



(12)发明专利

(10)授权公告号 CN 103109274 B

(45)授权公告日 2016.08.17

(21)申请号 201180044782.3

(22)申请日 2011.09.19

(30)优先权数据

61/384,571 2010.09.20 US

61/515,182 2011.08.04 US

13/235,266 2011.09.16 US

13/235,236 2011.09.16 US

(85)PCT国际申请进入国家阶段日

2013.03.18

(86)PCT国际申请的申请数据

PCT/US2011/052196 2011.09.19

(87)PCT国际申请的公布数据

W02012/040121 EN 2012.03.29

(73)专利权人 高通股份有限公司

地址 美国加利福尼亚州

(72)发明人 阿列克谢·V·布尔德

科林·克里斯托弗·夏普

戴维·里赫尔·加西亚 加西亚

张弛红

(74)专利代理机构 北京律盟知识产权代理有限公司 11287

代理人 宋献涛

(51)Int.Cl.

G06F 9/54(2006.01)

G06F 9/38(2006.01)

(56)对比文件

US 6088740 A,2000.07.11,

US 2010123717 A1,2010.05.20,

US 7526634 B1,2009.04.28,

审查员 梁艳

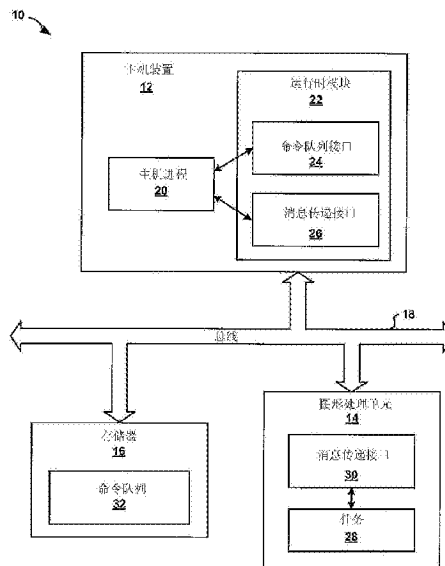
权利要求书5页 说明书41页 附图21页

(54)发明名称

多处理器计算平台中的处理器间通信技术

(57)摘要

本发明描述可用于多处理器计算平台内的通信技术。在一些实例中,所述技术可提供可用于在使用命令队列起始任务的多处理器计算平台内支持消息传递的软件接口。在额外的实例中,所述技术可提供可用于多处理器计算平台内的共享存储器处理器间通信的软件接口。在进一步的实例中,所述技术可提供图形处理单元GPU,所述图形处理单元包含用于支持所述GPU与主机CPU之间的消息传递和/或共享存储器通信的硬件。



1. 一种主机装置,其包括:

一个或多个处理器;

命令队列接口,其在所述一个或多个处理器上执行且经配置以响应于从在所述主机装置上执行的进程接收到一个或多个排队指令而将多个命令放置到命令队列中,所述多个命令包含第一命令,所述第一命令指令所述主机装置在与所述主机装置相关联的第一存储器空间和与图形处理单元GPU相关联的第二存储器空间之间传送数据,所述多个命令进一步包含第二命令,所述第二命令指令所述主机装置起始所述GPU上的任务的执行;以及

消息传递接口,其在所述一个或多个处理器上执行且经配置以在所述GPU上执行的任务正在所述GPU上执行时且响应于从在所述主机装置上执行的所述进程接收到一个或多个消息传递指令而在所述主机装置上执行的所述进程与所述任务之间传递一个或多个消息。

2. 根据权利要求1所述的装置,

其中所述一个或多个消息传递指令包括发送指令,所述发送指令指令所述消息传递接口将消息从在所述主机装置上执行的所述进程发送到在所述GPU上执行的所述任务,且

其中所述消息传递接口进一步经配置以在所述GPU上执行的所述任务正在所述GPU上执行时响应于接收到所述发送指令而将所述消息从在所述主机装置上执行的所述进程发送到所述任务。

3. 根据权利要求1所述的装置,

其中所述一个或多个消息传递指令包括寄存回调例程指令,所述寄存回调例程指令响应于从所述GPU接收到指示在所述GPU上执行的所述任务已发送消息的信号而指令所述消息传递接口调用回调例程,且

其中所述消息传递接口进一步经配置以响应于从所述GPU接收到指示在所述GPU上执行的所述任务已发送消息的所述信号而起始在所述寄存回调例程指令中所指定的所述回调例程的执行。

4. 根据权利要求1所述的装置,

其中所述一个或多个消息传递指令包括轮询指令,所述轮询指令指令所述消息传递接口针对指示在所述GPU上执行的所述任务是否已发送消息的消息状态信息来轮询所述GPU,且

其中所述消息传递接口进一步经配置以响应于接收到所述轮询指令而针对所述消息状态信息来轮询所述GPU,且在所述消息状态信息指示在所述GPU上执行的所述任务已发送所述消息的情况下,从所述GPU获得所述消息。

5. 根据权利要求1所述的装置,其中在所述GPU上执行的所述任务包含指令所述GPU将消息从在所述GPU上执行的所述任务发送到在所述主机装置上执行的所述进程的指令。

6. 根据权利要求1所述的装置,其中在所述GPU上执行的所述任务包含指令所述GPU确定从在所述主机装置上执行的所述进程发送到所述任务的消息是否可用,并在所述消息可用的情况下向所述任务提供所述消息的指令。

7. 根据权利要求1所述的装置,其中所述消息传递接口进一步经配置以在不将任何命令放置于所述命令队列中的情况下执行所述一个或多个消息传递指令。

8. 一种方法,其包括:

响应于从在主机装置上执行的进程接收到一个或多个排队指令而用在所述主机装置

的一个或多个处理器上执行的命令队列接口将多个命令放置到命令队列中,所述多个命令包含第一命令,所述第一命令指令所述主机装置在与所述主机装置相关联的第一存储器空间和与图形处理单元GPU相关联的第二存储器空间之间传送数据,所述多个命令进一步包含第二命令,所述第二命令指令所述主机装置起始所述GPU上的任务的执行;以及

在所述GPU上执行的任务正在所述GPU上执行时且响应于从在所述主机装置上执行的所述进程接收到一个或多个消息传递指令而用在所述主机装置的所述一个或多个处理器上执行的消息传递接口在所述主机装置上执行的所述进程与所述任务之间传递一个或多个消息。

9. 根据权利要求8所述的方法,

其中所述一个或多个消息传递指令包括发送指令,所述发送指令指令所述消息传递接口将消息从在所述主机装置上执行的所述进程发送到在所述GPU上执行的所述任务,且

其中所述方法进一步包括在所述GPU上执行的所述任务正在所述GPU上执行时且响应于接收到所述发送指令而用所述消息传递接口将所述消息从在所述主机装置上执行的所述进程发送到所述任务。

10. 根据权利要求8所述的方法,

其中所述一个或多个消息传递指令包括寄存回调例程指令,所述寄存回调例程指令响应于从所述GPU接收到指示在所述GPU上执行的所述任务已发送消息的信号而指令所述消息传递接口调用回调例程,且

其中所述方法进一步包括响应于从所述GPU接收到指示在所述GPU上执行的所述任务已发送消息的所述信号而起始在所述寄存回调例程指令中所指定的所述回调例程的执行。

11. 根据权利要求8所述的方法,

其中所述一个或多个消息传递指令包括轮询指令,所述轮询指令指令所述消息传递接口针对指示在所述GPU上执行的所述任务是否已发送消息的消息状态信息来轮询所述GPU,且

其中所述方法进一步包括:

响应于接收到所述轮询指令而用所述消息传递接口针对所述消息状态信息来轮询所述GPU;以及

在所述消息状态信息指示在所述GPU上执行的所述任务已发送所述消息的情况下,从所述GPU获得所述消息。

12. 根据权利要求8所述的方法,其中在所述GPU上执行的所述任务包含指令所述GPU将消息从在所述GPU上执行的所述任务发送到在所述主机装置上执行的所述进程的指令。

13. 根据权利要求8所述的方法,其中在所述GPU上执行的所述任务包含指令所述GPU确定从在所述主机装置上执行的所述进程发送到所述任务的消息是否可用,并在所述消息可用的情况下向所述任务提供所述消息的指令。

14. 根据权利要求8所述的方法,其进一步包括:

用所述消息传递接口在不将任何命令放置于所述命令队列中的情况下执行所述一个或多个消息传递指令。

15. 一种设备,其包括:

用于响应于从在主机装置上执行的进程接收到一个或多个排队指令而将多个命令放

置到命令队列中的装置,所述多个命令包含第一命令,所述第一命令指令所述主机装置在与所述主机装置相关联的第一存储器空间和与图形处理单元GPU相关联的第二存储器空间之间传送数据,所述多个命令进一步包含第二命令,所述第二命令指令所述主机装置起始所述GPU上的任务的执行;以及

用于在所述GPU上执行的任务正在所述GPU上执行时且响应于从在所述主机装置上执行的所述进程接收到一个或多个消息传递指令而在所述主机装置上执行的所述进程与所述任务之间传递一个或多个消息的装置。

16.根据权利要求15所述的设备,

其中所述一个或多个消息传递指令包括发送指令,所述发送指令指令所述用于传递所述一个或多个消息的装置将消息从在所述主机装置上执行的所述进程发送到在所述GPU上执行的所述任务,且

其中所述设备进一步包括用于在所述GPU上执行的所述任务正在所述GPU上执行时响应于接收到所述发送指令而将所述消息从在所述主机装置上执行的所述进程发送到所述任务的装置。

17.根据权利要求15所述的设备,

其中所述一个或多个消息传递指令包括寄存回调例程指令,所述寄存回调例程指令响应于从所述GPU接收到指示在所述GPU上执行的所述任务已发送消息的信号而指令所述用于传递所述一个或多个消息的装置调用回调例程,且

其中所述设备进一步包括用于响应于从所述GPU接收到指示在所述GPU上执行的所述任务已发送消息的所述信号而起始在所述寄存回调例程指令中所指定的所述回调例程的执行的装置。

18.根据权利要求15所述的设备,

其中所述一个或多个消息传递指令包括轮询指令,所述轮询指令指令所述用于传递所述一个或多个消息的装置针对指示在所述GPU上执行的所述任务是否已发送消息的消息状态信息来轮询所述GPU,且

其中所述设备进一步包括:

用于响应于接收到所述轮询指令而针对所述消息状态信息来轮询所述GPU的装置;以及

用于在所述消息状态信息指示在所述GPU上执行的所述任务已发送所述消息的情况下从所述GPU获得所述消息的装置。

19.一种包括图形处理单元GPU的设备,所述GPU包括:

一个或多个处理器,其经配置以执行任务;

一个或多个寄存器,其可由主机装置存取;以及

消息传递模块,其经配置以在所述一个或多个处理器上执行的所述任务正在所述一个或多个处理器上执行时且响应于从在所述一个或多个处理器上执行的所述任务接收到一个或多个消息传递指令而经由所述一个或多个寄存器在所述任务与在所述主机装置上执行的进程之间传递一个或多个消息。

20.根据权利要求19所述的设备,

其中所述一个或多个消息传递指令包括发送指令,所述发送指令指令所述消息传递模

块将消息从在所述GPU上执行的所述任务发送到在所述主机装置上执行的所述进程,且

其中所述消息传递模块进一步经配置以将与所述消息相关联的消息数据存储在所述一个或多个寄存器中。

21. 根据权利要求19所述的设备,

其中所述一个或多个消息传递指令包括接收指令,所述接收指令指令所述消息传递模块确定从在所述主机装置上执行的所述进程发送到所述任务的消息是否可用,并在所述消息可用的情况下向所述任务提供所述消息,且

其中所述消息传递模块进一步经配置以从所述一个或多个寄存器获得与所述消息相关联的消息数据。

22. 一种方法,其包括:

用图形处理单元GPU的消息传递模块从在所述GPU上执行的任务接收一个或多个消息传递指令;以及

经由所述GPU内的可由主机装置存取的一个或多个寄存器在所述GPU上执行的所述任务正在所述GPU上执行时且响应于从在所述GPU上执行的所述任务接收到所述一个或多个消息传递指令而在所述任务与在所述主机装置上执行的进程之间传递一个或多个消息。

23. 根据权利要求22所述的方法,

其中所述一个或多个消息传递指令包括发送指令,所述发送指令指令所述消息传递模块将消息从在所述GPU上执行的所述任务发送到在所述主机装置上执行的所述进程,且

其中所述方法进一步包括将与所述消息相关联的消息数据存储在所述一个或多个寄存器中。

24. 根据权利要求22所述的方法,

其中所述一个或多个消息传递指令包括接收指令,所述接收指令指令所述消息传递模块确定从在所述主机装置上执行的所述进程发送到所述任务的消息是否可用,并在所述消息可用的情况下向所述任务提供所述消息,且

其中所述方法进一步包括从所述一个或多个寄存器获得与所述消息相关联的消息数据。

25. 一种设备,其包括:

用于从在图形处理单元GPU上执行的任务接收一个或多个消息传递指令的装置;以及

用于经由所述GPU内的可由主机装置存取的一个或多个寄存器在所述GPU上执行的所述任务正在所述GPU上执行时且响应于从在所述GPU上执行的所述任务接收到所述一个或多个消息传递指令而在所述任务与在所述主机装置上执行的进程之间传递一个或多个消息的装置。

26. 根据权利要求25所述的设备,

其中所述一个或多个消息传递指令包括发送指令,所述发送指令指令所述用于传递的装置将消息从在所述GPU上执行的所述任务发送到在所述主机装置上执行的所述进程,且

其中所述设备进一步包括用于将与所述消息相关联的消息数据存储在所述一个或多个寄存器中的装置。

27. 根据权利要求25所述的设备,

其中所述一个或多个消息传递指令包括接收指令,所述接收指令指令所述用于传递的

装置确定从在所述主机装置上执行的所述进程发送到所述任务的消息是否可用,并在所述消息可用的情况下向所述任务提供所述消息,且

其中所述设备进一步包括用于从所述一个或多个寄存器获得与所述消息相关联的消息数据的装置。

多处理器计算平台中的处理器间通信技术

技术领域

[0001] 本发明涉及计算平台,且更特定来说,涉及包含多个处理器的计算平台。

背景技术

[0002] 包含多个处理器的计算平台用于提高具有高计算密集要求和/或高数据处理量要求的应用的性能。多处理器计算平台可包含可充当主机装置的通用中央处理单元(CPU)以及主机CPU可用来卸载计算密集型任务的性能的一个或一个以上计算装置,进而提高整个系统的性能。在一些情况下,所述一个或一个以上计算装置可经特别设计以比主机CPU更高效地处理某些类型的任务,其可提供对整个系统的进一步的性能改进。举例来说,所述一个或一个以上计算装置可经特别设计以比主机CPU更高效地执行平行算法。

[0003] 可用于多处理器计算系统中的一种类型的计算装置是图形处理单元(GPU)。传统上,GPU包含固定功能硬件,其经特别设计以用于向显示装置实时地再现三维(3D)图形,但是通常不可编程,即,不可将经编译的程序下载到GPU并在GPU上执行。然而,近来,随着可编程着色器单元的发展,大多数GPU架构已经转变为可编程架构,所述可编程架构包含许多并行的处理元件。所述可编程架构允许GPU促进不仅仅是图形操作的执行,而且促进以高度并行的方式执行通用计算任务。

[0004] 使用GPU来执行通用非图形专有计算任务可在本文中被称作图形处理单元上的通用计算(GPGPU),或者被称作GPU计算。在一些情况下,GPU可使并非图形专有的应用编程接口(API)可用,进而减轻对GPU的编程以用于执行通用计算任务。GPU计算任务可包含计算密集的任务和/或包含高度并行的任务,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。

[0005] GPU是可用于多处理器计算平台中的仅一种类型的计算装置,且可使用其它类型的计算装置来补充或取代GPU。举例来说,可用于多处理器计算平台中的其它类型的计算装置包含(例如)额外的CPU、数字信号处理器(DSP)、小区宽带引擎(Cell/BE)处理器,或任何其它类型的处理单元。

[0006] 具有多个计算装置的多处理器计算平台可为同类平台或异类平台。在同类平台中,所有计算装置共享共同指令集架构(ISA)。相比而言,异类平台可包含具有不同ISA的两个或两个以上计算装置。一般来说,不同类型的计算装置可具有不同的ISA,且相同类型的不同品牌的计算装置也可具有不同的ISA。

[0007] 可通过利用多核计算装置和/或众核计算装置来进一步改进多处理器计算平台的性能。多核计算装置的一实例是上文所描述的GPU,其含有具有多个处理核心的可编程着色器单元。然而,CPU还可经设计以包含多个处理核心。一般来说,可将包含多个处理核心的任何芯片或裸片视为多核处理器。处理核心可指代能够对特定块数据执行指令的处理单元。举例来说,可将GPU内的单一算法逻辑单元(ALU)单元或向量处理器视为处理核心。众核处理器一般指代具有相对大量的核心的多核处理器,例如大于十个核心,且通常使用与用于设计具有少量核心的多核处理器的技术不同的技术来设计。多核处理器通过允许在单一芯

片上在多个核心上并行地(例如,同时地)执行软件程序来提供性能改进。

[0008] 并行编程模型指代经设计以允许在多个处理核心上同时地执行程序的编程模型。所述程序可为多线程程序,在这种情况下,单一线程可操作于每一处理核心上。在一些实例中,单一计算装置可包含用于执行程序的所有处理核心。在其它实例中,用于执行程序的一些处理核心可位于相同类型或不同类型的不同计算装置上。

[0009] 可使用跨平台、跨供应商、异类计算平台、并行编程模型应用编程接口(API)来提供共同语言规范,以用于对包含由实施不同ISA的不同供应商可能制成的不同类型的计算装置的异类多核计算平台进行并行编程。开放计算语言(OpenCL™)是跨平台、跨供应商、异类计算平台、并行编程API的一实例。此类API可经设计以允许GPU上的更一般化的数据处理。举例来说,除了经由计算语言暴露经扩展的着色器子系统能力之外,这些API可以非图形专有的方式将数据流和控制路径一般化到GPU中。然而,目前,由此类API提供的指令集是基于GPU的硬件架构,且因此受限于与现有GPU架构相容的功能性。

发明内容

[0010] 本发明描述可用于多处理器计算平台内的通信技术。在一些实例中,所述技术可提供软件接口,所述软件接口可用于支持在使用命令队列起始任务的多处理器计算平台内的消息传递。在额外的实例中,所述技术可提供软件接口,所述软件接口可用于多处理器计算平台内的共享存储器处理器间通信。在进一步的实例中,所述技术可提供图形处理单元GPU,所述图形处理单元包含用于支持所述GPU与主机CPU之间的消息传递和/或共享存储器通信的硬件。

[0011] 在一个实例中,本发明描述一种包含一个或一个以上处理器的主机装置。所述装置进一步包含命令队列接口,所述命令队列接口在一个或一个以上处理器上执行且经配置以响应于从在主机装置上执行的进程接收到一个或一个以上排队指令而将多个命令置于命令队列中。所述多个命令包含第一命令,所述第一命令指令所述主机装置在与所述主机装置相关联的第一存储器空间与和图形处理单元(GPU)相关联的第二存储器空间之间传送数据。所述多个命令进一步包含第二命令,所述第二命令指令所述主机装置起始GPU上的任务的执行。所述装置进一步包含消息传递接口,所述消息传递接口在一个或一个以上处理器上执行且经配置以在GPU上执行的任务正在GPU上执行时且响应于从在主机装置上执行的进程接收到一个或一个以上消息传递指令而在于所述主机装置上执行的进程与所述任务之间传递一个或一个以上消息。

[0012] 在另一实例中,本发明描述一种方法,所述方法包含响应于从在主机装置上执行的进程接收到一个或一个以上排队指令而用在主机装置的一个或一个以上处理器上执行的命令队列接口将多个命令放置到命令队列中。所述多个命令包含第一命令,所述第一命令指令所述主机装置在与所述主机装置相关联的第一存储器空间与和图形处理单元(GPU)相关联的第二存储器空间之间传送数据。所述多个命令进一步包含第二命令,所述第二命令指令所述主机装置起始GPU上的任务的执行。所述方法进一步包含在GPU上执行的任务正在GPU上执行时且响应于从在主机装置上执行的进程接收到一个或一个以上消息传递指令而用在主机装置的一个或一个以上处理器上执行的消息传递接口在于所述主机装置上执行的进程与所述任务之间传递一个或一个以上消息。

[0013] 在另一实例中,本发明描述一种设备,所述设备包含用于响应于从在主机装置上执行的进程接收到一个或一个以上排队指令而将多个命令放置到命令队列中的装置。所述多个命令包含第一命令,所述第一命令指令所述主机装置在与所述主机装置相关联的第一存储器空间与和图形处理单元(GPU)相关联的第二存储器空间之间传送数据。所述多个命令进一步包含第二命令,所述第二命令指令所述主机装置起始GPU上的任务的执行。所述设备进一步包含在于GPU上执行的任务正在GPU上执行时且响应于从在主机装置上执行的进程接收到一个或一个以上消息传递指令而在于所述主机装置上执行的进程与所述任务之间传递一个或一个以上消息的装置。

[0014] 在另一实例中,本发明描述一种包含指令的计算机可读存储媒体,所述指令致使一个或一个以上处理器响应于从在主机装置上执行的进程接收到一个或一个以上排队指令而将多个命令放置到命令队列中。所述多个命令包含第一命令,所述第一命令指令所述主机装置在与所述主机装置相关联的第一存储器空间与和图形处理单元(GPU)相关联的第二存储器空间之间传送数据。所述多个命令进一步包含第二命令,所述第二命令指令所述主机装置起始GPU上的任务的执行。所述计算机可读存储媒体进一步包含致使所述一个或一个以上处理器在于GPU上执行的任务正在GPU上执行时且响应于从在主机装置上执行的进程接收到一个或一个以上消息传递指令而在于所述主机装置上执行的进程与所述任务之间传递一个或一个以上消息的指令。

[0015] 在另一实例中,本发明描述一种图形处理单元(GPU),其包含经配置以执行任务的一个或一个以上处理器。所述GPU进一步包含可由主机装置存取的一个或一个以上寄存器。所述GPU进一步包含消息传递模块,所述消息传递模块经配置以在于所述一个或一个以上处理器上执行的任务正在所述一个或一个以上处理器上执行时且响应于从在所述一个或一个以上处理器上执行的任务接收到一个或一个以上消息传递指令而经由所述一个或一个以上寄存器在所述任务与在主机装置上执行的进程之间传递一个或一个以上消息。

[0016] 在另一实例中,本发明描述一种方法,所述方法包含用图形处理单元(GPU)的消息传递模块从在所述GPU上执行的任务接收一个或一个以上消息传递指令。所述方法进一步包含经由可由主机装置存取的所述GPU内的一个或一个以上寄存器在于所述GPU上执行的任务正在所述GPU上执行时且响应于从在所述GPU上执行的任务接收到一个或一个以上消息传递指令而在所述任务与在所述主机装置上执行的进程之间传递一个或一个以上消息。

[0017] 在另一实例中,本发明描述一种设备,所述设备包含用于从在图形处理单元(GPU)上执行的任务接收一个或一个以上消息传递指令的装置。所述设备进一步包含用于经由可由主机装置存取的所述GPU内的一个或一个以上寄存器在于所述GPU上执行的任务正在所述GPU上执行时且响应于从在所述GPU上执行的任务接收到一个或一个以上消息传递指令而在所述任务与在所述主机装置上执行的进程之间传递一个或一个以上消息的装置。

[0018] 在另一实例中,本发明描述一种包括指令的计算机可读媒体,所述指令致使一个或一个以上处理器从在图形处理单元(GPU)上执行的任务接收一个或一个以上消息传递指令。所述计算机可读存储媒体进一步包含致使所述一个或一个以上处理器经由可由主机装置存取的所述GPU内的一个或一个以上寄存器在于所述GPU上执行的任务正在所述GPU上执行时且响应于从在所述GPU上执行的任务接收到一个或一个以上消息传递指令而在所述任务与在所述主机装置上执行的进程之间传递一个或一个以上消息的指令。

[0019] 在另一实例中,本发明描述一种方法,所述方法包含用在主机装置的一个或一个以上处理器上执行的存储器缓冲器接口来接收包含指定是否应针对可由主机装置以及由图形处理单元(GPU)存取的共享存储器空间来启用直接模式的信息的指令。所述方法进一步包含基于指定是否应启用所述直接模式的所述信息而用所述存储器缓冲器接口针对所述共享存储器空间选择性地启用所述直接模式。

[0020] 在另一实例中,本发明描述一种包含一个或一个以上处理器的主机装置。所述装置进一步包含存储器缓冲器接口,所述存储器缓冲器接口在所述一个或一个以上处理器上执行且经配置以接收包含指定是否应针对共享存储器空间来启用直接模式的信息的指令,且基于指定是否应启用所述直接模式的所述信息而针对所述共享存储器空间选择性地启用所述直接模式,所述共享存储器空间可由主机装置以及由图形处理单元(GPU)存取。

[0021] 在另一实例中,本发明描述一种设备,所述设备包含用于接收包含指定是否应针对可由主机装置以及由图形处理单元(GPU)存取的共享存储器空间来启用直接模式的信息的指令的装置。所述设备进一步包含用于基于指定是否应启用所述直接模式的所述信息而针对所述共享存储器空间选择性地启用所述直接模式的装置。

[0022] 在另一实例中,本发明描述一种包括指令的计算机可读媒体,所述指令致使一个或一个以上处理器接收包含指定是否应针对可由主机装置以及由图形处理单元(GPU)存取的共享存储器空间来启用直接模式的信息的指令。所述计算机可读存储器媒体进一步包含致使一个或一个以上处理器基于指定是否应启用所述直接模式的所述信息而针对所述共享存储器空间选择性地启用所述直接模式的指令。

[0023] 在另一实例中,本发明描述一种图形处理单元(GPU),其包含与存储器相关联的GPU高速缓冲存储器。所述装置进一步包含一个或一个以上处理模块,所述一个或一个以上处理模块经配置以响应于接收到指定是否应使用高速缓存服务来用于相对于存储器的存储器空间执行读取操作和写入操作中的至少一者的信息而选择性地使用GPU高速缓冲存储器的高速缓存服务来相对于所述存储器空间执行读取操作和写入操作中的至少一者。

[0024] 在另一实例中,本发明描述一种方法,所述方法包含响应于接收到指定是否应使用高速缓存服务来用于相对于存储器的存储器空间执行读取操作和写入操作中的至少一者的信息而选择性地使用与存储器相关联的图形处理单元(GPU)高速缓冲存储器的高速缓存服务来相对于所述存储器空间执行读取操作和写入操作中的至少一者。

[0025] 在另一实例中,本发明描述一种设备,其包含与存储器相关联的GPU高速缓冲存储器。所述设备进一步包含用于响应于接收到指定是否应使用高速缓存服务来用于相对于存储器的存储器空间执行读取操作和写入操作中的至少一者的信息而选择性地使用GPU高速缓冲存储器的高速缓存服务来相对于所述存储器空间执行读取操作和写入操作中的至少一者的装置。

[0026] 在另一实例中,本发明描述一种包括指令的计算机可读媒体,所述指令致使一个或一个以上处理器响应于接收到指定是否应使用高速缓存服务来用于相对于存储器的存储器空间执行读取操作和写入操作中的至少一者的信息而选择性地使用与存储器相关联的图形处理单元(GPU)高速缓冲存储器的高速缓存服务来相对于所述存储器空间执行读取操作和写入操作中的至少一者。

附图说明

[0027] 图1是说明根据本发明的可用于执行消息传递技术的实例性计算系统的方框图。

[0028] 图2是说明根据本发明的可用于图1的计算系统中的实例性GPU的方框图。

[0029] 图3是说明根据本发明的用于多处理器平台环境中的消息传递的实例性技术的流程图。

[0030] 图4是说明根据本发明的用于执行由在主机装置上执行的进程发布的发送指令的实例性技术的流程图。

[0031] 图5和6是说明根据本发明的可用于实施图4中所说明的技术的若干部分的实例性技术的流程图。

[0032] 图7是说明根据本发明的用于处理例如GPU等计算装置中的所接收的消息的实例性技术的流程图。

[0033] 图8是说明根据本发明的用于执行由在例如GPU等计算装置上执行的任务发布的接收指令的实例性技术的流程图。

[0034] 图9和10是说明根据本发明的可用于实施图8中所说明的技术的若干部分的实例性技术的流程图。

[0035] 图11是说明根据本发明的用于执行由在例如GPU等计算装置上执行的进程发布的发送指令的实例性技术的流程图。

[0036] 图12和13是说明根据本发明的可用于实施图11中所说明的技术的若干部分的实例性技术的流程图。

[0037] 图14是说明根据本发明的用于执行由在主机装置上执行的进程发布的寄存回调例程指令的实例性技术的流程图。

[0038] 图15是说明根据本发明的用于处理从计算装置接收到的中断的实例性技术的流程图。

[0039] 图16和17是说明根据本发明的可用于实施图15中所说明的技术的若干部分的实例性技术的流程图。

[0040] 图18是说明根据本发明的用于执行由在主机装置上执行的进程发布的读取指令的实例性技术的流程图。

[0041] 图19是说明根据本发明的可用于实施图18中所说明的技术的若干部分的实例性技术的流程图。

[0042] 图20是说明根据本发明的可促进直接存储器对象的使用的实例性计算系统的方框图。

[0043] 图21是说明根据本发明的用于执行由在主机装置上执行的进程发布的存储器对象创建指令的实例性技术的流程图。

[0044] 图22是说明根据本发明的用于执行由在主机装置上执行的进程发布的存储器对象创建指令的另一实例性技术的流程图。

[0045] 图23到26是说明根据本发明的用于处理高速缓存模式指令和直接模式指令的实例性技术的流程图。

[0046] 图27是说明根据本发明的可用于图20的计算系统中的实例性GPU的方框图。

[0047] 图28是说明根据本发明的用于处理高速缓存模式指令和直接模式指令的实例性技术的流程图。

[0048] 图29是说明根据本发明的用于执行由在主机装置上执行的进程发布的存储器对象创建指令的另一实例性技术的流程图。

[0049] 图30是说明根据本发明GPU可如何处理根据第一编译技术而编译的指令序列的流程图。

[0050] 图31是说明根据本发明的用于编译用于任务的源代码的实例性技术的流程图。

[0051] 图32是说明根据本发明的可由GPU用来选择性地使用高速缓存服务的实例性技术的流程图。

具体实施方式

[0052] 本发明描述可用于多处理器计算平台内的通信技术。在一些实例中,所述技术可提供软件接口,所述软件接口可用于支持在使用命令队列起始任务的多处理器计算平台内的消息传递。在额外的实例中,所述技术可提供软件接口,所述软件接口可用于多处理器计算平台内的共享存储器处理器间通信。在进一步的实例中,所述技术可提供图形处理单元GPU,所述图形处理单元包含用于支持所述GPU与主机CPU之间的消息传递和/或共享存储器通信的硬件。

[0053] 近年来,最初经设计以用于处理实时3D图形的处理器(例如,图形处理单元(GPU))被一般化,从而执行通用计算任务(GPGPU)。已通过采用业界标准(例如,开放计算语言(OpenCL™)标准)来部分地证明GPGPU的价值。OpenCL是可用于在多处理器计算平台上执行具有任务级并行度和/或数据级并行度的跨平台、跨供应商、异类计算平台、并行编程API的一实例。所述API经特别设计以通过以非图形专有的方式来使GPU的数据流和控制路径一般化而允许GPU上的更一般化的数据处理。此方法的一个限制是主机CPU与计算装置(例如,GPU)之间的数据通信的粗糙粒度。

[0054] 举例来说,OpenCL API提供支持主机装置与一个或一个以上计算装置之间的任务级粒度的通信的命令队列接口。每一命令队列一般保持将由特定计算装置执行的命令。在主机装置上执行的主机进程可通过将指令主机装置执行存储器传送的命令放置在命令队列中而在主机存储器空间与装置存储器空间之间传送数据。类似地,主机进程可通过将指令主机装置在计算装置上执行任务的命令放置在命令队列中而致使任务开始在计算装置上执行。

[0055] 所述命令队列接口可经配置以提供对命令的按序执行或对命令的无序执行。当命令队列接口经配置以提供对命令的按序执行时,命令队列接口保证将以将命令放置到命令队列中的次序来执行命令,且直到前一命令已完成执行之后才开始对后续命令的执行。因此,当主机进程将命令放置在命令队列中来执行任务时,命令队列等待任务完成执行,之后执行可能被随后放置到命令队列中的任何额外的命令。

[0056] 在涉及主机CPU和GPU以及按序命令队列的简单环境中,主机CPU与GPU之间的通信方案可涉及以下操作:(1)主机CPU准备好数据且将所述数据放置到GPU可存取的存储器中;(2)主机CPU命令GPU执行任务;(3)主机CPU等待GPU完成对所述任务的执行;以及(4)主机CPU将数据从GPU可存取的存储器复制到主机存储器。在此类配置中,将在GPU上执行任务所

需的所有数据传送到GPU可存取的存储器,之后开始对所述任务的执行,且由在GPU上执行的任务产生的数据不可用于主机CPU,直到在GPU上执行的任务完成执行之后方可。主机CPU与GPU之间的数据共享上的此粗糙度可阻止对用于基于并行的应用的许多有用操作的实施,例如,在于主机装置上执行的进程与在GPU上执行的任务之间传递进程间消息。此些消息(例如)对于允许在GPU上运行的任务具有在主机CPU上执行远程过程调用(RPC)的能力可为有用的。

[0057] 当命令队列接口经配置以提供对命令的无序执行时,在特定任务的执行期间,主机进程不能够控制何时将发生对特定命令的执行。因此,用于命令队列的无序执行模式实际上也不允许在于主机装置上执行的进程与在GPU上执行的任务之间实施进程间消息传递。

[0058] 关于用于OpenCL中的存储器模型,API界定所谓的全局CL缓冲器和全局CL图像,其可用于在主机CPU与GPU之间共享数据或用于在多个OpenCL计算装置之间共享数据。然而,CPU和GPU无法同时从缓冲器进行读取或写入到缓冲器。通常,CPU准备好含有源数据的一个或一个以上缓冲器,且将所述缓冲器传递到GPU以供处理。GPU修改这些缓冲器或将结果放置在还曾由在CPU上执行的软件先验分配的其它缓冲器中,以用于接收GPU数据修改。

[0059] 虽然OpenCL中的存储器对象当前允许将主机存储器空间的区用于存储由计算装置使用的缓冲器数据,但所述规范允许计算装置对此数据进行高速缓存以用于对任务的更高效的执行。主机装置一般不能直接使用用于对缓冲器数据进行高速缓存的计算装置高速缓存存储器无效。因此,即使主机装置将盖写存储于主机存储器空间中的某些存储器缓冲器数据,也不能保证计算装置中的高速缓存存储器将得到更新以向计算装置提供对经修改数据的直接存取。另外,因为由计算装置执行的计算的结果可被存储在计算装置高速缓存存储器中,所以在主机装置上执行的主机进程不能从缓冲器读取任何部分结果,因为此类数据可能归因于计算装置高速缓存存储器中所存储的较新的数据而无效。因此,OpenCL中的存储器管理模型未容易地经由共享存储器实现运行中的数据共享。

[0060] 在一些实例中,可使用本发明中所描述的技术来克服OpenCL API的上文提及的限制中的一者或一者以上。举例来说,本发明的技术可提供软件接口,所述软件接口可用于支持在使用任务级粒度命令队列起始任务的多处理器计算平台内的进程间消息传递。作为另一实例,本发明的技术可提供软件接口,所述软件接口可用于支持经由多处理器计算平台内的共享存储器的运行中的数据共享。

[0061] 在一些实例中,本发明的技术可提供促进软件级消息传递的GPU硬件架构。举例来说,本发明的技术可提供经配置以支持对软件级消息传递指令的执行的GPU硬件架构。在进一步的实例中,本发明的技术可提供促进GPU与主机CPU之间的共享存储器通信的GPU硬件架构。举例来说,本发明的技术可提供经配置以针对共享存储器空间选择性地启用和停用高速缓存服务且/或针对共享存储器空间选择性地启用和停用高速缓存存储器-相关性机制。

[0062] 根据本发明的第一方面,提供一种消息传递接口,所述消息传递接口促进在由计算装置执行任务期间在主机装置与一个或一个以上计算装置之间执行消息传递指令。消息传递可指代一种形式的进程间以及潜在地装置间的通信,其中正通信的进程各自执行互补组的操作以成功地传递消息。举例来说,根据消息传递协议进行通信的进程中的每一者可

实施发送操作和接收操作。本发明中的消息传递技术可允许CPU和计算装置(例如,GPU)在于计算装置上执行任务期间在彼此之间传递消息。以此方式,实施任务级粒度命令队列通信方案的多处理器计算平台可以能够促进进程间和/或装置间通信。

[0063] 在一些实例中,本发明中所描述的消息传递技术可被称作“带外信令”技术,因为所述技术使用不同于命令队列接口的接口,命令队列接口通常用于OpenCL中以用于在主机装置与计算装置(例如,GPU)之间的通信。换句话说,本发明的技术可包含新的带外通信接口,其与OpenCL内所包含的带内命令队列接口在逻辑上分离。带外通信接口可不经受命令队列接口所经受的相同任务级粒度,进而提供对上文相对于命令队列的任务级粒度所描述的一个或一个以上限制的解决方案。

[0064] 根据本发明的技术在CPU与GPU之间传送的消息可为任何类型的消息。不同类型的消息的实例包含信号、存储器分配请求、存储器取消分配请求、通知消息、同步消息、远程过程调用消息(例如,作为远程过程调用(RPC)的部分的消息)、数据包、报告消息、断言机制消息,以及记录消息。

[0065] 在当前的OpenCL范例中,从主机CPU到GPU的所有请求均在OpenCL命令队列中排队等候,且随后被发送到GPU。具体来说,应用可能将大量内核执行和缓冲器操作排在命令队列中。同时,如果首先被排队的任务(例如,内核执行)需要(例如)向CPU请求额外的存储器分配,那么就出现问题。首先,GPU如何在运行内核中向CPU通知其需要进行存储器分配?其次,CPU如何向GPU通知存储器分配的完成以及新分配的存储器块的地址?然而,本发明的消息传递接口技术可能通过允许含有上述通知和信息的一个或一个以上消息在CPU与GPU之间传递来解决这些问题。

[0066] 在一些实例中,可使用本发明的带外信令技术在主机CPU与一个或一个以上计算装置(例如,OpenCL计算装置)之间实施信令。带外信令可例如使用推拉机制来提供快速的带外通知。在一些实例中,带外信令技术可携带相对少量的数据。

[0067] 根据本发明的第二方面,提供能够将消息发送到在不同于GPU的处理器上执行的进程以及从所述进程接收消息的GPU。举例来说,GPU可包含经配置以实施用于发送和接收消息的一个或一个以上操作的硬件。在一些实例中,根据本发明而设计的GPU可包含一个或一个以上主机可存取的寄存器,所述寄存器经配置以存储与消息传递协议相关联的状态和数据信息。所述一个或一个以上寄存器可经配置以促进在GPU上执行的任务与在不同于GPU的装置上执行的进程之间的消息传递。在进一步的实例中,GPU的ALU处理块(例如,可编程着色器单元)可通信地耦合到主机可存取的寄存器以经由所述主机可存取的寄存器来发送和接收消息。GPU还可经设计以包含各种轮询和/或中断机制以实施同步和/或异步消息传递技术。

[0068] 根据本发明的第三方面,提供存储器缓冲器接口,其允许创建直接存储器对象。直接存储器对象可用于实施非可高速缓存共享存储器空间和/或高速缓冲存储器相干共享存储器空间,以便在于计算装置上执行的任务正在计算装置上执行时在于主机装置上执行的进程与所述任务之间共享数据。所述共享存储器空间可为可由主机装置以及计算装置(例如,GPU)两者在计算装置执行任务期间存取的存储器空间。如本文中所使用的非可高速缓存的共享存储器空间可指代针对所述存储器空间而停用主机装置和计算装置中的一者或两者中的一个或一个以上对应高速缓冲存储器的共享存储器空间。如本文中所使用的高速

缓冲存储器相干共享存储器空间可指代其中使用共享存储器高速缓冲存储器相干技术来维持主机装置和计算装置中的一者或两者中的一个或一个以上对应高速缓冲存储器内的高速缓冲存储器相干的共享存储器空间。所述非可高速缓存共享存储器空间以及高速缓冲存储器相干共享存储器空间可在任何时间允许数据共享。在一些实例中,可将直接存储器对象实施为非可高速缓存易失性共享存储器和/或实施为高速缓冲存储器相干易失性共享存储器来用于主机装置和计算装置。

[0069] 在一些实例中,本发明的直接存储器对象可被集成在包含存储器对象存储器管理方案的跨平台、跨供应商、异类计算平台、并行编程API内。举例来说,可将直接存储器对象集成到OpenCL中以作为OpenCL存储器对象的额外属性,例如OpenCL缓冲器对象或OpenCL图像对象。在这些实例中,可通过修改存储器对象创建功能以包含一参数或旗标来创建直接存储器对象,所述参数或旗标指定由功能调用创建的所得的存储器对象是否应为标准模式存储器对象或直接模式存储器对象。以此方式,本发明的技术可允许实施包含若干存储器对象存储器管理方案(例如,OpenCL)以经由不经受高速缓冲存储器相干性问题的共享存储器空间来实施运行中的数据共享的API的多处理器计算平台。

[0070] 在进一步的实例中,本发明的直接存储器对象可用于主机CPU与OpenCL计算装置之间或者不同的OpenCL计算装置之间的运行中的数据共享。在额外的实例中,直接存储器对象可含有内部同步标记。在进一步的实例中,可与带外信号一起使用直接存储器对象以用于同步。

[0071] 根据本发明的第四方面,提供包含对应于共享存储器空间的高速缓冲存储器的GPU,所述共享存储器空间可针对特定存储器地址空间而被选择性地停用以便提供非可高速缓存共享存储器空间。举例来说,GPU可响应于接收到指定是否应使用高速缓存服务来相对于共享存储器空间执行读取操作和/或写入操作的信息而启用和停用由与共享存储器空间相关联的高速缓冲存储器提供的高速缓存服务。在一些实例中,指定是否应使用高速缓存服务来相对于共享存储器空间执行读取操作和/或写入操作的信息可为高速缓存模式指令或直接模式指令,其指定是否应使用高速缓存模式或直接模式来执行特定指令。在进一步的实例中,指定是否应使用高速缓存服务来相对于共享存储器空间执行读取操作和/或写入操作的信息可为直接模式存储器对象属性,其指定是否针对存储器对象启用直接模式。

[0072] 在进一步的实例中,本发明的技术可提供包含高速缓冲存储器相干性模式的GPU,所述高速缓冲存储器相干模式可被选择性地启用以提供高速缓冲存储器相干共享存储器空间。在一些实例中,GPU可基于从主机装置接收到的一个或一个以上指令来选择性地启用高速缓冲存储器相干模式以用于对应于共享存储器空间的高速缓冲存储器的一部分。在主机装置基于由主机进程指定的直接模式参数而分配共享存储器空间之后,主机装置可即刻向GPU发布一个或一个以上指令以选择性地启用共享存储器空间高速缓冲存储器相干性模式以用于对应于共享存储器空间的高速缓冲存储器的一部分。

[0073] 与可通过单独使用OpenCL命令队列接口而获得的主机CPU与GPU之间或两个OpenCL计算装置之间的任务耦合相比,本发明的带外信令和直接缓冲技术可提供更精细粒度的任务耦合。本发明的技术可允许多处理器计算平台执行多种操作以便辅助并行和/或多线程程序的高效执行。举例来说,本发明的技术可允许在GPU上执行的任务启动RPC。作为

另一实例,本发明的技术可允许在GPU上执行的任务经由CPU来启动另一GPU任务。作为进一步的实例,本发明的技术可允许在GPU上执行的任务向CPU和/或在CPU上执行的驱动器发布资源管理请求,例如存储器分配和/或存储器取消分配请求。作为又一实例,本发明的技术可允许在GPU上执行的任务执行状态检查和到CPU的一般消息传递,例如断言机制的实施、进展报告,和/或诊断记录。

[0074] 图1是说明根据本发明的实例性计算系统10的方框图。计算系统10经配置以在多个处理装置上处理一个或一个以上软件应用。在一些实例中,所述一个或一个以上软件应用可包含主机进程,且计算系统10可经配置以执行主机进程且分布由在计算系统10内的其它计算装置上的主机进程起始的一个或一个以上任务的执行。在进一步的实例中,可根据并行编程模型来编程由计算系统10执行的主机进程和/或任务。举例来说,所述应用可包含经设计以充分利用基础硬件系统的任务级并行度和/或数据级并行度的指令。

[0075] 计算系统10可为个人计算机、桌上型计算机、膝上型计算机、计算机工作站、视频游戏平台或控制台、移动电话(例如,蜂窝式或卫星电话)、移动电话、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))、数字媒体播放器(例如,个人音乐播放器)、视频播放器、显示装置,或电视、电视机顶盒、服务器、中间网络装置、大型计算机或处理信息的任何其它类型的装置。

[0076] 计算系统10包含主机装置12、图形处理单元(GPU)14、存储器16和互连网络18。主机装置12经配置以提供用于执行用于多处理器计算平台API的主机进程和运行时模块的平台。通常,主机装置12是通用CPU,但主机装置12可为能够执行程序的任何类型的装置。主机装置12经由互连网络18通信地耦合到GPU14和存储器16。主机装置12包含主机进程20和运行时模块22,主机进程20和运行时模块22中的每一者可在一个或一个以上可编程处理器的任何组合上执行。

[0077] 主机进程20包含形成用于在计算系统10的计算系统平台上执行的软件程序的一组指令。所述软件程序可经设计以执行用于终端用户的一个或一个以上特定任务。在一些实例中,这些任务可涉及可利用由计算系统10提供的多个处理装置和并行架构的计算密集算法。

[0078] 运行时模块22可为在主机装置12上执行的软件模块,其实施经配置以服务于主机进程20中所包含的指令中的一者或一者以上的一个或一个以上接口。由运行时模块22实施的接口包含命令队列接口24和主机消息传递接口26。在一些实例中,运行时模块22可实施除了本发明中所描述的接口之外的标准多处理器系统API内所包含的一个或一个以上接口。在一些实例中,所述标准API可为异类计算平台API、跨平台API、跨供应商API、并行编程API、任务级并行编程API和/或数据级并行编程API。在进一步的实例中,所述标准API可为OpenCLAPI。在此些实例中,可将运行时模块22设计成遵照OpenCL规范中的一者或一者以上。在额外的实例中,可将运行时模块22实施为驱动器程序(例如,GPU驱动器)的一部分或实施为驱动器程序。

[0079] 命令队列接口24经配置以从主机进程20接收一个或一个以上排队指令,且执行由所接收的指令指定的功能。在一些实例中,可根据OpenCL规范来设计命令队列接口24。举例来说,命令队列接口24可实施OpenCL规范中所指定的排队指令中的一者或一者以上以用于与命令队列交互。

[0080] 根据本发明,主机消息传递接口26经配置以从主机进程20接收一个或一个以上消息传递指令,且执行由所接收的指令指定的功能。在一些实例中,可将主机消息传递接口26实施为对现有标准API(例如,OpenCLAPI)的扩展。在额外的实例中,可将主机消息传递接口26集成到现有标准API(例如,OpenCLAPI)中。

[0081] GPU14经配置以响应于从主机装置12接收到的指令来执行一个或一个以上任务。GPU14可为包含一个或一个以上可编程处理元件的任何类型的GPU。举例来说,GPU14可包含经配置以并行地执行任务的多个执行实例的一个或一个以上可编程着色器单元。可编程着色器单元可包含顶点着色器单元、片段着色器单元、几何着色器单元和/或统一着色器单元。GPU14经由互连网络18通信地耦合到主机装置12和存储器16。GPU14包含任务28和装置消息传递接口30。任务28和装置消息传递接口30可在一个或一个以上可编程处理元件的任何组合上执行。

[0082] 任务28包括形成用于在计算系统10中的计算装置上执行的任务的一组指令。在一些实例中,用于任务28的所述组指令可在主机进程20中界定,且在一些情况下,由在主机装置12上执行的主机进程20中所包含的指令编译。在进一步的实例中,任务28可为具有在GPU14上并行地执行的多个执行实例的内核程序。在此些实例中,主机进程20可界定用于内核的索引空间,其将内核执行实例映射到用于执行内核执行实例的相应处理元件,且GPU14可根据为内核界定的索引空间来执行用于任务28的多个内核执行实例。

[0083] 根据本发明,装置消息传递接口30经配置以从主机进程20接收一个或一个以上消息传递指令,且执行由所接收的指令指定的功能。在一些实例中,可将装置消息传递接口30实施为对现有标准API的扩展。举例来说,所述标准API可为标准计算装置API,例如OpenCL C API。在额外的实例中,可将装置消息传递指令30集成到现有标准API(例如,OpenCL C API)中。

[0084] 存储器16经配置以存储数据以供主机装置12和GPU14中的一者或两者使用。存储器16可包含一个或一个以上易失性或非易失性存储器或存储装置的任何组合,所述易失性或非易失性存储器或存储装置例如为随机存取存储器(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、磁性数据存储媒体或光学存储媒体。存储器16经由互连网络18通信地耦合到主机装置12和GPU14。存储器16包含命令队列32。

[0085] 命令队列32可为实施于存储器16中的数据结构,存储器16存储并检索从命令队列接口24接收到的命令。在一些实例中,命令队列32可为以特定次序存储命令以用于执行的缓冲器。

[0086] 互连网络18经配置以促进主机装置12、GPU14与存储器16之间的通信。互连网络18可为此项技术中已知的任何类型的互连网络。在图1的实例性计算系统10中,互连网络18是总线。所述总线可包含多种总线结构中的任一者中的一者或一者以上,例如第三代总线(例如,超传输总线或不限带宽总线)、第二代总线(例如,高级图形端口总线、外围组件互连快速(PCIe)总线,或高级可扩展接口(AXI)总线),或任何其它类型的总线。互连网络18耦合到主机装置12、GPU14和存储器16。

[0087] 现在将进一步详细地描述计算系统10中的组件的结构和功能性。如上文所论述,主机进程20包含一组指令。所述组指令可包含(例如)一个或一个以上排队指令,以及一个

或一个以上主机消息传递指令。在额外的实例中,所述组指令可包含指定将在GPU14上执行的任务或内核的指令、创建命令队列且使命令队列与特定装置相关联的指令、编译并捆绑程序的指令、设置内核自变量的指令、界定索引空间的指令、界定装置背景的指令,以及支持由主机进程20提供的功能性的其它指令。

[0088] 主机进程20可通过向命令队列接口24发布指令命令队列接口24将一个或一个以上命令放置到命令队列32中的一个或一个以上排队指令而与命令队列接口24交互。所述一个或一个以上排队指令可包含指令命令队列接口24将存储器传送命令排到命令队列32中的存储器传送排队指令。举例来说,所述一个或一个以上排队指令可包含用以将一命令排队的指令,所述命令指令主机装置12(例如,在主机装置12上执行的运行时模块22)在与主机装置12相关联的存储器空间与和GPU14相关联的存储器空间之间传送数据。

[0089] 如果存储器空间在主机装置12执行主机进程20期间可由主机装置12存取,那么存储器空间可与主机装置12相关联。类似地,如果存储器空间在GPU14执行任务28期间可由GPU14存取,那么存储器空间可与GPU14相关联。与主机装置12相关联的存储器空间可在本文中被称作主机存储器空间,且与GPU14相关联的存储器空间可在本文中被称作装置存储器空间。在一些实例中,存储器16可包含主机存储器空间和装置存储器空间两者的部分。在进一步的实例中,主机存储器空间和装置存储器空间中的一者或两者的部分可位于图1的计算系统10中未展示的一个或一个以上其它存储器装置上。

[0090] 在一些实例中,指令主机装置12在与主机装置12相关联的存储器空间与和GPU14相关联的存储器空间之间传送数据的命令可为指令运行时模块22将存储于主机存储器空间的一部分中的数据传送到分配于装置存储器空间中的缓冲器对象的命令。由主机进程20发布以将此命令排队的指令可在本文中被称作写入缓冲器排队指令。在一些情况下,写入缓冲器排队指令可采取由OpenCL API规范指定的`clEnqueueWriteBuffer()`功能的形式。

[0091] 在额外的实例中,指令主机装置12在与主机装置12相关联的存储器空间与和GPU14相关联的存储器空间之间传送数据的命令可为指令运行时模块22将存储于分配于装置存储器空间中的缓冲器对象中的数据传送到主机存储器空间的一部分的命令。由主机进程20发布以将此命令排队的指令可在本文中被称作读取缓冲器排队指令。在一些情况下,读取缓冲器排队指令可采取由OpenCL API规范制定的`clEnqueueReadBuffer()`功能的形式。

[0092] 所述一个或一个以上排队指令还可包含指令命令队列接口24将任务执行命令排队到命令队列32中的任务执行排队指令。举例来说,所述一个或一个以上排队指令可包含用以将一命令排队的指令,所述命令指令主机装置12(例如,在主机装置12上执行的运行时模块22)在GPU14上执行任务。在一些实例中,用以执行任务的命令可为在GPU14的多个处理元件上并行地执行任务的多个执行实例的命令。举例来说,所述任务可为内核,主机进程20可界定用于内核的索引空间,其将内核执行实例映射到GPU14中的用于执行内核执行实例的相应处理元件。在此实例中,用以执行任务的命令可为用以根据为GPU14界定的索引空间在GPU14上执行内核的命令。在一些情况下,任务执行排队指令可采取由OpenCL API指定的`clEnqueueNDRangeKernel()`功能的形式。

[0093] 根据本发明,主机进程20还可通过向主机消息传递接口26发布指令主机消息传递接口26在于主机装置12上执行的主机进程20与在GPU14上执行的任务28之间传递一个或一

个以上消息的一个或一个以上主机消息传递指令而与主机消息传递接口26交互。所述主机消息传递指令可由主机装置12执行。

[0094] 在一些实例中,主机消息传递指令可包含指令主机装置12将指定数据发送到指定装置的发送指令。举例来说,所述发送指令可指令主机消息传递接口26将消息从在主机装置12上执行的主机进程20发送到在GPU14上执行的任务28。在一些实例中,所述发送指令可包含指定应将消息发送到其的特定装置的第一输入参数,以及指定将发送的消息的内容的第二输入参数。

[0095] 所述发送指令可为封锁发送指令或非封锁发送指令。在一些实例中,所述发送指令可包含指定所述发送指令是封锁发送指令还是非封锁发送指令的第三输入参数。封锁发送指令可在完成发送操作之前一直进行等待,之后返回到调用进程,例如在主机装置12上执行的主机进程20。非封锁发送指令可返回到调用进程,而不在完成发送操作之前一直等待。举例来说,非封锁发送指令返回到特定发送操作的句柄,可由调用进程随后询问所述句柄以确定发送操作是否成功。非封锁发送操作可能失败,且在失败的情况下,调用进程可需要再次发布发送指令以重试发送操作。

[0096] 在一些实例中,用于发送指令的接口可采取以下形式:

```
clSendOutOfBandData(  
    cl_device *deviceId,  
    int OOB_data,  
    bool blocking)
```

[0098] 其中clSendOutOfBandData是指令识别符,cl_device*deviceId是指定应将消息发送到其的特定OpenCL装置的输入参数,int OOB_data是指定将发送的消息的内容的输入参数,且bool blocking是指定指令是封锁发送指令还是非封锁发送指令的输入参数。在封锁指令的情况下,指令可返回指示发送操作是否成功完成的参数。在非封锁指令的情况下,指令可返回用于由调用进程进行后续状态询问的句柄参数。

[0099] 在一些实例中,主机消息传递指令可包含寄存回调例程指令,寄存回调例程指令指令主机装置12寄存回调,从而以异步的方式从指定装置接收数据。举例来说,寄存回调例程指令可响应于从GPU14接收到指示在GPU14上执行的任务已将一消息发送到主机进程20的信号而指令主机消息传递接口26调用回调例程。寄存回调例程指令可包含指定应寄存回调例程的特定装置的第一输入参数,以及指定回调例程的存储器位置的第二输入参数。

[0100] 在一些实例中,用于寄存回调例程的接口可采取以下形式:

```
clRegisterOutOfBandDataCallback(  
    cl_device*deviceId,  
    void(*)(int)callbackPtr)
```

[0104] 其中clRegisterOutOfBandDataCallback是指令识别符,cl_device*deviceId是指定应将消息发送到其的特定OpenCL装置的输入参数,且void(*)(int)callbackPtr是指定回调例程的存储器位置的输入参数。寄存回调例程指令可返回指示回调例程寄存操作是否成功完成的参数。

[0105] 在一些实例中,主机消息传递指令可包含指令主机装置12尝试从指定装置读取数据的轮询指令。举例来说,轮询指令可指令主机消息传递接口26针对指示在GPU14上执行的

任务28是否已发送消息的消息状态信息来轮询GPU14。轮询指令可包含指定将轮询的特定装置的输入参数,以及指定由于轮询而获得的数据的输出参数。

[0106] 在一些实例中,用于轮询指令的接口可采取以下形式:

[0107] `clTryReadOutOfBandData(`

[0108] `cl_device*deviceId,`

[0109] `int*oob_data)`

[0110] 其中`ClTryReadOutOfBandData`是指令识别符,`cl_device*deviceId`是指定将轮询的特定OpenCL装置的输入参数,且`int*oob_data`是指定由于轮询而获得的数据的输出参数。轮询指令可返回指示是否从轮询操作成功地获得数据的参数。

[0111] 类似于主机进程20,任务28可包含由计算装置执行的一个或一个以上装置消息传递指令。所述装置消息传递指令可包含指令计算装置将指定数据发送到主机装置12的发送指令。举例来说,发送指令可指令GPU14将消息从在GPU14上执行的任务28发送到在主机装置12上执行的主机进程20。

[0112] 所述发送指令可为封锁发送指令或非封锁发送指令。在一些实例中,所述发送指令可包含指定所述发送指令是封锁发送指令还是非封锁发送指令的第一输入参数。封锁发送指令可中止调用进程(例如,在GPU14上执行的任务28),且等待第二操作完成,之后返回到调用进程。非封锁发送指令可返回到调用进程,而不在完成发送操作之前一直等待。举例来说,非封锁发送指令返回到特定发送操作的句柄,可由调用进程随后询问所述句柄以确定发送操作是否成功。非封锁发送操作可能失败,且在失败的情况下,调用进程可能需要再次发布发送指令以重试发送操作。发送指令可包含指定将发送到主机装置的消息的内容的第二输入参数。

[0113] 在一些实例中,用于发送指令的接口可采取以下形式:

[0114] `send_oobdata(`

[0115] `bool blocking,`

[0116] `int data)`

[0117] 其中`send_oobdata`是指令识别符,`bool blocking`是指定指令是封锁发送指令还是非封锁发送指令的输入参数,且`int data`是指定将发送的消息的内容的输入参数。在封锁指令的情况下,指令可返回指示发送操作是否成功完成的参数。在非封锁指令的情况下,指令可返回用于由调用进程进行后续状态询问的句柄参数。

[0118] 在一些实例中,装置消息传递指令可包含指令计算装置从主机装置12接收数据的接收指令。举例来说,接收指令可指令GPU14(例如,装置消息传递接口30)向在GPU14上执行的任务28提供从在主机装置12上执行的主机进程20发送到任务28的消息(如果可用)。此指令可用于支持轮询机制。

[0119] 所述接收指令可为封锁接收指令或非封锁接收指令。在一些实例中,所述接收指令可包含指定所述接收指令是封锁接收指令还是非封锁接收指令的输入参数。封锁接收指令可中止调用进程(例如,在GPU14上执行的任务28),且一直等到消息可用,之后返回到调用进程。非封锁接收指令可返回到调用进程,而不一直等到消息可用。举例来说,如果消息可用,那么非封锁接收指令可返回所述消息。然而,如果消息不可用,那么非封锁接收指令可失败。在失败的情况下,调用进程可能需要再次发布接收指令以重试接收操作。接收指令可

包含指定由于接收操作而获得的数据(如果有)的输出参数。

[0120] 在一些实例中,用于接收指令的接口可采取以下形式:

[0121] receive_oobdata(

[0122] bool blocking,

[0123] int data)

[0124] 其中receive_oobdata是指令识别符,bool blocking是指定指令是封锁接收指令还是非封锁接收指令的输入参数,且int data是指定由于接收操作而获得的数据(如果有)的输出参数。所述指令可返回指示接收操作是否成功的参数。

[0125] 命令队列接口24经配置以将命令排到命令队列32中。举例来说,命令队列接口24可从主机进程20接收一个或一个以上排队指令,且将响应于从主机进程20接收一个或一个以上排队指令而将一个或一个以上命令放置到命令队列32中。所述一个或一个以上排队指令可包含任务执行排队指令以及数据传送排队指令,其分别指令命令队列接口24对任务执行命令和数据传送命令进行排队。

[0126] 命令队列接口24还经配置以执行存储于命令队列32中的命令。对于数据传送命令,命令队列接口24可在主机存储器空间与装置存储器空间之间传送数据。举例来说,对于写入缓冲器命令,命令队列接口24可将存储于主机存储器空间的一部分中的数据传送到在装置存储器空间中分配的缓冲器对象。作为另一实例,对于读取缓冲器命令,命令队列接口24可将存储于在装置存储器空间中分配的缓冲器对象中的数据传送到主机存储器空间的一部分。装置存储器空间可对应于命令队列32与其相关联的装置。

[0127] 对于任务执行命令,命令队列接口24可致使在与命令队列相关联的装置上开始任务的执行。举例来说,在图1的实例中,命令队列32与运行时模块22的背景内的GPU14相关联。因此,当执行任务执行命令时,命令队列接口24可致使任务开始在GPU14上执行。在一些实例中,命令队列接口24可通过将一个或一个以上命令放置到GPU14内所包含的本地命令队列中而致使任务开始在GPU14上执行。在其它实例中,命令队列接口24可通过将指令GPU14开始执行任务的一个或一个以上指令发送到GPU14而致使任务开始在GPU14上执行。命令队列接口24可使用互连网络18来与GPU14、存储器16和主机存储器空间和装置存储器空间通信。

[0128] 在一些实例中,命令队列接口24可按序执行命令。在此些实例中,如果将第一命令排在第二命令之前,那么第二命令的执行在第一命令已完成执行后开始。在进一步的实例中,命令队列接口24可无序执行命令。在此些实例中,如果将第一命令排在第二命令之前,那么第二命令的执行不一定在第一命令已完成执行后才开始。

[0129] 主机消息传递接口26经配置以执行从主机进程20接收的一个或一个以上消息传递指令。举例来说,响应于从主机进程20接收到一个或一个以上消息传递指令,主机消息传递接口26可在于GPU14上执行的任务28正在GPU14上执行时在于主机装置12上执行的主机进程20与任务28之间传递一个或一个以上消息。在一些实例中,主机消息传递接口26可在未将任何命令放置到命令队列32中的情况下执行一个或一个以上消息传递指令。

[0130] 根据第一实例,响应于从主机进程20接收到发送指令,主机消息传递接口26可在任务28正在GPU14上执行时将消息从主机进程20发送到任务28。举例来说,主机消息传递接口26可基于在发送指令内所包含的消息数据来撰写传出消息,且经由互连网络18将传出消

息传送到在发送指令中指定的装置(例如,GPU14),以用于递送到在指定装置上执行的任务(例如,任务28)。

[0131] 根据第二实例,响应于从主机进程20接收到寄存回调例程指令,主机消息传递接口26可使指令中所指定的回调例程与来自指令中所指定的装置(例如,GPU14)的信号相关联,从而指示在指定装置上执行的任务(例如,任务28)已发送消息。在一些实例中,来自装置的信号可为中断信号。在一些实例中,可经由专用中断信号线来递送中断信号。响应于从指定装置接收到指示在装置上执行的任务已发送消息的信号,主机消息传递接口26可起始在寄存回调例程指令中指定的回调例程的执行。回调例程可从指定装置(例如,GPU14)获得由任务(例如,任务28)发送的消息,且将所述消息返回到主机进程20以供进一步处理。

[0132] 根据第三实例,响应于接收到轮询指令,主机消息传递接口26可针对消息状态信息来轮询指令中所指定的装置(例如,GPU14)。主机消息传递接口26可使用互连网络18或另一基于硬件的通信路径来轮询装置。如果消息状态信息指示在指定装置(例如,GPU14)上执行的任务(例如,任务28)已发送消息,那么主机消息传递接口26可从指定装置获得消息,且将所述消息返回到主机进程20以供进一步处理。

[0133] 装置消息传递接口30经配置以执行从任务28接收的一个或一个以上装置消息传递指令。举例来说,响应于从任务28接收到一个或一个以上装置消息传递指令,装置消息传递接口30可在于GPU14上执行的任务28正在GPU14上执行时在任务28与在主机装置12上执行的主机进程20之间传递一个或一个以上消息。

[0134] 根据第一实例,响应于接收到发送指令,装置消息传递接口30可将消息从在GPU14上执行的任务28发送到在主机装置12上执行的主机进程20。举例来说,装置消息传递接口30可基于在发送指令内所包含的消息数据来撰写传出消息,且经由互连网络18将传出消息传送到在主机装置12以供递送到主机进程20。

[0135] 根据第二实例,响应于从任务28接收到接收指令,装置消息传递接口30可确定来自主机进程20的消息是否可用。在一些实例中,装置消息传递接口30可检查一个或一个以上主机可存取的寄存器以确定消息是否可用。如果来自主机进程20的消息可用,那么装置消息传递接口30可将所述消息提供到任务28。

[0136] 虽然将命令队列接口24和主机消息传递接口26说明为与图1中的主机进程20分离的组件,但在一些实例中,命令队列接口24和主机消息传递接口26中的一者或两者的功能性可部分地且/或完全编译到主机进程20中。类似地,在一些实例中,装置消息传递接口30的功能性可部分地且/或完全编译到任务28中。

[0137] 为了易于说明,图1中所说明的实例性计算系统10描述将GPU14用作计算装置的本发明的消息传递技术。然而,应认识到,可将本发明的技术应用于具有不同于除了GPU14之外或作为GPU14的替代的GPU的计算装置的多处理器计算系统。在一些实例中,计算装置可为OpenCL计算装置。OpenCL计算装置包含一个或一个以上计算单元。计算单元中的每一者包含一个或一个以上处理元件。举例来说,计算单元可为具有可由计算单元中的所有处理元件使用的芯片上共享存储器的处理元件(例如,ALU)的群集。工作项目可为由放置到命令队列中的命令在OpenCL计算装置上调用的内核或任务的多个并行执行中的一者。每一工作项目可在计算单元中的个别处理元件上与在其它处理元件上执行的其它工作项目并行地执行。工作群组可为在计算装置内的单一计算单元上作为单一内核执行命令的部分而处理

的一个或一个以上工作项目的集合。OpenCL主机可为平台的用于运行OpenCL运行时层的中央CPU。

[0138] OpenCL API可提供一组共同接口以用于主机装置与不同类型的计算装置之间的交互。举例来说,OpenCL API可提供共同接口以用于主机与GPU计算装置以及主机与非GPU计算装置之间的交互。OpenCL API允许主机使用共同接口来用于在各种计算装置上执行任务(例如,OpenCL内核)。在一些实例中,任务可为通用计算任务,且OpenCLAPI可允许主机致使通用计算任务在GPU计算装置上执行。

[0139] 图1中所示的实例性计算系统10说明用于促进主机装置与计算装置之间的消息传递和/或带外信令的基础结构和技术。然而,在其它实例性计算系统中,可容易将所述技术扩展以提供在具有一个以上计算装置的计算系统中的不同计算装置(例如,OpenCL计算装置)之间的运行中的消息传递。在此些实例中,可在不同的计算装置之间布线一个或一个以上中断线。

[0140] 图2是说明根据本发明的可用于图1的计算系统10中的实例性GPU40的方框图。在一些实例中,GPU40可用于实施图1中所说明的GPU14。GPU40包含GPU处理块42、主机可存取的GPU寄存器44和总线控制器46。GPU40可经由互连网络18通信地耦合到一个或一个以上其它主机装置或计算装置。

[0141] GPU处理块42经配置以执行任务,且促进在GPU处理块42上执行的任务与在其它主机或计算装置上执行的进程之间的消息传递。GPU处理块42(例如)经由一个或一个以上控制和/或数据线而通信地耦合到主机可存取的GPU寄存器44。在一些实例中,GPU处理块42可被称作算术逻辑单元(ALU)块。GPU处理块42包含任务48、消息传递模块50、传入数据寄存器52和传出数据寄存器54。

[0142] 主机可存取的GPU寄存器44经配置以存储可被传送到主机装置或从主机装置传送的数据。主机可存取的GPU寄存器44包含消息状态寄存器56、消息计数寄存器58、传入消息寄存器60、传出消息寄存器62、中断状态寄存器64和中断确认寄存器66。主机可存取的GPU寄存器44中的每一者可由主机装置(例如,图1中的主机装置12)存取的。在一些实例中,主机可存取的GPU寄存器44可为存储器映射的寄存器,即被映射到主机装置的存储器空间且可在所述存储器空间中寻址的寄存器。在进一步的实例中,主机可存取的GPU寄存器44可为输入/输出映射(I/O映射)的寄存器,即被映射到主机装置的I/O空间的寄存器。主机可存取的GPU寄存器44经由一个或一个以上控制和/或数据线而通信地耦合到GPU处理块42。主机可存取的GPU寄存器44还经由互连网络18而通信地耦合到总线控制器46。

[0143] 任务48可在一个或一个以上可编程处理器上执行。在一些实例中,GPU处理块42可包含经配置以执行任务48的多个执行实例的多个处理器或处理元件。任务48可实质上类似于上文相对于图1所描述的任务28,且因此将不进一步详细地描述。

[0144] 消息传递模块50经配置以控制由GPU40执行的消息传递操作。可以硬件、软件、固件或其任何组合来实施消息传递模块50。在一些实例中,如果消息传递模块50的功能性中的一些或全部是以软件实施,那么用于此实施方案的软件指令可包含在与含有用于任务48的软件指令的可执行文件相同的可执行文件内。消息传递模块50通信地耦合到任务48、消息传递模块50、传入数据寄存器52和传出数据寄存器54。

[0145] 消息传递模块50可在于一个或一个以上处理器上执行的任务48正在一个或一个

以上处理器上执行且响应于从任务48接收到一个或一个以上消息传递指令而经由主机可存取的GPU寄存器44在任务48与在主机装置上执行的进程之间传递一个或一个以上消息。在一些实例中,所述一个或一个以上消息传递指令可包含指令消息传递模块50将消息从任务48发送到在主机装置上执行的进程的发送指令。在这些实例中,消息传递模块50可将与所述消息相关联的消息数据存储在主可存取的GPU寄存器44中的一者中。在进一步的实例中,所述一个或一个以上消息传递指令可包含指令消息传递模块50向任务48提供从在主机装置上执行的进程发送到任务48的消息的接收指令。在这些实例中,消息传递模块50可从主可存取的GPU寄存器44中的一者或一者以上获得与所述消息相关联的消息数据。

[0146] 在图2的实例中,传入数据寄存器52是存储经由传入消息寄存器60从外部装置接收的传入数据的硬件寄存器。传入数据寄存器52还可存储指示传入数据寄存器52中的数据是否已被消耗且/或传入数据寄存器52中的数据是否以用于读取的状态位。传入数据寄存器52经由一个或一个以上数据线而通信地耦合到传入消息寄存器60。在一些实例中,数据线的数目可等于传入数据寄存器52中的位的数目,这两个数目可等于消息中的位的数目。在进一步的实例中,位的数目可为32位。在一些实例中,GPU处理块42可实施内部先入先出(FIFO)缓冲器来存储从传入数据寄存器52接收的多个传入消息。

[0147] 在图2的实例中,传出数据寄存器54是存储从由任务48发布的一个或一个以上消息传递指令接收的传出数据的硬件寄存器。传出数据寄存器54经由一个或一个以上数据线而通信地耦合到传出消息寄存器62。在一些实例中,数据线的数目可等于传出数据寄存器54中的位的数目,这两个数目可等于消息中的位的数目。在一些实例中,传出数据寄存器54和传出消息寄存器62可经配置以使得当消息传递模块50将数据写入到传出数据寄存器54时,自动地用写入到传出数据寄存器54的数据来更新传出消息寄存器62。在一些实例中,GPU处理块42可实施内部先入先出(FIFO)缓冲器来存储将写入到传出数据寄存器54的多个传出消息。

[0148] 在图2的实例中,消息状态寄存器56经配置以存储指示传入消息是否被GPU40接受的数据。消息状态寄存器56可由主机装置使用以确定消息是否被成功传输,且在一些实例中,消息状态寄存器56可由主机装置使用以实施后退和/或溢出机制。在接受传入消息之后,消息传递模块50可将消息状态寄存器56设定为指示传入消息被接受的特定值。

[0149] 在图2的实例中,消息计数寄存器58经配置以存储指示传入消息寄存器60是否含有传入消息的数据。在一些实例中,当消息计数寄存器58已被主机装置递增时,消息计数寄存器58可将指示消息到达的信号发送到消息传递模块50。在一些情况下,所述信号可为1位脉冲线。在进一步的实例中,消息传递模块50可在从传入数据寄存器52读取消息之后递减消息计数寄存器58。

[0150] 在图2的实例中,传入消息寄存器60经配置以存储传入消息数据。举例来说,主机装置可将传入消息数据放置到传入消息寄存器60中以便将消息发送到任务48。传入消息寄存器60通信地耦合到传入数据寄存器52。

[0151] 在图2的实例中,传出消息寄存器62经配置以存储从传出数据寄存器54接收的传出消息数据。当将新数据写入到传出数据寄存器54时,传出消息寄存器62可自动地更新传出消息寄存器62中的数据以对应于传出数据寄存器54。在一些实例中,消息传递模块50可响应于传出消息被写入到传出消息寄存器62而产生中断信号。中断信号可被发送到主机装

置且指示消息传递模块50已发送消息。

[0152] 在图2的实例中,中断状态寄存器64经配置以存储指示传出消息是否已被写入到传出消息寄存器62的状态位。举例来说,中断状态寄存器64和传出消息寄存器62可经配置以使得当传出消息被写入到传出消息寄存器62时设定中断状态寄存器64中的状态位。状态位可允许在主机装置上执行的进程轮询GPU40以查看消息是否可用。

[0153] 在图2的实例中,中断确认寄存器66经配置以存储指示主机装置是否已读取存储于传出消息寄存器62中的传出消息的确认位。举例来说,传出消息寄存器62和中断确认寄存器66可经配置以使得当传出消息被写入到传出消息寄存器62时,设定中断确认寄存器66中的确认位。在此实例中,在主机装置读取传出消息寄存器62时,主机装置可清除确认位,进而指示主机装置已读取传出消息,且可将新的传出消息写入到传出消息寄存器62。可使用确认位来实施用于传出消息数据的流控制方案。

[0154] 在图2的实例中,总线控制器46经配置以允许外部装置经由互连网络18对主机可存取的GPU寄存器44的存取。举例来说,总线控制器46可对总线信号进行多路复用和多路分用,且执行由总线信号指定的各种接收和传输操作。总线控制器46可根据一个或一个以上公共的或专有的总线标准来操作。

[0155] 现在将根据本发明的某些方面来描述用于多处理器计算系统中的消息传递的各种技术。在一些实例中,图1的计算系统10可用于实施图3到19中所示的实例性技术。为了易于阐释,可相对于图1中所示的实例性计算系统10的组件来描述所述技术,但应理解,可以相同或不同的配置在具有相同或不同的组件的其它系统上执行所述技术。在额外的实例中,可相对于图2的GPU40的特定组件来描述图3到19中所示的技术中的一些。再者,应理解,图2是可以能够实施本发明的技术的GPU的一个实例,且可以相同或不同的配置在具有相同或不同的组件的其它GPU上执行所述技术。

[0156] 图3说明根据本发明的用于多处理器平台环境中的消息传递的实例性技术。在一些实例中,图1的计算系统10可用于实施图3中所示的实例性技术。命令队列接口24将存储器传送命令放置到命令队列32中(70)。命令队列接口24将任务执行命令放置到命令队列32中(72)。命令队列接口24执行任务执行命令以起始GPU14上的任务的执行(74)。在任务28正在GPU14上执行时,主机消息传递接口26在主机装置12与GPU14之间传递一个或一个以上消息(76)。举例来说,主机消息传递接口26可将源自由主机进程20发布的一个或一个以上发送指令的消息传递到GPU14。所述一个或一个以上发送指令可指定GPU14或在GPU14上执行的任务是消息的目的地。

[0157] 图4是根据本发明的用于执行由在主机装置上执行的进程发布的发送指令的实例性技术。在一些实例中,图1的计算系统10可用于实施图4中所示的实例性技术。主机消息传递接口26从主机进程20接收发送指令(78)。主机消息传递接口26基于与发送指令包含在一起的消息数据而产生传出消息(80)。在一些实例中,传出消息可等同于发送指令中所包含的消息数据。在额外的实例中,主机消息传递接口26可将一条或一条以上标头信息和/或路由信息附加到发送指令中所包含的消息数据以产生传出消息。在进一步的实例中,主机消息传递接口26可对发送指令中所包含的消息数据执行一个或一个以上译码或变换操作以产生传出消息。主机消息传递接口26可将传出消息发送到GPU14(82)。

[0158] 主机消息传递接口26可确定发送指令是封锁指令还是非封锁指令(84)。在一些实

例中,主机消息传递接口26可基于发送指令中指定的输入参数来确定发送指令是封锁指令还是非封锁指令。在其它实例中,可使用两种不同类型的发送指令,且主机消息传递接口26可基于指令的类型(例如,指令的操作码(操作码))来确定发送指令是封锁指令还是非封锁指令。如果主机消息传递接口26确定发送指令是非封锁指令,那么主机消息传递接口26可将句柄返回到调用进程(86)。所述句柄可允许调用进程询问句柄来确定是否已在稍后时间成功发送所述消息。如果后续询问指示发送失败,那么调用进程可能需要发布后续发送指令来重试发送操作。在一些实例中,调用进程可响应于失败的发送操作而实施后退例程或溢出机制。

[0159] 如果主机消息传递接口26确定发送指令是封锁指令,那么主机消息传递接口26可确定传出消息是否被GPU14成功接收(88)。如果主机消息传递接口26确定传出消息被成功接收,那么主机消息传递接口26可将指示发送指令中所包含的消息被成功发送的值返回到调用进程(90)。否则,如果主机消息传递接口26确定传出消息未被成功接收,那么主机消息传递接口26可前进到过程方框82且向GPU14重发传出消息。在一些实例中,当主机消息传递接口26确定消息被成功接收或已达到不成功递送尝试的阈值数目时,封锁指令可完成。

[0160] 图5是说明根据本发明的可用于实施图4中的过程方框82的实例性技术的流程图。在一些实例中,图1的计算系统10和/或图2的GPU40可用于实施图5中所示的实例性技术。主机消息传递接口26可将传出消息放置或存储在GPU40的传入消息寄存器60中(92)。主机消息传递接口26可递增GPU40的消息计数寄存器58以向GPU14中的消息传递模块50指示新的消息已到达(94)。在一些实例中,主机消息传递接口26可使用此项技术中已知的存储器映射寄存器硬件和/或I/O映射寄存器硬件来执行过程方框92和94中的一者或一者以上。

[0161] 图6是说明根据本发明的可用于实施图4中的决策方框88的实例性技术的流程图。在一些实例中,图1的计算系统10和/或图2的GPU40可用于实施图6中所示的实例性技术。主机消息传递接口26可检查GPU40的消息状态寄存器56中的状态位(96)。主机消息传递接口26可基于消息状态寄存器56中的状态位来确定所发送的消息是否被GPU14接受(98)。如果所述状态位指示所发送的消息被GPU14接受,那么主机消息传递接口26可确定传出消息被成功接收(100)。另一方面,如果所述状态位指示所发送的消息未被GPU14接受,那么主机消息传递接口26可确定传出消息未被成功接收(102)。

[0162] 图7是说明用于处理例如GPU等计算装置中的所接收的消息的实例性技术的流程图。在一些实例中,图2的GPU40可用于实施图7中所示的实例性技术。GPU40中的消息传递模块50接收消息到达信号(104)。举例来说,消息计数寄存器58可经配置以使得每当主机装置递增消息计数寄存器58时,便将消息到达脉冲发送到消息传递模块50。消息传递模块50可致使将存储于传入消息寄存器60中的数据传送到传入数据寄存器52(106)。举例来说,消息传递模块50可向传入数据寄存器52发布控制信号,从而致使传入数据寄存器52用存储于传入消息寄存器60中的数据来盖写存储于传入数据寄存器52中的当前数据。消息传递模块50可设定传入数据寄存器52中的状态位以指示数据在传入数据寄存器52中可用,例如,未被消耗(108)。消息传递模块50可设定消息状态寄存器56中的状态位以指示传入消息已被GPU40接受(110)。

[0163] 图8是说明根据本发明的用于执行由在计算装置上执行的任务发布的接收指令的实例性技术的流程图。在一些实例中,图1的计算系统10可用于实施图8中所示的实例性技

术。装置消息传递接口30从任务28接收接收指令(112)。装置消息传递接口30确定可从主机装置得到消息(114)。

[0164] 如果消息传递模块50确定消息不可用,那么消息传递模块50可确定所述接收指令是封锁接收指令还是非封锁接收指令(116)。在一些实例中,消息传递模块50可基于接收指令中指定的输入参数来确定接收指令是封锁指令还是非封锁指令。在其它实例中,可使用两种不同类型的接收指令,且消息传递模块50可基于指令的类型(例如,指令的操作码(操作码))来确定接收指令是封锁指令还是非封锁指令。如果消息传递模块50确定接收指令是封锁指令,那么消息传递模块50可返回到决策方框114以确定传入消息是否可用。否则,如果消息传递模块50确定接收指令是非封锁指令,那么消息传递模块50可将指示接收指令失败的值返回到调用进程(118)。

[0165] 如果消息传递模块50确定可从主机装置得到消息,那么消息传递模块50可将消息数据返回到调用进程(120)。消息传递模块50确定是否应将所述消息数据标记为被消耗(122)。消息传递模块50可基于一个或一个以上消耗模式来确定是否应将所述数据标记为被消耗。在一些实例中,可将所述消耗模式硬连线到GPU14中。在额外的实例中,所述消耗模式可由任务28和/或主机进程20编程。举例来说,任务28或主机进程20中的发送和/或接收指令可含有指定特定消耗模式的参数。举例来说,一个消耗模式可指定当任务的至少一个执行实例已读取数据时应将消息数据标记为被消耗。作为另一实例,一个消耗模式可指定当任务的至少阈值数目的执行实例已读取数据时应将消息数据标记为被消耗。

[0166] 如果消息传递模块50确定应将消息数据标记为被消耗,那么消息传递模块50可清除消息数据(124)。举例来说,消息传递模块50可清除传入数据寄存器52中的状态位。另一方面,如果消息传递模块50确定不应将消息数据标记为被消耗,那么消息传递模块50可保持消息数据(126)。举例来说,消息传递模块50可不清除传入数据寄存器52中的状态位。

[0167] 图9是说明根据本发明的可用于实施图8中的决策方框114的实例性技术的流程图。在一些实例中,图1的计算系统10和/或图2的GPU40可用于实施图9中所示的实例性技术。消息传递模块50可读取GPU40的传入数据寄存器52中的状态位(128)。消息传递模块50可确定所述状态位是否被设定(130)。如果传入数据寄存器52中的状态位被设定,那么消息传递模块50可确定所述传入消息可用(132)。另一方面,如果传入数据寄存器52中的状态位未被设定,那么消息传递模块50可确定所述传入消息不可用(134)。

[0168] 图10是说明根据本发明的可用于实施图8中的过程方框120的实例性技术的流程图。在一些实例中,图1的计算系统10和/或图2的GPU40可用于实施图10中所示的实例性技术。消息传递模块50可从GPU40中的传入数据寄存器52检索传入消息数据(136)。消息传递模块50可基于从传入数据寄存器52检索到的消息数据而产生用于任务48的返回消息数据(138)。在一些实例中,所返回的消息数据可等同于传入数据寄存器52中所包含的消息数据。在额外的实例中,消息传递模块50可从传入数据寄存器52中所包含的消息数据移除一条或一条以上标头信息和/或路由信息以产生返回消息数据。在进一步的实例中,消息传递模块50可对传入数据寄存器52中所包含的消息数据执行一个或一个以上解码或变换操作以产生返回消息数据。消息传递模块50将消息数据提供给任务48(140)。

[0169] 图11是根据本发明的用于执行由在计算装置(例如,GPU14)上执行的进程发布的发送指令的实例性技术。在一些实例中,图1的计算系统10可用于实施图11中所示的实例性

技术。消息传递模块50从任务28接收发送指令(142)。消息传递模块50基于与发送指令包含在一起的消息数据而产生传出消息(144)。在一些实例中,传出消息可等同于发送指令中所包含的消息数据。在额外的实例中,消息传递模块50可将一条或一条以上标头信息和/或路由信息附加到发送指令中所包含的消息数据以产生传出消息。在进一步的实例中,消息传递模块50可对发送指令中所包含的消息数据执行一个或一个以上译码或变换操作以产生传出消息。消息传递模块50可将传出消息发送到主机装置12(146)。

[0170] 消息传递模块50可确定发送指令是封锁指令还是非封锁指令(148)。在一些实例中,消息传递模块50可基于发送指令中指定的输入参数来确定发送指令是封锁指令还是非封锁指令。在其它实例中,可使用两种不同类型的发送指令,且消息传递模块50可基于指令的类型(例如,指令的操作代码(操作码))来确定发送指令是封锁指令还是非封锁指令。如果消息传递模块50确定发送指令是非封锁指令,那么消息传递模块50可将句柄返回到调用进程(例如,任务28)(150)。所述句柄可允许调用进程询问句柄来确定是否已在稍后时间成功发送所述消息。如果后续询问指示发送操作失败,那么调用进程可能需要发布后续发送指令来重试发送操作。

[0171] 如果消息传递模块50确定发送指令是封锁指令,那么消息传递模块50可确定传出消息是否被主机装置12成功接收(152)。举例来说,消息传递模块50可轮询主机装置12内所包含的指示消息是否被接受的状态寄存器。如果消息传递模块50确定传出消息被成功接收,那么消息传递模块50可将指示发送指令中所包含的消息被成功发送的值返回到调用进程(154)。否则,如果消息传递模块50确定传出消息未被成功接收,那么消息传递模块50可前进到过程方框146且向主机装置12重发传出消息。在一些实例中,当消息传递模块50确定消息被成功接收或已达到不成功递送尝试的阈值数目时,封锁指令可完成。

[0172] 图12是说明根据本发明的可用于实施图11中的过程方框146的实例性技术的流程图。在一些实例中,图1的计算系统10和/或图2的GPU40可用于实施图12中所示的实例性技术。消息传递模块50可将传出消息放置或存储在传出数据寄存器54中(156)。响应于新数据被放置到传出数据寄存器54中,传出消息寄存器62可更新传出消息寄存器62中的数据以对应于传出数据寄存器54(158)。消息传递模块50可产生中断信号且将中断信号发送到主机装置12,从而指示可从GPU40的任务28得到消息(160)。

[0173] 图13是根据本发明的可用于实施图11中的过程方框146的另一实例性技术。在一些实例中,图1的计算系统10和/或图2的GPU40可用于实施图13中所示的实例性技术。消息传递模块50可将传出消息放置或存储在传出数据寄存器54中(162)。响应于新数据被放置到传出数据寄存器54中,传出消息寄存器62可更新传出消息寄存器62中的数据以对应于传出数据寄存器54(164)。消息传递模块50可设定中断状态寄存器64中的状态位以指示可从GPU40的任务28得到消息。所述状态位可经设定以允许主机装置12轮询GPU40来确定消息是否可用(166)。

[0174] 图14是说明根据本发明的用于执行由在主机装置上执行的进程发布的寄存回调例程指令的实例性技术的流程图。在一些实例中,图1的计算系统10可用于实施图14中所示的实例性技术。主机消息传递接口26从主机进程20接收寄存回调例程指令(168)。主机消息传递接口26使寄存回调例程指令中所指定的回调例程与来自所述指令中所指定的装置(例如,GPU14)的中断信号相关联(170)。在一些实例中,所述中断信号可指示在指定装置上执

行的任务(例如,在GPU14上执行的任务28)已发送消息。在一些实例中,可经由耦合于主机装置12与GPU14之间的专用中断信号线来递送中断信号。在进一步的实例中,中断信号可指示除了任务28发送消息之外的其它事件。在此些实例中,主机消息传递接口26可在接收到中断信号之后执行额外的处理以确定多个事件中的哪一者是由所述信号表示。

[0175] 主机消息传递接口26确定回调例程是否与中断信号成功关联(172)。如果回调例程与中断信号成功关联,那么主机消息传递接口26可将指示寄存回调例程操作成功完成的值返回到调用进程(174)。否则,如果回调例程未与中断信号成功关联(例如,发生错误),那么主机消息传递接口26可将指示寄存回调例程操作失败的值得返回到调用进程(176)。

[0176] 图15是说明根据本发明的用于处理从计算装置接收到的中断的实例性技术的流程图。在一些实例中,图1的计算系统10可用于实施图15中所示的实例性技术。主机消息传递接口26从计算装置(例如,GPU14)接收中断信号(178)。主机消息传递接口26确定是否响应于消息接收事件而发送中断信号(180)。换句话说,主机消息传递接口26可确定中断信号是否指示在装置上执行的任务(例如,在GPU14上执行的任务28)已发送消息。

[0177] 在一些实例中,中断信号可为用信号通知消息接收事件和无其它事件的专用中断信号。在此些实例中,主机消息传递接口26可通过接收到中断信号本身且不一定需要执行其它操作而确定曾响应于消息接收事件而发送了中断信号。在其中中断信号用信号通知多个可能事件的实例中,主机消息传递接口26可能需要询问计算装置以确定用信号通知哪一事件。

[0178] 如果主机消息传递接口26确定未曾响应于消息接收事件而发送中断信号,那么主机消息传递接口26可检查其它类型的事件(182)。否则,如果主机消息传递接口26确定曾响应于消息接收事件而发送中断信号,那么主机消息传递接口26可执行与从其接收到消息的装置相关联的回调例程(184)。

[0179] 图16是说明根据本发明的可用于实施图15中的决策方框180的实例性技术的流程图。在一些实例中,图1的计算系统10和/或图2的GPU40可用于实施图16中所示的实例性技术。主机消息传递接口26可读取GPU40中的中断状态寄存器64(186)。主机消息传递接口26可确定中断状态寄存器64中的状态位是否指示新的消息可用于主机装置(188)。举例来说,消息传递模块50可在消息可用时设定中断状态寄存器64中的状态位,且主机消息传递接口26可轮询中断状态寄存器64来确定所述状态位是否被设定以便确定新的消息是否可用于主机装置。如果状态位指示新的消息可用于主机装置,那么主机消息传递接口26可确定曾响应于消息接收事件而发送了中断信号(190)。另一方面,如果状态位指示新的消息不可用于主机装置,那么主机消息传递接口26可确定未曾响应于消息接收事件而发送中断信号(192)。

[0180] 图17是说明根据本发明的可用于实施图15中的过程方框184的实例性技术的流程图。在一些实例中,图1的计算系统10和/或图2的GPU40可用于实施图17中所示的实例性技术。主机消息传递接口26可从GPU40中的传出消息寄存器62检索消息(194)。主机消息传递接口26可清除中断确认寄存器66中的确认位(196)。清除确认位可辅助对GPU40的流控制。举例来说,GPU40可在传出消息被写入到传出消息寄存器62时设定中断确认寄存器66中的确认位,且一直等到确认位已被清除,之后将额外数据写入到传出消息寄存器62。

[0181] 图18是说明根据本发明的用于执行由在主机装置上执行的进程发布的读取指令

的实例性技术的流程图。在一些实例中,图1的计算系统10可用于实施图18中所示的实例性技术。主机消息传递接口26接收指定从其读取数据的特定装置的读取指令(198)。主机消息传递接口26轮询读取指令中所指定的装置(200)。主机消息传递接口26基于从轮询操作接收的轮询数据而确定是否可从接收指令中所指定的装置得到消息(202)。如果主机消息传递接口26确定可从接收指令中所指定的装置得到消息,那么主机消息传递接口26可从读取指令中所指定的装置检索消息(204)。在一些实例中,主机消息传递接口26可从主机装置12可存取的装置中的寄存器(例如,GPU40中的传出消息寄存器62)检索消息。主机消息传递接口26可将消息数据返回到调用进程(例如,主机进程20)(206)。如果主机消息传递接口26确定不可从接收指令中所指定的装置得到消息,那么主机消息传递接口26可返回指示读取指令失败的值(208)。调用进程可能需要再次发布读取指令以重试读取操作。

[0182] 图19是说明根据本发明的可用于实施图18中的决策方框202的实例性技术的流程图。在一些实例中,图1的计算系统10和/或图2的GPU40可用于实施图19中所示的实例性技术。主机消息传递接口26可读取GPU40中的中断状态寄存器64(210)。主机消息传递接口26可确定中断状态寄存器64中的状态位是否指示新的消息可用于主机装置(212)。举例来说,消息传递模块50可在消息可用时设定中断状态寄存器64中的状态位,且主机消息传递接口26可轮询中断状态寄存器64来确定所述状态位是否被设定以便确定新的消息是否可用于主机装置。如果状态位被设定,那么主机消息传递接口26可确定消息可用(214)。另一方面,如果状态位未被设定,那么主机消息传递接口26可确定消息不可用(216)。

[0183] 虽然已在上文将由主机消息传递接口26和装置消息传递接口30实施的消息传递技术描述为提供主机装置12与GPU14之间的带外信令,但在其它实例性系统中,可使用其它技术来提供带外信令。举例来说,在一些实例中,可界定特殊的高优先级队列,其可用于发送带外消息。

[0184] 图20是说明根据本发明的可促进直接存储器对象的使用的实例性计算系统310的方框图。计算系统310经配置以在多个处理装置上处理一个或一个以上软件应用。在一些实例中,所述一个或一个以上软件应用可包含主机进程,且计算系统310可经配置以执行主机进程且分布由在计算系统310内的其它计算装置上的主机进程起始的一个或一个以上任务的执行。在进一步的实例中,可根据并行编程模型来编程由计算系统310执行的主机进程和/或任务。举例来说,所述应用可包含经设计以充分利用基础硬件系统的任务级并行度和/或数据级并行度的指令。

[0185] 计算系统310可包括个人计算机、桌上型计算机、膝上型计算机、计算机工作站、视频游戏平台或控制台、移动电话(例如,蜂窝式或卫星电话)、移动电话、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))、数字媒体播放器(例如,个人音乐播放器)、视频播放器、显示装置,或电视、电视机顶盒、服务器、中间网络装置、大型计算机或处理信息的任何其它类型的装置。

[0186] 计算系统310包含主机装置312、GPU314、存储器316和互连网络318。主机装置312经配置以提供用于执行用于多处理器计算平台API的主机进程和运行时模块的平台。通常,主机装置312是通用CPU,但主机装置12可为能够执行程序的任何类型的装置。主机装置12经由互连网络318通信地耦合到GPU314和存储器316。主机装置312包含主机进程320、运行时模块322、主机高速缓冲存储器324和主机高速缓冲存储器控制模块326。主机进程320和

运行时模块322可在一个或一个以上可编程处理器的任何组合上执行。

[0187] 主机进程320包含形成用于在计算系统310平台上执行的软件程序的一组指令。所述软件程序可经设计以执行用于终端用户的一个或一个以上特定任务。在一些实例中, 此些任务可涉及可利用由计算系统310提供的多个处理装置和并行架构的计算密集算法。

[0188] 运行时模块322可为实施经配置以服务于主机进程320中所包含的指令中的一者或一者以上的一个或一个以上接口的软件模块。由运行时模块322实施的接口包含存储器缓冲器接口328。在一些实例中, 运行时模块322除了存储器缓冲器接口328之外还可实施图1中所示的命令队列接口24以及图1中所示的主机消息传递接口26中的一者或一者以上。在进一步的实例中, 运行时模块322可实施除了本发明中所描述的接口之外的标准多处理器系统API内所包含的一个或一个以上接口。在一些实例中, 所述标准API可为异类计算平台API、跨平台API、跨供应商API、并行编程API、任务级并行编程API和/或数据级并行编程API。在进一步的实例中, 所述标准API可为OpenCLAPI。在此些实例中, 可将运行时模块322设计成遵照OpenCL规范中的一者或一者以上。在额外的实例中, 可将运行时模块322实施为驱动器程序(例如, GPU驱动器)的一部分或实施为驱动器程序。

[0189] 存储器缓冲器接口328经配置以从主机进程20接收一个或一个以上存储器对象创建指令, 且执行由所接收的指令指定的功能。在一些实例中, 可将存储器缓冲器接口328实施为对现有标准API(例如, OpenCLAPI)的扩展。在额外的实例中, 可将主命令队列接口24集成到现有标准API(例如, OpenCLAPI)中。

[0190] 主机高速缓冲存储器324经配置以存储数据以供在主机装置312内执行的进程使用。在一些实例中, 与存储于主机高速缓冲存储器324中的数据相关联的存储器空间可与存储器316中的存储器空间的一部分重叠。主机高速缓冲存储器324可为此项技术中已知的任何类型的高速缓冲存储器。举例来说, 主机高速缓冲存储器324可包含若干高速缓冲存储器级(例如, L1、L2等)和/或若干映射机制(例如, 直接映射、全关联、组关联等)的任何组合。主机高速缓冲存储器控制模块326经配置以控制主机高速缓冲存储器324的操作。

[0191] GPU314经配置以响应于从主机装置312接收到的指令来执行一个或一个以上任务。GPU314可为包含一个或一个以上可编程处理器或处理元件的任何类型的GPU。举例来说, GPU314可包含经配置以并行地执行任务的多个执行实例的一个或一个以上可编程着色器单元。可编程着色器单元可包含顶点着色器单元、片段着色器单元、几何着色器单元和/或统一着色器单元。GPU314经由互连网络318通信地耦合到主机装置312和存储器316。GPU314包含任务330、GPU高速缓冲存储器332和GPU高速缓冲存储器控制模块334。任务330可在一个或一个以上可编程处理元件的任何组合上执行。

[0192] 任务330包括形成用于在计算系统310中的计算装置上执行的任务的一组指令。在一些实例中, 用于任务330的所述组指令可在主机进程320中界定, 且在一些情况下, 由在主机进程320中所包含的指令编译。在进一步的实例中, 任务330可为具有在GPU314上并行地执行的多个执行实例的内核程序。在此些实例中, 主机进程320可界定用于内核的索引空间, 其将内核执行实例映射到用于执行内核执行实例的相应处理元件, 且GPU314可根据为内核界定的索引空间来执行用于任务330的多个内核执行实例。

[0193] GPU高速缓冲存储器332经配置以存储数据以供在GPU314内执行的任务使用。在一些实例中, 与存储于GPU高速缓冲存储器332中的数据相关联的存储器空间可与存储器316

中的存储器空间的一部分重叠。GPU高速缓冲存储器332可为此项技术中已知的任何类型的高速缓冲存储器。举例来说, GPU高速缓冲存储器332可包含若干高速缓冲存储器级(例如, L1、L2等)和/或若干映射机制(例如, 直接映射、全关联、组关联等)的任何组合。GPU高速缓冲存储器控制模块334经配置以控制GPU高速缓冲存储器332的操作。

[0194] 存储器316经配置以存储数据以供主机装置312和GPU314中的一者或两者使用。存储器316可包含一个或一个以上易失性或非易失性存储器或存储装置的任何组合, 所述易失性或非易失性存储器或存储装置例如为随机存取存储器(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、磁性数据存储媒体或光学存储媒体。存储器316经由互连网络318通信地耦合到主机装置312和GPU314。存储器316包含共享存储器空间336。共享存储器空间336可为可由主机装置312和GPU314存取的存储器空间。

[0195] 互连网络318经配置以促进主机装置312、GPU314与存储器316之间的通信。互连网络318可为此项技术中已知的任何类型的互连网络。在图20的实例性计算系统310中, 互连网络318是总线。所述总线可包含多种总线结构中的任一者中的一者或一者以上, 例如第三代总线(例如, 超传输总线或不限带宽总线)、第二代总线(例如, 高级图形端口总线、外围组件互连快速(PCIe)总线, 或高级可扩展接口(AXI)总线), 或任何其它类型的总线。互连网络318耦合到主机装置312、GPU314和存储器316。

[0196] 现在将进一步详细地描述计算系统310中的组件的结构和功能性。如上文所论述, 主机进程320包含一组指令。所述组指令可包含(例如)一个或一个以上存储器对象创建指令。在额外的实例中, 所述组指令可包含指定将在GPU314上执行的任务或内核的指令、创建命令队列且使命令队列与特定装置相关联的指令、编译并捆绑程序的指令、设置内核自变量的指令、界定索引空间的指令、界定装置背景的指令, 排队指令、消息传递指令以及支持由主机进程320提供的功能性的其它指令。

[0197] 根据本发明, 主机进程320可通过将一个或一个以上存储器对象创建指令发布到存储器缓冲器接口328而与存储器缓冲器接口328交互, 所述一个或一个以上存储器对象创建指令指令存储器缓冲器接口328基于所述指令中所包含的指定是否针对存储器对象启用直接模式的信息而创建所述存储器对象。如本文中所使用, 存储器对象可指代表示可由GPU314存取的存储器空间的区的软件对象。在一些实例中, 存储器空间的所述区还可由主机装置312存取。存储器对象可包含存储器空间中的与存储器对象相关联的数据。存储器对象可进一步包含与存储器空间相关联的一个或一个以上特性。在一些实例中, 存储器对象可包含全局存储器(例如, 存储器316)的参考计数区的句柄。

[0198] 所述存储器对象可包含缓冲器对象和图像对象。缓冲器对象可为存储一维字节集合的存储器对象。所述一维字节集合可为与存储器对象相关联的数据。缓冲器对象还可包含信息, 例如以字节计的与缓冲器对象相关联的存储器空间的大小、缓冲器对象的使用信息, 以及为缓冲器对象分配的存储器空间的区。图像对象存储数据的二维或三维阵列, 例如纹理、帧缓冲器或图像。图像对象还可包含信息, 例如图像的尺寸、图像中的每一元素的描述、图像对象的使用信息, 以及为图像对象分配的存储器空间的区。

[0199] 根据本发明的一些方面, 存储器对象创建指令可包含指定是否应针对将创建的存储器对象而启用直接模式的输入参数。如本文中进一步详细地论述, 当启用直接模式时, 可

将存储器对象实施为非可高速缓存共享存储器且/或实施为高速缓冲存储器相干共享存储器。当停用直接模式时,可没有必要将存储器对象实施为非可高速缓存共享存储器或实施为高速缓冲存储器相干共享存储器。

[0200] 在一些实例中,存储器对象可包含指示存储器对象是否为直接模式存储器对象的直接模式属性。在此实施例中,存储器缓冲器接口328可经配置以基于指定是否应针对存储器对象启用直接模式的信息而将创建的存储器对象的直接模式属性设定为指示是否针对存储器对象启用直接模式的值。存储器对象的直接模式属性可由计算系统310用来确定是否将存储器对象实施为非可高速缓存共享存储器且/或实施为高速缓冲存储器相干共享存储器。

[0201] 在一些实例中,存储器对象创建指令可包含缓冲器对象创建指令,所述缓冲器对象创建指令基于指令中所包含的指定是否针对缓冲器对象启用直接模式的信息而指令存储器缓冲器接口328创建缓冲器对象。在进一步的实例中,存储器对象创建指令可包含图像对象创建指令,所述图像对象创建指令基于指令中所包含的指定是否针对图像对象启用直接模式的信息而指令存储器缓冲器接口328创建图像对象。

[0202] 在一些实例中,用于缓冲器对象创建指令的接口可采取以下形式:

```
cl_mem clCreateBuffer (  
    cl_context context,  
    cl_mem_flags flags,  
[0203]    size_t size,  
    void *host_ptr,  
    cl_int *errcode_ret)
```

[0204] 其中clCreateBuffer是指令识别符,cl_context context是用于创建缓冲器对象的有效上下文(例如,OpenCL上下文),cl_mem_flags flags是用于指定缓冲器对象的分配和使用信息的位字段,size_t size是指定将分配的缓冲器存储器对象的以字节计的大小的参数,void*host_ptr是到可能已由应用分配的缓冲器数据的指针,且cl_int*errcode_ret返回一个或一个以上错误代码。所述指令可将所创建的缓冲器对象返回为cl_mem存储器对象。在此实例中,指定是否应针对图像对象启用直接模式的输入参数可为(例如)在cl_mem_flags flags字段中指定的CL_IMMEDIATE旗标。

[0205] 在进一步的实例中,用于图像对象创建指令的接口可采取以下形式:

```
cl_mem clCreateImage2D (  
    cl_context context,  
[0206]    cl_mem_flags flags,  
    const cl_image_format *image_format,
```

```
    size_t image_width,  
    size_t image_height,  
[0207]    size_t image_row_pitch,  
    void *host_ptr,  
    cl_int *errcode_ret)
```

[0208] 其中cl_CreateImage2D是指令识别符,cl_context context是用于创建缓冲器对象的有效上下文(例如,OpenCL上下文),cl_mem_flags flags是用于指定图像对象的分配和使用信息的位字段,const cl_image_format*image_format是到描述将分配的图像的格式性质的结构的指针,size_t image_width是以像素计的图像的宽度,size_t image_height是以像素计的图像的高度,size_t image_row_pitch是以字节计的扫描线间距,void*host_ptr是到可能已由应用分配的图像数据的指针,且cl_int*errcode_ret返回一个或一个以上错误代码。所述指令可将所创建的图像对象返回为cl_mem存储器对象。在此实例中,指定是否应针对图像对象启用直接模式的输入参数可为(例如)在cl_mem_flagsflags字段中指定的CL_IMMEDIATE旗标。

[0209] 在一些实例中,存储器对象创建接口可经配置以在读取/写入属性方面仅允许WRITE_ONLY属性或READ_ONLY属性。换句话说,在此些实例中,存储器缓冲器接口328可不允许READ_WRITE属性。非直接CL图像可已经具有由OpenCL规范提供的此特征。不允许READ_WRITE属性可减小在维持高速缓冲存储器相干性中的复杂性。

[0210] 根据本发明,存储器缓冲器接口328经配置以接收指定是否应针对可由主机装置312和GPU314存取的共享存储器空间336启用直接模式的指令,且基于指定是否应针对共享存储器空间336启用直接模式的所接收的指令而针对共享存储器空间336选择性地启用直接模式。举例来说,存储器缓冲器接口328可在所述指令指定应针对共享存储器空间336启用直接模式的情况下针对共享存储器空间336启用直接模式,且在所述指令指定应针对共享存储器空间336停用直接模式的情况下针对共享存储器空间336停用直接模式。所述指令可为(例如)存储器对象创建指令、缓冲器对象创建指令或图像对象创建指令中的一者。共享存储器空间336可对应于(例如)存储器对象、缓冲器对象或图像对象。

[0211] 在一些实例中,当存储器缓冲器接口328针对共享存储器空间336启用直接模式时,存储器缓冲器接口328可致使停用用于共享存储器空间336的高速缓存服务。类似地,当存储器缓冲器接口328针对共享存储器空间336停用直接模式时,存储器缓冲器接口328可致使针对共享存储器空间336启用用于共享存储器空间336的高速缓存服务。高速缓存服务可由主机高速缓冲存储器324和GPU高速缓冲存储器332中的一者或两者执行。如本文中所使用,高速缓存服务可指代通常由此项技术中已知的高速缓冲存储器执行的服务。

[0212] 在进一步的实例中,存储器缓冲器接口328可通过将与共享存储器空间336相关联的直接模式属性设定为指示是否针对共享存储器空间启用直接模式的值而针对共享存储器空间336启用和停用直接模式。举例来说,存储器缓冲器接口328可通过将与共享存储器空间336相关联的直接模式属性设定为指示针对共享存储器空间336启用直接模式的值(即,直接模式属性=true)而针对共享存储器空间336启用直接模式。类似地,存储器缓冲器接口328可通过将与共享存储器空间336相关联的直接模式属性设定为指示针对共享存

存储器空间336停用直接模式的值(即,直接模式属性=false)而针对共享存储器空间336停用直接模式。在一些情况下,直接模式属性可为可由在GPU314上执行的任务330存取的全局变量,例如,布尔变量。在一些实例中,直接模式属性可存储于共享存储器空间336内。在其它实例中,直接模式属性可存储于可由在GPU314上执行的任务存取的除了共享存储器空间336之外的位置中。在共享存储器空间336是存储器对象的部分的情况下,直接模式属性可存储于其中存储有存储器对象的其它属性的存储器空间的位置中。

[0213] 在其中存储器缓冲器接口328通过设定与共享存储器空间336相关联的直接模式属性而针对共享存储器空间336启用和停用直接模式的实例中,在一些情况下,用于任务330的源代码可经编译以使得在相对于共享存储器空间336执行存储器读取或写入操作之前,任务330存取与共享存储器空间336相关联的直接模式属性,且基于用于共享存储器空间336的直接模式属性而确定是否针对共享存储器空间336启用直接模式。如果针对共享存储器空间336启用直接模式,那么任务330可经编程以执行直接模式读取或写入指令以从共享存储器空间336读取数据或将数据写入到共享存储器空间336。另一方面,如果未针对共享存储器空间启用直接模式,那么任务330可经编程以执行高速缓存模式读取或写入指令(例如,高速缓存读取或写入指令)以从共享存储器空间336读取数据或将数据写入到共享存储器空间336。

[0214] 直接模式读取和写入指令可(例如)在不使用高速缓存服务的情况下分别执行读取和写入操作。举例来说,直接模式读取指令可致使高速缓冲存储器无效,之后执行读取操作,且/或可在执行读取操作时绕过高速缓冲存储器。直接模式写入指令(例如)可致使高速缓冲存储器在执行写入操作时执行直接回写,且/或可在执行写入操作时绕过高速缓冲存储器。高速缓存读取和写入指令可(例如)使用GPU高速缓冲存储器332中的一者或两者的高速缓存服务来分别执行读取和写入操作。

[0215] 在额外情况下,当编译用于任务330的源代码时,用于任务330的编译器可具有对指示是否针对共享存储器空间336启用直接模式的信息的存取权。举例来说,用于任务330的源代码(例如,内核源代码)可包含指示是否针对由任务330使用且与共享存储器空间336相关联的存储器对象而启用直接模式的旗标。在一些实例中,所述旗标可采取OpenCL属性限定符的形式,例如a_cl_immediate属性限定符。如果针对与共享存储器空间336相关联的存储器对象启用直接模式,那么编译器可编译任务330以使得用于任务330的经编译代码包含用于相对于共享存储器空间336发生的读取或写入操作的直接模式读取和/或写入指令。否则,如果未针对与共享存储器空间336相关联的存储器对象启用直接模式,那么编译器可编译任务330以使得用于任务330的经编译代码不包含用于相对于共享存储器空间336发生的读取或写入操作的直接模式读取和/或写入指令。举例来说,编译器可编译任务330以使得用于任务330的经编译代码包含用于相对于共享存储器空间336发生的读取或写入操作的高速缓存读取和/或写入指令。

[0216] 在进一步的实例中,存储器缓冲器接口328可通过启用和停用由主机装置312中的主机高速缓冲存储器324以及GPU314中的GPU高速缓冲存储器332中的至少一者对用于共享存储器空间336的高速缓存服务的执行而针对共享存储器空间336启用和停用直接模式。举例来说,存储器缓冲器接口328可通过停用由主机装置312中的主机高速缓冲存储器324以及GPU314中的GPU高速缓冲存储器332中的至少一者对用于共享存储器空间336的高速缓存

服务的执行而针对共享存储器空间336启用直接模式。类似地,存储器缓冲器接口328可通过启用由主机装置312中的主机高速缓冲存储器324以及GPU314中的GPU高速缓冲存储器332中的至少一者对用于共享存储器空间336的高速缓存服务的执行而针对共享存储器空间336停用直接模式。

[0217] 在此些实例中,存储器缓冲器接口328可通过配置与执行用于共享存储器空间336的高速缓存服务的高速缓冲存储器相关联的基于硬件的高速缓冲存储器控制模块和/或基于硬件的存储器管理单元而启用和停用对用于共享存储器空间336的高速缓存服务的执行。举例来说,为了由主机高速缓冲存储器324启用对用于共享存储器空间336的高速缓存服务的执行,存储器缓冲器接口328可配置主机高速缓冲存储器控制模块326以使得由主机高速缓冲存储器328针对共享存储器空间336提供高速缓存服务。为了由主机高速缓冲存储器324停用对用于共享存储器空间336的高速缓存服务的执行,存储器缓冲器接口328可(例如)配置主机高速缓冲存储器控制模块326以使得主机高速缓冲存储器328不针对共享存储器空间336提供高速缓存服务。类似地,为了由GPU高速缓冲存储器332启用对用于共享存储器空间336的高速缓存服务的执行,存储器缓冲器接口328可(例如)配置GPU高速缓冲存储器控制模块334以使得由主机高速缓冲存储器324针对共享存储器空间336提供高速缓存服务。为了由GPU高速缓冲存储器332停用对用于共享存储器空间336的高速缓存服务的执行,存储器缓冲器接口328可(例如)配置GPU高速缓冲存储器控制模块334以使得GPU高速缓冲存储器332不针对共享存储器空间336提供高速缓存服务。

[0218] 在一些实例中,存储器缓冲器接口328可通过将与共享存储器空间336相关联的一个或一个以上基于硬件的直接旗标设定为指示是否应针对共享存储器空间336提供高速缓存服务的值来配置主机高速缓冲存储器控制模块326和GPU高速缓冲存储器控制模块334中的一者或两者。在一些实例中,所述一个或一个以上基于硬件的直接旗标可为一个或一个以上寄存器。在进一步的实例中,所述基于硬件的直接旗标可为直接旗标的表格的部分,其中直接旗标的表格中的每一直接旗标对应于存储器316内的特定地址空间。在任何情况下,当与共享存储器空间336相关联的一个或一个以上直接旗标被设定为指示应提供高速缓存服务的值时,主机高速缓冲存储器控制模块326和/或GPU高速缓冲存储器控制模块334可使用主机高速缓冲存储器324和/或GPU高速缓冲存储器332来提供用于共享存储器空间336的高速缓存服务。类似地,当与共享存储器空间336相关联的一个或一个以上直接旗标被设定为指示不应提供高速缓存服务的值时,主机高速缓冲存储器控制模块326和/或GPU高速缓冲存储器控制模块334可使不提供用于共享存储器空间336的高速缓存服务。

[0219] 在此些实例中,GPU高速缓冲存储器控制模块334可经配置以处理用于存储器316的地址空间内的存储器地址的读取指令和/或写入指令。所述读取和写入指令可(例如)由在GPU314上执行的任务330发布到GPU高速缓冲存储器控制模块334。响应于接收到用以从存储器316的给定地址空间内的存储器位置读取数据或将数据写入到所述存储器位置的读取或写入指令,GPU高速缓冲存储器控制模块334可识别与地址空间相关联的基于硬件的旗标,且确定在基于基于硬件的旗标的值来处理读取或写入指令时是否使用GPU高速缓冲存储器332的高速缓存服务。如果GPU高速缓冲存储器控制模块334确定使用GPU高速缓冲存储器332的高速缓存服务,那么GPU高速缓冲存储器控制模块334可(例如)尝试从GPU高速缓冲存储器332读取数据(如果所述数据有效)且/或将数据写入到GPU高速缓冲存储器332。如果

GPU高速缓冲存储器控制模块334确定不使用GPU高速缓冲存储器332的高速缓存服务,那么在一些实例中, GPU高速缓冲存储器控制模块334可绕过GPU高速缓冲存储器332且直接从存储器316读取数据或将数据写入到存储器316。在额外实例中, 如果GPU高速缓冲存储器控制模块334确定不使用GPU高速缓冲存储器332的高速缓存服务, 那么GPU高速缓冲存储器控制模块334可使与地址空间相关联的高速缓冲存储器332的一部分无效, 之后执行读取指令且/或在执行写入指令时执行高速缓冲存储器回写或高速缓冲存储器通写技术。主机高速缓冲存储器控制模块334可响应于从在主机装置312上执行的进程320接收到的读取和写入指令而以与主机高速缓冲存储器324类似的方式进行操作。

[0220] 在额外实例中, 存储器缓冲器接口328可通过针对主机装置312中的主机高速缓冲存储器324以及GPU314中的GPU高速缓冲存储器332中的至少一者启用和停用共享存储器高速缓冲存储器相干模式而针对共享存储器空间336启用和停用直接模式。举例来说, 为了针对共享存储器空间336启用直接模式, 存储器缓冲器接口328可针对主机装置312中的主机高速缓冲存储器324以及GPU314中的GPU高速缓冲存储器332中的至少一者启用共享存储器高速缓冲存储器相干模式。类似地, 为了针对共享存储器空间336停用直接模式, 存储器缓冲器接口328可针对主机装置312中的主机高速缓冲存储器324以及GPU314中的GPU高速缓冲存储器332中的至少一者停用共享存储器高速缓冲存储器相干模式。在此些实例中, 在一些情况下存储器缓冲器接口328可通过配置主机高速缓冲存储器控制模块326和GPU高速缓冲存储器控制模块334中的一者或两者来启用共享存储器高速缓冲存储器相干模式而针对主机高速缓冲存储器324启用共享存储器高速缓冲存储器相干模式, 且通过配置主机高速缓冲存储器控制模块326和GPU高速缓冲存储器控制模块334中的一者或两者来停用共享存储器高速缓冲存储器相干模式而针对主机高速缓冲存储器324停用共享存储器高速缓冲存储器相干模式。

[0221] 当针对主机高速缓冲存储器324启用共享存储器高速缓冲存储器相干模式时, 主机高速缓冲存储器控制模块326可根据已知方法相对于共享存储器空间336执行共享存储器高速缓冲存储器相干技术。当针对主机高速缓冲存储器324停用共享存储器高速缓冲存储器相干模式时, 主机高速缓冲存储器324可不相对于共享存储器空间336执行共享存储器高速缓冲存储器相干技术。类似地, 当针对GPU高速缓冲存储器332启用共享存储器高速缓冲存储器相干模式时, GPU高速缓冲存储器控制模块334可根据已知方法相对于共享存储器空间336执行共享存储器高速缓冲存储器相干技术。当针对GPU高速缓冲存储器332停用共享存储器高速缓冲存储器相干模式时, GPU高速缓冲存储器控制模块334可不相对于共享存储器空间336执行共享存储器高速缓冲存储器相干技术。

[0222] 为了易于说明, 图20中所说明的实例性计算系统310描述将GPU314用作计算装置的本发明的直接缓冲技术。应认识到, 可将本发明的技术应用于具有不同于除了GPU314之外或作为GPU314的替代的GPU的计算装置的多处理器计算系统。在一些实例中, 计算系统310中的计算装置可为OpenCL计算装置。另外, 图20中所示的实例性计算系统310说明用于促进主机装置与计算装置之间的运行中的数据共享的基础结构和技术。然而, 在其它实例性计算系统中, 可容易将所述技术扩展以提供在具有一个以上计算装置的计算系统中的不同计算装置(例如, OpenCL计算装置)之间的运行中的数据共享。在此些实例中, 可在不同的计算装置之间布线一个或一个以上中断线。

[0223] 图21是说明根据本发明的用于执行由在主机装置上执行的进程发布的存储器对象创建指令的实例性技术的流程图。在一些实例中,图20的计算系统310可用于实施图21中所示的实例性技术。存储器对象创建指令可为缓冲器对象创建指令或图像对象创建指令。存储器缓冲器接口328接收存储器对象创建指令(340)。存储器缓冲器接口328确定存储器对象创建指令是否指定应针对存储器对象启用直接模式(342)。举例来说,存储器缓冲器接口328可确定直接旗标参数是否包含于用于存储器对象创建指令的参数列表中。

[0224] 如果存储器缓冲器接口328确定存储器对象创建指令未指定应针对存储器对象启用直接模式,那么存储器缓冲器接口328可针对将创建的存储器对象分配共享存储器空间336(344),致使针对共享存储器空间336启用由主机高速缓冲存储器324和GPU高速缓冲存储器332中的一者或两者对高速缓存服务的执行(346),且返回对所创建的存储器对象的参考(348)。存储器对象创建指令可(例如)通过不包含直接旗标参数或通过用不应启用直接模式的另一参数值进行指定而指定不应启用直接模式。相反,如果存储器缓冲器接口328确定存储器对象创建指令指定应针对存储器对象启用直接模式,那么存储器缓冲器接口328可针对将创建的存储器对象分配共享存储器空间336(350),致使针对共享存储器空间336停用由主机高速缓冲存储器324和GPU高速缓冲存储器332中的一者或两者对高速缓存服务的执行(352),且返回对所创建的存储器对象的参考(354)。存储器对象创建指令可(例如)通过包含直接旗标参数或通过用应启用直接模式的另一参数值进行指定而指定应启用直接模式。

[0225] 在一些实例中,存储器缓冲器接口328可通过将与共享存储器空间336相关联的存储器对象的直接模式属性设定为指示应针对与共享存储器空间336相关联的存储器对象提供高速缓存服务的值而针对共享存储器空间336启用高速缓存服务的执行。同样,存储器缓冲器接口328可通过将与共享存储器空间336相关联的存储器对象的直接模式属性设定为指示不应针对与共享存储器空间336相关联的存储器对象提供高速缓存服务的值而针对共享存储器空间336停用高速缓存服务的执行。所返回的存储器对象可包含直接模式属性。在此些实例中,用于存储器对象的直接模式属性可为可由在主机装置312上执行的主机进程320和在GPU314上执行的任务330中的一者或两者存取的。主机进程320和/或任务330可基于与共享存储器空间336相关联的存储器对象的直接模式属性而确定在相对于共享存储器空间336执行特定读取和写入指令时是否使用高速缓存服务。

[0226] 在进一步的实例中,存储器缓冲器接口328可通过将与共享存储器空间336相关联的基于硬件的直接旗标配置为指示应针对共享存储器空间336提供高速缓存服务的值而针对共享存储器空间336启用高速缓存服务的执行。同样,存储器缓冲器接口328可通过将与共享存储器空间336相关联的基于硬件的直接旗标配置为指示不应针对共享存储器空间336提供高速缓存服务的值而针对共享存储器空间336停用高速缓存服务的执行。所述一个或一个以上基于硬件的直接旗标可位于主机高速缓冲存储器控制模块326和GPU高速缓冲存储器控制模块334或另一本地或全局存储器管理单元(未图示)中的一者或一者以上中。

[0227] 在额外实例中,存储器缓冲器接口328可使存储器对象返回到调用进程(例如,主机进程320),之后分配存储器316中的物理存储器空间以存储数据。在此些实例中,存储器缓冲器接口328可包含所返回的存储器对象中的直接模式属性。随后,当在稍后时间为存储器对象分配存储器316时,存储器缓冲器接口328可基于存储器对象的直接模式属性来配置

一个或一个以上基于硬件的直接旗标。

[0228] 图22是说明根据本发明的用于执行由在主机装置上执行的进程发布的存储器对象创建指令的另一实例性技术的流程图。在一些实例中,图20的计算系统310可用于实施图22中所示的实例性技术。存储器对象创建指令可为缓冲器对象创建指令或图像对象创建指令。存储器缓冲器接口328接收存储器对象创建指令(356)。存储器缓冲器接口328确定存储器对象创建指令是否指定应针对存储器对象启用直接模式(358)。举例来说,存储器缓冲器接口328可确定直接旗标参数是否包含于用于存储器对象创建指令的参数列表中。

[0229] 如果存储器缓冲器接口328确定存储器对象创建指令未指定应针对存储器对象启用直接模式,那么存储器缓冲器接口328可针对将创建的存储器对象分配共享存储器空间336(360),针对共享存储器空间336停用共享存储器高速缓冲存储器相干模式(362),且返回对所创建的存储器对象的参考(364)。相反,如果存储器缓冲器接口328确定存储器对象创建指令指定应针对存储器对象启用直接模式,那么存储器缓冲器接口328可针对将创建的存储器对象分配共享存储器空间336(366),针对共享存储器空间336启用共享存储器高速缓冲存储器相干模式(368),且返回对所创建的存储器对象的参考(370)。

[0230] 在一些实例中,存储器缓冲器接口328可使存储器对象返回到调用进程(例如,主机进程320),之后分配存储器316中的物理存储器空间以存储数据。在此些实例中,存储器缓冲器接口328可包含所返回的存储器对象中的直接模式属性。随后,当在稍后时间为存储器对象分配存储器316时,存储器缓冲器接口328或另一模块可基于存储器对象的直接模式属性而启用或停用共享存储器空间高速缓冲存储器相干模式。

[0231] 图23到28说明根据本发明的GPU可用来处理直接模式和高速缓存模式加载和存储指令的实例性技术。如上文所论述,在一些实例中,用于任务330的源代码可经编译以使得经编译的代码可包含高速缓存模式指令和直接模式指令两者,以便支持直接存储器对象和高速缓冲存储器对象两者。高速缓存模式指令可使用与基础存储器相关联的高速缓冲存储器的高速缓存服务来相对于存储器执行读取和写入操作,且直接模式指令可不使用与基础存储器相关联的高速缓冲存储器的高速缓存服务来相对于存储器执行读取和写入操作。高速缓存模式指令可或者在本文中被称作非直接模式指令。加载和存储指令可或者在本文中分别被称作读取指令和写入指令。

[0232] 在一些实例中,加载或存储指令的高速缓存模式版本以及加载或存储指令的直接模式版本可为不同的指令,例如各自具有不同的操作代码,即操作码。在进一步的实例中,加载或存储指令的高速缓存模式版本以及加载或存储指令的直接模式版本可为相同的指令,例如两者都具有相同的操作码。在此实例中,向指令提供的参数可指定所述指令是高速缓存模式还是直接模式。

[0233] 图23是说明根据本发明的用于处理高速缓存模式指令和直接模式指令的实例性技术的流程图。在一些实例中,图20的计算系统310可用于实施图23中所示的实例性技术。在图23的实例中,直接模式被称作绕过高速缓冲存储器模式,且直接模式指令对应于绕过高速缓冲存储器模式指令。GPU高速缓冲存储器控制模块334接收指定存储器位置以及是否启用绕过高速缓冲存储器模式的加载指令(372)。GPU高速缓冲存储器控制模块334确定加载指令是否指定启用绕过高速缓冲存储器模式(374)。在一些情况下,GPU高速缓冲存储器控制模块334可基于指令的类型(例如,指令的操作码)来确定加载指令是否指定启用绕过

高速缓冲存储器模式。在额外情况下, GPU高速缓冲存储器控制模块334可基于与加载指令一起包含的指示是否启用绕过高速缓冲存储器模式的参数来确定加载指令是否指定启用绕过高速缓冲存储器模式。如果GPU高速缓冲存储器控制模块334确定不启用绕过高速缓冲存储器模式, 那么GPU高速缓冲存储器控制模块334从高速缓冲存储器(例如, GPU高速缓冲存储器332)在与加载指令中所指定的存储器位置相关联的高速缓冲存储器位置处检索数据(376)。另一方面, 如果GPU高速缓冲存储器控制模块334确定启用绕过高速缓冲存储器模式, 那么GPU高速缓冲存储器控制模块334从存储器(例如, 共享存储器空间336)在加载指令中所指定的存储器位置处检索数据(378)。

[0234] 图24是说明根据本发明的用于处理高速缓存模式指令和直接模式指令的另一实例性技术的流程图。在一些实例中, 图20的计算系统310可用于实施图24中所示的实例性技术。在图24的实例中, 直接模式被称作绕过高速缓冲存储器模式, 且直接模式指令对应于绕过高速缓冲存储器模式指令。GPU高速缓冲存储器控制模块334接收指定存储器位置、要存储的数据以及是否启用绕过高速缓冲存储器模式的存储指令(380)。GPU高速缓冲存储器控制模块334确定存储指令是否指定启用绕过高速缓冲存储器模式(382)。在一些情况下, GPU高速缓冲存储器控制模块334可基于指令的类型(例如, 指令的操作码)来确定存储指令是否指定启用绕过高速缓冲存储器模式。在额外情况下, GPU高速缓冲存储器控制模块334可基于与加载指令一起包含的指示是否启用绕过高速缓冲存储器模式的参数来确定存储指令是否指定启用绕过高速缓冲存储器模式。如果GPU高速缓冲存储器控制模块334确定不启用绕过高速缓冲存储器模式, 那么GPU高速缓冲存储器控制模块334将存储指令中所指定的数据存储在高速缓冲存储器(例如, GPU高速缓冲存储器332)中在与存储指令中所指定的存储器位置相关联的高速缓冲存储器位置处(384)。另一方面, 如果GPU高速缓冲存储器控制模块334确定启用绕过高速缓冲存储器模式, 那么GPU高速缓冲存储器控制模块334将存储指令中所指定的数据存储在存储器(例如, 共享存储器空间336)在加载指令中所指定的存储器位置处(386)。

[0235] 图25是说明根据本发明的用于处理高速缓存模式指令和直接模式指令的另一实例性技术的流程图。在一些实例中, 图20的计算系统310可用于实施图25中所示的实例性技术。GPU高速缓冲存储器控制模块334接收指定存储器位置、要存储的数据以及是否启用直接模式的存储指令(388)。GPU高速缓冲存储器控制模块334将存储指令中所指定的数据存储在高速缓冲存储器(例如, GPU高速缓冲存储器332)中在与存储指令中所指定的存储器位置相关联的高速缓冲存储器位置处(390)。GPU高速缓冲存储器控制模块334基于存储指令中的指定是否启用直接模式的信息而确定是否启用直接模式(392)。在一些实例中, 是否启用直接模式的信息可为指令的类型(例如, 用于指令的操作码)和/或与指令一起包含的指定是否针对所述指令启用直接模式的参数。如果未启用直接模式, 那么GPU高速缓冲存储器控制模块334不执行直接高速缓冲存储器回写操作(394)。另一方面, 如果启用直接模式, 那么GPU高速缓冲存储器控制模块334执行直接高速缓冲存储器回写操作(396)。

[0236] 图26是说明根据本发明的用于处理高速缓存模式指令和直接模式指令的另一实例性技术的流程图。在一些实例中, 图20的计算系统310可用于实施图26中所示的实例性技术。GPU高速缓冲存储器控制模块334接收指定存储器位置以及是否启用直接模式的加载指令(398)。GPU高速缓冲存储器控制模块334基于加载指令中的指定是否启用直接模式的信

息而确定是否启用直接模式(400)。在一些实例中,是否启用直接模式的信息可为指令的类型(例如,用于指令的操作码)和/或与指令一起包含的指定是否针对所述指令启用直接模式的参数。如果未启用直接模式,那么GPU高速缓冲存储器控制模块334不清洗高速缓冲存储器且使高速缓冲存储器无效(402)。GPU高速缓冲存储器控制模块334在可在高速缓冲存储器中得到数据的情况下从高速缓冲存储器(例如,GPU高速缓冲存储器332)检索加载指令中所指定的数据,或在高速缓冲存储器中得不到数据的情况下从基础存储器检索加载指令中所指定的数据(404)。如果启用直接模式,那么GPU高速缓冲存储器控制模块334清洗高速缓冲存储器且使高速缓冲存储器无效(406)。GPU高速缓冲存储器控制模块334从基础存储器检索加载指令中所指定的数据(408)。高速缓冲存储器不返回数据,因为高速缓冲存储器已被清洗且无效。

[0237] 图27是说明根据本发明的可用于图20的计算系统310中的实例性GPU420的方框图。在一些实例中,GPU420可用于实施图20中所说明的GPU314。GPU420包含GPU处理模块422、GPU高速缓冲存储器控制模块424、GPU高速缓冲存储器426、高速缓冲存储器总线428和绕过总线430。GPU处理模块422经由高速缓冲存储器总线428通信地耦合到GPU高速缓冲存储器控制模块424。GPU处理模块422还经由绕过总线430通信地耦合到存储器316。GPU高速缓冲存储器控制模块424和GPU高速缓冲存储器426实质上类似于图20中的GPU高速缓冲存储器控制模块334和GPU高速缓冲存储器332,且将不进一步详细地描述。GPU处理模块422包含处理元件432和总线控制器434。处理元件432经配置以将加载和存储指令发布到总线控制器434。

[0238] 总线控制器434可经配置以经由高速缓冲存储器总线428和绕过总线430将加载和存储指令转发到适当位置。总线控制器434可经配置以基于加载或存储指令中的指示指令是直接模式指令还是高速缓存模式指令的信息而在直接模式或非直接模式中操作。当总线控制器434经配置以在非直接模式(即,高速缓存模式)中操作时,总线控制器434可使用高速缓冲存储器总线428将加载和存储指令转发到GPU高速缓冲存储器控制模块424以供执行。另一方面,当总线控制器434经配置以在直接模式中操作时,总线控制器434可使用绕过总线430将加载和存储指令转发到存储器316以供执行。

[0239] 图28是说明根据本发明的用于处理高速缓存模式指令和直接模式指令的实例性技术的流程图。在一些实例中,图27的GPU420可用于实施图28中所示的实例性技术。总线控制器434接收加载或存储指令(440)。总线控制器434基于加载或存储指令中的指定是否启用直接模式的信息而确定是否启用直接模式(442)。在一些实例中,是否启用直接模式的信息可为指令的类型(例如,用于指令的操作码)和/或与指令一起包含的指定是否针对所述指令启用直接模式的参数。如果总线控制器434确定未启用直接模式,那么总线控制器434使用高速缓冲存储器总线428将所接收的指令转发到GPU高速缓冲存储器控制模块424(444)。否则,如果总线控制器434确定启用直接模式,那么总线控制器434使用绕过总线430将所接收的指令转发到存储器316(446)。

[0240] 图29是说明根据本发明的用于执行由在主机装置上执行的进程发布的存储器对象创建指令的另一实例性技术的流程图。在一些实例中,图20的计算系统310可用于实施图29中所示的实例性技术。存储器对象创建指令可为缓冲器对象创建指令或图像对象创建指令。存储器缓冲器接口328接收存储器对象创建指令(448)。存储器缓冲器接口328确定存储

器对象创建指令是否指定应针对存储器对象启用直接模式(450)。举例来说,存储器缓冲器接口328可确定直接旗标参数是否包含于用于存储器对象创建指令的参数列表中。

[0241] 如果存储器缓冲器接口328确定存储器对象创建指令未指定应针对存储器对象启用直接模式,那么存储器缓冲器接口328将用于所创建的存储器对象的直接模式属性设定为指示不启用直接模式的值(例如,“false”)(452)。另一方面,如果存储器缓冲器接口328确定存储器对象创建指令指定应针对存储器对象启用直接模式,那么存储器缓冲器接口328将用于所创建的存储器对象的直接模式属性设定为指示启用直接模式的值(例如,“true”)(454)。在一些实例中,存储器对象的直接模式属性可由主机装置312和/或GPU314使用以在存取存储于特定存储器对象中的数据时确定是执行高速缓存模式还是直接模式读取和写入操作。

[0242] 在一些实例中,主机进程320和/或任务330可能希望将一些存储器对象编程为直接存储器对象,且将其它对象编程为高速缓冲存储器对象,即,非直接存储器对象。在一些实例中,本发明的技术可包含专用编译技术,所述专用编译技术允许经编译任务330相对于高速缓冲存储器对象以及直接存储器对象两者执行读取和写入操作。第一实例性编译技术可将给定读取操作或写入操作编译为指令序列。所述指令序列可检查从其进行读取或对其进行写入的存储器对象的直接模式属性的值,且基于所述直接模式属性的值来确定是执行高速缓存模式指令还是直接模式指令。第二实例性编译技术可使用源代码中的指示存储器对象是否为直接模式对象的信息,以选择高速缓存模式指令或直接模式指令来用于经编译代码中以用于存取存储器对象。

[0243] 根据第一实例性编译技术,编译器可编译用于任务330的源代码,以使得用于任务330的经编译代码包含根据以下实例性伪代码的读取序列:

```
        if( isImmediate ){  
            immediate_read(...) }  
[0244]  else{  
            cached_read(...)  
        }
```

[0245] 其中“isImmediate”表示将从其读取数据的存储器对象的布尔直接模式属性,“immediate_read(...)”表示直接模式读取指令,且“cached_read(...)”表示高速缓存模式读取指令。

[0246] GPU高速缓冲存储器控制模块334可通过(例如)在从GPU高速缓冲存储器332读取数据之前使GPU高速缓冲存储器332(在使用的情况下)无效而处理immediate_read(...)指令。GPU高速缓冲存储器控制模块334可通过(例如)以正常方式从GPU高速缓冲存储器332读取数据(例如,而不在执行读取之前使高速缓冲存储器无效)来处理cached_read(...)指令。

[0247] 根据第一实例性编译技术,编译器可编译用于任务330的源代码,以使得用于任务330的经编译代码包含根据以下实例性伪代码的写入序列:

```
        if( isImmediate ){  
            immediate_write(...)  
        }  
[0248]    else{  
        cached_write(...)  
    }  
}
```

[0249] 其中“isImmediate”表示将向其写入数据的存储器对象的布尔直接模式属性,“immediate_write(...)”表示直接模式写入指令,且“cached_write(...)”表示高速缓存模式写入指令。

[0250] 在一些实例中, GPU高速缓冲存储器控制模块334可通过在使用高速缓冲存储器的情况下针对GPU高速缓冲存储器332使用通写模式来处理immediate_write(...)指令。在进一步的实例中, GPU高速缓冲存储器控制模块334可通过在使用高速缓冲存储器的情况下将数据写入到GPU高速缓冲存储器332来处理immediate_write(...)指令, 且响应于将数据写入到GPU高速缓冲存储器332而针对GPU高速缓冲存储器332执行高速缓冲存储器清洗。GPU高速缓冲存储器控制模块334可通过以正常方式将数据写入到GPU高速缓冲存储器332(例如, 而不响应于写入操作而清洗高速缓冲存储器)来处理cached_write(...)指令。

[0251] 图30是说明根据GPU可如何处理根据上述第一编译技术而编译的指令序列的流程图。在一些实例中, 图30中所说明的技术可用于实施上文针对读取和写入序列而提供的实例性伪代码。任务330开始读取序列或写入序列(456)。举例来说, 任务330可在任务330到达任务330的执行中的一点时(其中应发生对特定存储器对象的读取或写入指令)开始读取序列或写入序列。任务330存取与将从其读取数据或将向其写入数据的存储器对象相关联的直接模式属性(458)。任务330确定用于存储器对象的属性是否被设定为指示启用直接模式的值(例如, “true”)(460)。如果任务330确定用于存储器对象的属性被设定为指示不启用直接模式的值, 那么任务330使用高速缓存读取或写入指令针对存储器对象执行高速缓存读取或写入操作(462)。否则, 如果任务330确定用于存储器对象的属性被设定为指示启用直接模式的值, 那么任务330使用直接读取或写入指令针对存储器对象执行直接读取或写入操作(464)。

[0252] 根据第二实例性编译技术, 当编译源代码时, 编译器可具有对指示是否针对任务330从其进行读取或对其进行写入的特定存储器对象而启用直接模式的信息的存取权。当任务330从所述特定存储器对象进行读取或向其进行写入时, 编译器使用此信息来编译用于任务330的源代码, 以在高速缓存模式读取和写入指令或直接模式读取和写入指令之间进行选择。

[0253] 在一些实例中, 指示是否针对特定存储器对象启用直接模式的信息可为指示是否针对由任务330的源代码存取的一个或一个以上存储器对象启用直接模式的编译时间属性。举例来说, 用于任务330的源代码(例如, 内核源代码)可包含指示是否针对由任务330使用的一个或一个以上存储器对象启用直接模式的编译时间属性。在一些情况下, 编译时间属性可采取OpenCL属性限定符(例如, _cl_immediate)的形式。所述属性限定符可与一个或一个以上特定存储器对象和/或存储于所述一个或一个以上存储器对象内的一个或一个以

上变量相关联。当属性限定符与特定存储器对象相关联时,那么编译器可确定针对存储器对象启用直接模式。类似地,当属性限定符不与特定存储器对象相关联时,那么编译器可确定不针对存储器对象启用直接模式。使用此属性可减少编译器的工作且潜在地减小内核大小。在一些实例中,软件应用可将直接缓冲器的使用限制到需要此些缓冲器的情况。在此些实例中,是否使用直接缓冲器的决策可为编译时间决策。

[0254] 如果编译时间属性指示针对与共享存储器空间336相关联的存储器对象启用直接模式,那么编译器可编译任务330以使得用于任务330的经编译代码包含用于相对于共享存储器空间336发生的读取或写入操作的直接模式读取和/或写入指令。否则,如果未针对与共享存储器空间336相关联的存储器对象启用直接模式,那么编译器可编译任务330以使得用于任务330的经编译代码不包含用于相对于共享存储器空间336发生的读取或写入操作的直接模式读取和/或写入指令。举例来说,编译器可编译任务330以使得用于任务330的经编译代码包含用于相对于共享存储器空间336发生的读取或写入操作的高速缓存读取和/或写入指令。

[0255] 图31是说明根据本发明的用于编译用于任务的源代码的实例性技术的流程图。在一些实例中,使用图31中的技术编译的所得代码对应于图20中的任务330。在图31的实例性技术中,任务330被称作内核。编译器处理由存储器对象实施的内核自变量(466)。编译器确定存储器对象是否为直接模式存储器对象(468)。在一些实例中,编译器可基于内核的源代码中所包含的信息(例如,与内核自变量相关联的编译时间属性)来确定存储器对象是否为直接模式存储器对象。如果编译器确定存储器对象不是直接模式存储器对象,那么编译器使用高速缓存读取和写入指令来编译与特定内核自变量相关联的读取操作和写入操作(470)。另一方面,如果编译器确定存储器对象是直接模式存储器对象,那么编译器使用直接模式读取和写入指令来编译与特定内核自变量相关联的读取操作和写入操作(472)。

[0256] 图32是说明根据本发明的可由GPU用来选择性地使用高速缓存服务的实例性技术的流程图。举例来说,所述技术可允许GPU响应于接收到指定是否应使用高速缓存服务来用于相对于存储器的存储器空间执行读取操作和写入操作中的至少一者的信息而选择性地使用与存储器相关联的GPU高速缓冲存储器来相对于所述存储器空间执行读取操作和写入操作中的至少一者。在一些实例中,图20中所说明的GPU314和/或图27中所说明的GPU420可用于实施图32中所说明的技术。

[0257] GPU314接收读取指令或写入指令中的至少一者进行处理(474)。所接收的指令可指令GPU314相对于存储器的存储器空间执行读取操作和写入操作中的至少一者。GPU314接收指定是否应使用高速缓存服务来用于相对于存储器空间执行读取操作和写入操作中的至少一者的信息(476)。在一些实例中,高速缓冲存储器模式信息可包含在所接收的指令内。在进一步的实例中,高速缓冲存储器模式信息可为与存储器空间相关联的存储器对象的直接模式属性。GPU314确定是否基于高速缓冲存储器模式信息(478)来使用高速缓存服务。响应于接收到指定应使用高速缓存服务来用于执行所接收的指令的信息,GPU314可使用高速缓存服务来执行所接收的指令(480)。响应于接收到指定不应使用高速缓存服务来用于执行所接收的指令的信息,GPU314可不使用高速缓存服务来执行所接收的指令(482)。在一些实例中,GPU314可使用图23到28以及30中所说明的技术中的一者或一者以上来实施决策框478以及过程框480和482中的一者或一者以上。在一些情况下,可使用GPU高速缓冲

存储器控制模块或存储器管理单元(例如,图20中所说明的GPU高速缓冲存储器控制模块334)来实施图32中所示的技术。在额外情况下,可使用总线控制器(例如,图27中所说明的总线控制器434)来实施图32中所示的技术。

[0258] 在一些实例中,为了实施直接存储器对象,GPUALU可经设计以执行ALU指令,所述ALU指令使指令中所指定的全局存储器高速缓冲存储器和/或全局存储器高速缓冲存储器的特定部分无效。一般来说,主机装置312可使用现有的GPU能力来实施直接存储器对象。

[0259] 现在将进一步详细地论述本发明中所描述的带外信令技术(例如,本文中所描述的消息传递技术)和本发明中所描述的直接存储器对象的各种使用情况。根据第一使用情况,可将带外信令用作独立特征,而没有必要除了带外信令技术之外还使用直接存储器对象。带外信令可用于同步以及快速地传递相对少量的数据。在一些实例中,带外信令可具有比直接存储器对象低的等待时间,但具有比直接存储器对象低的带宽。

[0260] 还可根据存储器分配操作的第一使用情况使用带外信令。举例来说,GPU可使用带外信令来请求主机CPU分配新的缓冲器。GPU还可使用带外信令来向主机CPU指定所请求的缓冲器长度。作为另一实例,CPU可在分配缓冲器之后使用带外信令以向GPU发送指定用于缓冲器的存储器位置的指针。

[0261] 还可根据远程过程调用的第一使用情况(其中将交换少量的数据)使用带外信令。举例来说,在其中在计算装置内的计算单元上执行的内核使用RPC以在同一计算装置或在另一计算装置中的另一计算单元上启动另一内核的情况下,用于所述RPC的数据可能被存储在启动的计算单元的本地存储器中。本发明的带外信令技术可用于将数据从启动的计算单元的本地存储器传送到执行新启动的内核的计算单元的本地存储器。

[0262] 还可根据进展报告的第一使用情况使用带外信令。举例来说,GPU可使用带外信令来向主机CPU报告当前任务的完成百分比。

[0263] 还可根据错误报告的第一使用情况使用带外信令。举例来说,GPU可使用带外信令来向主机CPU报告错误代码。

[0264] 还可根据辅助上下文切换的第一使用情况使用带外信令。举例来说,主机CPU可使用带外信令来请求GPU保存状态以准备上下文切换。

[0265] 根据第二使用情况,可将直接存储器对象用作独立特征,而没有必要除了直接存储器对象之外还使用带外信令。举例来说,可使用直接缓冲器来完成相对大量数据的交换。直接缓冲器可不仅含有数据,而且还有同步标记。在此情况下,数据产生器可首先将数据写入到缓冲器,且随后写入同步标记,所述同步标记向消费者指示数据的就绪和/或位置。消费者在先验决定的位置中寻找同步数据,例如通过轮询此存储器位置而在缓冲器的标头区段中寻找同步数据。一旦获得同步标记,消费者便读取数据。可将类似技术应用于直接图像对象。

[0266] 可针对这些技术使用多种同步协议。举例来说,可将同步标记嵌入在数据缓冲器中,或可定位在单独的缓冲器中。可将这些技术应用于使用可变长度编码或行程长度编码方案压缩的经压缩数据的传输。

[0267] 根据第三使用情况,可与带外信令联合地使用直接存储器对象(例如)以完成对相对大量数据的交换。在此情况下,可在直接存储器对象存储数据的同时将带外信令用于同步。举例来说,数据产生器可将数据放置到直接缓冲器中,且使用带外信令向消费者通知数

据的就绪以及位置和/或大小。在流控制情形中,消费者读取数据且通知产生器可再次使用缓冲器。还可使用带外信令来完成所述通知。

[0268] 这些技术可用于需要流控制数据流水线的算法中。对于主机CPU和GPU,可使用这些技术(例如)用于诊断记录。对于多个OpenCL计算装置,可使用这些技术将多个装置连接到异步流控制数据管线中。这可允许应用被分解为更适合于每一CPU或GPU的若干块,启动多个装置上的各种管线处理级,且/或从主机CPU卸载大多数或甚至全部的数据同步。

[0269] 在一些实例中,本发明的技术可为使用命令队列起始任务的多处理器计算平台提供消息传递接口,所述消息传递接口促进在主机装置上执行的进程与在计算装置上执行的任务之间的消息的发送和接收。在一些情况下,所述计算装置可为GPU。在额外情况下,所述计算装置可为由跨平台、跨供应商、异类计算平台API界定的任何类型的计算装置。

[0270] 在进一步的实例中,本发明的技术可提供包含可由主机装置存取的一个或一个以上寄存器的GPU。所述一个或一个以上寄存器可经配置以促进在GPU上执行的任务与在不同于GPU的装置上执行的进程之间的消息传递。

[0271] 在额外实例中,本发明的技术可提供允许创建直接存储器对象的存储器缓冲器接口。所述直接存储器对象可用于实施非可高速缓存共享存储器空间和/或高速缓冲存储器相干共享存储器空间,以便在于计算装置上执行的任务正在计算装置上执行时在于主机装置上执行的进程与所述任务之间共享数据。在一些情况下,所述计算装置可为图形处理单元(GPU)。在额外情况下,所述计算装置可为由跨平台、跨供应商、异类计算平台API界定的任何类型的计算装置。

[0272] 在又进一步的实例中,本发明的技术可提供包含用于共享存储器空间的高速缓冲存储器的GPU,所述共享存储器空间可被选择性地停用以便提供非可高速缓存共享存储器空间。在额外实例中,本发明的技术可提供包含高速缓冲存储器相干模式的GPU,所述高速缓冲存储器相干模式可被选择性地启用以提供高速缓冲存储器相干共享存储器空间。

[0273] 本发明中所描述的技术可至少部分以硬件、软件、固件,或其组合来实施。举例来说,所描述技术的各种方面可实施于以下各项内:一个或一个以上处理器(包含一个或一个以上微处理器)、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,以及此些组件的任何组合。术语“处理器”或“处理电路”通常可指代单独或与其它逻辑电路组合的前述逻辑电路中的任一者,或任何其它等效电路,例如执行处理的离散硬件。

[0274] 此类硬件、软件和固件可实施于相同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实施,或单独地实施为离散但可互操作的逻辑装置。将不同特征描绘为模块或单元意在突出不同功能方面,且未必暗示必须通过单独的硬件或软件组件来实现此些模块或单元。而是,与一个或一个以上模块或单元相关联的功能性可由单独的硬件、固件和/或软件组件执行,或集成于共用或单独的硬件或软件组件内。

[0275] 本发明中所描述的技术还可存储、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(例如)在所述指令被一个或一个以上处理器执行时可致使一个或一个以上处理器执行本文中所描述的技术。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)、可编程只读存储器

(PROM)、可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)、快闪存储器、硬盘、CD-ROM、软盘、盒式磁带、磁性媒体、光学媒体或有形的其它计算机可读存储媒体。

[0276] 计算机可读媒体可包含对应于有形存储媒体(例如上文所列举的有形存储媒体)的计算机可读存储媒体。计算机可读媒体还可包括通信媒体,通信媒体包括(例如)根据通信协议促进将计算机程序从一处传送到另一处的任何媒体。以此方式,短语“计算机可读媒体”一般可对应于(1)非暂时性的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的计算机可读通信媒体。

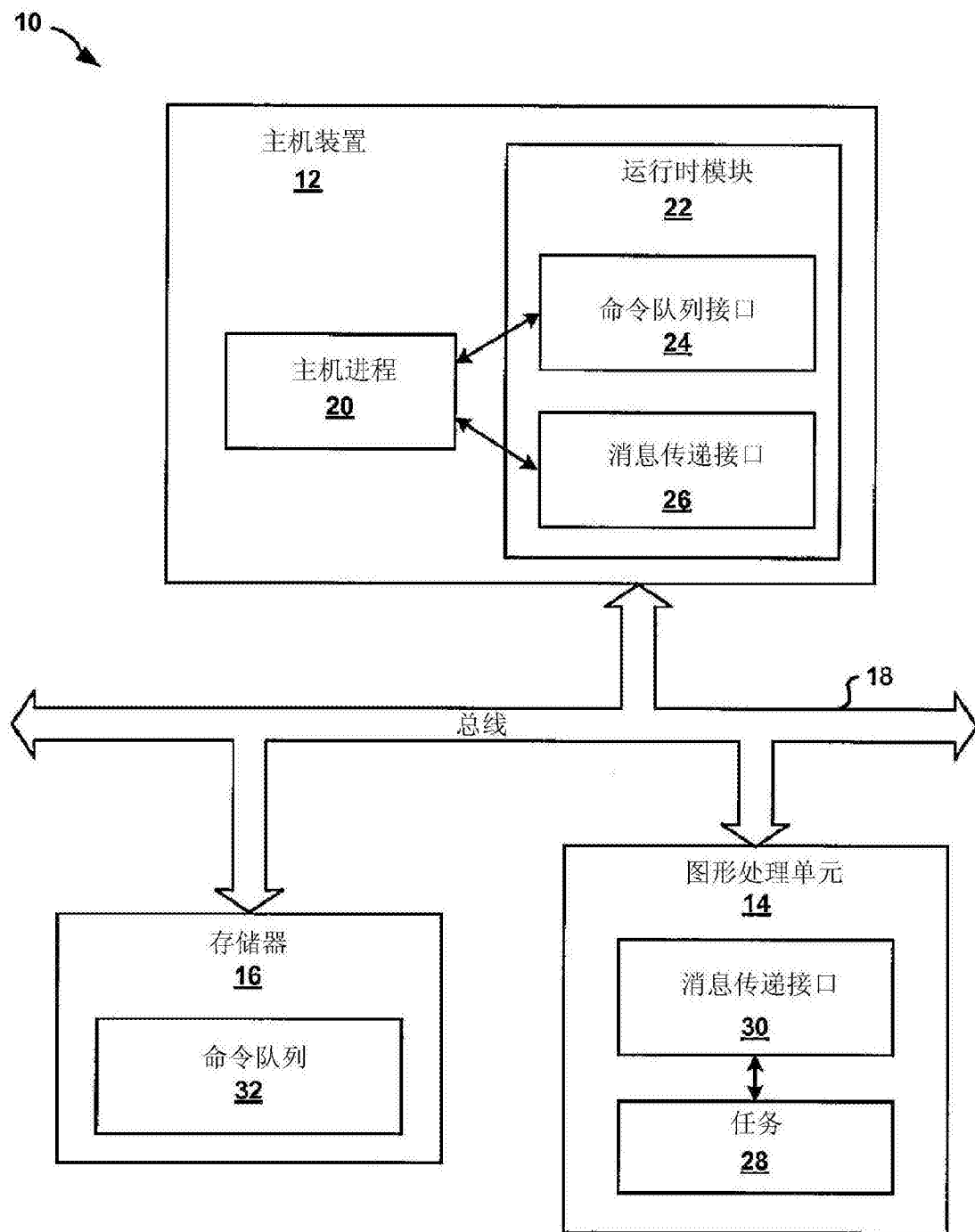


图1

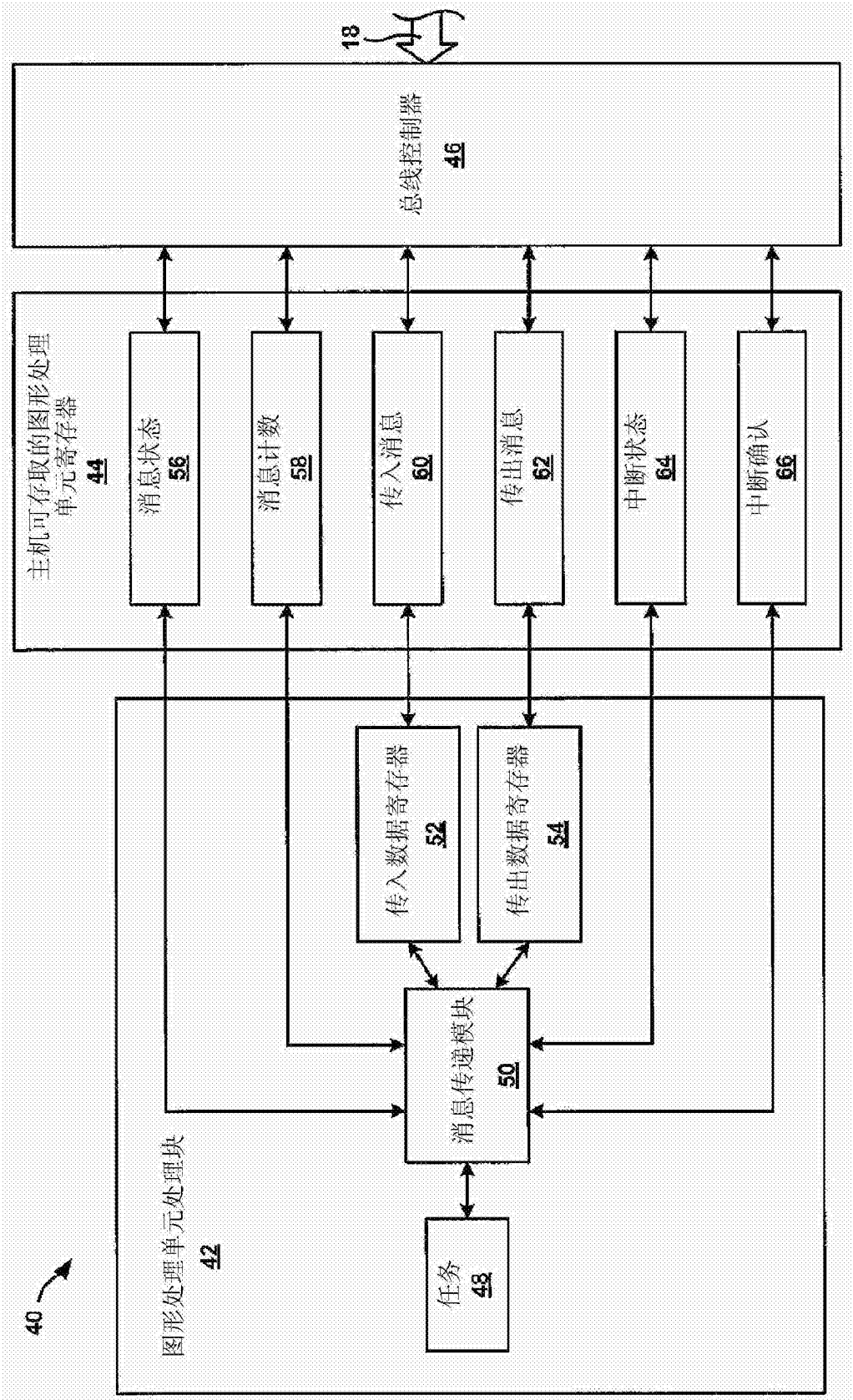


图2

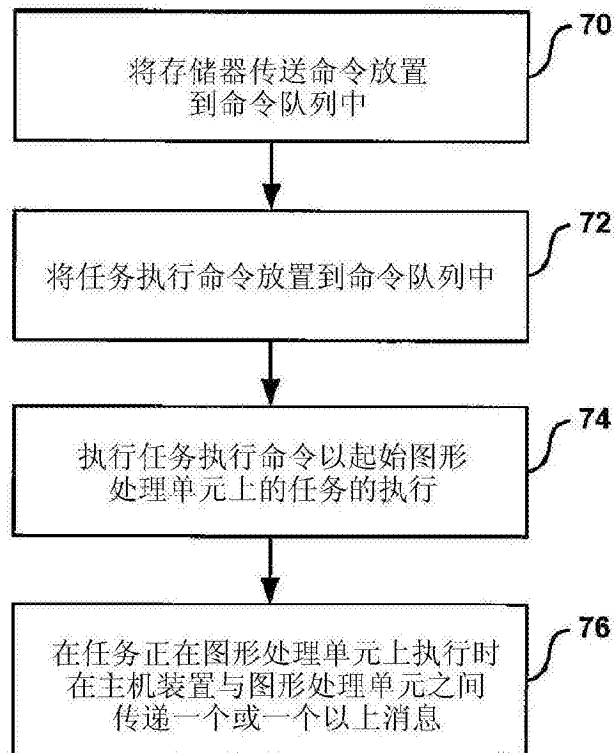


图3

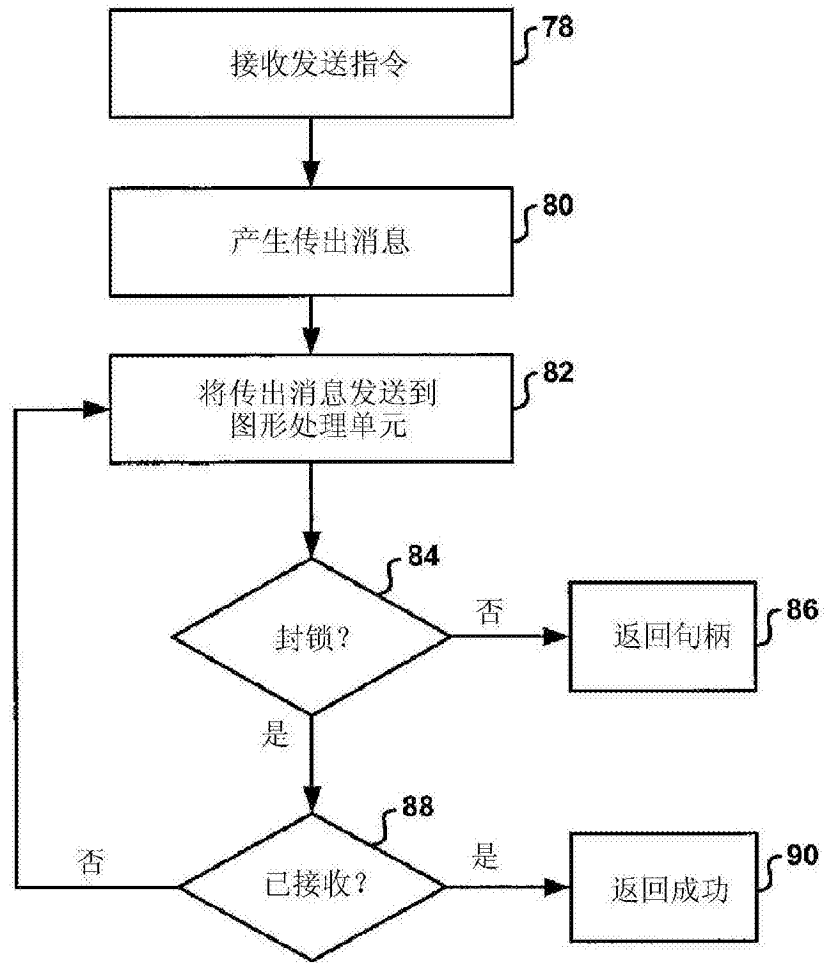


图4

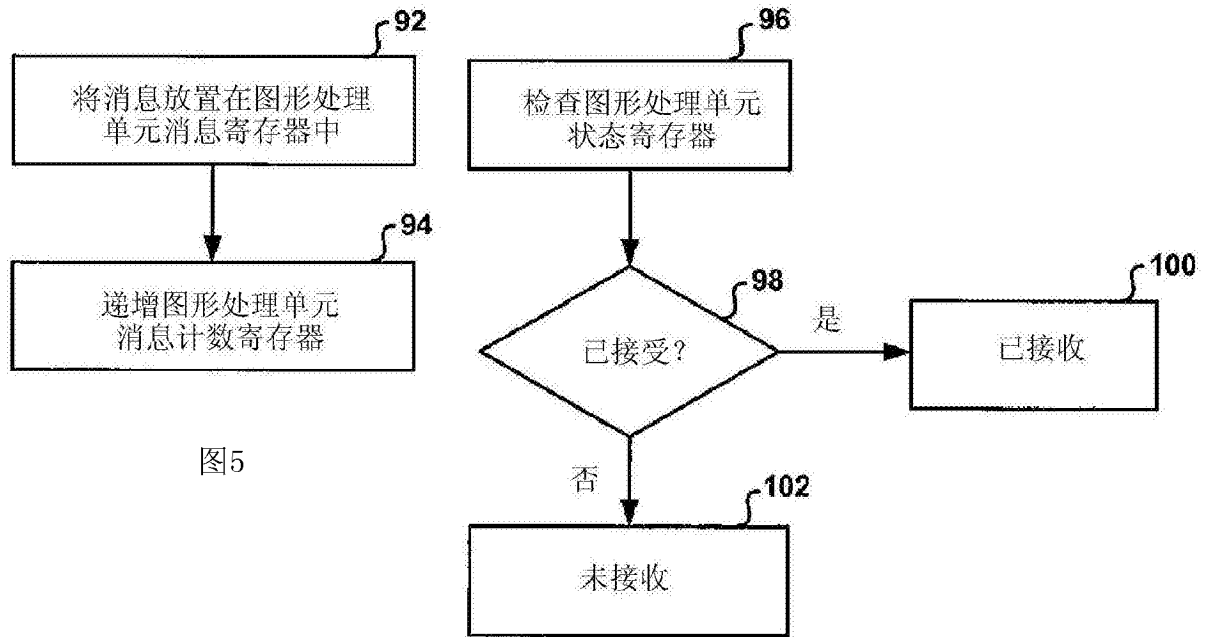
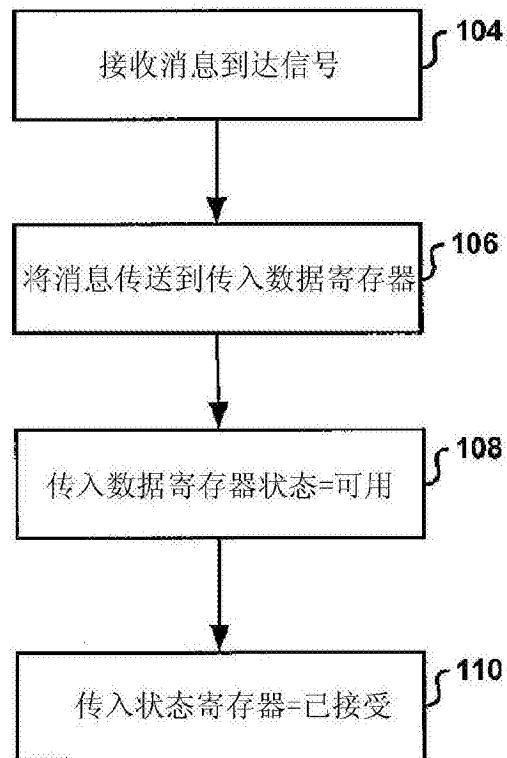


图6



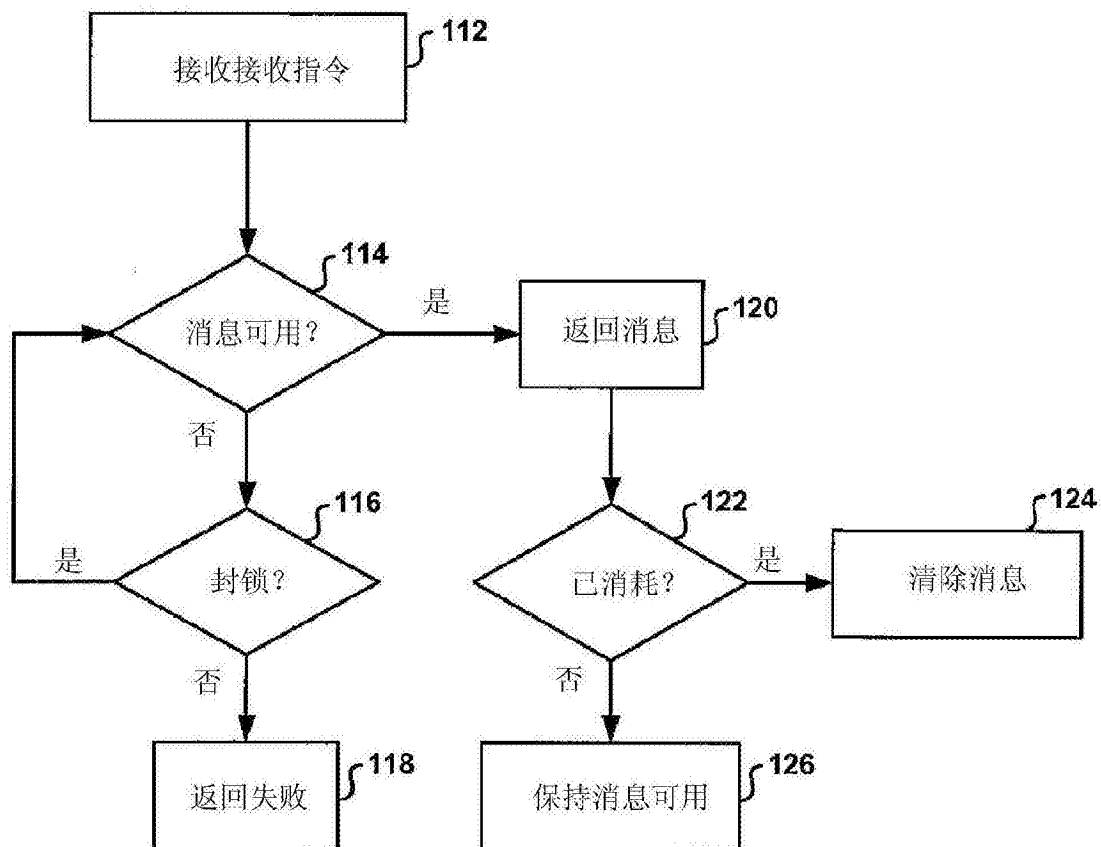


图8

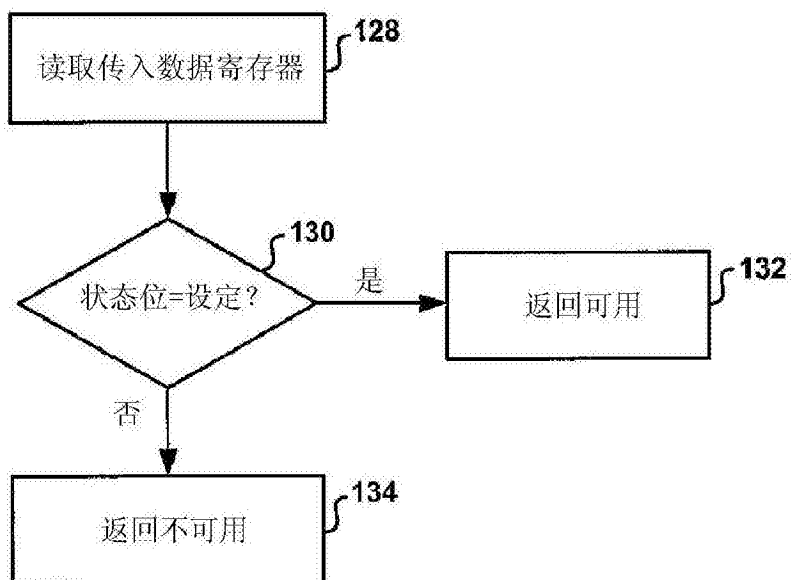


图9

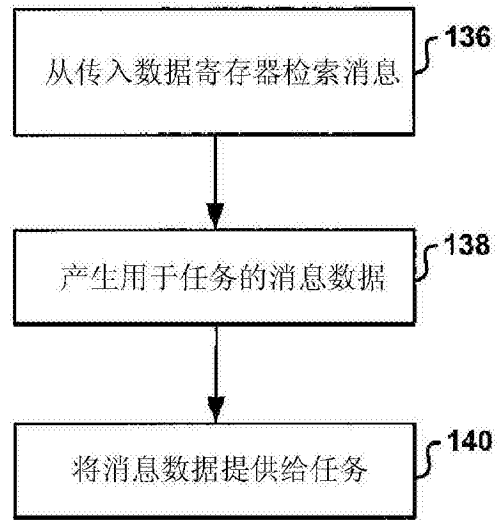


图10

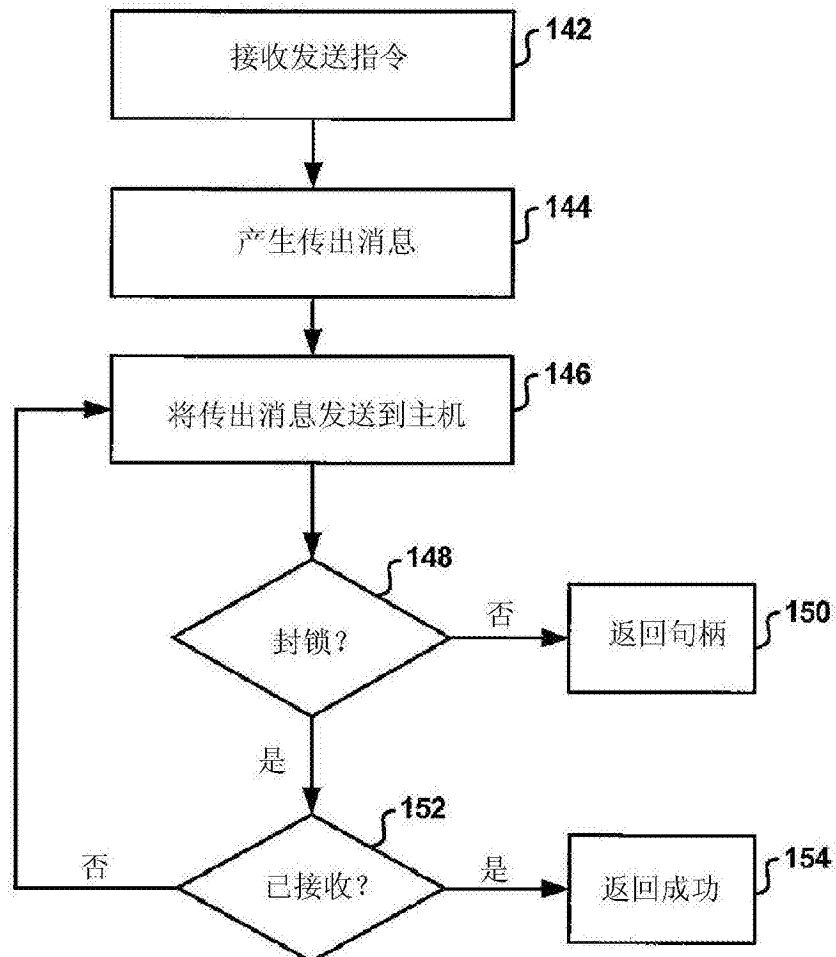


图11

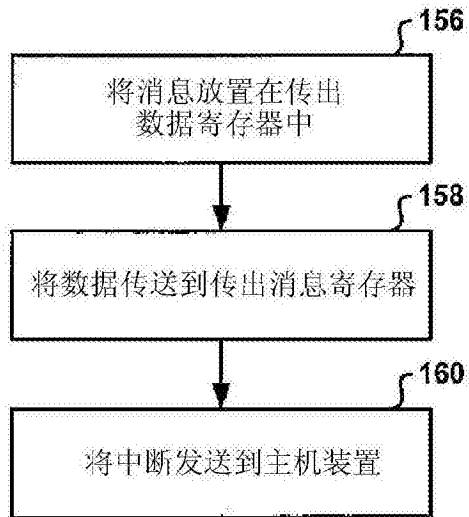


图12

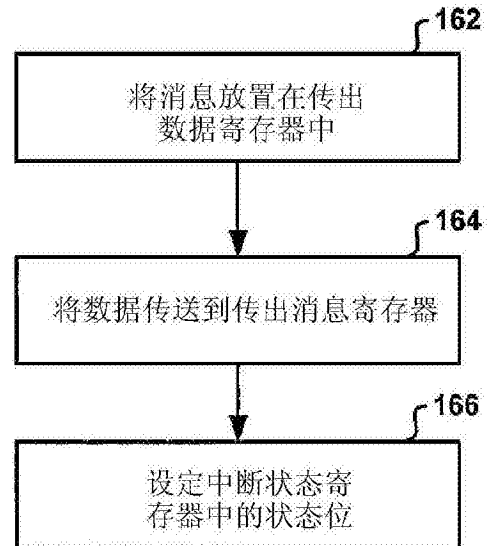


图13

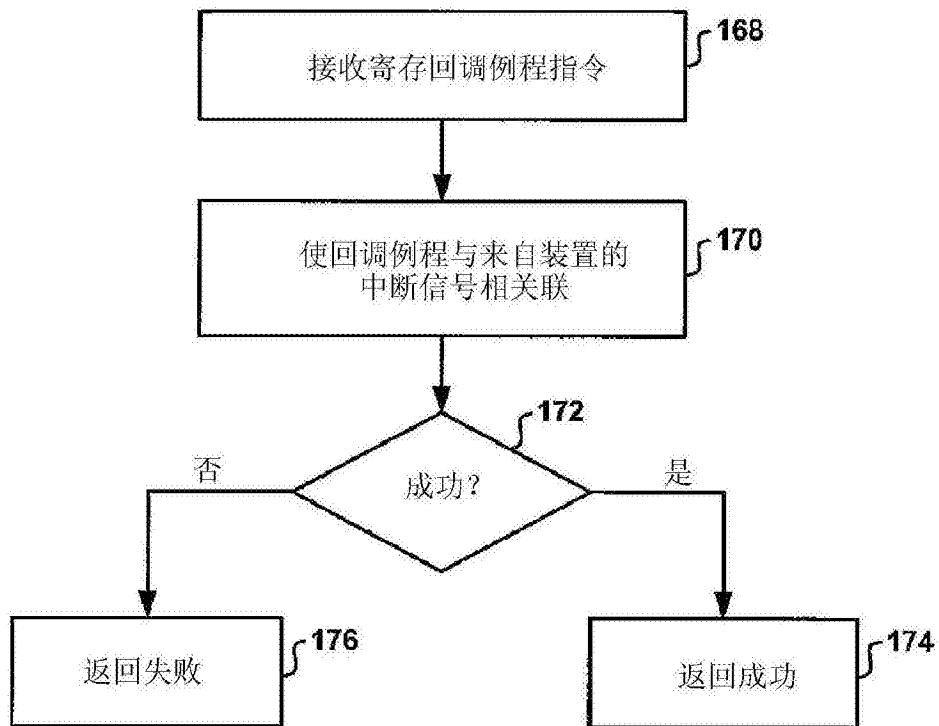


图14

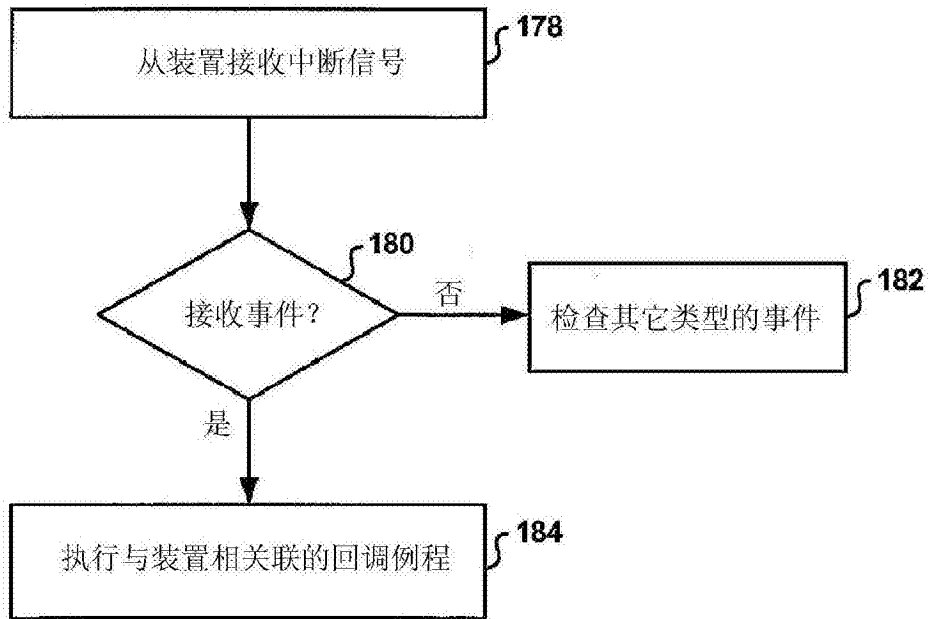


图15

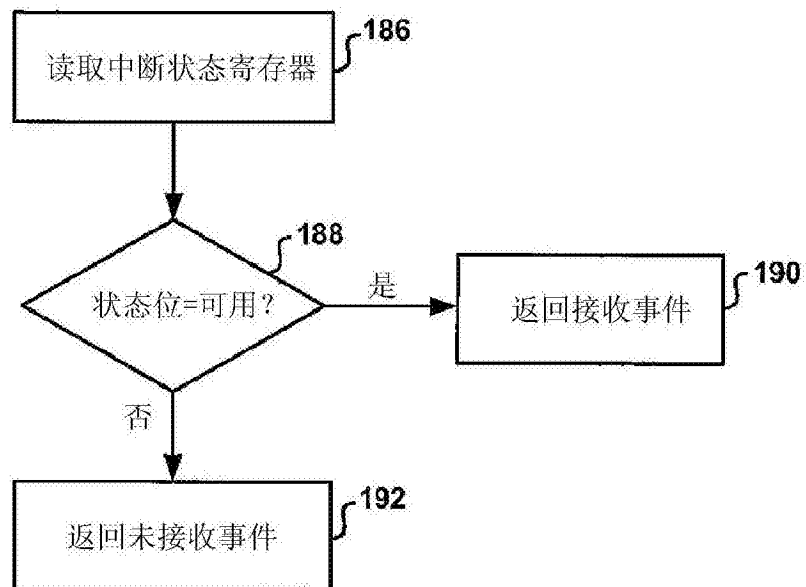


图16

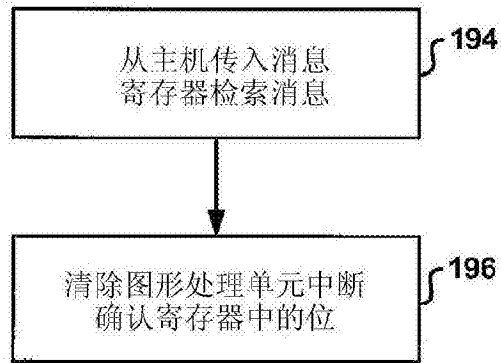


图17

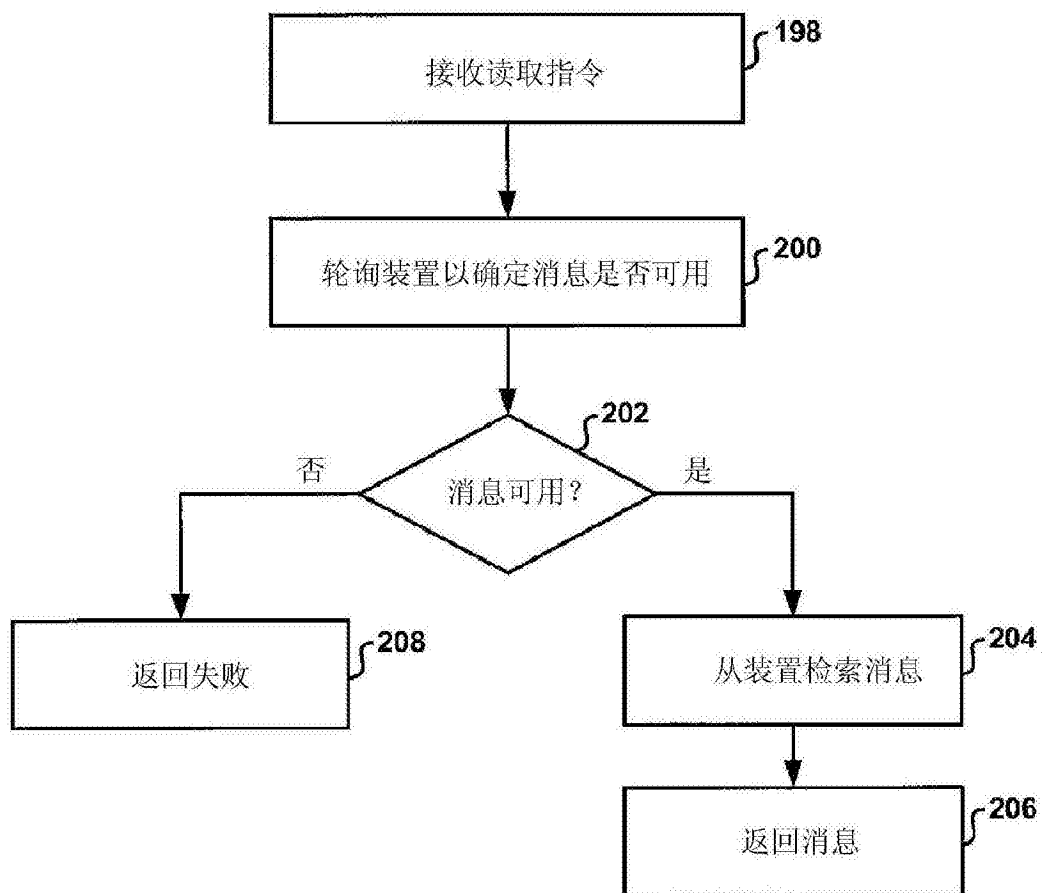


图18

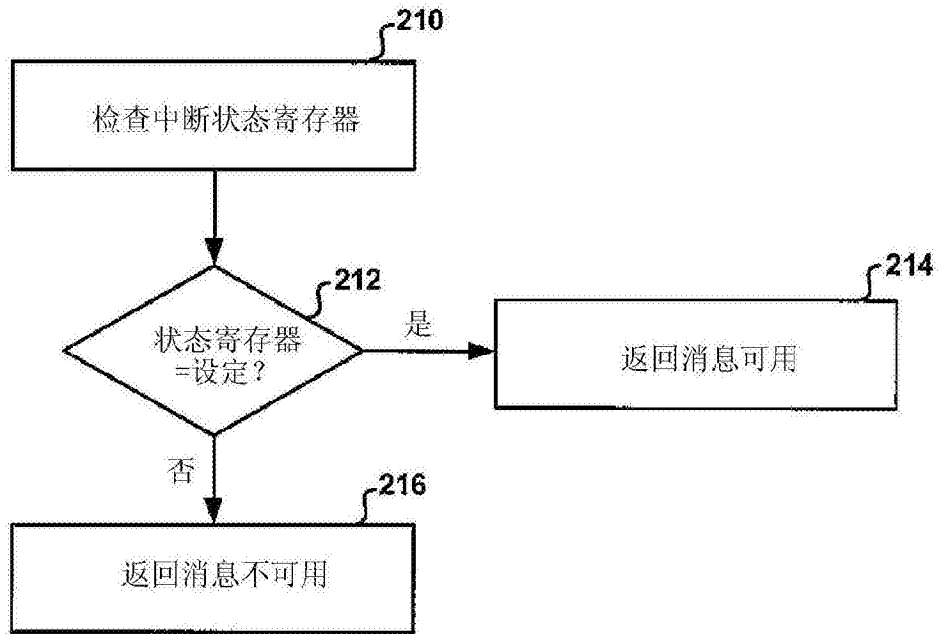


图19

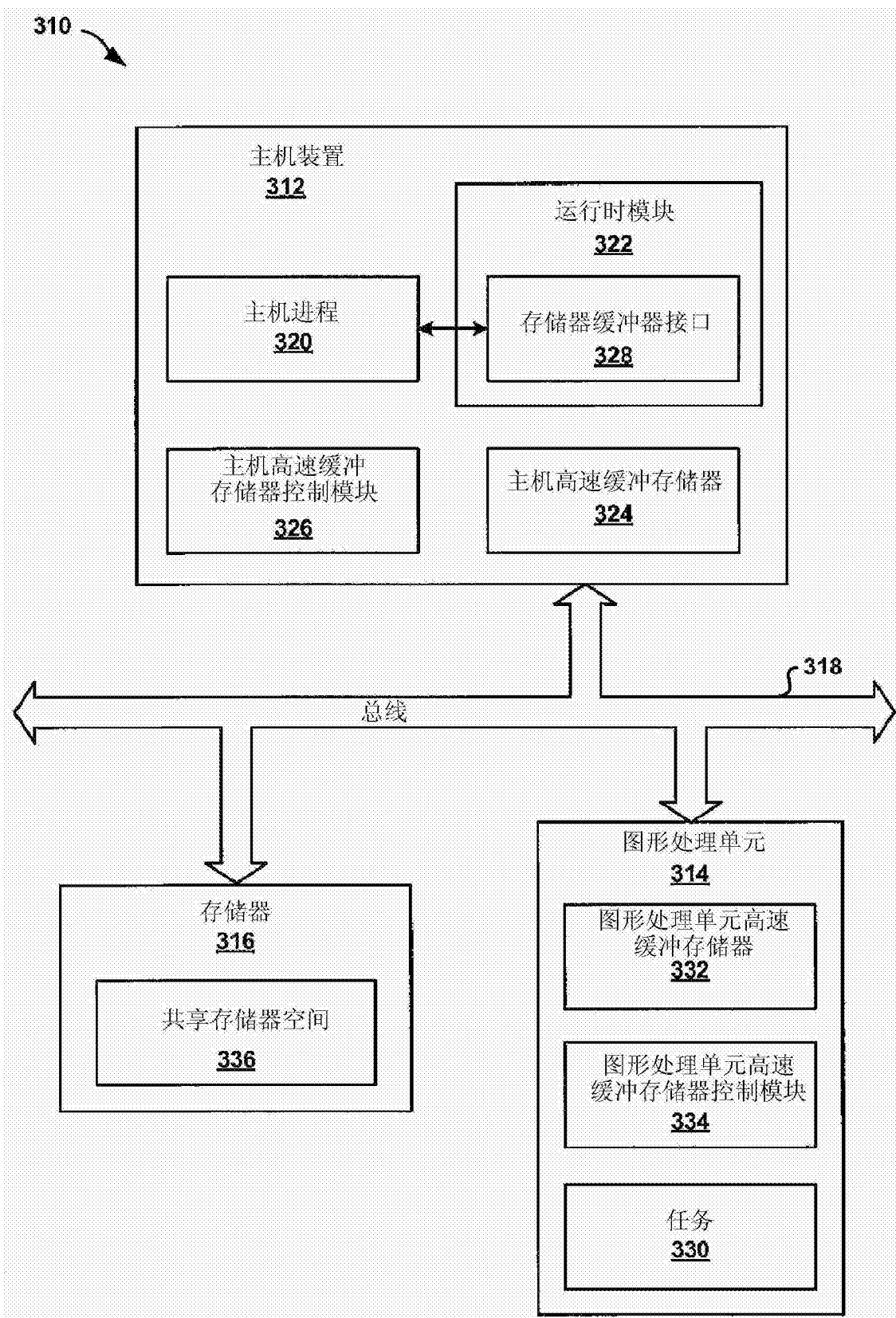


图20

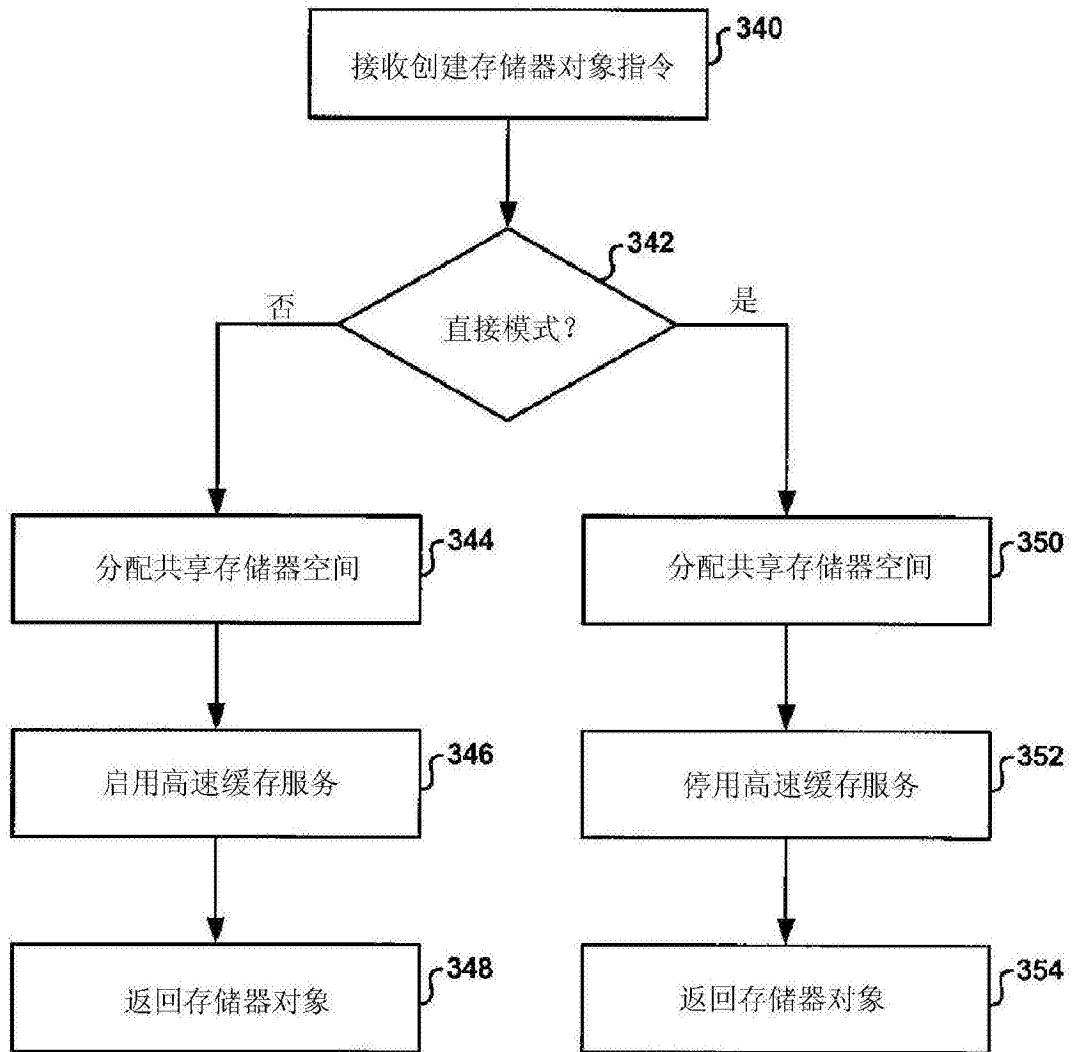


图21

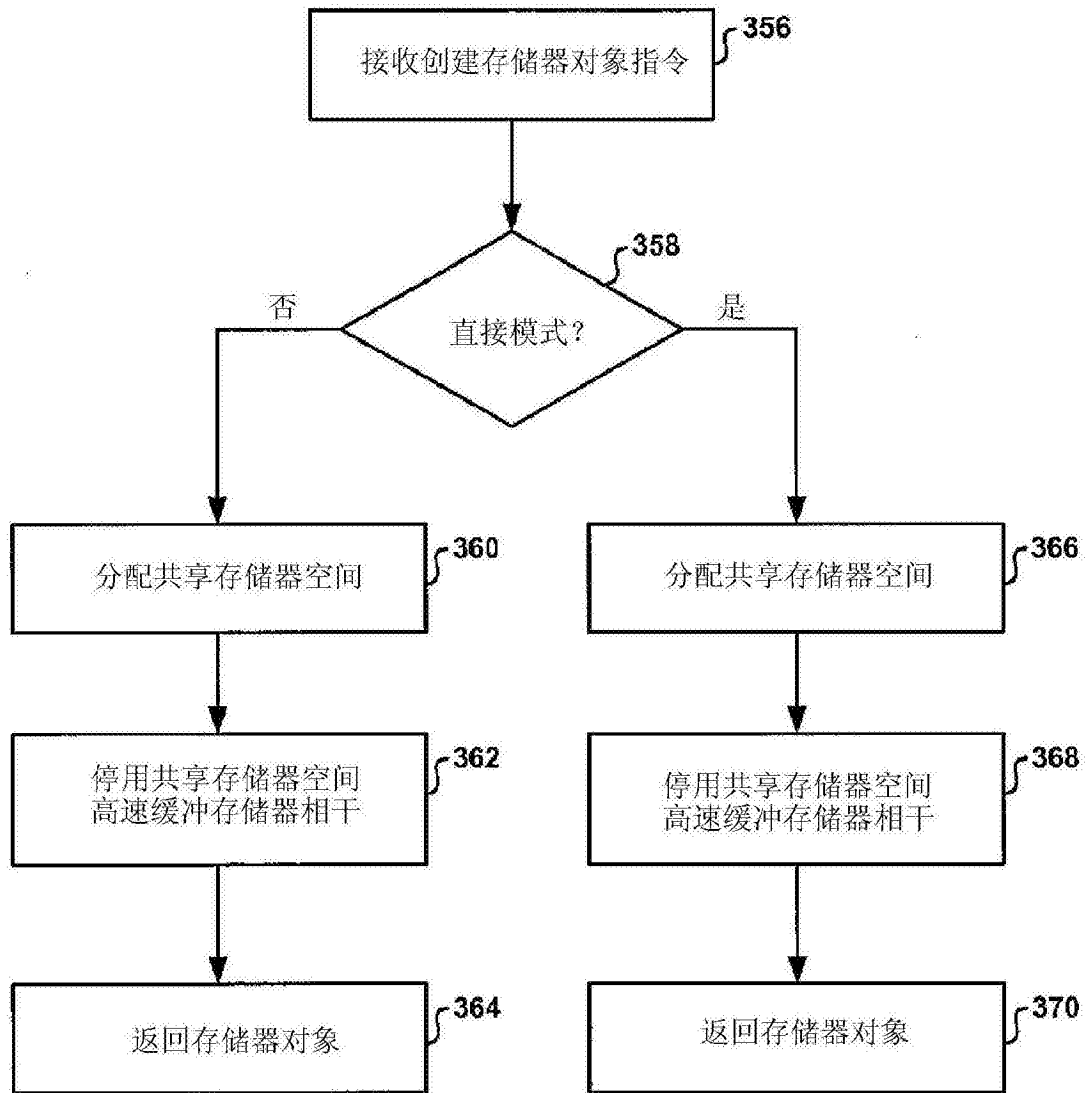


图22

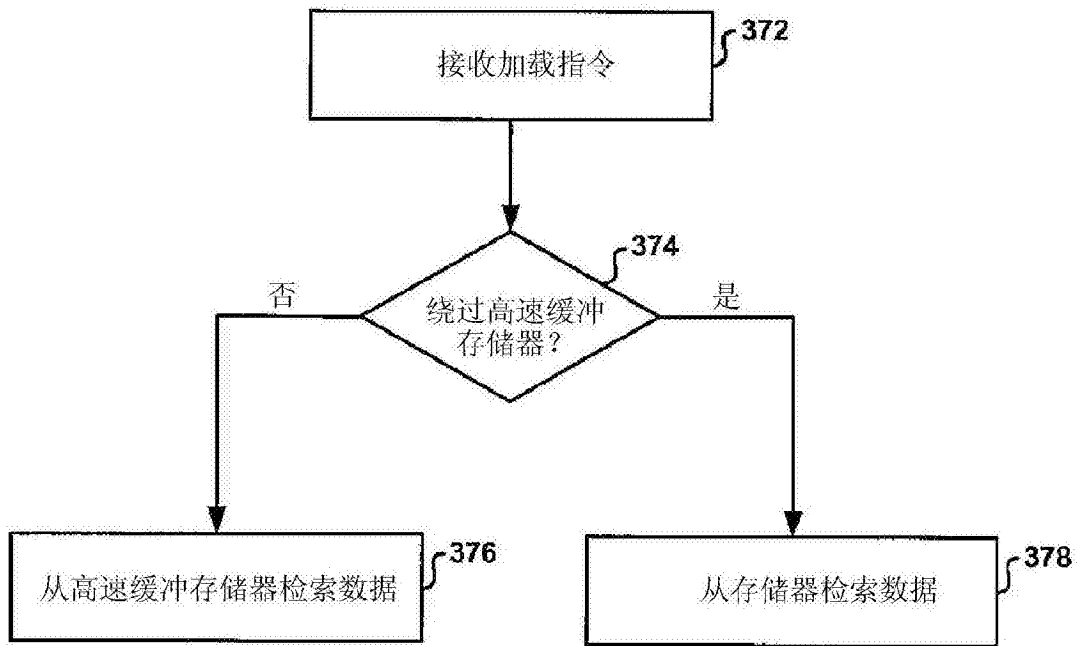


图23

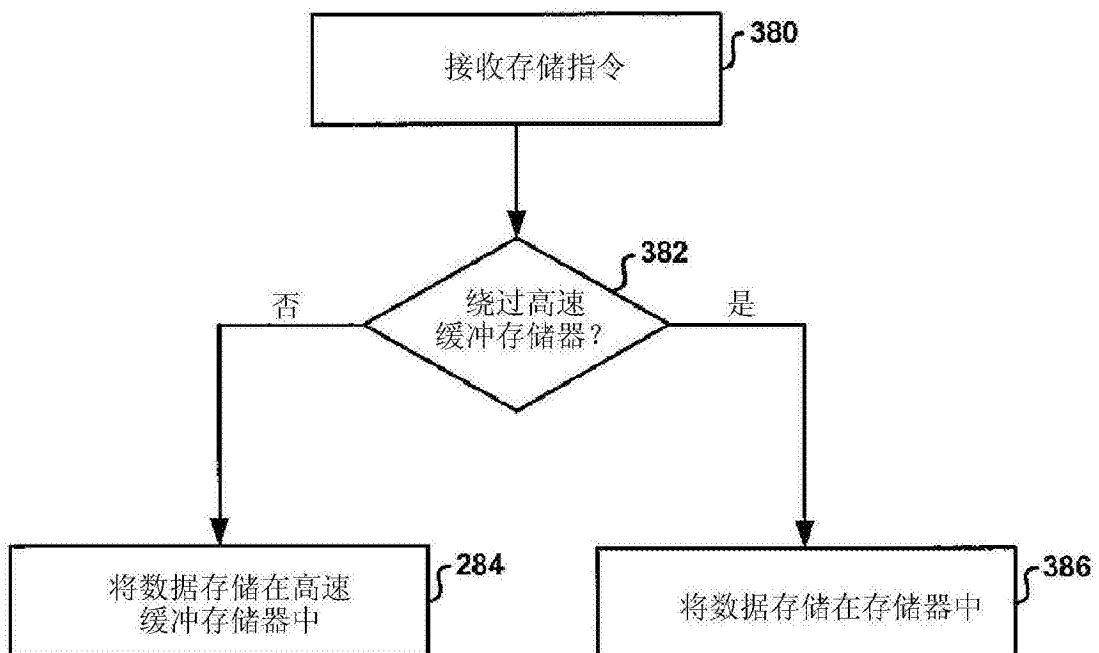


图24

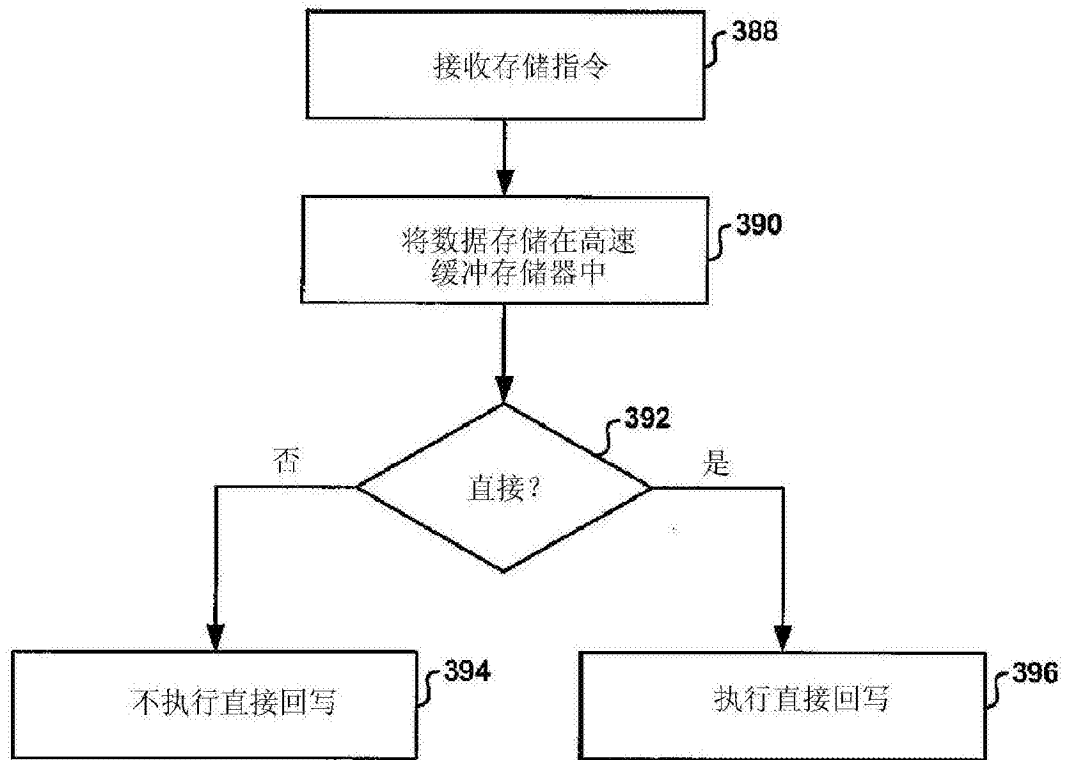


图25

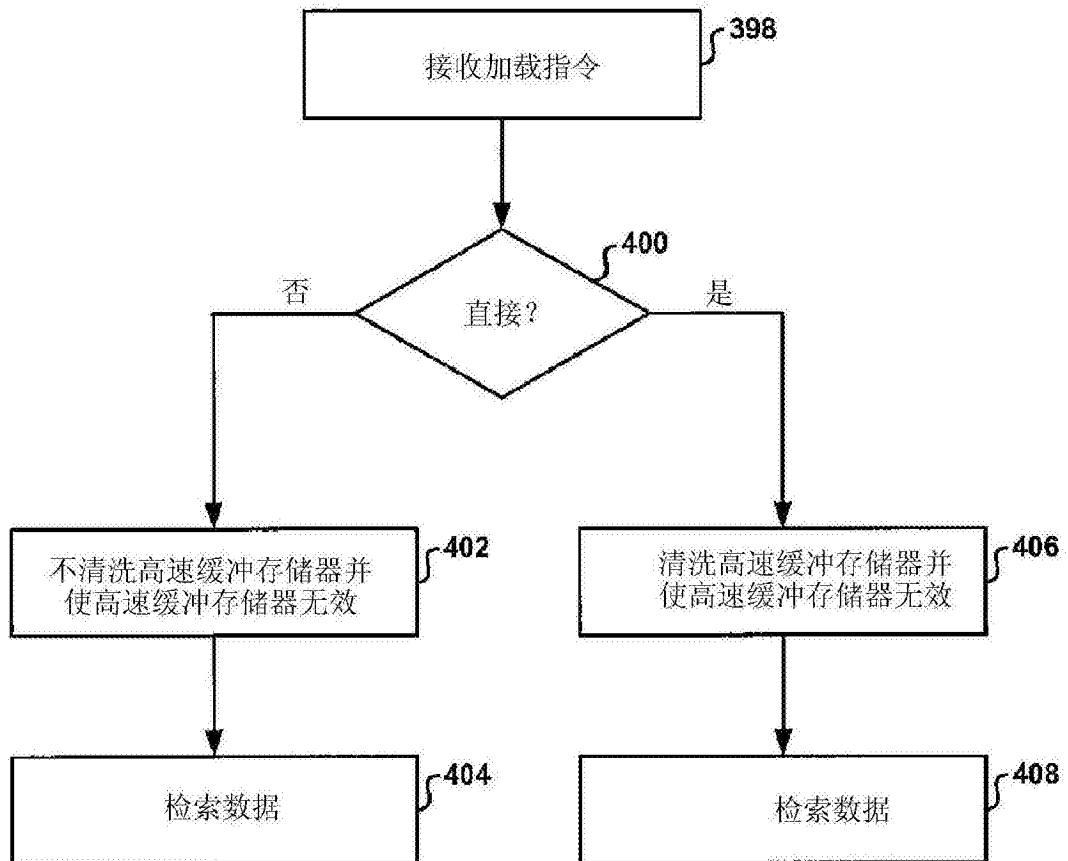


图26

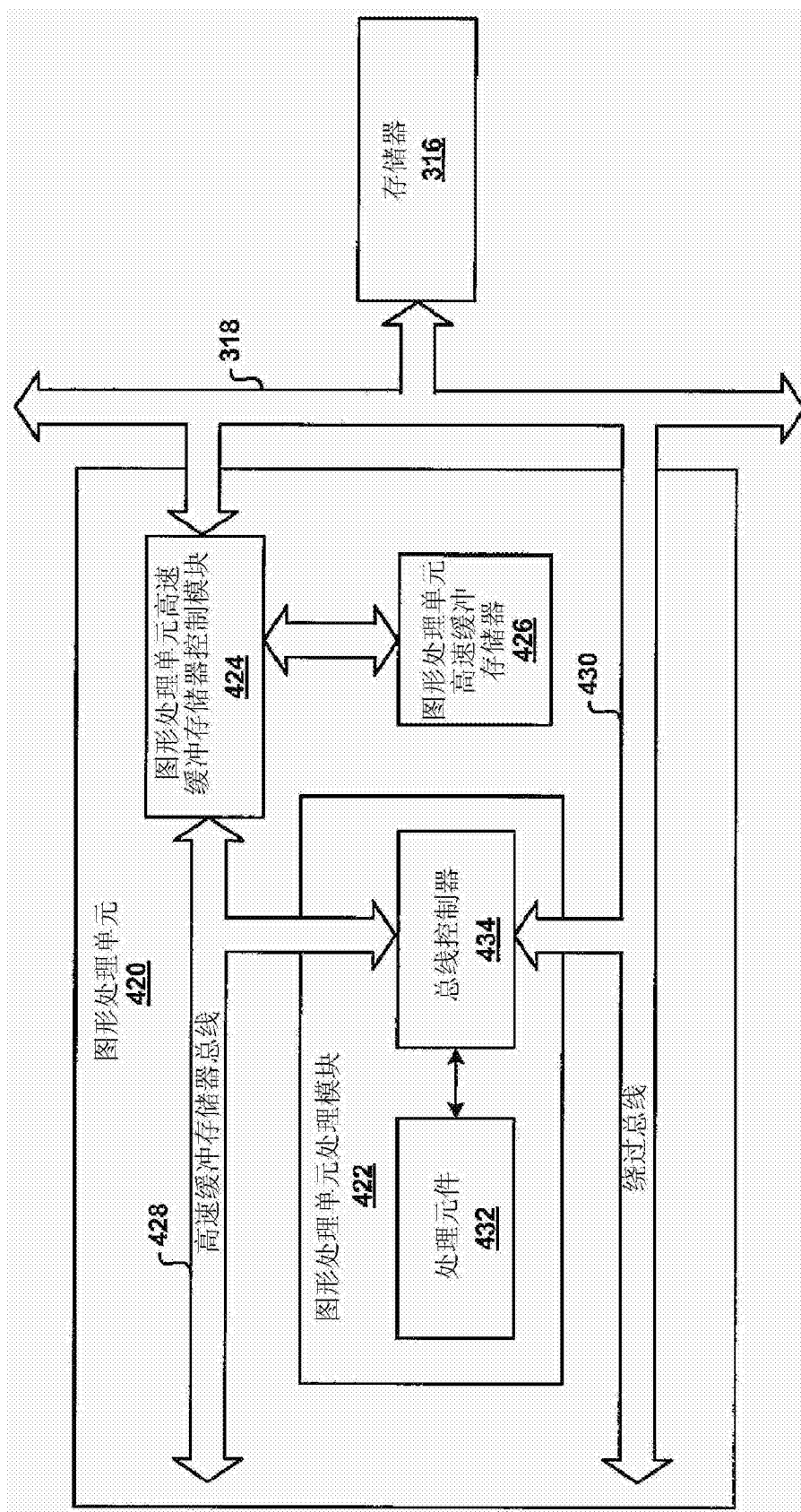


图27

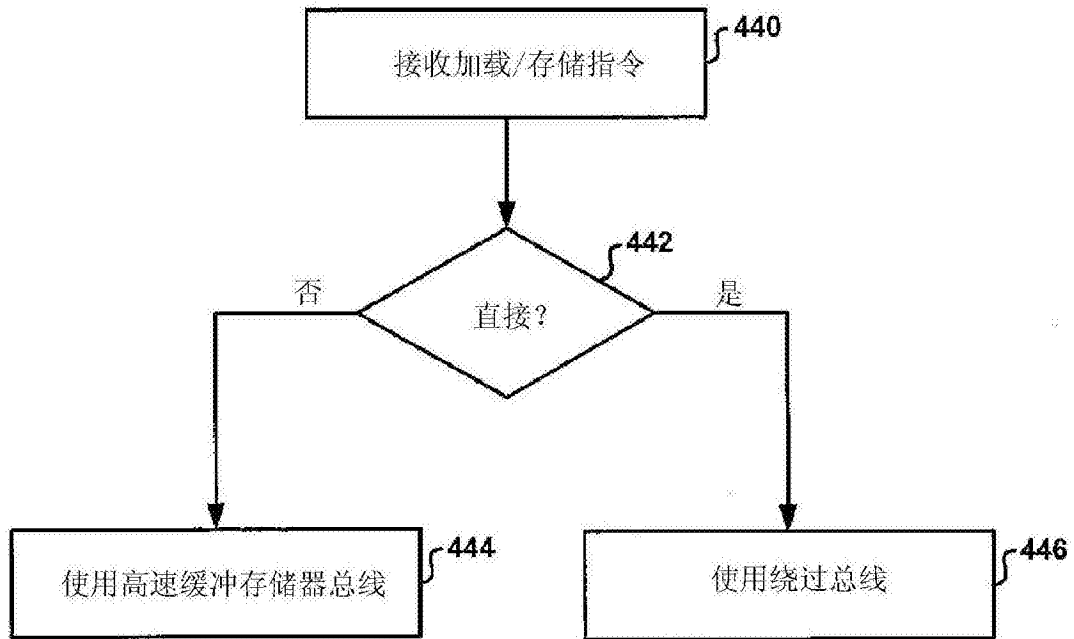


图28

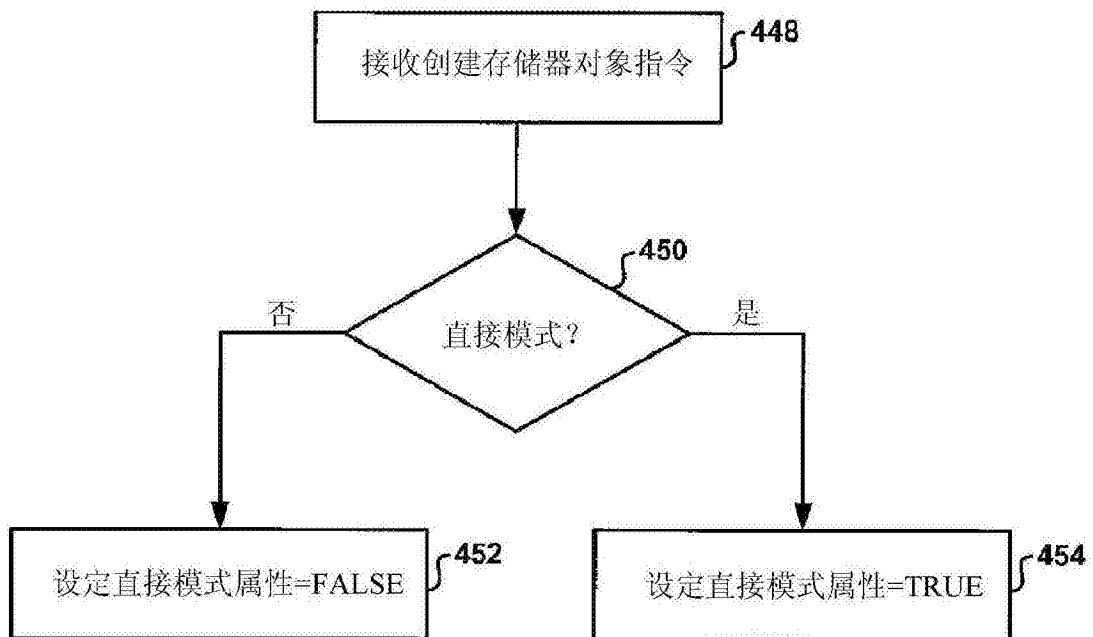


图29

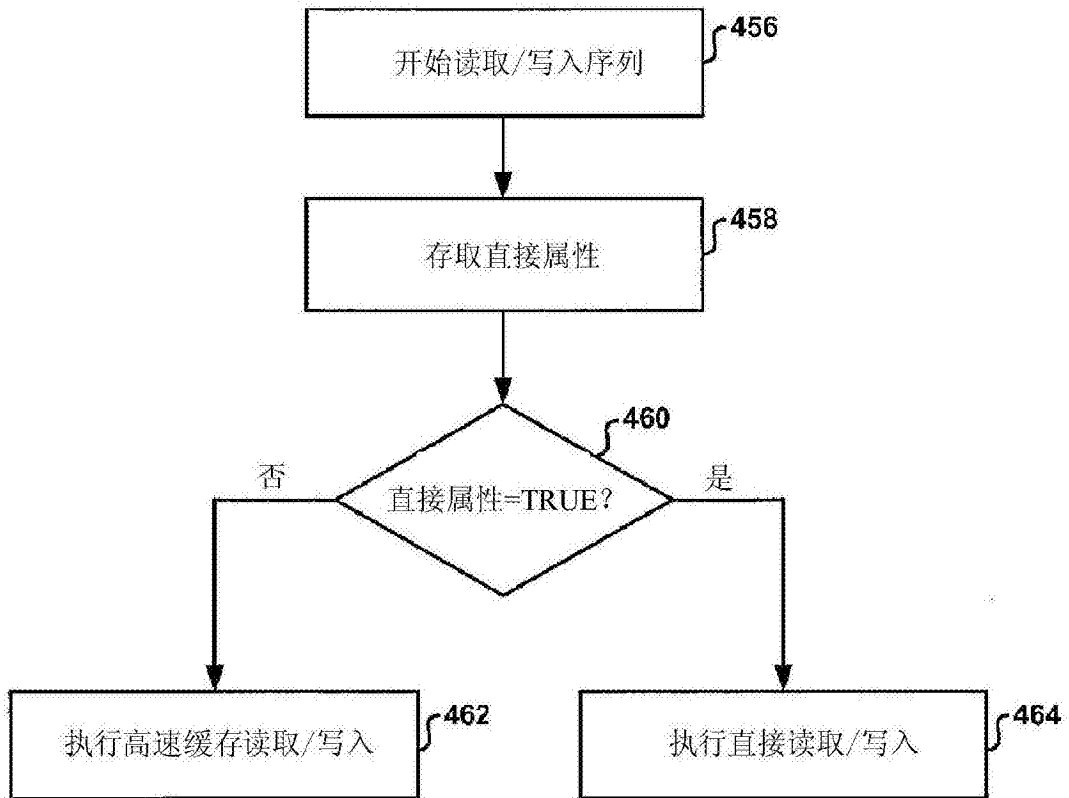


图30

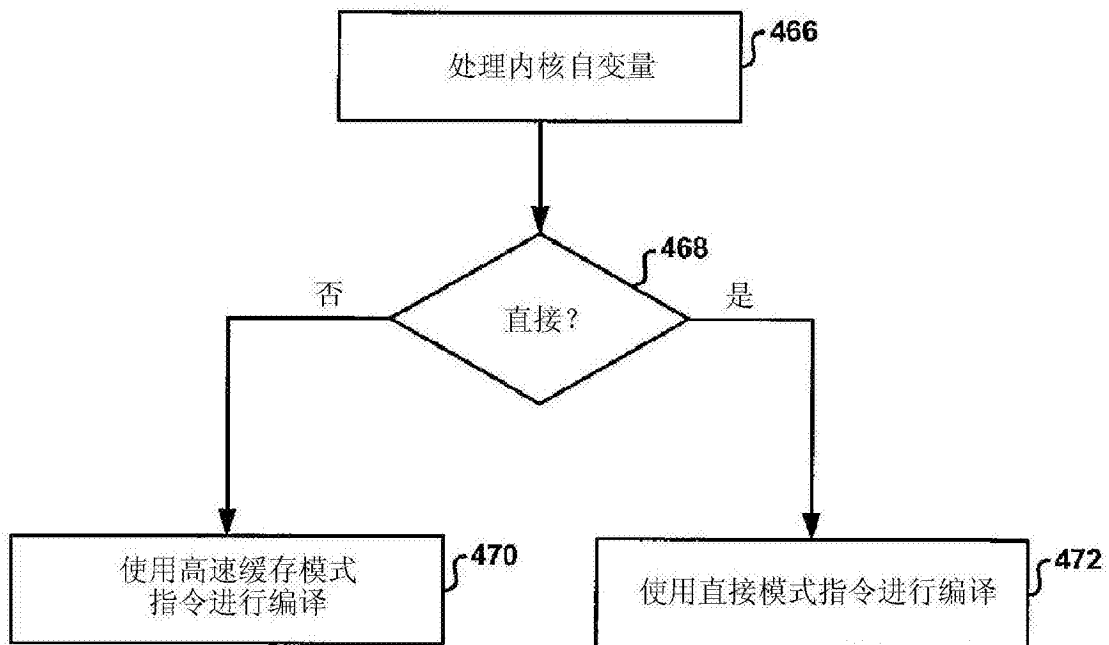


图31

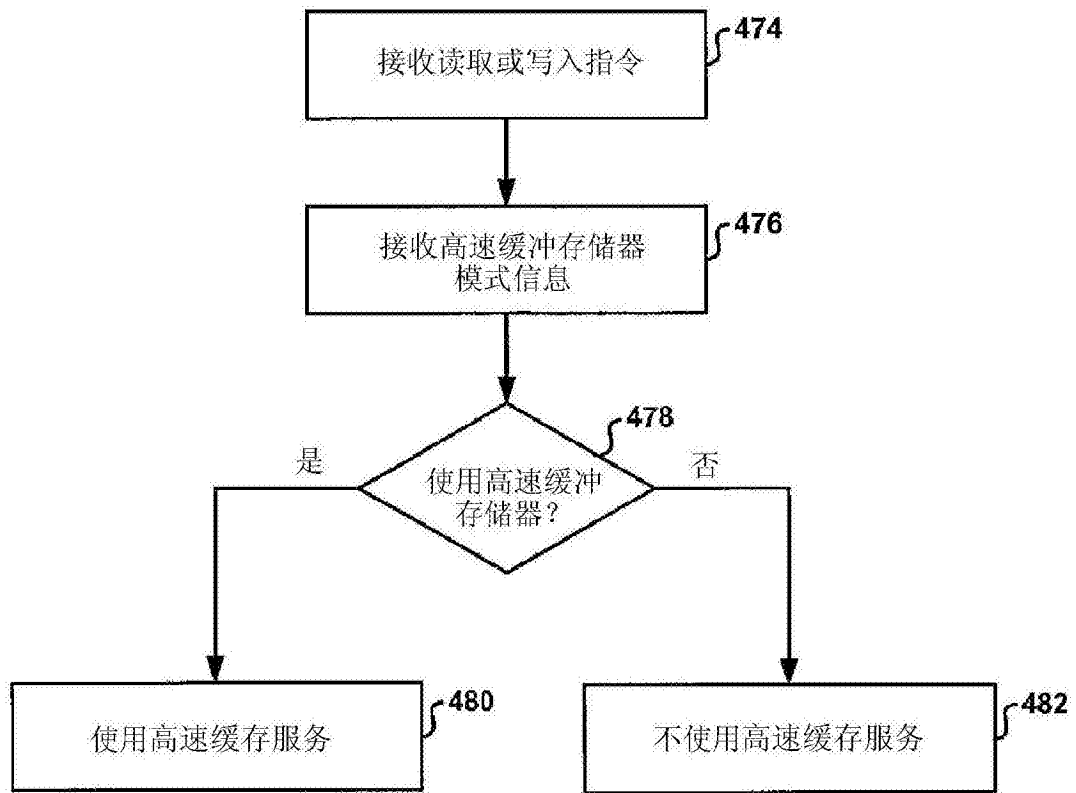


图32