务。在某些情况下,计算单元12可并行地执行任务。计算单元12中的每一者可执行相同任务或不同任务。由计算单元12执行的任务可利用数据并行度。因此,任务的执行可分布到多个线程(即,任务的实例),其中所述线程中的每一者相对于不同数据执行同一程序/任务。所述多个线程可在线程群组中执行,其中对于每一线程群组,线程是并行地执行的。因此,计算单元12中的每一者可在给定时间点执行一个任务,但可同时执行所述任务的多个实例(即,线程)。所述线程可替代地称为工作项

目,且所述线程群组可替代地称为工作群组。

[0066] 如图1所示,计算单元12经配置以执行一或多个线程。例如,计算单元12A经配置以执行线程18A、18B,且计算单元12B经配置以执行线程20A、20B。

[0067] 计算单元12中的每一者可包含一或多个处理元件。计算单元12中的处理元件中的每一者可执行单个线程。在其中计算单元12中的一或多者包含多个处理元件的实例中,单个一个或计算单元12中的处理元件可共同地经配置以并行地执行线程群组(例如,工作群组),且处理元件中的每一者可执行线程群组(即,工作群组)中的线程中的相应一个。在一些实例中,计算单元12中的每一者还可包含局部存储器、指令存储装置、恒定存储器等中的一或多者。

[0068] 在一些实例中,计算单元12中的单个一者中的处理单元可共同地实施单指令多数据(SIMD)处理器和/或单程序多数据(SPMD)处理器。两个类型的处理器可在多个处理元件上同时执行任务的多个实例。在SIMD处理器中,全部有源处理元件可在给定指令循环期间相对于不同数据执行相同指令,且处理元件中的每一者可共享共同程序计数器。在SPMD处理器中,全部处理元件可相对于不同数据执行相同程序,且处理元件中的每一者可具有其自身的程序计数器。对于SPMD处理器,处理元件可同时执行同一程序的不同指令。

[0069] 计算单元12中的每一者可为可编程计算单元。计算单元在所述计算单元可包含经配置以执行程序或任务的硬件的意义上可为可编程的。相比之下,非可编程或固定功能硬件可指代未经配置以执行程序的硬件。虽然固定功能硬件可为可配置的,但硬件的可配置性是基于由固定功能硬件接收的一或多个控制信号,而不是基于由硬件执行的用户定义的程序。

[0070] 在一些实例中,计算单元12中的一者或两者可根据并行编程应用程序编程接口(API)、异构计算平台API和/或协处理器计算平台API而操作,例如开放式计算语言(OpenCL™)API。在其它实例中,计算单元12中的一者或两者可对应于根据OpenCL™规范界定的OpenCL™计算单元。关于OpenCL™API的进一步细节可参见“OpenCL规范”版本2.0文献修正19(克洛诺斯OpenCL工作组,2013年11月14日);“OpenCL C规范”版本2.0文献修正19(克洛诺斯OpenCL工作组,2013年11月14日);以及“OpenCL扩展规范”版本2.0文献修正19(克洛诺斯OpenCL工作组,2013年11月14日)。

[0071] 在一些实例中,计算单元12中的每一者可包含在一或多个处理单元中。处理单元可包含例如中央处理单元(CPU)、多核CPU、图形处理单元(GPU)、数字信号处理器(DSP)、Cell宽带引擎(单元/B.E.)处理器、并行处理器、协处理器等。在一些实例中,处理单元可为异构计算平台、并行处理计算平台和/或协处理器计算平台中的计算装置。在其它实例中,处理单元可对应于根据OpenCL™规范界定的装置和/或计算装置。

[0072] 在一些实例中,计算单元12中的每一者可包含在同一处理单元和/或计算装置中。例如,计算单元12中的每一者可包含在单个图形处理单元(GPU)中。在其它实例中,计算单元12中的每一者可包含在不同处理单元和/或计算装置中。例如,计算单元12A可包含在GPU中,且计算单元12B可包含在多核CPU中。

[0073] 在其中计算单元12中的一者或两者包含在GPU中的实例中,计算单元12中的每一者可在一些实例中对应于GPU的相应着色器单元。GPU的着色器单元可经配置以执行着色器程序,例如图形特定着色器程序和/或通用着色器程序(例如,计算着色器程序)。

[0074] 在一些实例中,由计算单元12执行的程序或任务连同主机程序可形成计算应用程序。主机程序可在与包含计算单元12的处理单元中的一或者相同的处理单元上执行或在不同处理单元上执行。执行主机程序的处理单元可被称为主机装置。用于执行计算应用程序的实例配置可包含CPU和GPU,其中CPU执行主机程序且GPU包含执行任务和/或线程的计算单元12。

[0075] 在其中由计算单元12执行的任务是包含主机程序的计算应用程序的部分的实例中,用于由计算单元12执行哦任务的程序代码可由主机程序提供。在某些情况下,主机处理器可将用于任务的程序代码加载到包含在计算单元12中的芯片上存储器中和/或加载到包含计算单元12的一或多个处理单元的芯片上存储器中以使得计算单元12可更高效地执行任务。在一些实例中,主机程序可将对应于任务的线程映射到工作群组和工作群组内的处理元件,且在包含计算单元12中的一者或两者的计算装置上调用任务的执行。包含在计算单元12中的一者或两者作为其部分的处理单元(例如,计算装置)中的控制单元可将特定工作群组指派到计算单元12中的特定计算单元用于执行。

[0076] 出于示范性目的将图1的实例计算系统10说明为包含两个计算单元12,其各自执行两个线程。实施本发明的技术的其它实例计算系统可具有相同或不同数目的计算单元,且可执行相同数目或不同数目的线程。由计算单元中的每一者同时执行的线程的数目可相同或不同。在一些实例中,实施本发明的技术的计算系统可具有单个计算单元。在其它实例中,实施本发明的技术的计算系统可具有执行单个线程的一或多个计算单元。

[0077] 原子操作单元14可经配置以执行本发明中所描述的基于硬件的原子操作。在一些实例中,原子操作单元14可执行维持管线的状态的一或多个基于硬件的原子读取-修改-写入操作。例如,原子操作单元14可执行修改指示管线的状态的两个或更多个管线状态参数的基于硬件的原子读取-修改-写入操作,其中所述管线对应于用于支持任务间通信的FIFO组织的数据缓冲器。在一些实例中,所述基于硬件的原子读取-修改-写入操作可包含读取保留操作、写入保留操作、读取提交操作和写入提交操作。

[0078] 原子操作单元14可从计算单元12中的一或者接收执行各种原子操作的命令,且处理所述命令。在一些实例中,原子操作单元14可循序地处理所述命令以确保原子操作的原子性。换句话说,原子操作单元14可从起始到结束处理单个命令,然后处理后续命令。在一些实例中,所述命令可包含原子保留读取命令、原子保留写入命令、原子提交读取命令和原子提交写入命令。

[0079] 原子操作单元14可当执行原子读取-修改-写入操作时与数据存储系统16交互。例如,原子操作单元14可向数据存储系统16发出读取请求和写入请求以便执行原子读取-修改-写入操作的读取和写入分量。

[0080] 在一些实例中,原子操作单元14可响应于处理原子命令而返回一或多个管线状态参数。例如,作为保留操作的部分,原子操作单元14可将一或多个管线状态参数返回到请求任务。在其它实例中,原子操作单元14可不响应于执行原子命令而返回任何数据。

[0081] 原子操作单元14可为在一或多个处理器上实施的硬件单元。所述硬件单元可包含用于实施根据本发明归于原子操作单元14的功能性的电路。例如,在一些实例中,所述硬件单元可包含数字电路、模拟电路、算术逻辑单元(ALU)、数据路径等中的一或者。

[0082] 在一些实例中,原子操作单元14可为固定功能(即,非可编程)硬件单元。换句话

说,在此些实例中,原子操作单元14不一定经配置以接收且执行用户指定的程序。

[0083] 原子操作单元14可与计算单元12中的一者或两者在相同处理单元上或在不同处理单元上实施。在一些实例中,原子操作单元14可包含在还包含计算单元12中的一者或两者的图形处理单元(GPU)中。在其它实例中,原子操作单元14可包含在包含于异构计算平台、并行处理计算平台和/或协处理器计算平台中的装置和/或计算装置(例如,根据OpenCL™规范界定的装置和/或计算装置)中。

[0084] 在一些实例中,原子操作单元14可包含在包含数据存储系统16的全部或部分的存储器子系统中。例如,原子操作单元14可为处理单元的高速缓冲存储器单元(例如,GPU的高速缓冲存储器单元)的部分。在额外实例中,原子操作单元14可集成到与包含于异构计算平台、并行处理计算平台和/或协处理器计算平台中的装置和/或计算装置(例如,根据OpenCL™规范界定的装置和/或计算装置)相关联的存储器子系统的部分中。

[0085] 数据存储系统16可经配置以存储指示将用于任务间通信的管线的状态和内容的数据。数据存储系统16包含管线状态参数22和管线数据缓冲器24。数据存储系统16可包含一或多个数据存储单元,例如易失性存储器、非易失性存储器、高速缓冲存储器、随机存取存储器(RAM)、静态RAM(SRAM)、动态RAM(DRAM)等。

[0086] 在一些实例中,数据存储系统16可包含单个存储单元,例如存储器存储单元。存储器存储单元可为位于包含计算单元12的处理单元上的芯片上存储器或在包含计算单元12的处理单元外部的芯片外存储器。

[0087] 在其它实例中,数据存储系统16可包含多个存储单元。例如,数据存储系统16可包含高速缓冲存储器存储单元和底层存储器存储单元。底层存储器存储单元可存储管线状态参数22和/或管线数据缓冲器24的源复制,且高速缓冲存储器存储单元可存储管线状态参数22和/或管线数据缓冲器24的高速缓冲存储复制。所述高速缓冲存储复制可为源复制的不完整版本。

[0088] 在其中数据存储系统16包含高速缓冲存储器存储单元和底层存储器存储单元的实例中,高速缓冲存储器存储单元可在一些实例中是芯片上高速缓冲存储器且底层存储器存储单元可为芯片外存储器存储单元。换句话说,高速缓冲存储器存储单元可位于包含计算单元12的处理单元上,或且存储器存储单元可在包含计算单元12的处理单元外部。

[0089] 管线状态参数22指示管线的状态。管线状态参数22可包含以下各项中的一或更多者:指示多少包当前存储在管线中的包计数参数,指示存储管线的起始包的包条目的第一包偏移参数,指示多少未提交保留当前对于管线待决的保留计数参数,以及指示多少包条目当前保留在管线中的保留包计数参数。

[0090] 包计数参数和第一包偏移参数可例如用以界定实施为环形缓冲器的管线的状态。保留计数参数和保留包计数参数可例如用以界定支持保留的管线的保留状态。

[0091] 上述管线状态参数22仅是可与本发明的技术一起使用的管线状态参数的集合的实例。在其它实例中,可使用相同数目或不同数目的管线状态参数,且使用的管线状态参数的类型可相同或不同。

[0092] 管线数据缓冲器24可包含经配置以存储对应于管线的数据内容的数据包的多个包条目。对于非空的管线,管线数据缓冲器24可包含存储在包条目的邻接集合中的包序列。在某些情况下,包条目的邻接集合可从管线数据缓冲器24的末尾绕回到管线数据缓冲器24

的开始。对于空的管线，管线数据缓冲器24中的包条目中的每一者可为空的。

[0093] 在一些实例中，管线数据缓冲器24可经组织为环形缓冲器。环形缓冲器可指代使用单个固定大小缓冲器如同所述缓冲器是端到端连接的数据结构。换句话说，位于管线数据缓冲器24的相对末端处的包条目可概念地认为彼此邻近以用于实施环形缓冲器。

[0094] 管线状态参数22和管线数据缓冲器24可用以实施FIFO组织的缓冲器。FIFO组织的缓冲器可指代其中写入接口和读取接口根据FIFO原理操作的缓冲器。根据FIFO原理操作的写入接口可指代附加数据以写入到已经存储在缓冲器中的任何数据的末尾的接口。根据FIFO原理操作的读取接口可指代从缓冲器选择且移除最早的未读数据以写入到缓冲器的接口。

[0095] 在一些实例中，计算单元12、原子操作单元14和数据存储系统16中的一或多者的全部或部分可为同一处理器单元的部分。例如，计算单元12、原子操作单元14和对应于数据存储系统16的部分的高速缓冲存储器单元可全部包含在GPU中。在此实例中，底层存储器存储单元可包含在不与GPU位于同一芯片上的芯片外存储器中。

[0096] 在其中原子操作单元14是计算装置的部分的实例中，如果计算单元12A和计算单元12B是不同计算装置的部分，那么每一计算装置可包含原子操作单元14。在此些实例中，并非如图1中所示计算单元12A和计算单元12B两者以通信方式耦合到单个原子操作单元14，计算单元12A和计算单元12B可以通信方式耦合到其相应计算装置中的其相应原子操作单元。

[0097] 现将描述计算系统10的操作。在下文描述的实例操作中，管线（例如，管线状态参数22和管线数据缓冲器24）已经在数据存储系统16中分配，且任务的执行已在计算单元12中的每一者上调用，以使得线程18A、18B在计算单元12A上同时执行且线程20A、20B在计算单元12B上同时执行。

[0098] 线程18、20中的一或者将原子操作命令发出到原子操作单元14。原子操作单元14执行原子操作命令。例如，原子操作单元14可执行对应于原子操作命令的原子操作。

[0099] 由原子操作单元14执行的原子操作可为原子读取-修改-写入操作，其以原子方式读取、修改和写入两个或更多个管线状态参数。为了执行原子读取-修改-写入操作，原子操作单元14可执行非可中断操作集合。所述非可中断操作集合可包含从数据存储系统16读取对应于管线状态参数22的值的操作、产生管线状态参数22的经修改值的操作，以及将管线状态参数22的经修改值写入到数据存储系统16的操作。

[0100] 为了执行读取操作，原子操作单元14可将读取请求发出到数据存储系统16。数据存储系统16可将管线状态参数22提供到原子操作单元14。在一些实例中，可响应于单个读取请求而提供管线的全部管线状态参数22。在额外实例中，可作为单个读取循环的部分而提供管线的全部管线状态参数22。在其它实例中，全部管线状态参数22可形成在单个存储器事务期间就绪的单个数据字。为了执行修改操作，原子操作单元14可使用一或多个数据路径。所述数据路径可包含例如一或多个ALU。为了执行写入操作，原子操作单元14可将写入请求发出到数据存储系统16，且将管线状态参数22的经修改值提供到数据存储系统16。

[0101] 原子操作单元14可基于正执行的原子操作的类型而修改所接收的管线状态参数值。原子操作的实例类型包含原子保留读取操作、原子保留写入操作、原子提交写入操作和原子提交读取操作。

[0102] 在一些实例中,在执行读取操作之后,原子操作单元14可执行返回操作,其将从数据存储系统16读取的管线状态参数值中的一或者返回到发出所述命令的请求线程。返回的值可对应于紧接在执行原子读取-修改-写入操作之前的管线的状态。

[0103] 原子操作单元14可基于从计算单元12接收的原子操作命令的类型而执行不同类型的原子读取-修改写入操作。例如,原子操作单元14可响应于从计算单元12接收到原子保留写入命令而执行原子保留写入操作,响应于从计算单元12接收到原子保留读取命令而执行原子保留读取操作,响应于从计算单元12接收到原子提交写入命令而执行原子提交写入操作,和/或响应于从计算单元12接收到原子提交读取命令而执行原子提交读取操作。

[0104] 原子命令可替代地在本文中被称作请求,且发出命令的线程可替代地在本文中被称作请求程序。例如,当线程18A将保留写入命令发出到原子操作单元14时,线程18A可称为发出写入保留请求的请求程序。

[0105] 现将描述本发明中所描述的原子操作和原子命令的实例使用情况。应用程序编程接口 (API) 可为允许在多个任务和/或线程可同时执行的计算环境中的任务间通信的管线提供支持。为了避免管线中的交错,API可允许管线中的包条目的邻接集合由线程保留以用于读取或写入。此管线可称为支持保留。

[0106] 为了支持保留,API可提供保留管线命令、读取和写入管线命令,以及提交管线命令。这些命令可由编程者和/或其它源代码产生器放置到源代码中以允许任务和/或线程从管线进行写入和读取。

[0107] 从编程者的角度,用以将数据写入到管线的操作的一般序列可为如下:保留用于写入的包条目,将数据写入到保留包条目,以及将保留包条目提交到管线。为了执行上文所提到的用于将数据写入到管线的操作序列,可将写入命令序列放置到源代码中。所述写入命令序列可包含保留写入管线命令、写入管线命令以及提交写入管线命令。在一些实例中,上文所提到的写入命令序列中的命令可分别对应于reserve_write_pipe命令、write_pipe命令和commit_write_pipe,如根据OpenCL™规范界定。

[0108] 保留写入管线命令可在管线中保留所请求数目的包条目用于写入,且返回指示保留的数据(例如,保留句柄)。写入管线命令可将数据(例如,包)写入到由保留句柄指示的保留包条目。提交写入管线命令可将保留包条目提交到管线。将保留包条目提交到管线可指示产生者任务已结束将数据写入到管线的保留包条目,且这些包条目准备好可用于消费者任务。

[0109] 从编程者的角度,用以从管线读取数据的操作的一般序列可为如下:保留用于读取的包条目,从保留包条目读取数据,以及将保留包条目提交到管线。为了执行上文所提到的用于从管线读取数据的操作序列,可将读取命令序列放置到源代码中。所述读取命令序列可包含保留读取管线命令、读取管线命令以及提交读取管线命令。在一些实例中,上文所提到的读取命令序列中的命令可分别对应于reserve_read_pipe命令、read_pipe命令和commit_read_pipe,如根据OpenCL™规范界定。

[0110] 保留读取管线命令可保留所请求数目的包条目用于读取,且返回指示保留的数据(例如,保留句柄)。读取管线命令可从由保留句柄指示的保留包条目读取数据(例如,包)。提交读取管线命令可将保留包条目提交到管线。将保留包条目提交到管线可指示消费者任务已结束从管线的保留包条目读取数据,且可从管线清除和/或移除此些包条目。

[0111] 上文所论述的读取和写入命令序列(包含保留管线和提交管线命令)可为源代码层级命令。编译器可编译所述源代码层级命令以产生包含经编译代码层级命令的经编译代码。经编译代码层级命令可包含本发明中所描述的原子保留命令和原子提交命令。

[0112] 现将针对线程18、20中的一者使用管线保留来将数据写入到管线且使用管线保留来从管线读取数据的情况描述计算系统10的实例操作。在此实例中,线程18、20中的每一者可执行包含本发明的原子保留和原子提交命令的经编译代码。

[0113] 为了此阐释的目的,将线程18A描述为将数据写入到管线且将线程20A描述为从管线读取数据。然而,在其它实例中,线程18、20中的任一者可以类似方式将数据写入到管线且从管线读取数据。

[0114] 为了将数据写入到管线,线程18A可将原子保留写入命令发出到原子操作单元14。原子保留写入命令可包含指示数据将写入到的管线的参数以及指示将保留用于写入的包条目的所请求数目的参数中的一者或两者。指示数据将写入到的管线的参数在一些实例中可为管线地址参数。例如,管线地址参数可为对应于存储器子系统(例如,数据存储系统16)中存储用于正存取的特定管线的管线状态参数22的位置的存储器地址。管线地址参数可用以识别当数据存储系统16含有用于多于一个管线的数据时在数据存储系统16中将修改哪一管线,且识别数据存储系统16中存储管线数据的位置。

[0115] 响应于接收到原子保留写入命令,原子操作单元14可执行原子保留写入操作。原子保留写入操作可为如本发明中所描述修改两个或更多个管线状态参数的原子读取-修改-写入操作的实例。

[0116] 为了执行原子保留写入操作,原子操作单元14可从数据存储系统16获得管线状态参数,且将从数据存储系统16获得的管线状态参数中的一或多者返回到线程18A。在一些实例中,由原子操作单元14返回的管线状态参数可包含第一包偏移参数、包计数参数以及保留包计数参数。

[0117] 原子操作单元14可进一步基于管线状态参数中的一或者而确定管线是否具有足够的未提交和未保留的条目来支持写入保留请求。响应于确定管线具有足够的未提交和未保留条目来支持写入保留请求,原子操作单元14可产生管线状态参数的经修改值以使得所述经修改值反映针对所述管线已进行写入保留。响应于确定管线不具有足够的未提交和未保留的条目来支持写入保留请求,原子操作单元14可产生管线状态参数的经修改值以使得经修改值等于管线状态参数的先前值(即,使得经修改值反映针对管线尚未进行写入保留)。

[0118] 在确定管线状态参数的经修改值之后,原子操作单元14可将管线状态参数22的经修改值写入到数据存储系统16。归于原子保留写入命令的功能性在一些实例中可作为非可中断操作集合的部分而执行。换句话说,原子操作单元14可从起始到结束执行归于原子保留写入命令的功能性而无需中断执行以执行可与其它原子操作命令相关联的其它操作(例如,由与发出原子保留写入命令的任务或线程同时执行的任务和/或线程发出的其它原子操作命令)。以此方式执行非可中断操作集合允许原子操作单元14实施原子保留写入命令,其甚至在同时执行的线程和/或任务可能同时尝试保留管线用于写入的情况下也确保管线的恰当功能性。下文相对于图2描述关于原子保留写入命令的实例实施方案的进一步细节。

[0119] 线程18A可基于返回的管线状态参数确定写入保留是否成功。为了确定写入保留

是否成功,线程18A可在一些实例中以类似于原子操作单元14可如何做出同一确定的方式,基于返回管线状态参数确定管线是否具有足够的未提交和未保留的条目来支持写入保留请求,如下文进一步详细所论述。返回的管线状态参数可指示在保留写入操作的执行之前的管线的状态。

[0120] 如果写入保留不成功,那么线程18A可响应于写入保留操作的完成而不将数据写入到管线或不将数据提交到管线。在一些实例中,如果写入保留不成功,那么线程18A可以错误条件终止执行。在其它实例中,如果写入保留不成功,那么线程18A可执行一或多个操作以处置所述条件。例如,线程18A可停止尝试执行写入保留且继续执行其它操作。作为另一实例,线程18A可尝试在某一时间周期之后重复写入保留操作。这可允许消费者线程时间以从管线读取一或多个包,进而释放额外包条目,所述条目可在后续写入保留操作中保留。

[0121] 如果写入保留成功,那么线程18A可确定管线数据缓冲器24的物理包条目,其中将基于当处理原子写入保留命令时由原子操作单元14返回的一或多个管线状态参数而写入数据。在其中线程18A接收到第一包偏移参数、包计数参数和保留包计数参数的实例中,线程18A可确定管线数据缓冲器24的第一物理包条目,其中将例如通过将这些参数中的每一者的值相加在一起写入数据。线程18A可将数据写入到所确定的第一物理包条目和随后的包条目直到保留的包条目的数目。如果在写入保留数目的包条目之前到达管线数据缓冲器24的末尾,那么线程18A可绕回到管线数据缓冲器24的开始以写入到额外物理包条目。

[0122] 为了将数据写入到管线数据缓冲器24中的保留包条目,线程18A在一些实例中可基于与待写入管线相关联的管线地址和基于所确定的将写入数据的物理包条目和/或基于返回的管线状态参数而确定包条目中的一或者者的物理存储器地址。在一些实例中,管线地址可由在数据存储系统16中分配管线的主机程序提供到线程18A。在此些实例中,管线地址在一些实例中可存储在计算单元12A的芯片上恒定存储器中。例如,在主机处理器上执行的主机程序在一些实例中可提供管线地址作为函数的自变量,所述函数调用计算单元12A上的线程18A的执行(例如,可将恒定参数作为内核自变量提供到线程18A)。

[0123] 在一些实例中,线程18A可将一或多个存储器写入请求发出到数据存储系统16以将包数据写入到包含于数据存储系统16中的保留包条目。写入请求中的每一者可指定数据存储系统16中将写入数据的物理位置(例如,存储器地址)以及待写入的数据。写入请求可共同地将包数据写入到全部保留包条目。

[0124] 线程18A在一些实例中可将存储器写入请求直接发出到数据存储系统16和/或包含数据存储系统16的存储器子系统。换句话说,在此些实例中,计算系统10可包含计算单元12A与数据存储系统16之间绕过原子操作单元14的通信接口,且线程18A可经由所述通信接口将存储器写入请求发出到数据存储系统16,以使得存储器写入请求由数据存储系统16接收但并不通过原子操作单元14。

[0125] 在包数据已写入到管线数据缓冲器24中的保留包条目之后,原子操作单元14可将原子提交写入命令发出到原子操作单元14。所述原子提交写入命令可指示线程18A已结束将数据写入到管线的保留包条目,且此些包条目准备好可用于消费者任务。

[0126] 所述原子提交写入命令可包含指示数据写入到的管线的参数。指示数据写入到的管线的参数在一些实例中可为管线地址参数。例如,管线地址参数可为对应于存储器子系统(例如,数据存储系统16)中存储用于正存取的特定管线的管线状态参数22的位置的存储

器地址。管线地址参数可用以识别当数据存储系统16含有用于多于一个管线的数据时在数据存储系统16中将修改哪一管线,且识别数据存储系统16中存储管线数据的位置。

[0127] 响应于接收到原子提交写入命令,原子操作单元14可执行原子提交写入操作。原子提交写入操作可为如本发明中所描述修改两个或更多个管线状态参数的原子读取-修改-写入操作的实例。

[0128] 为了执行原子提交写入操作,原子操作单元14可从数据存储系统16获得管线状态参数。不同于原子保留写入操作,原子操作单元14在一些实例中当执行原子提交写入操作时可不将从数据存储系统16获得的任何管线状态参数返回到线程18A。

[0129] 原子操作单元14可产生第一管线状态参数的经修改值以使得所述经修改值反映所述管线具有相对于在执行原子提交写入操作之前管线具有的未提交保留的数目来说少一个的未提交保留。在一些实例中,第一管线参数可为保留计数参数,且原子操作单元14可递减保留计数参数以反映管线具有相对于在执行原子提交写入操作之前管线具有的未提交保留的数目来说少一个的未提交保留。

[0130] 原子操作单元14可进一步基于管线状态参数中的一或者确定是否管线的全部保留已经提交。响应于确定管线的全部保留已经提交,原子操作单元14可产生管线状态参数的一或多个经修改值以使得所述经修改值反映提交条目已经置于管线中。在一些实例中,原子操作单元14可将包计数参数递增等于保留包计数参数的量,且可将保留包计数参数复位到零以便反映提交的条目已经放置到管线中。响应于确定尚未提交管线的全部保留,原子操作单元14可产生除第一管线状态参数(例如,保留计数参数)外的管线状态参数的经修改值以使得所述经修改值等于管线状态参数的先前值。

[0131] 在确定管线状态参数的经修改值之后,原子操作单元14可将管线状态参数22的经修改值写入到数据存储系统16。归于原子提交写入命令的功能性在一些实例中可作为非可中断操作集合的部分而执行。换句话说,原子操作单元14可从起始到结束执行归于原子提交写入命令的功能性而无需中断执行以执行可与其它原子操作命令相关联的其它操作(例如,由与发出原子提交写入命令的任务或线程同时执行的任务和/或线程发出的其它原子操作命令)。以此方式执行非可中断操作集合允许原子操作单元14实施原子提交写入命令,其甚至在同时执行的线程和/或任务可能同时尝试将写入包条目提交到管线的情况下也确保管线的恰当功能性。下文相对于图3描述关于原子提交写入命令的实例实施方案的进一步细节。

[0132] 为了从管线读取数据,线程20A可将原子保留读取命令发出到原子操作单元14。原子保留读取命令可包含指示将从其读取数据的管线的参数以及指示将保留用于读取的包条目的数目的参数中的一者或两者。指示将从其读取数据的管线的参数在一些实例中可为管线地址参数。例如,管线地址参数可为对应于存储器子系统(例如,数据存储系统16)中存储用于正存取的特定管线的管线状态参数22的位置的存储器地址。管线地址参数可用以识别当数据存储系统16含有多个管线的数据时在数据存储系统16中将存取哪一管线,且识别数据存储系统16中存储管线数据的位置。

[0133] 响应于接收到原子保留读取命令,原子操作单元14可执行原子保留读取操作。原子保留读取操作可为如本发明中所描述修改两个或更多个管线状态参数的原子读取-修改-写入操作的实例。

[0134] 为了执行原子保留读取操作,原子操作单元14可从数据存储系统16获得管线状态参数,且基于管线状态参数中的一或者者而确定管线是否具有足够的未读取和未保留包条目,其中存储包数据以支持读取保留请求。响应于确定管线具有足够的其中存储包数据以支持读取保留请求的未读取和未保留包条目,原子操作单元14可产生管线状态参数的经修改值以使得所述经修改值反映针对所述管线已进行读取保留。响应于确定管线不具有足够的其中存储包数据以支持读取保留请求的未读取和未保留包条目,原子操作单元14可产生管线状态参数的经修改值以使得所述经修改值等于管线状态参数的先前值(即,使得所述经修改值反映针对管线尚未进行读取保留)。

[0135] 在确定管线状态参数的经修改值之后,原子操作单元14可将管线状态参数22的经修改值写入到数据存储系统16。归于原子保留读取命令的功能性在一些实例中可作为非可中断操作集合的部分而执行。换句话说,原子操作单元14可从起始到结束执行归于原子保留读取命令的功能性而无需中断执行以执行可与其它原子操作命令相关联的其它操作(例如,由与发出原子保留读取命令的任务或线程同时执行的任务和/或线程发出的其它原子操作命令)。以此方式执行非可中断操作集合允许原子操作单元14实施原子保留读取命令,其甚至在同时执行的线程和/或任务可能同时尝试保留管线用于读取的情况下也确保管线的恰当功能性。下文相对于图4描述关于原子保留读取命令的实例实施方案的进一步细节。

[0136] 线程20A可确定读取保留是否成功。为了确定写入保留是否成功,线程20A可在一些实例中以类似于原子操作单元14可如何做出同一确定的方式,基于返回管线状态参数确定管线是否具有足够的未读取和未保留的包条目来支持读取保留请求,如下文进一步详细所论述。返回的管线状态参数可指示在保留读取操作的执行之前的管线的状态。

[0137] 如果读取保留不成功,那么线程20A响应于读取保留操作的完成可不从管线读取数据或不将读取包提交到管线。在一些实例中,如果读取保留不成功,那么线程20A可以错误条件终止执行。在其它实例中,如果读取保留不成功,那么线程20A可执行一或多个操作以处置所述条件。例如,线程20A可停止尝试执行读取保留且继续执行其它操作。作为另一实例,线程20A可尝试在某一时间周期之后重复读取保留操作。这可允许产生者线程时间来将一或多个包写入到管线,进而添加可在后续读取保留操作中保留的额外包条目。

[0138] 如果读取保留成功,那么线程20A可基于当处理原子读取保留命令时由原子操作单元14返回的一或多个管线状态参数而确定将从其读取数据的管线数据缓冲器24的物理包条目。在其中线程20A接收到第一包偏移参数、包计数参数和保留包计数参数的实例中,线程20A可例如通过使用对应于第一包偏移参数的包条目而确定将从其读取数据的管线数据缓冲器24的第一物理包条目。线程20A可从所确定的第一物理包条目和随后的包条目读取数据直到保留的包条目的数目。如果在读取保留数目的包条目之前到达管线数据缓冲器24的末尾,那么线程20A可绕回到管线数据缓冲器24的开始以读取额外物理包条目。

[0139] 为了从管线数据缓冲器24中的保留包条目读取数据,线程20A在一些实例中可基于与待读取管线相关联的管线地址和基于所确定的将从其读取数据的物理包条目和/或基于返回的管线状态参数而确定包条目中的一或者者的物理存储器地址。在一些实例中,管线地址可由在数据存储系统16中分配管线的主机程序提供到线程20A。在此些实例中,管线地址在一些实例中可存储在计算单元12B的芯片上恒定存储器中。例如,在主机处理器上执行的主机程序在一些实例中可提供管线地址作为函数的自变量,所述函数调用计算单元

12B上的线程20A的执行(例如,可将恒定参数作为内核自变量提供到线程12B)。

[0140] 在一些实例中,线程20A可将一或多个存储器读取请求发出到数据存储系统16以从包含于数据存储系统16中的保留包条目读取包数据。读取请求中的每一者可指定数据存储系统16中将从其读取数据的物理位置(例如,存储器地址)。读取请求可从全部保留包条目共同地读取包数据。

[0141] 线程20A在一些实例中可将存储器读取请求直接发出到数据存储系统16和/或包含数据存储系统16的存储器子系统。换句话说,在此些实例中,计算系统10可包含计算单元12B与数据存储系统16之间绕过原子操作单元14的通信接口,且线程20A可经由所述通信接口将存储器读取请求发出到数据存储系统16,以使得存储器读取请求由数据存储系统16接收但并不通过原子操作单元14。

[0142] 在已从管线数据缓冲器24中的保留包条目读取包数据之后,原子操作单元14可将原子提交读取命令发出到原子操作单元14。原子提交读取命令可指示线程20A已结束从管线的保留包条目读取数据,且此些包条目准备好从管线移除。

[0143] 所述原子提交读取命令可包含指示数据读取到的管线的参数。指示数据读取到的管线的参数在一些实例中可为管线地址参数。例如,管线地址参数可为对应于存储器子系统(例如,数据存储系统16)中存储用于正存取的特定管线的管线状态参数22的位置的存储器地址。管线地址参数可用以识别当数据存储系统16含有多于一个管线的数据时在数据存储系统16中将存取哪一管线,且识别数据存储系统16中存储管线数据的位置。

[0144] 响应于接收到原子提交读取命令,原子操作单元14可执行原子提交读取操作。原子提交读取操作可为如本发明中所描述修改两个或更多个管线状态参数的原子读取-修改-写入操作的实例。

[0145] 为了执行原子提交读取操作,原子操作单元14可从数据存储系统16获得管线状态参数。不同于原子保留读取操作,原子操作单元14在一些实例中当执行原子提交读取操作时可不将从数据存储系统16获得的任何管线状态参数返回到线程20A。

[0146] 原子操作单元14可产生第一管线状态参数的经修改值以使得所述经修改值反映管线具有相对于在执行原子提交读取操作之前管线具有的未提交保留的数目来说少一个的未提交保留。在一些实例中,所述第一管线参数可为保留计数参数,且原子操作单元14可递减保留计数参数以反映所述管线具有相对于在执行原子提交读取操作之前管线具有的未提交保留的数目来说少一个的未提交保留。

[0147] 原子操作单元14可进一步基于管线状态参数中的一或者确定是否管线的全部保留已经提交。响应于确定管线的全部保留已经提交,原子操作单元14可产生管线状态参数的一或多个经修改值以使得所述经修改值反映提交条目已经从管线移除。在一些实例中,原子操作单元14可将包计数参数递减等于保留包计数参数的量,且可将保留包计数参数复位到零以便反映提交的条目已经从管线移除。在此些实例中,原子操作单元14还可将第一包偏移参数递增等于保留包计数参数的量。如果保留包条目的数目造成第一包偏移参数递增超过管线数据缓冲器24的末尾,那么第一包偏移参数可以环形缓冲器状方式绕回到管线数据缓冲器24的开始。响应于确定尚未提交管线的全部保留,原子操作单元14可产生除第一管线状态参数(例如,保留计数参数)外的管线状态参数的经修改值以使得所述经修改值等于管线状态参数的先前值。

[0148] 在确定管线状态参数的经修改值之后,原子操作单元14可将管线状态参数22的经修改值写入到数据存储系统16。归于原子提交读取命令的功能性在一些实例中可作为非可中断操作集合的部分而执行。换句话说,原子操作单元14可从起始到结束执行归于原子读取提交命令的功能性而无需中断执行以执行可与其它原子操作命令相关联的其它操作(例如,由与发出原子提交读取命令的任务或线程同时执行的任务和/或线程发出的其它原子操作命令)。以此方式执行非可中断操作集合允许原子操作单元14实施原子提交读取命令,其甚至在同时执行的线程和/或任务可能同时尝试将读取包条目提交到管线的情况下也确保管线的恰当功能性。下文相对于图5描述关于原子提交读取命令的实例实施方案的进一步细节。

[0149] 管线可包含多个包条目,其中所述包条目中的每一者可配置以存储用于管线的数据包。管线的大小可指代包含在管线中的包条目的数目。在一些实例中,支持管线的API可允许编程者指定特定限制内的管线的任何大小。然而,支持API的运行时间程序(例如,驱动程序)在一些实例中可实施管线以使得包含在为管线分配的物理存储器中的包条目的数目不同于由编程者指定的包条目的数目。例如,运行时间程序可实施管线以使得包含在为管线中的每一者分配的物理存储器中的包条目的数目始终为二的幂。即使由运行时间程序在物理存储器中分配的包条目的数目可不同于由主机程序指定的包条目的数目,运行时间程序也可以一方式控制管线的状态以使得从正在使用管线的任务的角度来看,所述管线表现为仅具有由主机程序指定的包条目的数目。

[0150] 例如,如果运行时间程序实施管线以使得包含在为每一管线分配的物理存储器中的包条目的数目始终为二的幂,且如果编程者指定11个包条目的管线大小,那么运行时间程序可分配物理存储器以使得物理存储器包含用于管线的16个包条目。在此实例中,运行时间程序可以一方式控制管线的状态以使得即使在物理存储器中已经为管线分配16个包条目,从正在使用所述管线的任务的角度来看,管线也表现为仅具有11个包条目。

[0151] 从源代码和/或编程者的角度来看的管线的包条目可在本文中被称作逻辑包条目,且在物理存储器中实际上为管线分配的包条目可在本文中被称作物理包条目。类似地,由源代码和/或编程者指定的管线的大小(例如,包条目的数目)可在本文中被称作管线的逻辑大小,且在存储器中实际上为管线分配的包条目的数目可在本文中被称作管线的物理大小。

[0152] 返回到原子保留写入命令,除包含指示数据将写入到的管线的参数和指示将保留用于写入的包条目的数目的参数之外,在一些实例中,原子保留写入命令还可进一步包含指示管线的逻辑大小的参数和/或指示管线的物理大小减一的参数。原子保留写入操作可使用指示管线的逻辑大小的参数以确定管线是否具有足够的未提交和未保留条目来支持写入保留请求。例如,原子操作单元14可从管线的逻辑大小减去提交到管线的包的总数和保留用于写入的包条目的总数,且确定所述差是否大于或等于由保留请求所请求以保留用于写入的包的数目。如果所述差大于或等于由保留请求所请求以保留用于写入的包的数目,那么原子操作单元14可确定所述管线具有足够的未提交和未保留条目来支持写入保留请求。否则,如果所述差不大于或等于由保留请求指定保留用于写入的包的数目,那么原子操作单元14可确定所述管线不具有足够的未提交和未保留条目来支持写入保留请求。

[0153] 类似地,对于原子保留读取命令,在一些实例中,除包含指示将从其读取数据的管

线的参数和指示将保留用于读取的包条目的数目的参数之外,原子保留读取命令还可进一步包含指示管线的逻辑大小的参数和/或指示管线的物理大小减一的参数。原子保留读取操作可使用指示管线的逻辑大小的参数以确定管线是否具有足够的未读取和未保留包条目,其中存储包数据以支持读取保留请求。例如,原子操作单元14可将先前保留用于读取的包的数目和由保留请求所请求以保留用于读取的包的数目相加,且确定所述总和是否小于或等于存储在管线中的包的总数(即,写入且提交到管线的包的总数)。如果所述总和小于或等于存储在管线中的包的总数,那么原子操作单元14可确定所述管线具有足够包条目来支持读取保留请求。否则,如果所述总和不小于或等于存储在管线中的包的总数,那么原子操作单元14可确定所述管线不具有足够包条目来支持读取保留请求。

[0154] 而且,对于原子提交读取命令,在一些实例中,除包含指示从其读取数据的管线的参数之外,原子读取命令还可进一步包含指示管线的物理大小减一的参数。当全部读取保留已经提交时,管线的物理大小减一参数可由原子提交读取命令使用以更新第一包偏移参数。例如,管线的物理大小减一参数可用以实施模数操作,其允许当第一包偏移参数递增的量致使第一包偏移参数递增超过管线数据缓冲器24的末尾时第一包偏移参数从物理包条目的末尾绕回到物理包条目的开始。

[0155] 在一些实例中,指示数据将写入到的管线的参数、指示管线的大小的参数以及指示管线的大小减一的参数中的一或多者可由分配数据存储系统16中的管线的主机程序提供到线程18A和/或线程20A。此些参数可被称为恒定参数,因为此些参数可当分配管线时确定且贯穿线程18、20的执行可保持恒定。在此些实例中,所述恒定参数在一些实例中可存储在计算单元12A和/或计算单元12B的芯片上恒定存储器中。在主机处理器上执行的主机程序在一些实例中可将所述恒定参数作为自变量提供到调用计算单元12A上的线程18A和/或计算单元12B上的线程20A的执行的函数(例如,所述恒定参数可作为内核自变量提供到线程18A和线程20A)。

[0156] 现将相对于图2到5描述关于由原子操作单元14针对原子保留命令和原子提交命令执行的实例功能性的进一步细节。图2到5中说明的原子操作命令中的每一者的功能性可以硬件实施,且所述命令中的每一者可为如本发明中所描述修改两个或更多个管线状态参数的基于硬件的原子读取-修改-写入操作的实例。

[0157] 图2是说明根据本发明的界定实例原子保留写入命令的接口和功能性的实例伪码的概念图。所述伪码的1到4行将在此实例中将存取和/或修改的管线状态参数映射到阵列的相应元素。管线状态参数包含指示存储管线的起始包的包条目的第一包偏移参数(即,“FST_PKT”)、指示多少包当前存储在管线中的包计数参数(即,“PKT_CNT”)、指示多少包条目当前保留在管线中的保留包计数参数(即,“RES_CNT”),以及指示多少未提交保留当前对于管线待决的保留计数参数(即,“NUM_RES”)。所述映射将第一包偏移参数映射到阵列的第一元素,将包计数参数映射到阵列的第二元素,等等。

[0158] 所述伪码的5到11行指定用于原子保留写入命令的接口。第5行指定命令的名称,且7到10行指定当调用所述命令时将使用的自变量或参数。第7行指定可指向数据存储系统16中存储待存取管线的管线状态参数22的存储器区的管线地址参数。在一些实例中,管线地址参数可为基于机器地址的32位地址或64位地址。第8行指定物理管线大小减一参数,其对应于为数据存储系统16中的管线分配的物理包条目的数目减一。第9行指定指示当管线

产生时由源代码从主机程序所请求的逻辑管线条目数目的逻辑管线大小参数(即,从编程者观点来看的管线的大小)。第10行指定指示请求任务请求保留用于写入的包的数目的所请求包参数的数目。在一些实例中,用于原子保留写入命令的接口可不包含物理管线大小减一参数。

[0159] 当原子保留写入命令由线程调用时,原子操作单元14可执行读取操作,其从数据存储系统16读取在此实例中使用的四个管线状态参数(第13行)。如图2中指示,原子操作单元14可从数据存储系统16读取含有全部四个管线状态参数的阵列。在一些实例中,参数中的每一者可为32位,且读取操作可为单个128位加载操作。换句话说,在一些实例中,可相对于数据存储系统16使用单个读取操作和/或加载操作以加载在原子操作期间存取和/或修改的全部管线状态参数。以此方式,原子读取-修改-写入操作可能够修改多个管线状态参数而不需要多于一个存储器读取事务。

[0160] 在读取管线状态参数之后,原子操作单元14可将从数据存储系统16读取的管线状态参数中的三个返回到请求线程(14到15行)。在此实例中返回的所述三个管线状态参数是第一包偏移参数、包计数参数和保留包计数参数。请求线程(例如,线程18A)可基于返回的管线状态参数确定其中将写入数据的管线数据缓冲器24的物理包条目。例如,线程18A可例如通过将这些参数中的每一者的值相加在一起而确定其中将写入数据的管线数据缓冲器24的第一物理包条目。线程18A可将数据写入到所确定的第一物理包条目和随后的包条目直到保留的包条目的数目。如果在写入保留数目的包条目之前到达管线数据缓冲器24的末尾,那么线程18A可绕回到管线数据缓冲器24的开始以写入到额外物理包条目。

[0161] 在读取管线状态参数之后,原子操作单元14可执行修改操作,其产生管线状态参数的经修改值(16到20行)。为了执行修改操作,原子操作单元14可基于管线状态参数中的一或者更多者而确定管线是否具有足够未提交和未保留条目来支持保留请求(第16行)。更具体来说,在此实例中,原子操作单元14可确定逻辑管线大小减包计数参数减保留包计数参数是否大于或等于所请求包数目(第16行)。响应于确定管线具有足够未提交和未保留条目来支持写入保留请求,原子操作单元14可如第17到20中指定产生管线状态参数的经修改值。

[0162] 例如,原子操作单元14可执行递增操作,其将保留包计数参数递增等于所请求包数目的量(第17行),且如果所请求包数目不等于零,那么原子操作单元14可执行递增一操作,其将保留计数参数递增一(18到20行)。在第18行的if语句中包含递增一操作可以确保当所请求包数目等于零时不修改保留计数参数。用户可将所请求包数目设定为等于零,且发出原子保留写入命令以获得管线的当前状态而无需做出保留(即,无需修改参数以反映写入保留已发生)。

[0163] 概念地,响应于确定管线具有足够未提交和未保留条目来支持写入保留请求,原子操作单元14可取决于所请求包数目是否大于零而以不同方式修改管线状态参数。如果所请求包数目大于零,那么原子操作单元14可产生管线状态参数的经修改值以使得所述经修改值反映已针对管线做出写入保留(17到19行)。如果所请求包数目等于零,那么原子操作单元14可产生管线状态参数的经修改值以使得所述经修改值等于管线状态参数的先前值(即,以使得所述经修改值反映尚未针对管线做出写入保留)(17到19行)。

[0164] 响应于确定管线不具有足够未提交和未保留条目来支持写入保留请求或响应于保留计数参数等于零,原子操作单元14可不更新管线状态参数值以使得管线状态保持相

同。不更新管线状态参数值以使得管线状态保持相同可对应于产生管线状态参数的经修改值以使得所述经修改值等于管线状态参数的先前值(即,使得所述值反映尚未针对管线做出写入保留)。

[0165] 在产生经修改管线状态参数值之后,原子操作单元14可执行写入操作(例如,存储操作),其将在此实例中使用的四个管线状态参数写入到数据存储系统16(第21行)。如图2中指示,原子操作单元14可将含有全部四个管线状态参数的阵列写入到数据存储系统16。在一些实例中,参数中的每一者可为32位,且写入操作可为单个128位存储操作。换句话说,在一些实例中,可相对于数据存储系统16使用单个写入操作和/或存储操作以存储在原子操作期间存取和/或修改的全部管线状态参数。以此方式,原子读取-修改-写入操作可能够修改多个管线状态参数而不需要多于一个存储器写入事务。原子保留写入操作的处理现在完成。

[0166] 在一些实例中,原子保留写入命令的语法可采取以下形式:

[0167] PIPE_RSV_WRITE Dest,Src0,Src1

[0168] 其中Dest对应于将接收由原子保留写入命令返回的输出管线状态参数的三个通用寄存器(GPR)寄存器的群组中的第一GPR的GPR识别(ID),Src0对应于含有所请求数目的包的寄存器的GPR ID,且Src1对应于含有物理管线大小减一、逻辑管线大小和管线地址的四个恒定寄存器的群组中的第一寄存器的恒定ID。管线地址可存储在含有指向管线状态参数22的32位或64位地址的一个或两个寄存器。在一些实例中,物理管线大小可为二的幂,以使得物理管线大小减一是二的幂减一。

[0169] 如上文所论述,原子保留写入操作可返回第一包偏移参数、包计数参数和保留包计数参数。因此,所述命令可为返回原子的实例。在一些实例中,计算单元12中的一者或两者可包含一或多个四重寄存器群组,其中每一四重寄存器群组包含四个寄存器,且用于所述四个寄存器的数据可作为单个高速缓冲存储器事务和/或存储器事务的部分从存储器读取和/或存储在存储器中。在此些实例中,由原子保留写入操作返回的三个管线状态参数在一些实例中可在四重寄存器群组内的三个连续寄存器中返回。以此方式,所述三个管线状态参数可作为单个高速缓冲存储器事务和/或存储器事务的部分由计算单元12中的一者或两者获取。

[0170] 在一些实例中,原子保留写入命令可用以实施如根据OpenCL™ C规范界定的reserve_write_pipe命令。在额外实例中,通过将保留计数参数设定为等于零,原子保留写入命令可返回管线的当前状态而无需对所述管线添加保留。以此方式,原子保留写入命令也可用以实施如根据OpenCL™ C规范界定的get_pip_num_packets命令。

[0171] 图3是说明根据本发明的界定实例原子提交写入命令的接口和功能性的实例伪码的概念图。所述伪码的1到4行将在此实例中将存取和/或修改的管线状态参数映射到阵列的相应元素。管线状态参数和映射与上文相对于图2描述的内容相同。

[0172] 所述伪码的第5行指定用于原子提交写入命令的接口,其包含命令的名称以及当调用所述命令时将使用的自变量或参数。更具体来说,原子提交写入命令可使用指向数据存储系统16中存储管线状态参数22的存储器区的管线地址参数。在一些实例中,管线地址参数可为基于机器地址的32位地址或64位地址。

[0173] 当原子提交写入命令由线程调用时,原子操作单元14可执行读取操作,其从数据

存储系统16读取在此实例中使用的四个管线状态参数(第7行)。如图3中指示,原子操作单元14可从数据存储系统16读取含有全部四个管线状态参数的阵列。在一些实例中,参数中的每一者可为32位,且读取操作可为单个128位加载操作。换句话说,在一些实例中,可相对于数据存储系统16使用单个读取操作和/或加载操作以加载在原子操作期间存取和/或修改的全部管线状态参数。以此方式,原子读取-修改-写入操作可能够修改多个管线状态参数而不需要多于一个存储器读取事务。

[0174] 在读取管线状态参数之后,原子操作单元14可执行修改操作,其产生管线状态参数的经修改值(8到12行)。为了执行修改操作,原子操作单元14可执行递减一操作,其将保留计数参数递减一(第8行)。将保留计数参数递减一可对应于产生第一管线状态参数的经修改值以使得所述经修改值反映管线具有相对于在执行原子提交写入操作之前管线具有的未提交保留的数目来说少一个的未提交保留。

[0175] 原子操作单元14可进一步基于管线状态参数中的一或者确定是否管线的全部保留已经提交(第9行)。更具体来说,在此实例中,在将保留计数参数递减一之后,原子操作单元14可确定保留计数参数是否等于零。

[0176] 响应于确定管线的全部保留已经提交,原子操作单元14可产生管线状态参数的一或多个经修改值以使得所述经修改值反映提交条目已经置于管线中(10到11行)。更具体来说,在此实例中,原子操作单元14可执行递增操作,其将包计数参数递增等于保留包计数参数的量(第10行),且可执行复位操作,其将保留包计数参数复位到零(第11行)。

[0177] 响应于确定管线的并非全部保留已经提交,除已经更新的保留计数参数以外原子操作单元14可不更新其它管线状态参数值,以使得剩余管线状态参数保持相同。不更新剩余管线状态参数值以使得剩余管线状态参数保持相同可对应于产生剩余管线状态参数的经修改值以使得所述经修改值等于管线状态参数的先前值。

[0178] 在产生经修改管线状态参数值之后,原子操作单元14可执行写入操作(例如,存储操作),其将在此实例中使用的四个管线状态参数写入到数据存储系统16(第13行)。如图3中指示,原子操作单元14可将含有全部四个管线状态参数的阵列写入到数据存储系统16。在一些实例中,参数中的每一者可为32位,且写入操作可为单个128位存储操作。换句话说,在一些实例中,可相对于数据存储系统16使用单个写入操作和/或存储操作以存储在原子操作期间存取和/或修改的全部管线状态参数。以此方式,原子读取-修改-写入操作可能够修改多个管线状态参数而不需要多于一个存储器写入事务。原子提交写入命令的处理现在完成。

[0179] 在一些实例中,原子提交写入命令的语法可采取以下形式:

[0180] PIPE_CMT_WRITE Src0

[0181] 其中不指定Dest寄存器,且Src0对应于含有物理管线大小减一、逻辑管线大小和管线地址的四个恒定寄存器的群组中的第一寄存器的恒定ID。管线地址可存储在含有指向管线状态参数22的32位或64位地址的一个或两个寄存器。在一些实例中,物理管线大小可为二的幂,以使得物理管线大小减一是二的幂减一。

[0182] 原子提交写入命令在此实例中不返回任何值。因此,所述命令可为不返回原子的实例。在一些实例中,原子提交写入命令可用以实施如根据OpenCL™ C规范界定的commit_write_pipe命令。

[0183] 图4是说明根据本发明的界定实例原子保留读取命令的接口和功能性的实例伪码的概念图。所述伪码的1到4行将在此实例中将存取和/或修改的管线状态参数映射到阵列的相应元素。管线状态参数和映射与上文相对于图2描述的内容相同。

[0184] 所述伪码的5到11行指定用于原子保留读取命令的接口。第5行指定命令的名称，且7到10行指定当调用所述命令时将使用的自变量或参数。第7行指定可指向数据存储系统16中存储管线状态参数22的存储器区的管线地址参数。在一些实例中，所述管线地址参数可为基于机器地址的32位地址或64位地址。第8行指定物理管线大小减一参数，其对应于为数据存储系统16中的管线分配的物理包条目的数目减一。第9行指定指示当管线产生时由源代码从主机程序所请求的逻辑管线条目数目的逻辑管线大小参数(即，从编程者观点来看的管线的大小)。第10行指定指示请求任务请求用于读取的包的数目的所请求包参数的数目。在一些实例中，用于原子保留读取命令的接口可不包含物理管线大小减一参数。

[0185] 当原子保留读取命令由线程调用时，原子操作单元14可执行读取操作，其从数据存储系统16读取在此实例中使用的四个管线状态参数(第13行)。如图4中指示，原子操作单元14可从数据存储系统16读取含有全部四个管线状态参数的阵列。在一些实例中，参数中的每一者可为32位，且读取操作可为单个128位加载操作。换句话说，在一些实例中，可相对于数据存储系统16使用单个读取操作和/或加载操作以加载在原子操作期间存取和/或修改的全部管线状态参数。以此方式，原子读取-修改-写入操作可能够修改多个管线状态参数而不需要多于一个存储器读取事务。

[0186] 在读取管线状态参数之后，原子操作单元14可将从数据存储系统16读取的管线状态参数中的三个返回到请求线程(14到15行)。在此实例中返回的所述三个管线状态参数是第一包偏移参数、包计数参数和保留包计数参数。请求线程(例如，线程20A)可基于返回的管线状态参数确定将从其读取数据的管线数据缓冲器24的物理包条目。例如，线程20A可例如通过使用对应于第一偏移参数的包条目作为第一包条目而确定将从其读取数据的管线数据缓冲器24的第一物理包条目。线程20A可从所确定的第一物理包条目和随后的包条目读取数据直到保留的包条目的数目。如果在读取保留数目的包条目之前到达管线数据缓冲器24的末尾，那么线程20A可绕回到管线数据缓冲器24的开始以读取额外物理包条目。

[0187] 在读取管线状态参数之后，原子操作单元14可执行修改操作，其产生管线状态参数的经修改值(16到20行)。为了执行修改操作，原子操作单元14可基于管线状态参数中的一或者而确定管线是否具有足够的其中存储包数据以支持读取保留请求的未读取和未保留包条目(第16行)。更具体来说，在此实例中，原子操作单元14可确定保留包计数参数和所请求包数目的总和是否小于或等于包计数参数(第16行)。响应于确定所述管线具有足够的其中存储包数据以支持读取保留请求的未读取和未保留包条目，原子操作单元14可产生管线状态参数的经修改值，如17到20行中指定。

[0188] 例如，原子操作单元14可执行递增操作，其将保留包计数参数递增等于所请求包数目的量(第17行)，且如果所请求包数目不等于零，那么原子操作单元14可执行递增一操作，其将保留计数参数递增一(18到20行)。在第18行的if语句中包含递增一操作可以确保当所请求包数目等于零时不修改保留计数参数。用户可将所请求包数目设定为等于零，且发出原子保留读取命令以获得管线的当前状态而无需做出保留(即，无需修改参数以反映读取保留已发生)。

[0189] 概念地,响应于确定管线具有足够的其中存储包数据以支持读取保留请求的未读取和未保留包条目,原子操作单元14可取决于所请求包数目是否大于零而以不同方式修改管线状态参数。如果所请求包数目大于零,那么原子操作单元14可产生管线状态参数的经修改值以使得所述经修改值反映已针对管线做出读取保留(17到19行)。如果所请求包数目等于零,那么原子操作单元14可产生管线状态参数的经修改值以使得所述经修改值等于管线状态参数的先前值(即,以使得所述经修改值反映尚未针对管线做出读取保留)(17到19行)。

[0190] 响应于确定管线不具有足够的其中存储包数据以支持读取保留请求的未读取和未保留包条目或响应于保留计数参数等于零,原子操作单元14可不更新管线状态参数值以使得管线状态保持相同。不更新管线状态参数值以使得管线状态保持相同可对应于产生管线状态参数的经修改值以使得所述经修改值等于管线状态参数的先前值(即,使得所述值反映尚未针对管线做出读取保留)。

[0191] 在产生经修改管线状态参数值之后,原子操作单元14可执行写入操作(例如,存储操作),其将在此实例中使用的四个管线状态参数写入到数据存储系统16(第21行)。如图4中指示,原子操作单元14可将含有全部四个管线状态参数的阵列写入到数据存储系统16。在一些实例中,参数中的每一者可为32位,且写入操作可为单个128位存储操作。换句话说,在一些实例中,可相对于数据存储系统16使用单个写入操作和/或存储操作以存储在原子操作期间存取和/或修改的全部管线状态参数。以此方式,原子读取-修改-写入操作可能够修改多个管线状态参数而不需要多于一个存储器写入事务。原子保留读取命令的处理现在完成。

[0192] 在一些实例中,原子保留读取命令的语法可采取以下形式:

[0193] PIPE_RSV_READ Dest,Src0,Src1

[0194] 其中Dest对应于将接收由原子保留读取命令返回的输出管线状态参数的三个通用寄存器(GPR)寄存器的群组中的第一GPR的GPR识别(ID),Src0对应于含有所请求数目的包的寄存器的GPR ID,且Src1对应于含有物理管线大小减一、逻辑管线大小和管线地址的四个恒定寄存器的群组中的第一寄存器的恒定ID。管线地址可存储在含有指向管线状态参数22的32位或64位地址的一个或两个寄存器。在一些实例中,物理管线大小可为二的幂,以使得物理管线大小减一是二的幂减一。

[0195] 如上文所论述,原子保留读取操作可返回第一包偏移参数、包计数参数和保留包计数参数。因此,所述命令可为返回原子的实例。在一些实例中,计算单元12中的一者或两者可包含一或多个四重寄存器群组,其中每一四重寄存器群组包含四个寄存器,且用于所述四个寄存器的数据可作为单个高速缓冲存储器事务和/或存储器事务的部分从存储器读取和/或存储在存储器中。在此些实例中,由原子保留读取操作返回的三个管线状态参数在一些实例中可在四重寄存器群组内的三个连续寄存器中返回。以此方式,所述三个管线状态参数可作为单个高速缓冲存储器事务和/或存储器事务的部分由计算单元12中的一者或两者获取。

[0196] 在一些实例中,原子保留读取命令可用以实施如根据OpenCL™ C规范界定的reserve_read_pipe命令。在额外实例中,通过将保留计数参数设定为等于零,原子保留读取命令可返回管线的当前状态而无需对所述管线添加保留。以此方式,原子保留读取命令

也可用以实施如根据OpenCL™ C规范界定的get_pip_num_packets命令。

[0197] 图5是说明根据本发明的界定实例原子提交读取命令的接口和功能性的实例伪码的概念图。所述伪码的1到4行将在此实例中将存取和/或修改的管线状态参数映射到阵列的相应元素。管线状态参数和映射与上文相对于图2描述的内容相同。

[0198] 所述伪码的第5行指定用于原子提交读取命令的接口，其包含命令的名称以及当调用所述命令时将使用的自变量或参数。更具体来说，原子提交读取命令可使用指向数据存储系统16中存储管线状态参数22的存储器区的管线地址参数，以及对应于为数据存储系统16中的管线分配的物理包条目的数目减一的物理管线大小减一参数。在一些实例中，管线地址参数可为基于机器地址的32位地址或64位地址。

[0199] 当原子提交读取命令由线程调用时，原子操作单元14可执行读取操作，其从数据存储系统16读取在此实例中使用的四个管线状态参数(第7行)。如图5中指示，原子操作单元14可从数据存储系统16读取含有全部四个管线状态参数的阵列。在一些实例中，参数中的每一者可为32位，且读取操作可为单个128位加载操作。换句话说，在一些实例中，可相对于数据存储系统16使用单个读取操作和/或加载操作以加载在原子操作期间存取和/或修改的全部管线状态参数。以此方式，原子读取-修改-写入操作可能够修改多个管线状态参数而不需要多于一个存储器读取事务。

[0200] 在读取管线状态参数之后，原子操作单元14可执行修改操作，其产生管线状态参数的经修改值(8到13行)。为了执行修改操作，原子操作单元14可执行递减一操作，其将保留计数参数递减一(第8行)。将保留计数参数递减一可对应于产生第一管线状态参数的经修改值以使得所述经修改值反映管线具有相对于在执行原子提交读取提交操作之前所述管线具有的未提交保留的数目来说少一个的未提交保留。

[0201] 原子操作单元14可进一步基于管线状态参数中的一或者确定是否管线的全部保留已经提交(第9行)。更具体来说，在此实例中，在将保留计数参数递减一之后，原子操作单元14可确定保留计数参数是否等于零。

[0202] 响应于确定管线的全部保留已经提交，原子操作单元14可产生管线状态参数的一或多个经修改值以使得所述经修改值反映提交条目已经从管线移除(10到12行)。更具体来说，在此实例中，原子操作单元14可执行模加法操作，其将第一包偏移参数递增等于保留包计数参数的量，且以管线的物理大小为模数对所述结果执行模操作(第10行)。原子操作单元14还可执行递减操作，其将包计数参数递减等于保留包计数参数的量(第11行)，且执行复位操作，其将保留包计数参数复位到零(第12行)。

[0203] 原子读取提交命令可使用管线的物理大小减一参数来执行模加法操作。更具体来说，使用第一包偏移参数和保留包计数参数的总和作为第一操作数且使用物理管线大小减一作为模数来执行逐位“与”操作。这可允许当第一包偏移参数递增的量致使第一包偏移参数递增超过管线数据缓冲器24的末尾时，第一包偏移参数以环形缓冲器状方式从物理包条目的末尾绕回到物理包条目的开始。

[0204] 响应于确定管线的并非全部保留已经提交，除已经更新的保留计数参数以外原子操作单元14可不更新其它管线状态参数值，以使得剩余管线状态参数保持相同。不更新剩余管线状态参数值以使得剩余管线状态参数保持相同可对应于产生剩余管线状态参数的经修改值以使得所述经修改值等于管线状态参数的先前值。

[0205] 在产生经修改管线状态参数值之后,原子操作单元14可执行写入操作(例如,存储操作),其将在此实例中使用的四个管线状态参数写入到数据存储系统16(第13行)。如图5中指示,原子操作单元14可将含有全部四个管线状态参数的阵列写入到数据存储系统16。在一些实例中,参数中的每一者可为32位,且写入操作可为单个128位存储操作。换句话说,在一些实例中,可相对于数据存储系统16使用单个写入操作和/或存储操作以存储在原子操作期间存取和/或修改的全部管线状态参数。以此方式,原子读取-修改-写入操作可能够修改多个管线状态参数而不需要多于一个存储器写入事务。原子提交读取命令的处理现在完成。

[0206] 在一些实例中,原子提交读取命令的语法可采取以下形式:

[0207] PIPE_CMT_READ Src0

[0208] 其中不指定Dest寄存器,且Src0对应于含有物理管线大小减一、逻辑管线大小和管线地址的四个恒定寄存器的群组中的第一寄存器的恒定ID。管线地址可存储在含有指向管线状态参数22的32位或64位地址的一个或两个寄存器。在一些实例中,物理管线大小可为二的幂,以使得物理管线大小减一是二的幂减一。

[0209] 原子提交读取命令在此实例中不返回任何值。因此,所述命令可为不返回原子的实例。在一些实例中,原子提交读取命令可用以实施如根据OpenCL™ C规范界定的commit_read_pipe命令。

[0210] 在一些实例中,为了允许使用单个读取/加载操作和单个写入/存储操作来存取管线状态参数,用以存储在原子操作期间存取和/或修改的管线状态参数的位数目可小于或等于包含于在单个存储器事务期间从数据存储系统16读取或写入到数据存储系统16的数据块中的位数目。例如,用以存储多个管线状态参数的位数目可小于或等于存储器字中的位数目和/或高速缓冲存储器线中的位数目。在某些情况下,存储于存储器子系统中的管线中的每一者的管线状态参数可经对准以使得管线的全部管线状态参数存储在存储器子系统的单个存储器字和/或单个高速缓冲存储器线中。以此方式,原子读取-修改-写入操作可能够修改多个管线状态参数而不需要多于一个存储器读取事务或多于一个存储器写入事务。

[0211] 在一些实例中,指示数据将写入到的管线的参数、指示管线的大小的参数以及指示管线的大小减一的参数中的一或更多者可由分配数据存储系统16中的管线的主机程序提供到线程18A和/或线程20A。此些参数可被称为恒定参数,因为此些参数可当分配管线时确定且贯穿线程18、20的使用期限可保持恒定。在此些实例中,所述恒定参数在一些实例中可存储在计算单元12A和/或计算单元12B的芯片上恒定存储器中。在主机处理器上执行的主机程序在一些实例中可将所述恒定参数作为自变量提供到调用计算单元12A上的线程18A和/或计算单元12B上的线程20A的执行的函数(例如,所述恒定参数可作为内核自变量提供到线程18A和线程20A)。

[0212] 图6是图1的实例计算系统10的框图,其中更详细说明图1中所示的原子操作单元14的实例。如图6中所示,原子操作单元14包含命令队列26、控制单元28和原子操作数据路径30。

[0213] 命令队列26可存储从在计算单元12上执行的线程18、20接收的多个原子操作命令。所述原子操作命令可包含例如原子保留写入命令、原子提交写入命令、原子保留读取命

令和原子提交读取命令。所述命令可替代地在本文中被称作请求。

[0214] 响应于从线程18、20中的一者接收到命令,命令队列26可将命令存储在命令队列26中,即使另一命令当前正由原子操作单元14处理也是如此。这可允许原子操作单元14从在计算单元12同时执行的多个线程和/或任务接收多个命令,而不需要线程18、20在发出命令之前等待直到原子操作单元14自由且不需要线程18、20由于原子操作单元14正忙而多次发出命令。

[0215] 控制单元28可处理存储在命令队列26中的原子操作命令。为了处理命令,控制单元28可执行对应于经处理命令的原子操作。例如,为了处理原子保留写入命令,控制单元28可执行如本发明中所描述的原子保留写入操作。为了处理原子提交写入命令,控制单元28可执行如本发明中所描述的原子提交写入操作。为了处理原子保留读取命令,控制单元28可执行如本发明中所描述的原子保留读取操作。为了处理原子提交读取命令,控制单元28可执行如本发明中所描述的原子提交读取操作。

[0216] 在一些实例中,控制单元28可循序地处理命令以使得每次执行一个原子读取-修改-写入操作,且使得每一原子读取-修改-写入操作的执行是在开始执行另一原子读取-修改-写入操作之前完成。以此方式处理命令可确保原子操作的原子性。

[0217] 在一些实例中,上文所提到的原子操作可为读取-修改-写入操作。在此些实例中,原子操作单元14可从并行地执行的多个线程接收执行基于硬件的原子读取-修改-写入操作的多个请求,且响应于接收到执行基于硬件的原子读取-修改-写入操作的所述多个请求而循序地执行多个基于硬件的原子读取-修改-写入操作。所述基于硬件的原子读取-修改-写入操作中的每一者可响应于所述多个请求中的相应一个而执行。

[0218] 当处理原子操作命令时执行的操作在所述操作是从开始到结束执行而无来自可由与请求原子操作的任务或线程同时执行的其它任务和/或线程请求的任何原子操作的中断的意义上可为原子的。换句话说,在原子操作的执行期间,没有其它线程和/或任务能够读取或修改由当前执行的原子操作修改和/或存取的管线状态参数。以此方式,在其中两个或更多个任务和/或线程同时尝试修改管线的管线状态参数的情况下可避免竞态条件。

[0219] 由原子操作单元14执行的原子操作可为基于硬件的原子操作。原子操作在所述原子操作可作为单个不可分操作而由原子操作单元14实施的意义上可为基于硬件的,所述单个不可分操作可由以通信方式耦合到原子操作单元14的线程和/或任务调用。所述单个不可分操作在原子操作单元14从起始到结束执行所述操作而无需中断操作以执行可由与请求所述原子操作的任务或线程同时执行的其它任务和/或线程请求的其它原子操作的意义上可为不可分的。

[0220] 在图6的实例原子操作单元14中,为了执行基于硬件的原子操作,原子操作单元14可包含实施一或多个原子操作数据路径30的电路。原子操作数据路径30中的每一者可为不可编程的数据路径。换句话说,原子操作数据路径30中的每一者可经配置以执行归于相应数据路径的功能性而不需要所述数据路径下载且执行用户指定的程序。

[0221] 原子操作数据路径30在一些实例中可实施用于本发明中所描述的原子操作中的每一者的基于硬件的数据路径。在一些实例中,电路可包含用于原子操作中的每一者的单独的基于硬件的数据路径。在其它实例中,共同基于硬件的数据路径可配置以执行本发明中所描述的原子操作中的两者或更多者。在一些实例中,所述共同基于硬件的数据路径可

为读取-修改-写入操作数据路径,其中修改操作可配置以基于将执行的原子操作的类型而执行不同修改操作。在一些实例中,原子操作数据路径30中的每一者可包含一或多个ALU以执行相关联原子操作命令的修改操作。

[0222] 图7是说明根据本发明的与数据存储系统16交互的实例原子操作数据路径32的概念图。原子操作数据路径32可经配置以执行基于硬件的原子读取-修改-写入操作,其修改指示管线的状态的两个或更多个管线状态参数。原子操作数据路径32可为上文相对于图6说明和描述的原子操作数据路径30中的一者的实例。原子操作数据路径32包含读取块34、修改块36和写入块38。

[0223] 读取块34可经配置以执行从包含数据存储系统16的存储器子系统读取对应于管线状态参数22的值的读取操作。修改块36可经配置以执行产生管线状态参数的经修改值的修改操作。写入块38可经配置以执行将管线状态参数的经修改值写入到包含数据存储系统16的存储器子系统的写入操作。

[0224] 读取块34、修改块36和写入块38中的每一者可以经配置以执行归于这些组件的功能性的电路来实施。在一些实例中,读取块34、修改块36和写入块38中的一或多者可为原子操作数据路径32中的单独硬件块。在其它实例中,共同硬件块可配置以实施读取块34、修改块36和写入块38中的一或多者。

[0225] 如图7中所示,读取块34、修改块36和修改块36中的一或多者可接收与发出到原子操作单元14的原子操作命令结合而提供到原子操作单元14的命令自变量(即,参数)。所述命令自变量可包含例如管线地址、物理管线大小减一、逻辑管线大小和所请求包数目中的一或多者。

[0226] 现将描述原子操作数据路径32的操作。控制单元28可响应于处理原子操作命令使用原子操作数据路径32起始原子读取-修改-写入操作的执行。控制单元28可将一或多个命令自变量提供到原子操作数据路径32以便执行原子读取-修改-写入操作。

[0227] 读取块34可基于管线地址执行读取操作。管线地址可为指向数据存储系统16中存储管线状态参数22的位置的存储器地址。为了执行读取操作,读取块34可将读取请求发出到数据存储系统16。所述读取请求可包含管线地址。响应于接收到读取请求,数据存储系统16可将对应于管线的当前状态的管线状态参数(PSP)值提供到读取块34。读取块34可将PSP值提供到修改块36。

[0228] 修改块36可接收PSP值且基于一或多个修改功能产生管线状态参数的经修改PSP值。在一些实例中,修改块36可基于逻辑管线大小、物理管线大小减一和所请求包数目中的一或多者而产生经修改PSP值。修改块36可将经修改PSP值提供到写入块38。

[0229] 在一些实例中,由修改块36使用以产生经修改PSP值的修改功能可对应于图2到5中所示的伪码中表达的修改功能。例如,当执行原子保留写入操作时,修改块36可基于图2中所示的伪码的16到20行中所示的修改功能而产生经修改PSP值。作为另一实例,当执行原子提交写入操作时,修改块36可基于图3中所示的伪码的8到11行中所示的修改功能而产生经修改PSP值。

[0230] 作为另一个实例,当执行原子保留读取操作时,修改块36可基于图4中所示的伪码的16到20行中所示的修改功能而产生经修改PSP值。作为额外实例,当执行原子提交读取操作时,修改块36可基于图5中所示的伪码的8到12行中所示的修改功能而产生经修改PSP值。

[0231] 写入块38可基于管线地址执行写入操作。管线地址可为指向数据存储系统16中存储管线状态参数22的位置的存储器地址。为了执行写入操作,写入块38可将写入请求发出到数据存储系统16,且将经修改PSP值提供到数据存储系统16用于存储在数据存储系统16中。所述写入请求可包含管线地址。响应于接收到写入请求,数据存储系统16可将经修改PSP值存储在数据存储系统16中(例如,作为管线状态参数22的部分)。原子读取-修改-写入操作现在完成。

[0232] 原子操作数据路径32可经配置以执行读取-修改-写入操作作为非可中断操作集合。换句话说,原子操作数据路径32可经配置以执行当前读取-修改-写入操作而不允许其它读取-修改-写入操作中断当前读取-修改-写入操作的处理。以此方式,可提供基于硬件的原子读取-修改-写入操作用于可能请求这些操作的同时执行的线程和/或任务。

[0233] 根据本发明的一些方面,修改块36可针对不同管线状态参数执行不同修改操作作为单个原子读取-修改-写入操作的部分。例如,修改块36可执行基于第一修改功能产生第一管线状态参数的经修改值的第一修改操作,以及基于第二修改功能产生第二管线状态参数的经修改值的第二修改操作。所述第二修改功能可不同于所述第一修改功能。以此方式,可作为单个原子操作的部分以不同方式更新不同管线状态参数。

[0234] 例如,为了支持原子保留写入操作或原子保留读取操作,第一修改功能可用以将保留包计数参数递增针对管线保留所请求的包的数目(图2,第17行;图4,第17行),且第二修改功能可用以将保留计数参数递增一的恒定值(图2,第19行;图4,第19行)。作为另一实例,为了支持原子提交写入操作或原子提交读取操作,第一修改功能可用以将保留计数递减一的恒定值(图3,第8行;图5,第8行),第二修改功能可用以将包计数参数递增或递减等于保留计数参数的量(图3,第10行;图5,第11行),且第三修改功能可用以将保留包计数参数复位到零(图3,第11行;图5,第12行)。其它实例是可能的且在本发明的范围内。

[0235] 根据本发明的额外方面,修改块36可基于在原子操作的执行期间从存储器子系统(例如,数据存储系统16)获得的一或多个额外管线状态参数的值而产生一或多个管线状态参数的经修改值。例如,读取块34可在单个原子读取-修改-写入操作期间读取对应于第一管线状态参数的第一值和对应于第二管线状态参数的第二值。作为同一原子读取-修改-写入操作的部分,修改块36可基于对应于第一管线状态参数的所述第一值产生第二管线状态参数的经修改值。以此方式,可基于管线的当前状态在单个原子操作中更新管线状态参数。

[0236] 例如,为了支持原子提交操作,取决于所述提交操作是否为读取提交操作或写入提交操作可将包计数参数递减或递增保留包计数参数(图3,第10行;图5,第11行)。作为另一实例,为了支持原子提交读取操作,可基于保留包计数参数调整第一包偏移参数(图5,第10行)。其它实例是可能的且在本发明的范围内。

[0237] 根据本发明的额外方面,修改块36可基于不同管线状态参数值有条件地更新管线状态参数值。有条件地更新管线状态参数值可涉及基于不同管线状态参数值确定是否更新或不更新管线状态参数值。以此方式,可作为单个原子操作的部分基于管线的当前状态而更新或不更新管线状态参数。

[0238] 例如,为了支持原子保留写入操作或原子保留读取操作,修改块36可基于包计数参数和保留包计数参数有条件地更新保留包计数参数和保留计数参数(图2,16到20行;图4,16到20行)。作为另一实例,为了支持原子提交写入操作或原子提交读取操作,修改块36

可基于保留计数参数有条件地更新包计数参数和保留包计数参数(图3,9到12行;图5,9到13行)。其它实例是可能的且在本发明的范围内。

[0239] 图8是说明可用以实施本发明的基于硬件的原子操作的实例计算装置40的框图。计算装置40可包括个人计算机、台式计算机、膝上型计算机、计算机工作站、视频游戏平台或控制台、无线通信装置(例如,移动电话、蜂窝式电话、卫星电话和/或移动电话手持机)、陆线电话、因特网电话、手持式装置(例如,便携式视频游戏装置或个人数字助理(PDA))、个人音乐播放器、视频播放器、显示装置、电视机、电视机顶盒、服务器、中间网络装置、主机计算机,或处理和/或显示图形数据的任何其它类型的装置。

[0240] 如图8的实例中所说明,计算装置40包含用户输入接口42、CPU 44、GPU 46、存储器控制器48、存储器50、显示器接口52、显示器54和总线56。用户输入接口42、CPU 44、GPU 46、存储器控制器48和显示器接口52可使用总线56彼此通信。应注意图8中所示的不同组件之间的总线和通信接口的特定配置仅是示范性的,且具有相同或不同组件的计算装置和/或其它图形处理系统的其它配置可用以实施本发明的技术。

[0241] 用户输入接口42可允许一或多个用户输入装置(未图示)以通信方式耦合到计算装置40。所述用户输入装置可允许用户经由用户输入接口42对计算装置40提供输入。实例用户输入装置包含键盘、鼠标、轨迹球、麦克风、触摸垫、触敏或存在敏感显示器,或另一输入装置。在其中触敏或存在敏感显示器用作用户输入装置的实例中,用户输入接口42的全部或部分可与显示器54集成。

[0242] CPU 44可包括控制计算装置40的操作的通用或专用处理器。如图8中所示,CPU 44可执行主机程序58和GPU驱动程序60。主机程序58可充当用于将各种任务的执行卸载到GPU 46的计算应用程序的主机程序。在一些实例中,所述任务可为计算上密集的任务和/或包含高度并行度的任务,例如矩阵计算、信号处理计算、统计算法、分子模型化应用、财务应用、医疗成像、密码分析应用等。在CPU 44上执行的主机程序58可为任何类型的软件程序,包含例如通用计算应用程序、并行处理应用程序、电子数据表应用程序、文字处理器应用程序、电子邮件应用程序、媒体播放器应用程序、视频游戏应用程序、图形用户接口应用程序、操作系统、图形应用程序或任何其它类型的应用程序。

[0243] GPU驱动程序60可从主机程序58接收指令,且致使GPU 46服务于所述指令。举例来说,GPU驱动器60可调配一或多个命令,将命令放置到存储器50中,且指示GPU46执行所述命令。在一些实例中,GPU驱动程序60可包含支持API的运行时间程序,所述API允许主机程序58产生管线且允许在计算装置(例如,GPU 46)上执行的任务通过对管线写入数据和从管线读取数据而彼此通信。

[0244] 虽然图8中未图示,但CPU 44还可执行编译器。所述编译器可接收用于一或多个不同类型的任务(例如,内核和/或计算着色器程序)的源代码且产生用于所述任务的经编译代码。GPU驱动程序60可将用于所述任务中的一或多者的经编译代码加载到GPU 46上(例如,加载到包含在GPU 46的计算单元中的指令存储中)用于由GPU 46执行。

[0245] 在一些实例中,用于主机程序58的源代码可符合GPGPU API、并行编程API和/或异构计算平台API,例如OpenCL™ API。在其它实例中,用于经编译的任务的源代码可符合GPGPU API、并行编程API和/或异构计算平台API,例如OpenCL™ C API。

[0246] GPU 46可经配置以执行由CPU 44发出到GPU 46的命令。由GPU 46执行的命令可包

含通用计算命令、任务执行命令(例如,内核执行命令)、存储器转移命令等。

[0247] GPU 46可经配置以执行用于在CPU 44上执行的应用程序的通用计算。例如,当在CPU 44上执行的主机程序58决定将计算任务卸载到GPU 46时,CPU 44可将通用计算数据提供到GPU 46,且将一或多个通用计算命令发出到GPU 46。所述通用计算命令可包含例如内核执行命令、存储器转移命令等。在一些实例中,CPU 44可通过将命令和数据写入到可由GPU 46存取的存储器50而将命令和通用计算数据提供到GPU 46。

[0248] 如图8中所示, GPU 46包含原子操作单元14和计算单元12。计算单元12可执行一或多个任务。在一些实例中,计算单元12可同时执行多个任务。在其它实例中,计算单元12中的一或者可同时执行任务的多个实例。由计算单元12中的一者执行的任务的实例可被称为线程和/或工作项目。任务可替代地称为内核。原子操作单元14可经配置以从计算单元12接收一或多个不同类型的原子操作命令,且执行对应于如本发明中所描述的不同类型原子操作命令中的每一者的基于硬件的原子操作。

[0249] 在一些实例中,除执行通用计算命令之外, GPU 46还可经配置以执行图形操作以将一或多个图形基元再现到显示器54。在此等实例中,当在CPU 44上执行的软件应用程序中的一者需要图形处理时,CPU 44可将图形数据提供到GPU 46且将一或多个图形命令发出到GPU 46。所述图形命令可包含例如绘制调用命令、GPU状态编程命令、存储器传送命令、传图命令等。所述图形数据可包含顶点缓冲器、纹理数据、表面数据等。在一些实例中,CPU 44可通过将命令和图形数据写入到可由GPU 46存取的存储器50而将命令和图形数据提供到GPU 46。

[0250] 在一些情况下, GPU 46可内置有高度并行的结构,其提供比CPU 44更有效的对向量操作的处理。例如, GPU 46可包含经配置以并行方式对多个顶点、控制点、像素和/或其它数据操作的多个处理元件。GPU 46的高度并行性质在一些情况下可允许GPU 46比CPU 44更快地处理包含高并行度的任务。另外, GPU 46的高度并行性质在一些实例中可允许GPU 46比使用CPU 44再现图像更快地将图形图像(例如, GUI和二维(2D)和/或三维(3D)图形场景)再现到显示器54上。

[0251] 在一些情况下, GPU 46可集成到计算装置40的母板中。在其它情况下, GPU 46可存在于安装在计算装置40的母板中的端口中的图形卡上或另外可并入在经配置以与计算装置40交互操作的外围装置内。在进一步例子中, GPU 46可与CPU 44位于同一微芯片上从而形成芯片上系统(SoC)。GPU 46可包含一或多个处理器,例如一或多个微处理器、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、数字信号处理器(DSP)或其它等效的集成或离散逻辑电路。

[0252] 在一些实例中, GPU 46可包含GPU高速缓冲存储器,其可为全部或一部分存储器50提供高速缓冲存储服务。在此些实例中, GPU 46可使用本地存储装置而不是芯片外存储器, 使用所述高速缓冲存储器来本地处理数据。此通过减少在每一读取和写入命令期间对于GPU 46经由总线56存取存储器50(其可经历繁重总线业务)的需要而允许GPU46以更有效方式操作。然而,在一些实例中, GPU 46可不包含单独的高速缓冲存储器,而是经由总线56利用存储器50。所述GPU高速缓冲存储器可包含一或多个易失性或非易失性存储器或存储装置,例如,随机存取存储器(RAM)、静态RAM(SRAM)、动态RAM(DRAM)等。

[0253] 存储器控制器48促进数据进出存储器50的传送。例如,存储器控制器48可接收存

储器读出和写入命令,以及相对于存储器50服务于此类命令,以便为计算装置40中的组件提供存储器服务。存储器控制器48以通信方式耦合到存储器50。虽然存储器控制器48在图8的实例计算装置40中被说明为与CPU 44和存储器50均分离的处理模块,但是在其它实例中,存储器控制器48中的一些或全部功能性可在CPU 44和存储器50中的一者或两者上实施。

[0254] 存储器50可存储可由CPU 44存取用于执行的程序模块和/或指令和/或供在CPU 44上执行的程序使用的数据。例如,存储器50可存储与在CPU 44执行的应用程序相关联的程序代码和图形数据。存储器50可另外存储供计算装置40的其它组件使用和/或由其产生的信息。例如,存储器50可充当用于GPU 46的装置存储器且可存储将由GPU 46操作的数据以及产生于由GPU 46执行的操作的数据。例如,存储器50可存储缓冲器对象、管线数据或类似物的任何组合。另外,存储器50可存储用于由GPU 46处理的命令流(例如,命令队列)。存储器50可包含一或多个易失性或非易失性存储器或存储装置,例如随机存取存储器(RAM)、静态RAM(SRAM)、动态RAM(DRAM)、只读存储器(ROM)、可擦除可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)、快闪存储器、磁性数据媒体或光学存储媒体。在一些实例中,存储器50可对应于图1中所示的数据存储系统16的全部或部分。

[0255] CPU 44和/或GPU 46可在存储器50内分配的帧缓冲器中存储光栅化图像数据。显示器接口52可从所述帧缓冲器检索所述数据并配置显示器54以显示由所述光栅化图像数据表示的所述图像。在一些实例中,显示器接口52可包含数/模转换器(DAC),其经配置以将从帧缓冲器检索的数字值转换为可由显示器54消耗的模拟信号。在其它实例中,显示器接口52可将数字值直接传递到显示器54以进行处理。

[0256] 显示器54可包含监视器、电视机、投影装置、液晶显示器(LCD)、等离子体显示器面板、发光二极管(LED)阵列、阴极射线管(CRT)显示器、电子纸、表面传导电子发射显示器(SED)、激光电视机显示器、纳米晶体显示器或另一类型的显示单元。显示器54可集成在计算装置40内。举例来说,显示器54可为移动电话手持机或平板计算机的屏幕。替代地,显示器54可为经由有线或无线通信链路而耦合到计算机装置2的独立装置。举例来说,显示器54可为经由电缆或无线链路而连接到个人计算机的计算机监视器或平板显示器。

[0257] 总线56可使用包含第一、第二和第三代总线结构和协议、共享总线结构和协议、点对点总线结构和协议、单向总线结构和协议以及双向总线结构和协议的总线结构和总线协议的任何组合来实现。可用以实现总线56的不同总线结构和协议的实例包含例如超传输总线、InfiniBand总线、高级图形端口总线、外围部件互连标准(PCI)总线、PCI高速总线、高级微控制器总线架构(AMBA)、高级高性能总线(AHB)、AMBA高级外围总线(APB)以及AMBA高级eXtensible接口(AXI)总线。也可以使用其它类型的总线结构和协议。

[0258] 根据本发明,GPU 46(例如,原子操作单元14)可执行修改指示管线状态的两个或更多个管线状态参数的基于硬件的原子读取-修改-写入操作。所述管线可对应于用于支持任务间通信的先入先出(FIFO)组织的数据缓冲器。以原子方式使用基于硬件的原子读取-修改-写入操作来修改两个或更多个管线状态参数值可允许同时执行的任务和/或线程修改管线的状态而不需要使用锁和/或关键代码区段。

[0259] 在一些实例中,为了执行基于硬件的原子读取-修改-写入操作,GPU 46(例如,原子操作单元14)可执行非可中断操作集合。所述非可中断操作集合可包含从存储器子系统

(例如,存储器50和/或包含在GPU 46中的高速缓冲存储器)读取对应于管线状态参数的值的操作、产生管线状态参数的经修改值的操作,以及将管线状态参数的经修改值写入到存储器子系统的操作。

[0260] 现将描述计算装置40的实例操作。主机程序58可发出指令到GPU驱动程序60以产生管线对象。所述指令可指定将产生的管线的逻辑大小。管线的逻辑大小可指代所述管线中将包含的逻辑包条目的数目。

[0261] 响应于接收到产生管线对象的指令,GPU驱动程序60可分配存储器50中的存储器空间用于所述管线对象。所述分配的存储器空间可包含为如图1中所说明的管线状态参数22和管线数据缓冲器24分配的存储器空间。在一些实例中,为管线分配的物理包条目的数目可不一定等于由主机程序58指定用于所述管线的逻辑包条目的数目。例如,GPU驱动程序60可为管线分配存储器以使得物理包条目的数目是大于或等于由主机程序58所请求的逻辑包条目的数目的第一个二的幂。在分配管线之后,GPU驱动程序60可返回识别刚产生的管线的句柄。

[0262] 主机程序58可基于将由GPU 46执行的任务而产生任务对象(例如,内核对象)。主机程序58可设定用于任务对象的自变量中的一者以对应于先前产生的管线(例如,将所述自变量设定为等于用于管线的句柄)。主机程序58可将致使GPU 46执行任务对象的指令发出到GPU驱动程序60。

[0263] 响应于接收到致使执行任务对象的指令,GPU驱动程序60可致使GPU 46在计算单元12中的一或者上执行对应于所述任务对象的任务。例如,GPU驱动程序60可在用于GPU 46的命令队列中放置致使GPU 46执行所述任务的命令。

[0264] 在一些实例中,GPU驱动程序60可为GPU 46提供指定管线的特性的一或多个参数和/或自变量。例如,GPU驱动程序60可为GPU 46提供管线的管线地址、管线的物理管线大小减一和/或管线的逻辑管线大小中的一或者。在一些实例中,GPU驱动程序60和/或GPU 46可致使计算单元12的正在执行任务的恒定存储器和/或恒定寄存器被填充指定管线的特性的参数。

[0265] 响应于接收到指示GPU 46执行任务的命令,GPU 46将所述任务指派到计算单元12中的一或者。计算单元12的被指派执行所述任务的所述一或者执行所述任务。在一些实例中,执行任务可涉及并行地执行多个线程(即,任务的实例)。

[0266] 在计算单元12上同时执行的任务和/或线程将一或多个原子操作命令发出到原子操作单元14。响应于接收到原子操作命令,原子操作单元14执行如本发明中所描述的一或多个基于硬件的原子操作。任务和/或线程的执行继续直到完成。

[0267] 图9是说明根据本发明可在GPU 46或另一计算装置中使用的实例计算单元12A的框图。计算单元12A可经配置以基于用于任务的对计算单元12A可存取的经编译程序代码而执行一或多个不同类型的任务。计算单元12A包含控制单元72、指令存储装置74、处理元件76A到76H(共同地为“处理元件76”)、寄存器78、恒定存储器80和局部存储器82。

[0268] 指令存储装置74经配置以存储装置将由计算单元12A执行的一或多个任务(例如,内核)的程序指令的全部或部分。指令存储装置74可为任何类型的存储单元,包含例如易失性存储器、非易失性存储器、高速缓冲存储器、随机存取存储器(RAM)、静态RAM(SRAM)、动态RAM(DRAM)等。当指令存储装置74是高速缓冲存储器时,指令存储装置74可高速缓冲存储存

储于计算单元12A外部的存储器中的程序指令。虽然将指令存储装置74说明为在计算单元12A的内部,但在其它实例中,指令存储装置74可在计算单元12A外部。

[0269] 处理元件76经配置以执行任务程序的线程。处理元件76中的每一者可执行不同线程。线程可指代相对于对线程特定的数据项目执行的任务的实例。因此,处理元件76中的每一者可称为相对于潜在地不同的数据项目执行任务的实例。在共同时间点在处理元件76A到76H上并行地执行的线程的集合可被称为线程波。

[0270] 在图9的实例计算单元12A中,处理元件76可为单指令多数据(SIMD)处理元件。SIMD处理元件指代当经激活时经配置以相对于不同数据同时执行同一指令的处理元件。这可允许处理元件76相对于不同数据项目并行地执行任务的多个线程。在某些情况下,处理元件76中的每一者可基于指向包含于指令存储装置74中的指令的共同程序计数器而执行任务的指令。

[0271] 如果处理元件76中的一或者经去活,那么这些处理元件76在给定指令周期中并不执行程序指令。在某些情况下,控制单元72可去活处理元件76中的一或者以实施条件性分支指令,其中分支条件对于一些线程得到满足且对于其它线程不得到满足。

[0272] 在一些实例中,处理元件76中的每一者可包含和/或对应于算术逻辑单元(ALU)。在其它实例中,处理元件76中的每一者可实施ALU功能性。ALU功能性可包含加法、减法、乘法等。在额外实例中,处理元件76中的每一者可为标量ALU或向量ALU。标量ALU可对标量数据项目操作,且向量ALU可对向量数据项目操作。标量数据项可包含对应于标量的单一量的单个值。向量数据项目可包含对应于向量的多个分量的多个值。

[0273] 处理元件76中的每一者可从指令存储装置74读取指令和/或从寄存器78、恒定存储器80、局部存储器82和外部存储器中的一或者读取数据项目。处理元件76中的每一者可将输出数据写入到寄存器78、局部存储器82和外部存储器中的一或者。

[0274] 寄存器78可动态地分配到各种处理元件76。在某些情况下,寄存器78中的一些或全部可充当用于在计算单元12A上执行的各种线程的输入寄存器和/或输出寄存器。输入寄存器可指代存储用于任务程序的输入数据项目的寄存器,且输出寄存器可指代存储用于任务程序的输出数据项目的寄存器。

[0275] 恒定存储器80可存储由在计算单元12A上执行的任务使用的恒定值。恒定存储器80可为任何类型的存储器,包含例如易失性存储器、随机存取存储器(RAM)、静态RAM(SRAM)、动态RAM(DRAM)、电可擦除可编程只读存储器(EEPROM)等。

[0276] 局部存储器82可提供用于由在计算单元12A上执行的任务使用的存储器存储。局部存储器82可为任何类型的存储器,包含例如易失性存储器、随机存取存储器(RAM)、静态RAM(SRAM)、动态RAM(DRAM)等。在一些实例中,用于局部存储器82的地址空间对包含在计算单元12A中的处理元件76可为局部的。换句话说, GPU 46的其它计算单元和/或其它部分不能够直接存取局部存储器82。类似地,主机装置不能够直接存取局部存储器82。在一些实例中,局部存储器82可与计算单元12A和/或GPU 46实施在同一芯片上。

[0277] 现将描述计算单元12A的实例操作。GPU 46可对计算单元12A指派任务用于执行。GPU 46可将用于任务的编译代码存储在指令存储装置74中。GPU 46可将用于任务的恒定参数存储在恒定存储器80中,且将用于任务的输入参数存储在寄存器78中。处理元件76并行地执行任务的多个实例(即,线程)。在处理元件76上执行的线程中的一或者发出本发明

中所描述的原子操作命令中的一或多者。

[0278] 为了发出原子操作命令,处理元件76A可指定含有用于原子操作命令的输入参数的寄存器78中的一或多者。处理元件76A可将原子操作命令发出到原子操作单元14。

[0279] 在一些实例中,对于原子提交写入和原子提交读取命令,原子操作单元14不提供返回值。对于原子保留写入和原子保留读取命令,可提供返回值。例如,原子操作单元14可返回第一包偏移参数、包计数参数和保留包计数参数。在一些实例中,寄存器78可包含一或多个四重寄存器群组,其中每一四重寄存器群组包含四个寄存器,且所述四个寄存器的数据可作为单个高速缓冲存储器和/或存储器事务的部分而从存储器读取和/或存储在存储器中。在此些实例中,由原子保留写入操作和/或原子保留读取操作返回的三个管线状态参数在一些实例中可在寄存器78的四重寄存器群组内的三个连续寄存器中返回。以此方式,可作为单个高速缓冲存储器和/或存储器事务的部分由计算单元12A获取所述三个管线状态参数。

[0280] 图10是说明图1的计算系统10的一部分的框图,其中更详细说明实例GPU 46和存储器配置。图10中说明的配置可用以实施图1的计算系统10和/或图8的计算装置40。

[0281] 如图10中所示, GPU 46包含高速缓冲存储器单元84。高速缓冲存储器单元84可为GPU 46中的计算单元12提供高速缓冲存储服务。高速缓冲存储器单元84包含命令队列26、高速缓冲存储器控制单元86、原子操作数据路径30和高速缓冲存储器存储单元88。高速缓冲存储器存储单元88可包含高速缓冲存储的管线状态参数90和高速缓冲存储的管线数据缓冲器92中的一者或两者。高速缓冲存储的管线状态参数90可为管线状态参数22的高速缓冲存储的副本,且高速缓冲存储的管线数据缓冲器92可为管线数据缓冲器24的高速缓冲存储的副本。在一些实例中,因为高速缓冲存储的副本是经高速缓冲存储的,所以这些副本可能不完整和/或是其相应源数据的部分副本。

[0282] 高速缓冲存储器单元84和存储器50可形成存储器子系统,且高速缓冲存储器存储单元88和存储器50可形成存储器子系统中的数据存储系统。例如,所述数据存储系统可对应于图1中的数据存储系统16。

[0283] 命令队列26可经配置以存储从计算单元12接收的存储器存取命令和从计算单元12接收的原子操作命令。高速缓冲存储器控制单元86可处理所述存储器存取命令和原子操作命令。在一些实例中,高速缓冲存储器控制单元86可循序地处理所述命令以使得每次处理一个命令。

[0284] 对于存储器存取命令,高速缓冲存储器控制单元86可将对应读取请求和写入请求发出到高速缓冲存储器存储单元88和存储器50中的一者或两者。对于原子操作命令,高速缓冲存储器控制单元86可致使原子操作数据路径30中的一或多者执行对应于所述命令的基于硬件的原子操作。由原子操作数据路径30实施的基于硬件的原子操作可将读取请求和写入请求发出到高速缓冲存储器存储单元88和存储器50中的一者或两者以执行原子操作。

[0285] 如图10中所示,由原子操作单元14执行的基于硬件的原子操作(图6)在一些实例中可并入到GPU 46的高速缓冲存储器单元84中。将由原子操作单元14执行的基于硬件的原子操作并入到GPU 46的高速缓冲存储器单元84中可允许执行类似功能性的硬件的某个部分在原子操作单元14与高速缓冲存储器单元84之间共享。以此方式,更紧凑的电路可以用于实施本发明的基于硬件的原子操作。

[0286] 图11是说明用于执行根据本发明的基于硬件的原子操作的实例技术的流程图。图11中说明的实例技术是相对于计算系统10的原子操作单元14而描述。在一些实例中，原子操作单元14可包含在处理器和/或处理单元中，例如GPU(例如，GPU 46)、CPU、DSP等。

[0287] 原子操作单元14从计算单元12A接收原子操作命令(100)。原子操作单元14执行修改指示管线的状态的两个或更多个管线状态参数的基于硬件的原子读取-修改-写入操作(102)。管线可对应于用于支持帧间线程通信的FIFO组织的数据缓冲器。

[0288] 读取-修改-写入操作在所述操作从开始到结束执行而无来自可由与请求读取-修改-写入操作的任务或线程同时执行的其它任务和/或线程请求的任何其它读取-修改-写入操作的中断的意义上可为原子的。换句话说，在原子读取-修改-写入操作的执行期间，没有其它线程和/或任务可能够读取或修改由当前执行的读取-修改-写入操作修改的管线状态参数。以此方式，在其中两个或更多个任务和/或线程同时尝试修改管线的管线状态参数的情况下可避免竞态条件。

[0289] 原子读取-修改-写入操作在所述原子读取-修改-写入操作可作为单个不可分操作由硬件单元实施的意义上可为基于硬件的，所述单个不可分操作可由以通信方式耦合到所述硬件单元的线程和/或任务调用。所述单个不可分操作在所述硬件单元从起始到结束执行操作而不中断所述操作以执行可由与请求读取-修改-写入操作的任务或线程同时执行的其它任务和/或线程请求的其它读取-修改-写入操作的意义上可为不可分的。例如，如果硬件单元接收到执行原子读取-修改-写入操作的多个请求，那么所述硬件单元可针对所述请求中的每一者循序执行原子读取-修改-写入操作以使得每次执行一个原子读取-修改-写入操作，且使得每一原子读取-修改-写入操作的执行是在开始执行另一原子读取-修改-写入操作之前完成。通过使用基于硬件的原子读取-修改-写入操作，可通过同时执行任务和/或线程而不需要任务和/或线程使用互斥锁来修改多个管线状态参数。

[0290] 在一些实例中，原子操作单元14可从并行地执行的多个线程接收执行基于硬件的原子读取-修改-写入操作的多个请求(即，命令)。在此些实例中，原子操作单元14可响应于接收到执行基于硬件的原子读取-修改-写入操作的所述多个请求而循序地执行多个基于硬件的原子读取-修改-写入操作。所述基于硬件的原子读取-修改-写入操作中的每一者可响应于所述多个请求中的相应一个而执行。循序地执行所述多个基于硬件的原子读取-修改-写入操作可涉及执行所述基于硬件的原子读取-修改-写入操作以使得与来自第一线程的第一请求相关联的全部管线状态参数修改是在执行与来自第二线程的第二请求相关联的管线状态参数修改中的任一者之前完成。

[0291] 例如，原子操作单元14可接收来自第一线程的执行原子操作的第一请求。响应于接收到第一请求，原子操作单元14可执行第一基于硬件的原子读取-修改-写入操作以服务于所述第一请求。在完成第一基于硬件的原子读取-修改-写入操作之前，原子操作单元14可从第二线程接收执行原子操作的第二请求。响应于接收到第二请求，原子操作单元14可等待开始用于服务于所述第二请求的第二基于硬件的原子读取-修改-写入操作的执行直到第一基于硬件的原子读取-修改-写入操作的执行完成为止。在第一基于硬件的原子读取-修改-写入操作的执行完成之后，原子操作单元14可执行修改两个或更多个管线状态参数的第二基于硬件的原子读取-修改-写入操作以服务于所述第二请求。

[0292] 以上述方式循序地执行基于硬件的原子读取-修改-写入操作可允许原子操作单

元14确保操作的原子性。以此方式,可在并行计算系统中提供基于管线的任务间通信而无与使用基于软件的互斥技术用于修改管线状态参数相关联的性能缺陷和/或电力消耗缺陷。

[0293] 在一些实例中,管线状态参数可包含指示管线的保留状态的一或多个管线保留状态参数。例如,保留状态参数可包含指示多少未提交保留当前对于管线待决的参数以及指示多少包条目当前保留在管线中的参数中的至少一者。在其它实例中,管线状态参数可包含指示多少包当前存储在管线中的参数、指示存储管线的起始包的包条目的参数、指示多少未提交保留当前对于管线待决的参数,以及指示多少包条目当前保留在管线中的参数。

[0294] 图12是说明用于执行根据本发明的基于硬件的原子操作的另一实例技术的流程图。图12中说明的实例技术是相对于计算系统10的原子操作单元14而描述。在一些实例中,原子操作单元14可包含在处理器和/或处理单元中,例如GPU(例如,GPU 46)、CPU、DSP等。在一些实例中,图13中说明的实例技术可实施图11中说明的过程框102。

[0295] 原子操作单元14执行从存储器子系统读取对应于两个或更多个管线状态参数的值的读取操作(104)。原子操作单元14执行产生所述两个或更多个管线状态参数的经修改值的修改操作(106)。原子操作单元14执行将管线状态参数的经修改值写入到存储器子系统的写入操作(108)。

[0296] 在一些实例中,存储器子系统可包含数据存储系统16。在此些实例中,将所述值写入到存储器子系统以及从存储器子系统读取所述值可涉及将值写入到包含在存储器子系统中的数据存储系统16以及从所述数据存储系统16读取值。

[0297] 在一些实例中,过程方框104、106和108可作为非可中断操作集合的部分而执行。在操作是从起始到结束执行而不中断执行以执行可与其它读取-修改-写入操作相关联的其它操作(例如,与请求当前执行的读取-修改-写入操作的任务或线程同时执行的任务和/或线程所请求的其它读取-修改写入操作)的意义上所述操作可为非可中断的。致使硬件单元以此方式执行非可中断读取-修改-写入序列允许所述硬件单元针对可请求此些操作的同时执行的线程和/或任务实施原子读取-修改-写入操作。

[0298] 图13是说明用于执行根据本发明的原子保留写入操作和原子保留读取操作的实例技术的流程图。图13中说明的实例技术是相对于计算系统10的原子操作单元14而描述。在一些实例中,原子操作单元14可包含在处理器和/或处理单元中,例如GPU(例如,GPU 46)、CPU、DSP等。在一些实例中,图13中说明的实例技术可实施图12中说明的实例技术。

[0299] 原子操作单元14执行从存储器子系统读取管线状态参数(PSP)值的读取操作(110)。在执行读取操作之后,原子操作单元14执行将管线状态参数值中的一或者返回到请求线程和/或任务的返回操作(112)。

[0300] 在执行读取操作之后,原子操作单元14执行产生管线状态参数的经修改值的修改操作(框114到118)。为了执行修改操作,原子操作单元14执行基于管线状态参数中的一或者而确定管线是否具有足够可用条目来支持保留请求的操作(114)。响应于确定管线具有足够可用条目来支持保留请求,原子操作单元14执行产生管线状态参数的经修改值以使得所述经修改值反映已针对所述管线做出保留的操作(116)。

[0301] 响应于确定所述管线不具有足够可用条目来支持保留请求,原子操作单元14可不执行产生管线状态参数的经修改值以使得所述经修改值反映已针对所述管线做出保留的

操作(118)。换句话说，原子操作单元14可产生管线状态参数的经修改值以使得所述经修改值等于管线状态参数的先前值(即，使得所述经修改值反映尚未针对所述管线做出保留)。

[0302] 在其中执行原子保留写入操作的实例中，为了确定管线是否具有足够可用条目来支持保留请求，原子操作单元14可基于管线状态参数中的一或者而确定所述管线是否具有足够未提交和未保留条目来支持写入保留请求。在其中执行原子保留读取操作的实例中，为了确定所述管线是否具有足够可用条目来支持保留请求，原子操作单元14可确定所述管线是否具有其中存储包数据以支持所述读取保留请求的足够未读和未保留包条目。

[0303] 在产生经修改值之后，原子操作单元14执行将管线状态参数的经修改值写入到存储器子系统的写入操作(108)。保留操作现在完成。

[0304] 在一些实例中，保留操作可为保留写入操作。在此些实例中，决策框114在一些实例中可通过评估图2的第16行上所示的表达式来实施。另外，为了如过程框116中所描述产生反映已针对管线做出保留的经修改管线状态参数值，原子操作单元14可基于图2的17到20行中指定的等式而更新管线状态参数值。

[0305] 在其它实例中，保留操作可为保留读取操作。在此些实例中，决策框114在一些实例中可通过评估图4的第16行上展示的表达式来实施。另外，为了如过程框116中所描述产生反映已针对管线做出保留的经修改管线状态参数值，原子操作单元14可基于图4的17到10行中指定的等式而更新管线状态参数值。

[0306] 图14是说明用于执行根据本发明的原子提交写入操作和原子提交读取操作的实例技术的流程图。图14中说明的实例技术是相对于计算系统10的原子操作单元14而描述。在一些实例中，原子操作单元14可包含在处理器和/或处理单元中，例如GPU(例如，GPU 46)、CPU、DSP等。在一些实例中，图14中说明的实例技术可实施图12中说明的实例技术。

[0307] 原子操作单元14执行从存储器子系统读取管线状态参数(PSP)值的读取操作(122)。在执行读取操作之后，原子操作单元14执行产生管线状态参数的经修改值的修改操作(框124到130)。为了执行修改操作，原子操作单元14可产生第一管线状态参数(例如，保留计数参数)的经修改值以反映已提交保留(124)。换句话说，原子操作单元14可执行产生第一管线状态参数的经修改值的操作以使得第一管线状态参数的经修改值反映所述管线具有相对于在执行所述提交操作之前所述管线具有的未提交保留的数目来说少一个的未提交保留。

[0308] 原子操作单元14执行基于管线状态参数中的一或者确定是否已经提交用于管线的全部保留的操作(126)。响应于确定已经提交用于管线的全部保留，原子操作单元14执行产生管线状态参数的一或多个经修改值以使得所述经修改值反映条目已经提交到管线的操作(128)。

[0309] 响应于确定尚未提交用于管线的全部保留，原子操作单元14不执行产生管线状态参数的一或多个经修改值以使得所述经修改值反映所提交条目已经提交到管线的操作(130)。换句话说，原子操作单元14可产生除第一管线状态参数(例如，保留计数参数)外的管线状态参数的经修改值以使得所述经修改值等于管线状态参数的先前值。

[0310] 在其中执行原子提交写入操作的实例中，修改管线状态参数值以反映条目已经提交到管线可涉及修改管线状态参数值以反映所提交条目已经放置到管线中且可用于消费者任务。在其中执行原子提交读取操作的实例中，修改管线状态参数值以反映条目已经提

交到管线可涉及修改管线状态参数值以反映所提交条目已经从管线移除。

[0311] 在产生经修改值之后,原子操作单元14执行将管线状态参数的经修改值写入到存储器子系统的写入操作(132)。提交操作现在完成。

[0312] 在一些实例中,提交操作可为提交写入操作。在此些实例中,过程框124在一些实例中可基于图3的第8行上所示的等式而实施。另外,决策框126在一些实例中可通过评估图3的第9行上所示的表达式而实施。进一步,为了如过程框128中所描述产生反映条目已经提交到管线的经修改管线状态参数值,原子操作单元14可基于图3的10到11行中指定的等式而更新管线状态参数值。

[0313] 在其它实例中,提交操作可为提交读取操作。在此些实例中,过程框124在一些实例中可基于图5的第8行上所示的等式而实施。另外,决策框126在一些实例中可通过评估图5的第9行上所示的表达式而实施。进一步,为了如过程框128中所描述产生反映条目已经提交到管线的经修改管线状态参数值,原子操作单元14可基于图5的10到12行中指定的等式而更新管线状态参数值。

[0314] 在一些实例中,本发明的技术可实施用于OpenCL™ 2.0管线的相对高效解决方案。所述技术在一些实例中可使用专用原子功能来实施OpenCL™ 2.0管线。

[0315] OpenCL™ 2.0管线充当FIFO,其可经由保留特征由多个产生者/多个消费者使用。用于存取管线的语义可包含调用reserve(cnt)函数,其中cnt指示用于保留的包的所请求数目。如果reserve(cnt)函数成功,那么reserve(cnt)函数可返回保留识别值(即, reserve_id)。用于存取管线的语义可进一步包含,如果reserve(cnt)函数成功,那么使用reserved_id以及从0到cnt-1的索引调用读取管线命令。所述索引可指示将从管线读取哪些包条目。用于存取管线的语义可进一步包含commit(reserve_id)命令,其指示读取管线完成以及包可从管线移除。

[0316] 维持管线的状态可能需要对多个计数器的同时(即,原子)修改。这可使用互斥锁来实现,所述互斥锁又可使用比较-交换原子操作来实施。然而使用互斥锁可为低效的,因为大量工作项目可能竞争所述锁。这可导致对于成功获取需要多次尝试,进而浪费电力。

[0317] 本发明的技术在一些实例中可使用硬件原子操作,其可经设计以使用每操作单个硬件指令而支持管线保留、提交和查询操作。在一些实例中,本发明中所描述的原子操作可在实施全局GPU原子的GPU管线的一部分中实施。在其它实例中,本发明中所描述的原子操作可在实施方案支持共享虚拟存储器的情况下允许在主机CPU与GPU之间使用管线。

[0318] 在一些实例中,本发明的技术可执行GPU的着色器块内的内建式管线功能,且用于着色器块的指令集架构(ISA)可包含本发明中所描述的原子操作命令中的一或者者。在其它实例中,本发明的技术可使用专用硬件电路来实施原子操作。

[0319] 在一些实例中,本发明的技术可将专用原子添加到GPU的统一高速缓冲存储器块(例如,L2高速缓冲存储器块)。在一些实例中,可以额外ISA指令存取所述专用原子。在一些实例中,可使用保留机制(例如,reserve_read、read、commit_read或reserve_write、write、commit_write)来存取管线。

[0320] 在一些实例中,管线的管线状态可由四个32位数表示:(1)第一包偏移;(2)包计数;(3)保留包计数;以及(4)保留计数。在此些实例中,这些32位数在一些实例中可对应于本发明中所描述的管线状态参数。在此些实例中,所述四个32位数(例如,管线状态参数)可

位于具有128位对准的128位块内,以允许以单个存储器事务的读取/写入。

[0321] 在一些实例中,管线的所分配大小可为二的幂。然而,管线的实际深度可小于所分配大小。这可能是由于OpenCL™ API的灵活性。

[0322] 本发明的技术在一些实例中可为OpenCL™管线保留、提交和查询提供支持。OpenCL™管线可用于读取(即,输入管线)或用于写入(即,输出管线)。在一些实例中,内核可从给定管线进行读取或对其进行写入,但不是这两者。

[0323] 在一些实例中,本发明的技术可在着色器处理器、流处理器和/或GPU的统一高速缓冲存储器中实施。在一些实例中,本发明的技术可允许支持每内核无限数目的管线(例如,仅受到用以保持管线状态的可用存储器和恒定寄存器限制)。在其它实例中,保持存储器中的管线状态的特征可与OpenCL™ 2.0的共享虚拟存储器特征结合以含有跨越主机/装置或跨越多个装置的管线。

[0324] 在一些实例中,管线可为存储经组织为FIFO的数据的存储器对象。管线可替代地称为管线对象。在某些情况下,可使用从管线进行读取和/或对管线进行写入的内建式功能来存取管线对象。在另外情况中,管线对象不可从主机装置存取。在一些实例中,管线对象可囊封以下信息:以字节计的包大小、以包计的管线的最大容量、指示当前管线中的包数目信息,以及包含在管线中的数据包。

[0325] 在一些实例中,管线存储器对象可概念上对应于数据项目的有序序列。例如,管线可具有两个端点:数据项目插入其中的写入端点,以及从其移除数据项目的读取端点。在一些实例中,在任何一个时间,可将仅一个内核实例(例如,任务)写入到管线中,且可从管线读取仅一个内核实例(例如,任务)。在一些实例中,管线可支持产生者-消费者设计模式,其中一个内核实例连接到管线的写入端点(作为产生者),而另一内核实例连接到管线的读取端点(作为消费者)。

[0326] 在某些情况下,管线可为一类存储器对象。存储器对象在一些实例中可包含全局存储器(例如,系统存储器)的内容。在一些实例中,存储器对象可指代到全局存储器的参考计数区的句柄。

[0327] 本发明中所描述的技术可至少部分实施于硬件、软件、固件或其任何组合中。举例来说,所描述技术的各种方面可实施于一或多个处理器中,包含一或多个微处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA),或任何其它等效集成或离散逻辑电路,以及此类组件的任何组合。术语“处理器”或“处理电路”可大体上指前述逻辑电路中的任一者(单独或结合其它逻辑电路)或例如执行处理的离散硬件等任何其它等效电路。

[0328] 此硬件、软件和固件可实施于同一装置内或单独装置内以支持本发明中所描述的各种操作和功能。另外,所描述的单元、模块或组件中的任一者可一起或单独作为离散但可互操作逻辑装置而实施。将不同特征描绘为模块或单元意图强调不同功能方面且未必暗示此类模块或单元必须由单独硬件或软件组件实现。而是,与一或多个模块或单元相关联的功能性可由单独硬件、固件及/或软件组件执行,或集成到共用或单独硬件或软件组件内。

[0329] 本发明中所描述的技术也可存储、体现或编码于计算机可读媒体(例如,存储指令的计算机可读存储媒体)中。嵌入或编码于计算机可读媒体中的指令可引起一或多个处理器执行本文中所描述的技术(例如,当由一或多个处理器执行指令时)。在一些实例中,计算

机可读媒体可为非暂时性计算机可读存储媒体。计算机可读存储媒体可包含随机存取存储器(RAM)、只读存储器(ROM)、可编程只读存储(PROM)、可擦除可编程只读存储器(EPROM)、电子可擦除可编程只读存储器(EEPROM)、快闪存储器、硬盘、CD-ROM、软盘、卡盒、磁性媒体、光学媒体或其它有形计算机可读存储媒体。

[0330] 计算机可读媒体可包含计算机可读存储媒体，其对应于例如上文所列的有形存储媒体等有形存储媒体。计算机可读媒体也可包括通信媒体，其包含促进计算机程序从一个地点到另一地点的传送(例如，根据通信协议)的任何媒体。以此方式，短语“计算机可读媒体”大体上可对应于(1)非暂时性有形计算机可读存储媒体，和(2)例如暂时性信号或载波等非有形计算机可读通信媒体。

[0331] 已描述各种方面和实例。然而，可在不脱离所附权利要求书的范围的情况下对本发明的结构或技术作出修改。

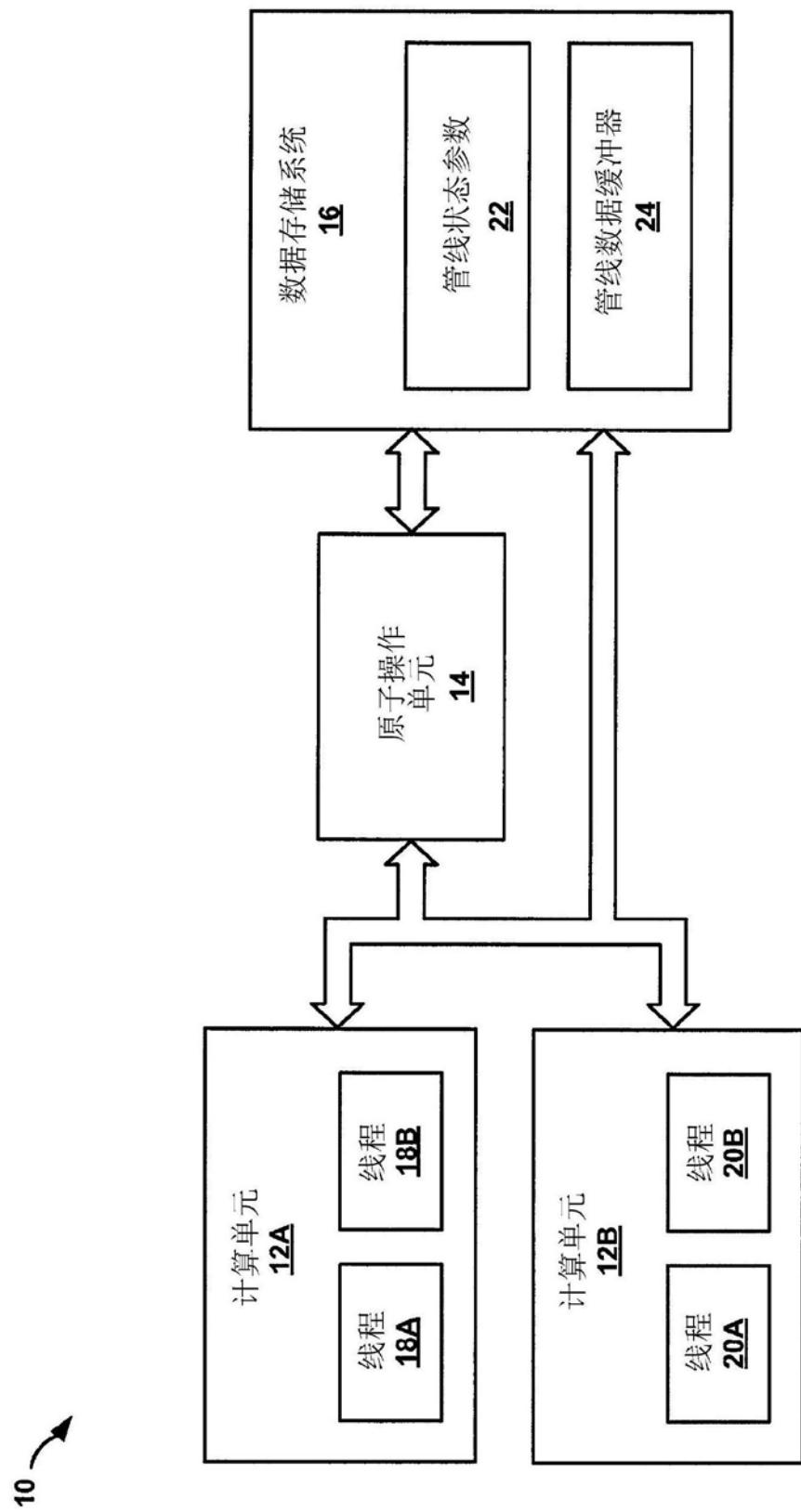


图1

```

1 #define FST_PKT 0 /* 第一包偏移 */
2 #define PKT_CNT 1 /* 包计数 */
3 #define RES_CNT 2 /* 保留包计数 */
4 #define NUM_RES 3 /* 保留的数目 */

5 atomic_reserve_write
6 {
7     uint32 *pipe_addr,
8     uint32 pipe_size_minus_one,
9     uint32 max_depth,
10    uint32 num_packets
11 }
12 {
13     uint32 cnts[4] = load(pipe_addr); // 128位加载
14     // 将96位返回给调用者
15     return cnts[FST_PKT], cnts[PKT_CNT], cnts[RES_CNT];
16     if( (max_depth - cnts[PKT_CNT] - cnts[RES_CNT]) >= num_packets) {
17         cnts[RES_CNT] += num_packets;
18         if( (num_packet != 0) {
19             cnts[NUM_RES]++;
20         }
21         store(pipe_addr, cnts); // 128位存储
22     }
23 }
```

图2

```
1 #define FST_PKT 0 /* 第一包偏移 */
2 #define PKT_CNT 1 /* 包计数 */
3 #define RES_CNT 2 /* 保留包计数 */
4 #define NUM_RES 3 /* 保留的数目 */

5 void atomic_commit_write(uint32 pipe_addr)
6 {
7     uint32 cnts[4] = load(pipe_addr); // 128位加载
8     cnts[NUM_RES]--;
9     if(cnts[NUM_RES] == 0){
10         cnts[PKT_CNT] += cnts[RES_CNT];
11         cnts[RES_CNT] = 0;
12     }
13     store(pipe_addr, cnts); // 将经修改管线计数器存储回到存储器。
14 }
```

图3

```
1 #define FST_PKT 0 /* 第一包偏移 */
2 #define PKT_CNT 1 /* 包计数 */
3 #define RES_CNT 2 /* 保留包计数 */
4 #define NUM_RES 3 /* 保留的数目 */

5 atomic_reserve_read
6 {
7     uint32 *pipe_addr,
8     uint32 pipe_size_minus_one,
9     uint32 max_depth,
10    uint32 num_packets
11 }
12 {
13     uint32 cnts[4] = load(pipe_addr); // 128位加载
14     // 将96位返回给调用者
15     return cnts[FST_PKT], cnts[PKT_CNT], cnts[RES_CNT];
16     if ((cnts[RES_CNT] + num_packets) <= cnts[PKT_CNT]) {
17         cnts[RES_CNT] += num_packets;
18         if (num_packets != 0) {
19             cnts[NUM_RES]++;
20         }
21         store(pipe_addr, cnts); // 128位存储
22     }
23 }
```

图4

```
1 #define FST_PKT 0 /* 第一包偏移 */
2 #define PKT_CNT 1 /* 包计数 */
3 #define RES_CNT 2 /* 保留包计数 */
4 #define NUM_RES 3 /* 保留的数目 */

5 void atomic_commit_read(uint32 *pipe_addr, uint32 pipe_size_minus_one)
6 {
7     uint32 cnts[4] = load(pipe_addr); // 128位加载
8     cnts[NUM_RES]--;
9     if (cnts[NUM_RES] == 0) {
10         cnts[FST_PKT] = (cnts[FST_PKT] + cnts[RES_CNT]) & Pipe_size_minus_one;
11         cnts[PKT_CNT] -= cnts[RES_CNT];
12         cnts[RES_CNT] = 0;
13     }
14     store(pipe_addr, cnts); // 将经修改管线计数器存储回到存储器。
15 }
```

图5

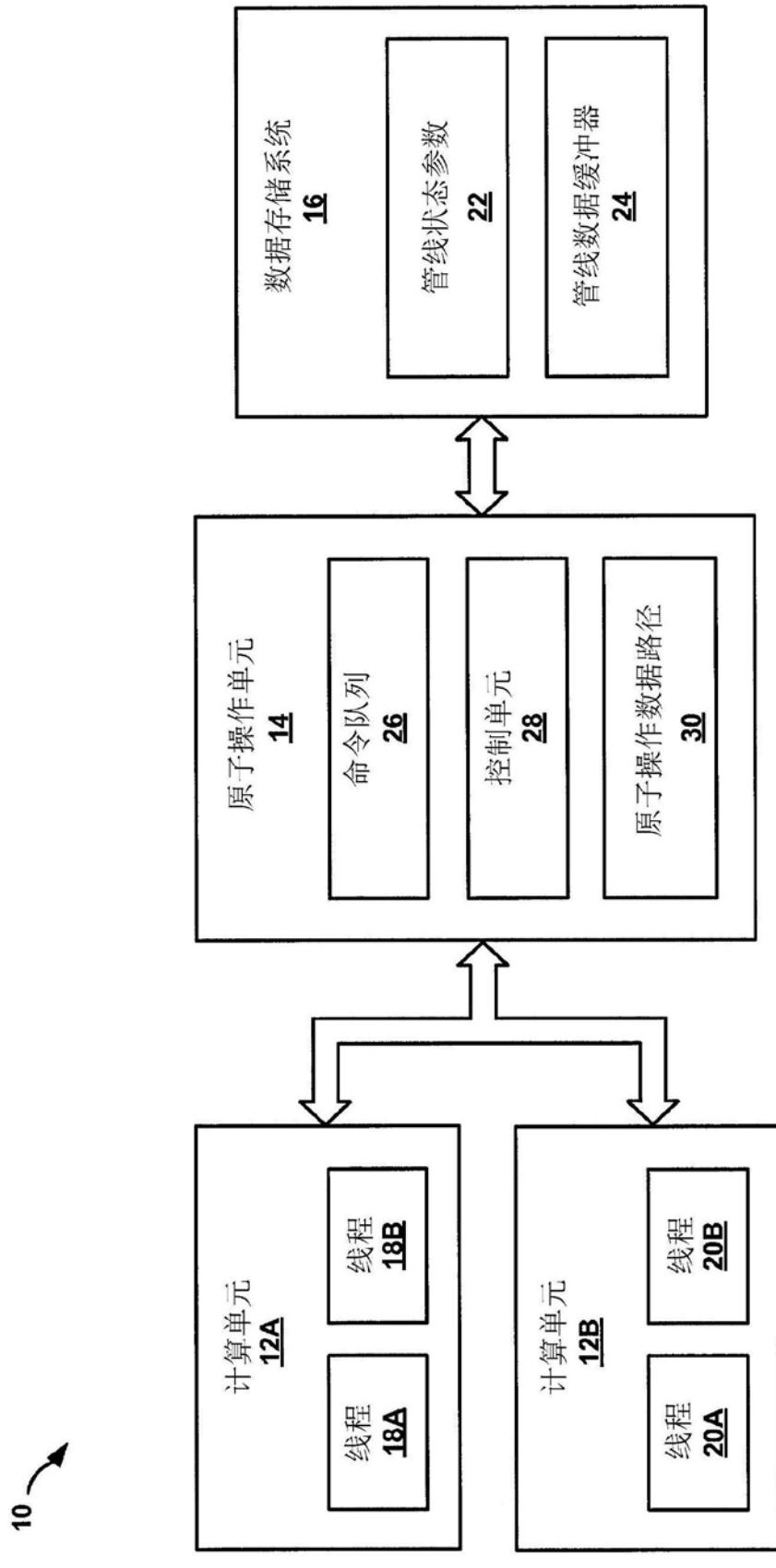


图6

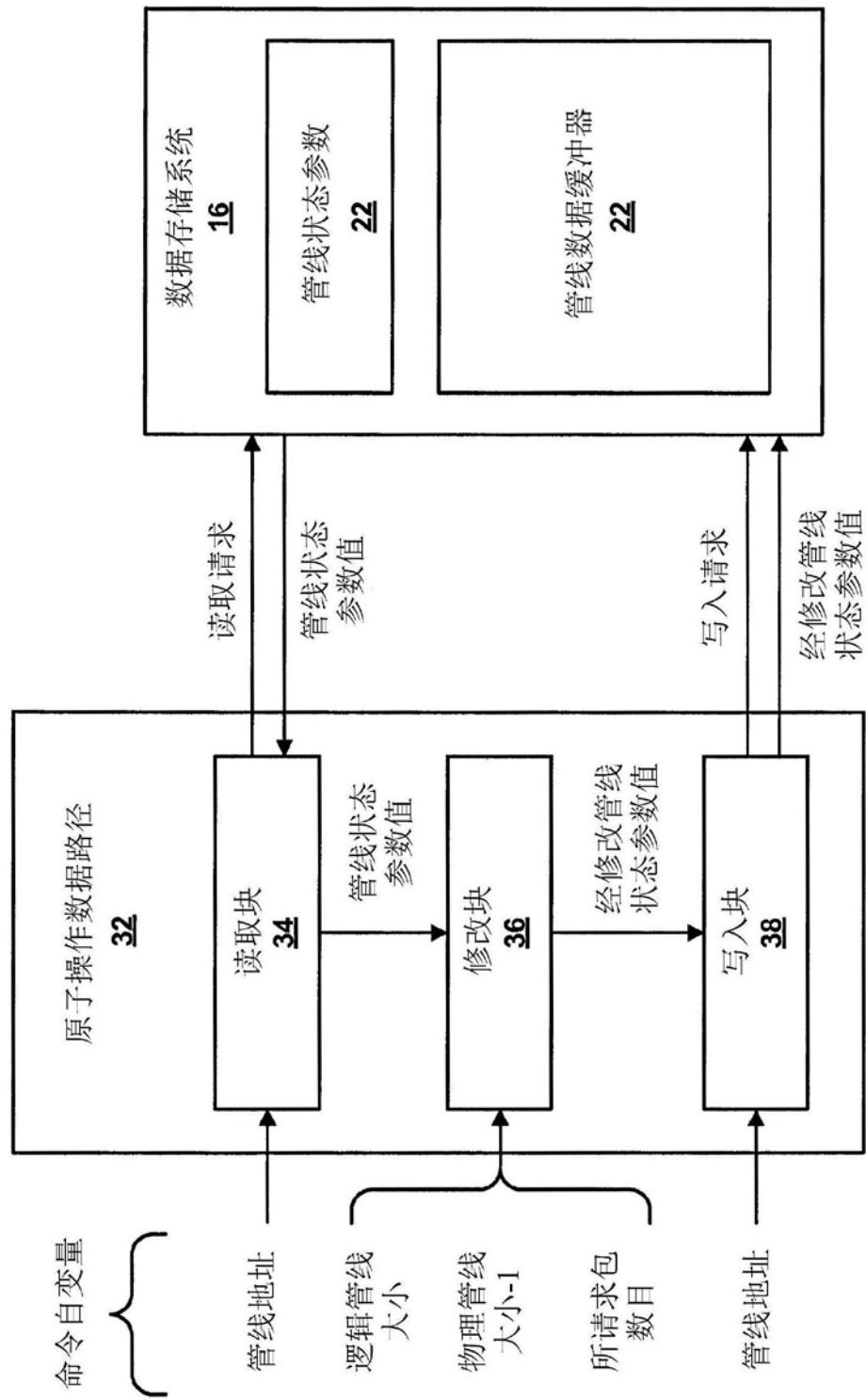


图7

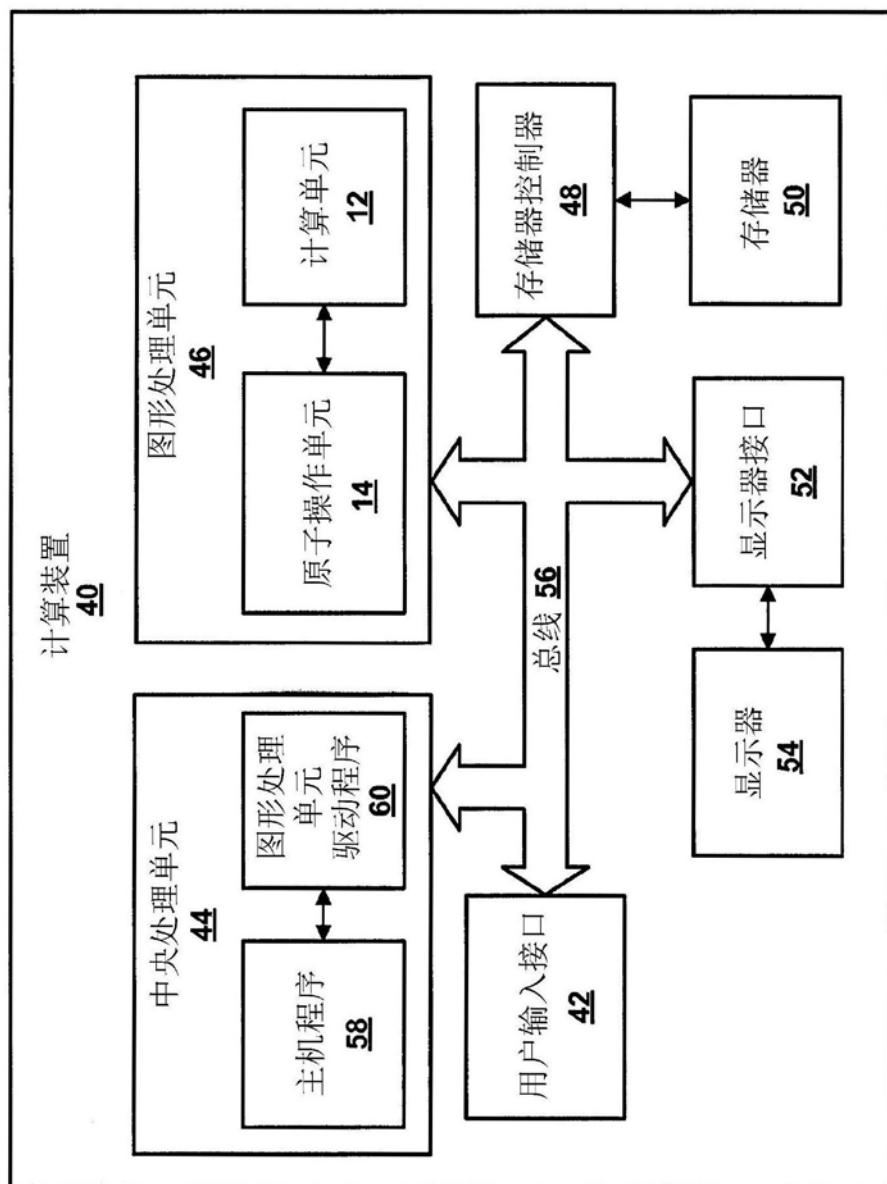


图8

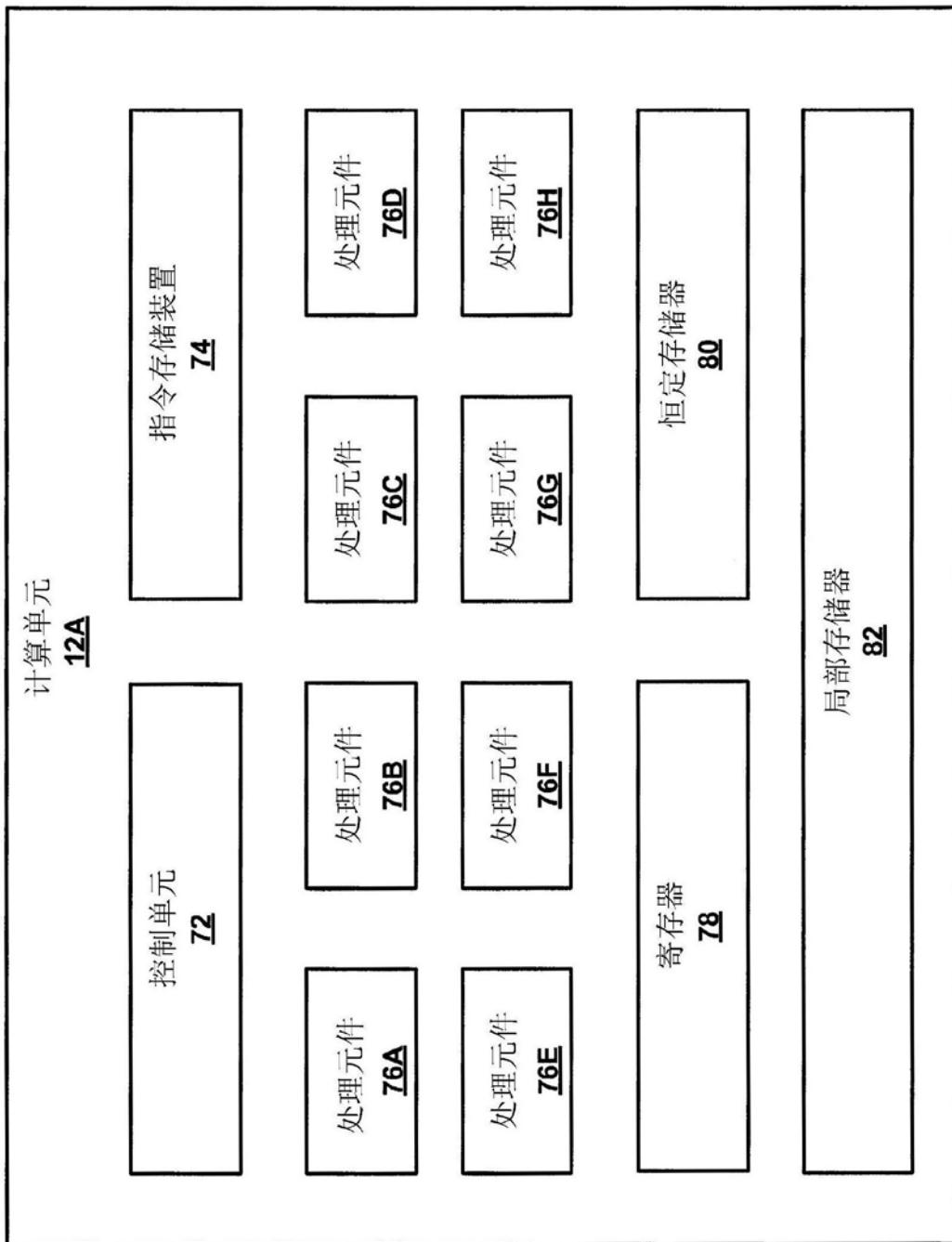


图9

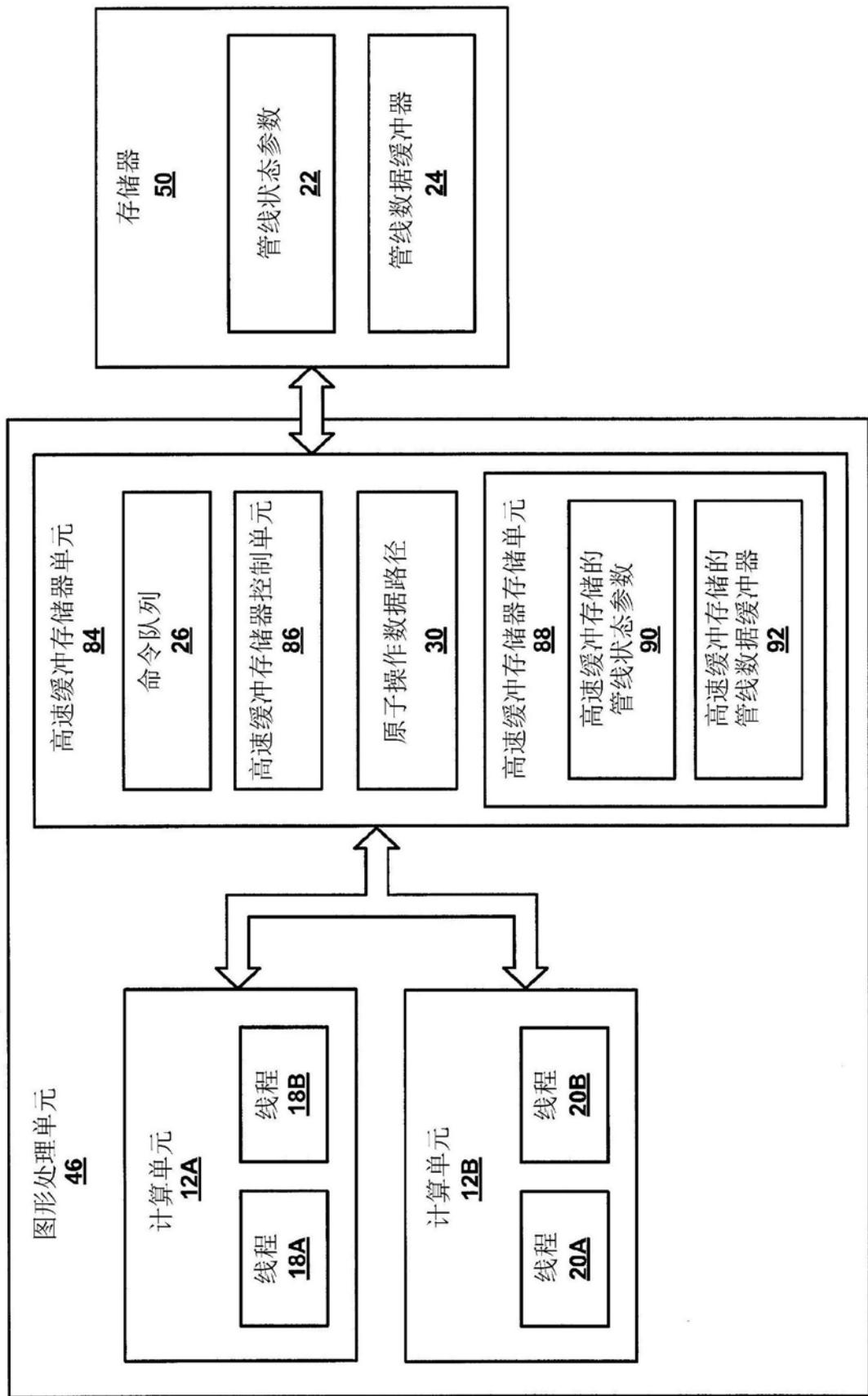


图10

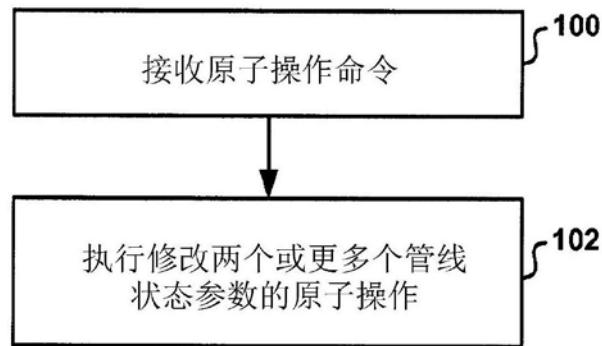


图11

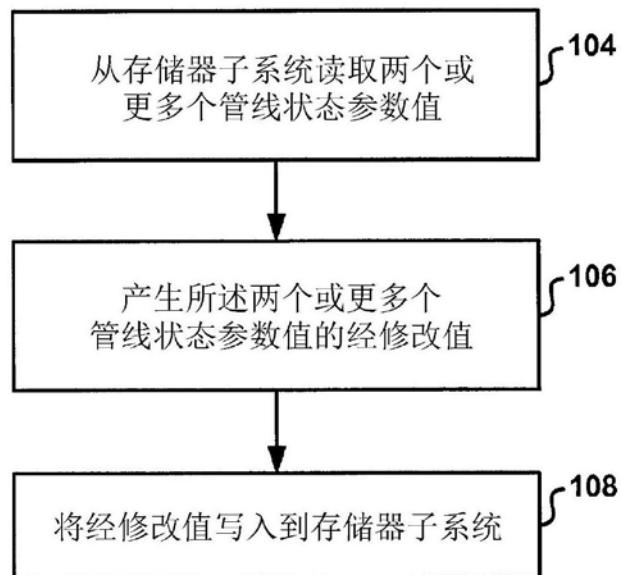


图12

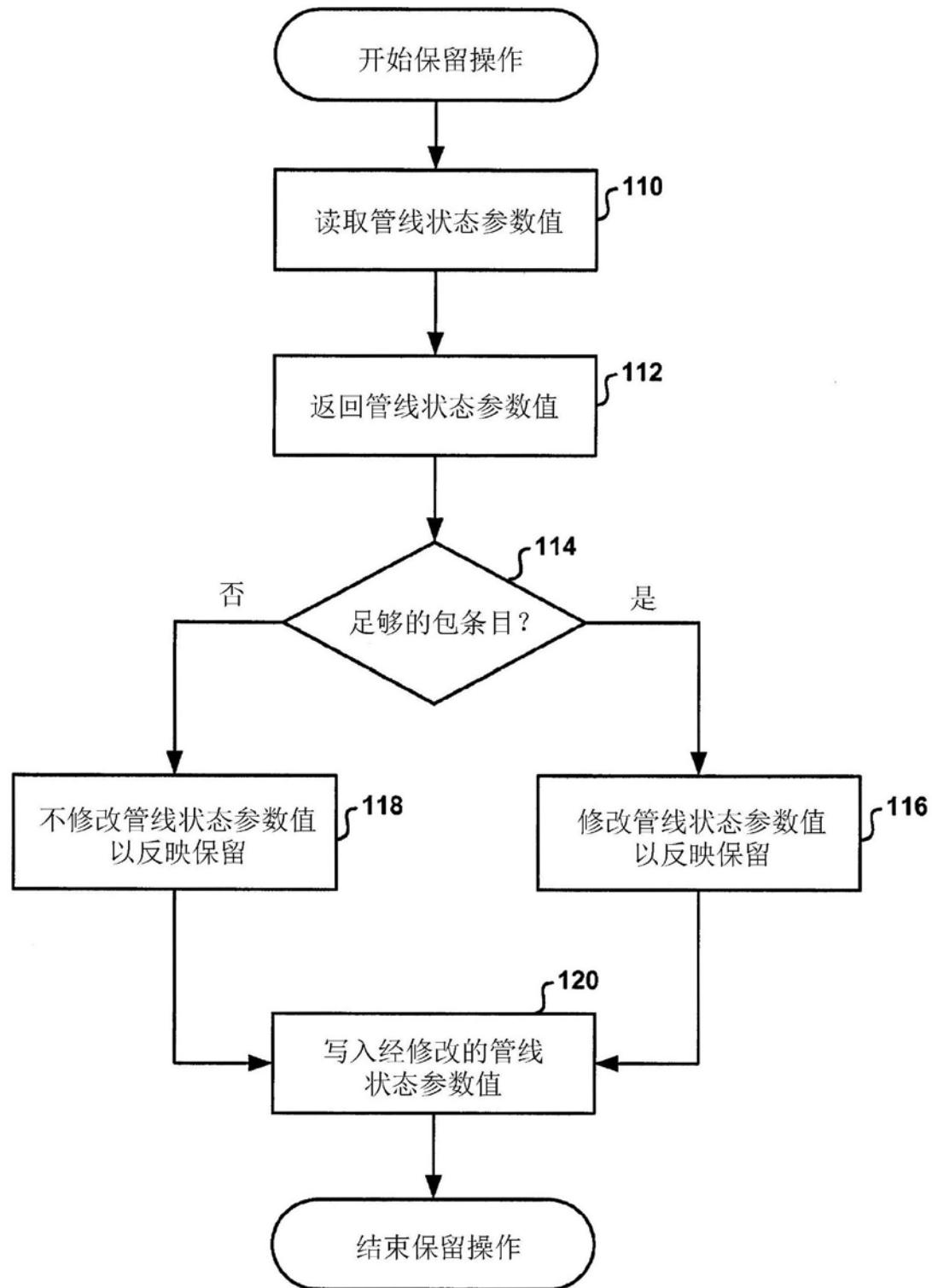


图13

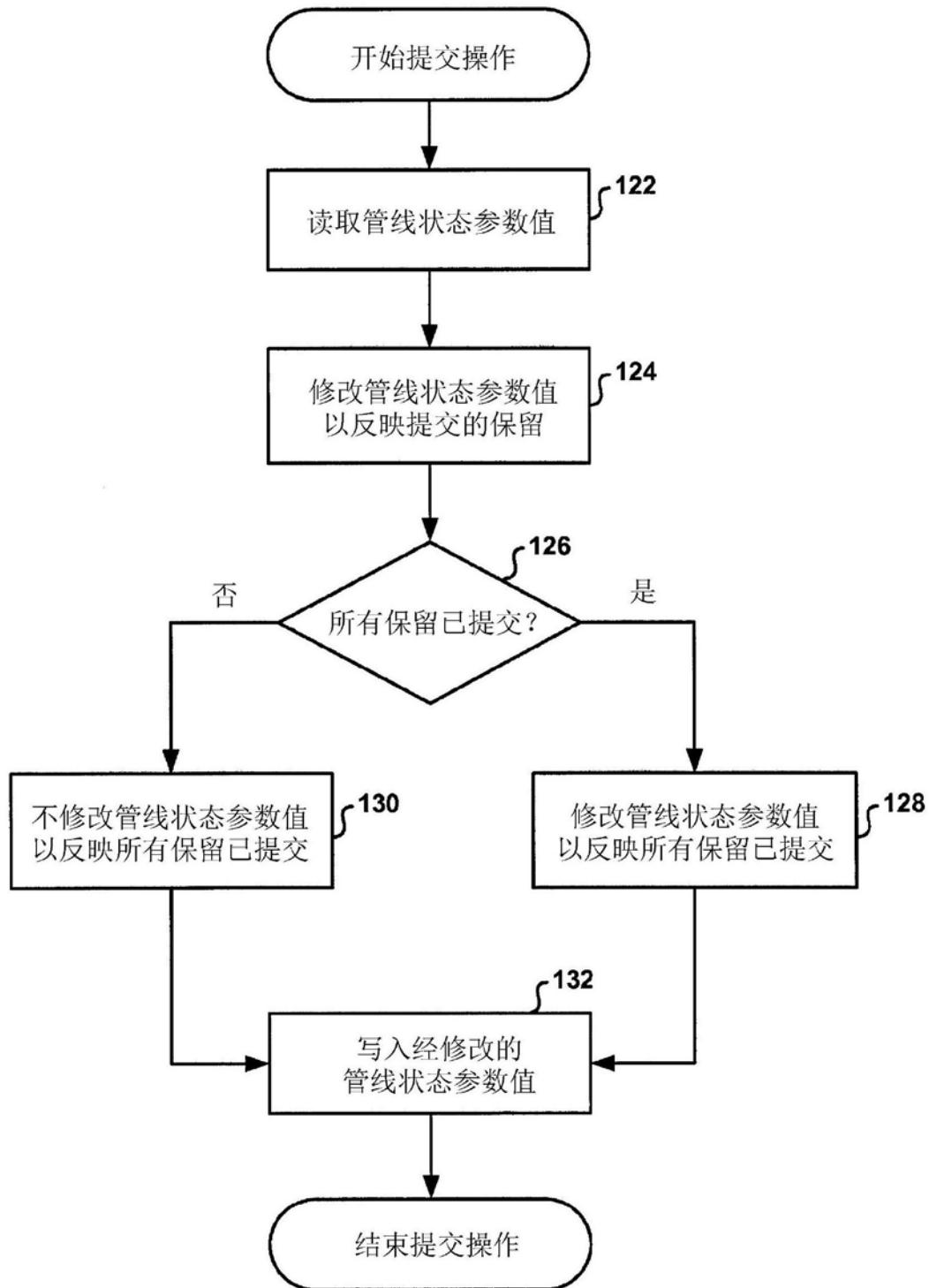


图14