



(12) 发明专利

(10) 授权公告号 CN 108292162 B

(45) 授权公告日 2021.08.31

(21) 申请号 201680070274.5

(22) 申请日 2016.12.12

(65) 同一申请的已公布的文献号

申请公布号 CN 108292162 A

(43) 申请公布日 2018.07.17

(30) 优先权数据

14/966,631 2015.12.11 US

(85) PCT国际申请进入国家阶段日

2018.05.31

(86) PCT国际申请的申请数据

PCT/US2016/066106 2016.12.12

(87) PCT国际申请的公布数据

W02017/100748 EN 2017.06.15

(73) 专利权人 图芯芯片技术有限公司

地址 美国加利福尼亚州

(72) 发明人 曼基特·洛

(74) 专利代理机构 上海光华专利事务所(普通
合伙) 31219

代理人 余明伟

(51) Int.CI.

G06F 3/00 (2006.01)

(56) 对比文件

CN 102388372 A, 2012.03.21

CN 103294753 A, 2013.09.11

CN 103106122 A, 2013.05.15

US 2011320680 A1, 2011.12.29

审查员 王晨霞

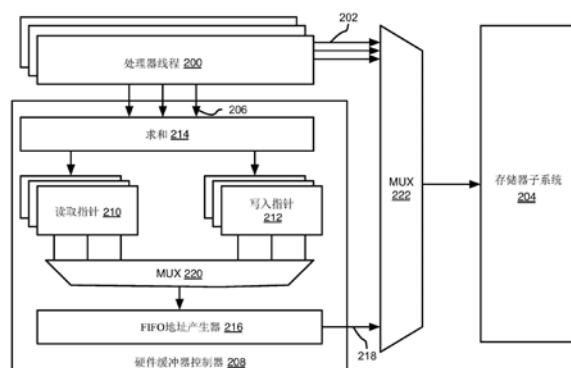
权利要求书4页 说明书8页 附图5页

(54) 发明名称

用于多线程访问的软件定义FIFO缓冲器

(57) 摘要

计算机系统包含硬件缓冲器控制器。对缓冲器的存储器访问请求不包含所述缓冲器内的地址，并且访问所述缓冲器的线程不访问或直接更新指向所述缓冲器内的位置的任何指针。所述存储器访问请求被寻址到所述硬件缓冲器控制器，所述硬件缓冲器控制器从其当前状态确定地址并且向所述地址发布存储器访问命令。所述硬件缓冲器控制器响应于所述存储器访问请求而更新其状态。响应于上溢或下溢条件或接近上溢或接近下溢条件，所述硬件缓冲器控制器评估其状态并且将事件输出到线程调度器。所述线程调度器随后可以阻止线程向所述硬件缓冲器控制器发布存储器访问请求。实施的所述缓冲器可以是FIFO或其它类型的缓冲器。



1. 一种实施FIFO缓冲器的方法,包括:
 - 提供存储可执行数据的存储器装置;
 - 提供处理装置,所述处理装置连接到所述存储器装置并且可用于检索和执行所述可执行数据;
 - 通过所述处理装置从应用程序的线程接收缓冲器初始化指令,
 - 响应于所述缓冲器初始化指令:
 - 通过所述处理装置将所述存储器装置的一部分分配为缓冲器;
 - 通过所述处理装置初始化硬件缓冲器控制器的状态以引用所述存储器装置分配为缓冲器的所述一部分;
 - 通过所述处理装置返回硬件缓冲器控制器的引用至应用程序的线程,存储器使用所述引用访问硬件缓冲器不包括对所述存储器装置的任何地址的任何引用;
 - 通过所述处理装置执行应用程序,执行所述应用程序包含执行从所述应用程序的线程访问所述硬件缓冲器控制器的请求,所述请求仅引用所述硬件缓冲器控制器并且不包含所述存储器装置的一部分内的任何地址;
 - 响应于所述请求,通过所述处理装置将所述请求输入到所述硬件缓冲器控制器;
 - 对于每个请求,由所述硬件缓冲器控制器通过以下操作处理:
 - 基于所述硬件缓冲器控制器的所述状态产生所述存储器装置的一部分内的地址;
 - 将包含所述地址的存储器访问指令输出到所述存储器装置;以及
 - 更新所述硬件缓冲器控制器的所述状态。
2. 根据权利要求1所述的方法,其中所述硬件缓冲器控制器实施先进先出(FIFO)缓冲器。
 3. 根据权利要求2所述的方法,其中所述硬件缓冲器控制器的所述状态包含读取指针和写入指针;
 - 其中更新所述硬件缓冲器控制器的所述状态包括:对于所述请求中的每个写入请求,使所述写入指针递增;
 - 其中更新所述硬件缓冲器控制器的所述状态包括:对于所述请求中的每个读取请求,使所述读取指针递增。
 4. 根据权利要求3所述的方法,
 - 其中使所述写入指针递增包括:
 - 确定在时钟循环期间接收到的多个写入请求的数目并使所述写入指针递增所述多个写入请求的所述数目;
 - 其中使所述读取指针递增包括:
 - 确定在所述时钟循环期间接收到的多个读取请求的数目并使所述读取指针递增所述多个读取请求的所述数目。
 5. 根据权利要求3所述的方法,
 - 其中基于所述硬件缓冲器控制器的所述状态产生所述存储器装置的一部分内的地址包括:
 - 对于所述请求中的每个写入请求,输出所述写入指针的当前值;
 - 其中基于所述硬件缓冲器控制器的所述状态产生所述存储器装置的一部分内的所述

地址包括：

对于所述请求中的每个读取请求,输出所述读取指针的当前值。

6.根据权利要求5所述的方法,其进一步包括:

通过处理装置计算所述写入请求的数量与所述读取请求的数量之间的差值;

通过所述处理装置确定所述差值小于第一阈值且非零;

响应于确定所述差值小于第一阈值且非零,通过所述处理装置输出指示所述缓冲器接近空的事件;

响应于指示所述缓冲器接近空的所述事件,通过所述处理装置阻止由所述处理装置执行的一个或多个执行线程产生读取请求。

7.根据权利要求5所述的方法,其进一步包括:

通过所述硬件缓冲器控制器计算所述写入请求的数量与所述读取请求的数量之间的差;

通过所述硬件缓冲器控制器确定所述差值大于第一阈值;

响应于确定所述差值大于第一阈值,通过所述硬件缓冲器控制器输出指示所述缓冲器接近满的事件;

响应于指示所述缓冲器接近满的所述事件,通过所述处理装置阻止由所述处理装置执行的一个或多个执行线程产生写入请求。

8.根据权利要求1所述的方法,其中通过硬件缓冲器控制器处理所述请求进一步包括:

集中所述请求中的多个写入请求;

将集中写入指令传输到所述存储器装置,所述集中写入指令包含对应于所述多个写入请求的所述缓冲器内的多个地址;

集中所述请求中的多个读取请求;以及

将集中读取指令传输到所述存储器装置,所述集中读取指令包含对应于所述多个读取请求的所述缓冲器内的多个地址。

9.根据权利要求8所述的方法,其中所述多个写入请求由多个执行线程以重叠方式发布,使得在所述请求中的其它请求输入到所述硬件缓冲器控制器之前接收所述写入请求中的至少一个写入请求。

10.根据权利要求9所述的方法,其进一步包括:

通过处理装置评估所述硬件缓冲器控制器的状态;

通过所述处理装置确定所述硬件缓冲器控制器的所述状态满足阈值条件;

响应于确定所述硬件缓冲器控制器的所述状态满足所述阈值条件,通过所述处理装置阻止所述多个执行线程发布引用所述硬件缓冲器控制器的额外请求。

11.一种实施FIFO缓冲器的系统,包括:

存储器装置,用于存储可执行数据;

处理装置,以及

多个硬件缓冲器控制器,

所述处理装置连接到所述存储器装置并且可用于检索和执行所述可执行数据;

所述处理装置被编程为从所述处理装置执行的应用程序的线程接收缓冲请求,

响应于所述缓冲请求,

将所述存储器装置的一部分分配至缓冲请求；

初始化多个硬件缓冲器控制器中的一个特定硬件缓冲器控制器的状态以引用所述存储器装置分配至缓冲请求的所述一部分；

返回所述特定硬件缓冲器控制器的引用至应用程序的线程，存储器使用所述引用访问硬件缓冲器不包括对所述存储器装置的任何地址的任何引用；

多个硬件缓冲器控制器中的每一个硬件缓冲器控制器连接到所述处理装置和所述存储器装置，每个所述硬件缓冲器控制器被编程为：

从所述处理装置执行的所述应用程序接收存储器访问请求，所述存储器访问请求不包括用于所述存储器装置中的任何存储位置的任何地址，仅指向引用每个硬件缓冲器控制器；

基于每个硬件缓冲器控制器的状态产生所述存储器装置的一部分内的地址；

将包含所述所产生地址的存储器访问指令输出到所述存储器装置；以及

更新所述硬件缓冲器控制器的所述状态。

12. 根据权利要求11所述的系统，其中所述硬件缓冲器控制器的所述状态实施先进先出(FIFO)缓冲器。

13. 根据权利要求12所述的系统，其中所述硬件缓冲器控制器进一步被编程为：

对于请求中的每个写入请求，通过使写入指针递增更新每个硬件缓冲器控制器的所述状态；

其中更新所述硬件缓冲器控制器的所述状态包括：对于所述请求中的每个读取请求，使读取指针递增。

14. 根据权利要求13所述的系统，其中每个硬件缓冲器控制器进一步被编程为通过确定在时钟循环期间接收到的多个写入请求的数目使所述写入指针递增并使所述写入指针递增所述多个写入请求的所述数目；以及

其中每个硬件缓冲器控制器进一步被编程为通过确定在所述时钟循环期间接收到的多个读取请求的数目使所述读取指针递增并使所述读取指针递增所述多个读取请求的所述数目。

15. 根据权利要求13所述的系统，其中每个硬件缓冲器控制器进一步被编程为通过针对所述请求中的每个写入请求输出所述写入指针的当前值，基于每个硬件缓冲器控制器的所述状态产生分配给缓冲器的存储器装置的区域内的所产生地址；

其中每个硬件缓冲器控制器进一步被编程为通过针对所述请求中的每个读取请求输出所述读取指针的当前值，基于每个硬件缓冲器控制器的所述状态产生分配给所述缓冲器的所述存储器装置的所述区域内的所述所产生地址。

16. 根据权利要求15所述的系统，其中每个硬件缓冲器控制器进一步被编程为：

计算所述写入请求的数量与所述读取请的数量求之间的差值；以及

如果所述差值小于第一阈值且非零，则输出指示所述缓冲器接近空的事件；以及

其中处理装置进一步被配置成响应于指示所述缓冲器接近空的所述事件，阻止由所述处理装置执行的一个或多个执行线程产生读取请求。

17. 根据权利要求16所述的系统，其中所述硬件缓冲器控制器进一步被编程为：

计算所述写入请求的数量与所述读取请求的数量之间的差值；以及

如果所述差值大于第二阈值，则通过所述处理装置输出指示所述缓冲器接近满的事件；以及

其中所述处理装置进一步被编程为响应于指示所述缓冲器接近满的所述事件，阻止由所述处理装置执行的一个或多个执行线程产生写入请求。

18. 根据权利要求11所述的系统，其中每个硬件缓冲器控制器进一步被编程为通过以下操作处理所述请求：

集中所述请求中的多个写入请求；

将集中写入指令传输到所述存储器装置，所述集中写入指令包含对应于所述多个写入请求的所述缓冲器内的多个地址；

集中所述请求中的多个读取请求；以及

将集中读取指令传输到所述存储器装置，所述集中读取指令包含对应于所述多个读取请求的所述缓冲器内的多个地址。

19. 根据权利要求11所述的系统，其中每个硬件缓冲器控制器被编程为以重叠方式从由所述处理装置执行的多个执行线程接收所述多个写入请求，使得在将所述请求中的其它请求输入到每个硬件缓冲器控制器之前接收所述写入请求中的至少一个写入请求。

20. 根据权利要求19所述的系统，其中每个硬件缓冲器控制器进一步被编程为：

评估每个硬件缓冲器控制器的所述状态；

如果每个硬件缓冲器控制器的所述状态满足阈值条件，则通过所述处理装置调用阻止所述多个执行线程发布引用所述硬件缓冲器控制器的额外请求。

用于多线程访问的软件定义FIFO缓冲器

技术领域

[0001] 本发明涉及用于实施FIFO缓冲器的系统和方法。

背景技术

[0002] 在常规的计算机系统中,先进先出(FIFO)缓冲器可以由多个线程访问。然而,在常规系统中,一次仅一个线程可以访问FIFO缓冲器。这会引入延迟和处理开销,以防止在等待FIFO缓冲器的数据结构由另一线程释放时由阻塞线程同时访问。

[0003] 本文所公开的系统和方法提供一种用于实施同时可由多个线程访问的FIFO缓冲器的改进方法。

附图说明

[0004] 为了使得将容易理解本发明的优点,将参考附图中所说明的具体实施例呈现对上文简述的本发明的更具体描述。在理解这些图式仅描绘本发明的典型实施例且因此不应认为其限制本发明的范围的情况下,将通过使用附图来以额外特异性和细节来描述并解释本发明,在附图中:

[0005] 图1是适合于实施根据本发明的实施例的方法的计算机系统的示意性框图;

[0006] 图2A到2B是根据本发明的实施例的用于实施硬件FIFO缓冲器的组件的示意性框图;

[0007] 图3A和3B是根据本发明的实施例的用于实施到硬件FIFO缓冲器的读取和写入的方法的过程流程图;以及

[0008] 图4是根据本发明的实施例用于使用硬件FIFO缓冲器防止缓冲器上溢和下溢的方法的过程流程图。

具体实施方式

[0009] 容易理解的是,如本文中大体描述且在图中说明的本发明的组件可以各种不同配置来布置和设计。因此,图中表示的本发明的实施例的以下更详细描述并不意图限制所要求的本发明的范围,而是仅表示根据本发明的当前设想的实施例的某些实例。参考图式将最好地理解当前所描述的实施例,其中通篇中用相同数字表示相同部分。

[0010] 根据本发明的实施例可以实施为设备、方法或计算机程序产品。因此,本发明可以采用完全硬件实施例、完全软件实施例(包含固件、驻留软件、微码等)或组合软硬件方面的实施例(在本文中可以全部总称为"模块"或"系统")的形式。另外,本发明可以采用实施于任何有形表达媒体中的计算机程序产品的形式,所述有形表达媒体具有实施于媒体中的计算机可用程序代码。

[0011] 可以利用一个或多个计算机可用媒体或计算机可读媒体的任何组合,包含非暂时性媒体。例如,计算机可读介质可以包含便携式计算机磁盘、硬盘、随机存取存储器(RAM)装置、只读存储器(ROM)装置、可擦除可编程只读存储器(EPROM或闪存存储器)装置、便携式压

缩光盘只读存储器 (CDROM)、光学存储装置和磁性存储装置中的一个或多个。在所选的实施例中,计算机可读介质可以包括任何非暂时性媒体,所述非暂时性介质可以容纳、存储、传送、传播或传递通过或结合指令执行系统、设备或装置使用的程序。

[0012] 可以通过一种或多种编程语言的任何组合编写用于执行本发明的操作的计算机程序代码,所述一种或多种编程语言包含例如Java、Smalltalk、C++等等面向对象的编程语言,以及例如“C”编程语言或类似编程语言的常规程序性编程语言。所述程序代码可以作为单独的软件包完全在计算机系统上执行,在单独硬件单元上执行,部分地在与计算机隔开某一距离的远程计算机上执行,或者完全在远程计算机或服务器上执行。在后一种场景中,远程计算机可以通过任何类型的网络,包含局域网 (LAN) 或广域网 (WAN) 连接到计算机,或者可以对外部计算机进行连接(例如,使用互联网服务提供商通过互联网)。

[0013] 下文参考根据本发明的实施例的方法、设备(系统)和计算机程序产品的流程图说明和/或框图描述本发明。应理解,可以通过计算机程序指令或代码实施流程图说明和/或框图中的每一块,以及流程图说明和/或框图中的块的组合。可以将这些计算机程序指令提供到通用计算机、专用计算机或其它可编程数据处理设备的处理器以产生机器,使得通过计算机或其它可编程数据处理设备的处理器执行的指令产生用于实施在流程图和/或框图块中指定的功能/动作的装置。

[0014] 这些计算机程序指令还可以存储在非暂时性计算机可读介质中,所述非暂时性计算机可读介质可以指导计算机或其它可编程数据处理设备用特定方式运行,使得存储在计算机可读介质中的指令产生包含实施在流程图和/或框图块中指定的功能/动作的指令装置的制品。

[0015] 计算机程序指令还可以加载到计算机或其它可编程数据处理设备上,以使在计算机或其它可编程设备上执行一系列操作步骤以产生计算机实施的过程,使得在计算机或其它可编程设备上执行的指令提供用于实施在流程图和/或框图块中指定的功能/动作的过程。

[0016] 图1是说明实例计算装置100的框图。计算装置100可以用于执行各种程序,例如本文中论述的那些程序。计算装置100可以充当服务器、客户端或任何其它计算实体。计算装置可以执行本文中论述的各种监测功能,并且可以执行一个或多个应用程序,例如本文所述的应用程序。计算装置100可以是多种多样的计算装置中的任一种,例如桌上型计算机、笔记本计算机、服务器计算机、手持型计算机、平板计算机等等。

[0017] 计算装置100包含一个或多个处理器102、一个或多个存储器装置104、一个或多个接口106、一个或多个大容量存储装置108、一个或多个输入/输出 (I/O) 装置110和显示装置130,所述装置全部连接到总线112。处理器102包含执行存储在存储器装置104和/或大容量存储装置108中的指令的一个或多个处理器或控制器。处理器102还可以包含各种类型的计算机可读媒体,例如,高速缓冲存储器。

[0018] 存储器装置104包含各种计算机可读媒体,例如,易失性存储器(例如,随机存取存储器 (RAM) 114) 和/或非易失性存储器(例如,只读存储器 (ROM) 116)。存储器装置104还可以包含可重写ROM,例如,闪存存储器。

[0019] 大容量存储装置108包含各种计算机可读媒体,例如,磁带、磁盘、光盘、固态存储器(例如,闪存存储器) 等等。如图1所示,特定的大容量存储装置是硬盘驱动器124。各种驱

动器还可以包含在大容量存储装置108中,以实现从不同计算机可读媒体的读取和/或到不同计算机可读媒体的写入。大容量存储装置108包含可装卸媒体126和/或非可装卸媒体。

[0020] I/O装置110包含允许数据和/或其它信息输入到计算装置100或从计算装置100检索的各种装置。实例I/O装置110包含光标控制装置、键盘、小键盘、麦克风、监视器或其它显示装置,例如,扬声器、打印机、网络接口卡、调制解调器、镜头、CCD或其它图像捕获装置等。

[0021] 显示装置130包含能够向计算装置100的一个或多个用户显示信息的任何类型的装置。显示装置130的实例包含监视器、显示终端、视频投影装置等。

[0022] 图形处理单元(GPU)132可以连接到处理器102和/或显示装置130。GPU可以用于再现计算机生成图像并且执行其它图形处理。GPU可以包含通用处理器,例如处理器102的功能性中的一些或全部。GPU还可以包含特定于图形处理的额外功能性。GPU可以包含与坐标变换、阴影化、纹理化、光栅化有关的硬编码和/或硬接线图形功能,以及有助于再现计算机生成图像的其它功能。

[0023] 接口106包含允许计算装置100与其它系统、装置或计算环境交互的各种接口。例如,接口106包含任何数目的不同网络接口120,例如局域网(LAN)、广域网(WAN)、无线网络和互联网的接口。其它接口包含用户接口118以及外围装置接口122。接口106还可以包含一个或多个用户接口元件118。接口106还可以包含一个或多个外围接口,例如,用于打印机、指向装置(鼠标、触控板等)、键盘等的接口。

[0024] 总线112允许处理器102、存储器装置104、接口106、大容量存储装置108和I/O装置110与彼此,以及连接到总线112的其它装置或组件通信。总线112表示若干类型的总线结构,例如,系统总线、PCI总线、IEEE 1394总线、USB总线等等中的一个或多个。

[0025] 出于说明的目的,程序和其它可执行程序组件在本文中示为离散块,但是应理解,此类程序和组件可以在不同时间驻留在计算装置100的不同存储组件中并且通过处理器102执行。或者,本文所描述的系统和程序可以在硬件,或硬件、软件和/或固件的组合中实施。例如,一个或多个专用集成电路(ASIC)可以被编程为执行本文所述的系统和程序中的一个或多个。

[0026] 参考图2A,处理装置102可以执行一个或多个线程200。如上所述,计算机系统100可以包含多个处理装置102。处理装置102中的任一个或处理装置102的组合可以执行下文归因于“处理装置”102的功能。线程可以通过由处理装置102执行的应用程序实例化。处理器可以通过常规方式分配处理资源并且将执行从一个线程200切换到另一线程。线程200可以向存储器子系统204发布存储器访问请求202(例如,读取和写入请求),所述存储器子系统204包含存储器装置104以及可能地插入于处理装置102与存储器装置104之间的存储器控制器。存储器访问请求202可以是本领域中已知的任何常规的存储器访问请求,例如,包含以下字段<操作码、源类型、源地址、目标类型、目标地址>中的一些或全部的指令,其中操作码是指示待执行的操作的代码,源类型和目标类型指示将访问哪个存储器装置104,源地址是将从其检索数据的存储器装置104内的位置,并且目标地址在数据将写入的存储器装置104内。同样,存储器请求202的处理方式可以是常规的。操作码可以指示读取操作,因此被执行为从源地址检索数据并返回到发布请求的处理器线程200。同样,操作码可以指示被执行为将数据写入到目标地址的写入操作。

[0027] 处理器线程200可以向硬件缓冲器控制器208发出存储器访问请求206。硬件缓冲

器控制器208是硬件装置,所述硬件装置包含被配置成执行本文中归因于硬件缓冲器控制器208的功能的电路。硬件缓冲器控制器208可以与处理装置102位于不同裸片或相同裸片上,或与存储器子系统204的一个或多个组件位于相同裸片上。与常规的存储器访问请求202相比,存储器访问请求206可以具有相同或不同格式。然而,存储器访问请求206不直接引用存储器装置104中的任何地址或地址范围。替代地,存储器访问请求202引用特定硬件缓冲器控制器208。硬件缓冲器控制器208可以被初始化以限定一个或多个缓冲器,每个缓冲器定义包含指派给缓冲器的存储器装置104中的一个内的地址分配。可以采用开始和结束地址或开始地址和缓冲器大小的形式为每个缓冲器记录分配。每个硬件缓冲器控制器208的初始化可以包含初始化读取指针210以指向初始读取位置,以及初始化写入指针212以指向初始写入位置。多个缓冲器可以在硬件缓冲器控制器208内初始化,或多个硬件缓冲器控制器208可以通过类似方式初始化。

[0028] 用于缓冲器的硬件缓冲器控制器208的初始化可以通过处理装置102调用,所述处理装置同样可以根据本领域中已知的任何方法为存储器装置104内的缓冲器分配存储器。缓冲器的初始化可以通过实例化一个或多个处理线程200的应用程序调用,并且可以将对初始化缓冲器的引用返回到应用程序,以供线程使用以读取以及写入到缓冲器。

[0029] 在所说明的实施例中,硬件缓冲器控制器208实施FIFO缓冲器。然而,还可以实施其它类型的缓冲器(例如,后进先出)。

[0030] 硬件缓冲器控制器208可以包含求和器214。求和器214接收存储器访问请求206并且确定对每个缓冲器的多个请求206。例如,可以在处理器102、总线112或存储器子系统204的一个时钟循环中接收多个请求206。具体来说,可以接收引用相同缓冲器的多个请求。在所说明的实施例中,求和器214将在每个时钟循环中针对每个缓冲器接收到的读取请求206的数目以及在每个时钟循环中针对每个缓冲器接收到的写入请求206的数目求和。每个请求206可以包含缓冲器标识符,使得求和器214可以识别具有相同缓冲器标识符的所有读取请求206以及具有相同缓冲器标识符206的所有写入请求。

[0031] 用于每个缓冲器的读取指针210随后通过硬件缓冲器控制器208递增在引用每个缓冲器的时钟循环中接收到的读取请求206的数目。同样,用于每个缓冲器的写入指针212通过硬件缓冲器控制器208递增在引用每个缓冲器的时钟循环中接收到的写入请求206的数目。

[0032] FIFO地址产生器216检测求和器214的输出或读取指针210的变化中的一个或两个,或另外被通知在时钟循环期间针对每个缓冲器接收到的任何读取请求206和写入请求。FIFO地址产生器216随后将存储器访问请求218输出到存储器子系统204。具体来说,在时钟循环期间接收到N个读取请求的情况下,FIFO地址产生器216输出N个读取请求218或包含一系列N个地址的读取请求218。同样,在时钟循环期间接收到N个读取请求的情况下,FIFO地址产生器216产生N个写入请求218或包含待写入的N个值和一系列N个地址的写入请求218。

[0033] FIFO地址产生器216在每个请求218中包含基于读取指针210的状态的地址。例如,对于单个读取或写入请求,FIFO发生器216可以取决于C指向缓冲器中的下一可用地址还是缓冲器中的先前访问地址来输出包含读取指针210或写入指针212的当前值(C)或C+1的读取或写入请求。对于N个读取或写入请求,FIFO发生器216输出包含地址C,C+1,C+2,……,C+N-1的N个读取或写入请求,或引用地址C到C+N-1的单个读取或写入请求。在一些实施例中,

在确定请求218的地址之前,指针210、212的值递增。因此,对于单个请求218,请求218的地址将是C+1、C+2、……、C+N或范围C+1到C+N。

[0034] 如先前所提及,硬件缓冲器控制器208可以包含多个缓冲器。因此,MUX 220可以选择性地将指针210、212连接到FIFO地址产生器216,所述FIFO地址产生器216随后又通过先前段落中描述的方式产生对每个缓冲器的请求218。同样,由于存储器访问请求202还被输入到存储器子系统204,因此可以将请求202、218输入到MUX 222,所述MUX根据轮询选择方案或其它负载平衡方法选择性地将请求202、218输入到存储器子系统204中。

[0035] 参考图2B,硬件缓冲器控制器208可以进一步基于其状态产生事件224。事件224可以被输入到由处理装置102执行的线程调度器226。线程调度器226确定哪些处理器线程200在处理装置102上执行,并且还阻塞线程、分支到其它线程以及执行本领域中已知的其它线程管理功能。每个事件224可以指示产生事件的缓冲器。

[0036] 在一些实施例中,线程调度器226接收事件224并且在其上执行线程管理任务。例如,事件224可以指示缓冲器已满或几乎满。因此,可以阻止线程200写入到缓冲器。例如,应用程序可以向线程调度器注册一个或多个线程200以响应于几乎满事件而进行阻塞。同样,调度器226可以被编程为响应于指示缓冲器为空或几乎为空的事件224而阻止某些线程从缓冲器读取。

[0037] 在所说明的实施例中,基于读取计数器226和写入计数器228产生事件。对于每个缓冲器,通过减法器230从写入计数器的值中减去读取计数器226的值。随后将计数器226、228值之间的差,以及一个或多个阈值234,输入到比较器232。

[0038] 零差指示缓冲器为空——读取刚好与写入一样多。在一些实施例中,在差为零的情况下,可以产生识别缓冲器的“缓冲器空”事件224。阈值234可以包含低阈值,使得如果参考值小于低阈值,则可以产生引用缓冲器的“接近空”事件224。

[0039] 等于缓冲器的大小的差指示缓冲器满,因为根据缓冲器中的条目总数,写入操作大于读取操作。在一些实施例中,在差等于缓冲器的大小的情况下,可以产生引用缓冲器的“缓冲器满”事件。阈值234可以包含小于缓冲器的大小的高阈值,例如,等于缓冲器的大小的80%与90%之间。如果差大于高阈值,则可以产生引用缓冲器的“接近满”事件224。

[0040] 在一些实施例中,为了避免计数器226、228的上溢,可以例如,周期性地或在计数器226、228中的一个接近可由计数器226、228表示的最大值之后,从计数器226、228中减去计数器226、228中的较小者的值。

[0041] 在一些实施例中,当空、接近空、满或接近满条件停止存在时,比较器232可以产生事件。例如,如果先前发现缓冲器满或接近满,并且随后发现未满或未接近满,则可以通过比较器232产生“缓冲器就绪”或“缓冲器满清除”事件。同样,如果先前发现缓冲器空或接近空,并且随后发现未空或未接近空,则可以通过比较器232产生“缓冲器就绪”或“缓冲器空清除”事件。

[0042] 如上所述,硬件缓冲器控制器208可以实施多个缓冲器。因此,在此类实施例中,多路复用器236、238可以选择性地将每个缓冲器的计数器226、228连接到减法器230和比较器232,以便评估其间的差并且根据上文所描述的方法视需要产生每个缓冲器的事件。

[0043] 在接收事件224之后,线程调度器226随后可以调用相对于处理器线程200的一个或多个动作。例如,在接收“缓冲器空”或“接近空”事件之后,应用程序可以通知调度器226

阻塞线程200。因此,在接收此事件之后,将阻止执行所述线程200。在清除“缓冲器空”或“缓冲器空”事件之后,例如,在接收“缓冲器就绪”或“缓冲器空清除”事件之后,停止通过线程调度器224阻塞线程。

[0044] 类似地,在接收“缓冲器满”或“接近满”事件之后,应用程序可以通知调度器226阻塞线程200。因此,在接收此事件之后,随后将阻止执行所述线程200。在清除“缓冲器满”或“接近满”事件之后,例如,在接收“缓冲器就绪”或“缓冲器空清除”事件之后,停止通过线程调度器224阻塞线程。

[0045] 因此,应用程序可以具有被编程为执行对缓冲器的写入请求的一些线程200,并且可以在接收引用缓冲器的“缓冲器满”或“接近满”事件之后指导线程调度器226阻塞此类线程200。应用程序可以具有被编程为执行对缓冲器的读取请求的一个或多个不同线程200,并且在接收所述缓冲器的“缓冲器空”或“接近空”事件之后,指导线程调度器226阻塞此类线程200。

[0046] 线程200可以对多于一个缓冲器发出请求,并且应用程序因此可以通知线程调度器226基于引用由硬件缓冲器控制器208实施的一个或多个缓冲器的事件来阻塞线程。

[0047] 参考图3A,响应于接收引用缓冲器的写入请求,所说明的方法300可以通过硬件缓冲器控制器208执行,所述写入请求包含缓冲器的标识符,但不包含实际上存储缓冲器的条目的存储器装置104中的地址。

[0048] 方法300包含接收302对缓冲器的一个或多个写入请求,以及求和304写入请求以确定在某一时间段期间缓冲器接收到的写入请求的数目。例如,在步骤304处可以计数在处理装置102、总线112、存储器装置104或硬件缓冲器208的一个时钟循环期间接收到的写入请求的数目。如上所述,多个缓冲器可以通过硬件缓冲器控制器208实施,使得针对在时钟循环期间接收写入请求的每个缓冲器执行方法300的步骤。出于图3A和3B的目的,“缓冲器”是指执行所说明方法的缓冲器。

[0049] 方法300可以包含更新306写入指针212和写入计数器228中的一个或两个。在一些实施例中,仅这些值中的一个用于表示缓冲器的状态。例如,硬件缓冲器控制器208可以仅存储写入指针212和读取指针210,以确定缓冲器而不是计数器的状态。在其它实施例中,仅使用计数器226、228,并且从计数器以及缓冲器的开始地址确定地址。具体来说,更新指针212和计数器228可以包含使其中的每一个递增在时钟循环期间接收到的写入请求的数目。

[0050] 方法300进一步包含基于写入指针212的状态以及接收到302的写入请求的数目产生308地址。如上所述,在接收302写入请求以及更新306写入指针212之前指针的当前值是C,并且接收到N个写入请求的情况下,产生的地址可以是C、C+1、……C+N-1,其中C指向将写入的下一地址。在C指向最近写入的地址的情况下,产生的地址是C+1、C+2、……C+N。地址可以以范围的形式产生,例如数据对<C, N-1>或<C+1, N>,其中一个值是开始地址并且另一值是从开始地址的偏移。

[0051] 方法300可以进一步包含将包含在步骤308处产生的地址的一个或多个写入命令输出310到存储器子系统204,写入命令有效地根据在存储器装置104中的一个存储器装置中的一个或多个写入命令中的地址处的写入请求调用数据的写入。如上所述,可以针对接收到302的每个写入请求输出310N个写入命令,并且每个写入命令包含在步骤308处产生的地址中的一个。或者,可以输出310单个写入命令,所述写入命令指定在步骤308处产生的所

有N个地址或跨越所有N个地址的一系列地址(例如,开始地址和偏移)。写入命令随后由所述写入命令被寻址到的存储器装置104执行,方法结束。

[0052] 参考图3B,响应于接收引用缓冲器的读取请求,所说明的方法312可以通过硬件缓冲器控制器208执行,所述读取请求包含缓冲器的标识符,但不包含实际上存储缓冲器的条目的存储器装置104中的地址。

[0053] 方法312包含接收314对缓冲器的一个或多个读取请求,以及求和316读取请求以确定在某一时间段期间针对缓冲器接收到的读取请求的数目。例如,可以在步骤316处对处理装置102、总线112、存储器装置104或硬件缓冲器208的一个时钟循环期间接收到的读取请求的数目进行计数。如上所述,多个缓冲器可以通过硬件缓冲器控制器208实施,使得针对在时钟循环期间接收写入请求的每个缓冲器执行方法312的步骤。

[0054] 方法312可以包含更新318读取指针210和读取计数器226中的一个或两个。在一些实施例中,仅这些值中的一个用于表示缓冲器的状态。例如,硬件缓冲器控制器208可以仅存储写入指针212和读取指针210,以确定缓冲器而不是计数器的状态。在其它实施例中,仅使用计数器226、228,并且从计数器以及缓冲器的开始地址确定地址。更新318指针210和计数器226可以包含使其中的每一个递增在时钟循环期间接收到的写入请求的数目。

[0055] 方法312进一步包含基于读取指针210的状态以及接收到314的读取请求的数目产生320地址。如上所述,在接收314读取请求以及更新318读取指针210之前指针的当前值是C,并且接收到N个读取请求的情况下,产生的地址可以是C、C+1、……C+N-1,其中C指向将写入的下一地址。在C指向最近读取的地址的情况下,产生的地址是C+1、C+2、……C+N。地址可以以范围的形式产生,例如数据对<C, N-1>或<C+1, N>,其中一个值是开始地址并且另一值是从开始地址的偏移。

[0056] 方法312可以进一步包括将包含在步骤308处产生的地址的一个或多个读取命令输出322到存储器子系统204,读取命令有效地在从存储器装置104中的一个产生的一个或多个写入命令中的地址处调用数据的检索,并且将此数据返回到处理装置102。如上所述,可以针对接收到302的每个读取请求输出322N个读取命令,并且每个读取命令包含在步骤308处产生的地址中的一个。或者,可以输出322单个读取命令,所述读取命令指定在步骤308处产生的所有N个地址或跨越所有N个地址的一系列地址(例如,开始地址和偏移)。读取命令随后由所述读取命令被寻址到的存储器装置104执行,方法结束。

[0057] 方法300和312可以基本上同时执行。例如,分离的读取和写入管线可以通过硬件缓冲器控制器208实施,使得可以针对读取请求和写入请求两者同时产生存储器访问命令。从步骤310和322产生的存储器访问命令可以通过存储器子系统204根据本领域中已知的任何命令执行方案按顺序缓冲和执行。

[0058] 参考图4,所说明的方法400可以通过硬件缓冲器控制器208执行,以便检测缓冲器的满、接近满、空和接近空情况。方法400可以针对由硬件缓冲器控制器208控制的每个缓冲器,下文称为“缓冲器”执行。

[0059] 方法400可以包含或前面有以下步骤:在时钟循环期间接收302写入指令,以及如上文相对于图3A所描述,求和304在时钟循环期间接收到的写入指令。求和步骤304可以针对方法300和方法400两者执行一次或可以针对每个方法单独地执行。

[0060] 方法400还可以包含或前面有以下步骤:在时钟循环期间接收314写入指令,以及

如上文相对于图3B所描述,求和316在时钟循环期间接收到的写入指令。求和步骤316可以针对方法312和方法400两者执行一次或可以针对每个方法单独地执行。

[0061] 方法400可以进一步包含更新402写入计数器238以及更新404读取计数器236。如上所述,这可以包含将在时钟循环期间接收到的写入请求的数目添加到写入计数器238,以及将在时钟循环期间接收到的读取请求的数目添加到读取计数器236。

[0062] 方法400可以进一步包含计算406写入计数器与读取计数器之间的差。在不存在读取超过写入的下溢条件的情况下,写入计数器与读取计数器之间的差将为正或零。在一些实施例中,为了避免超过由实施步骤406的差函数的电路以及存储计数器236、238的寄存器可处理的最大数,对于方法400的每次迭代,写入计数器可以设定成等于差的值并且读取指针可以设定成等于零。

[0063] 方法400可以包含相对于一个或多个阈值评估在步骤406处计算出的差。例如,如果得出408差大于接近满阈值,则可以产生410接近满事件。如果得出412差小于接近空阈值,则可以产生414接近空事件。如果得出416差大于缓冲器的大小,则可以产生418缓冲器满事件。如果得出420差为零,则可以产生缓冲器空事件。步骤408到420的阈值可以使用并行管线按任何次序或同时评估。

[0064] 接近满和接近空阈值可以是在执行方法400之前硬接线或编程的预定值。在一些实施例中,接近满和接近空阈值可以是执行线程的数目以及当发现存在接近满或接近空条件时与可以响应于事件阻塞线程之间的延迟的函数。例如,在被编程为发出写入请求的M个线程正执行并且响应于事件至少需要Q个时钟循环来阻塞线程的情况下,接近满阈值可以设定成等于 $S-Q*M-R$,其中S是缓冲器的大小并且R是大于零的值,以便保证不出现上溢。以此方式,如果检测到接近满条件,则在接近满事件导致可以发布写入请求的M个线程阻塞之前不出现缓冲器上溢。

[0065] 通过类似方式,在被编程为发布读取请求的M个线程正执行的情况下,接近空阈值可以设定成等于 $Q*M+R$,其中S是缓冲器的大小并且R是大于零的值,以便保证不出现下溢。以此方式,如果检测到接近空条件,则在接近空事件导致可以发布读取请求的M个线程阻塞之前不出现缓冲器下溢。

[0066] 在不脱离本发明的精神或基本特征的情况下,可以其它特定形式实施本发明。所描述的实施例应视为在所有方面均仅为说明性而非限制性的。因此,本发明的范围由所附权利要求书而不是由前述描述指示。在权利要求书等效物的含义和范围内的所有变化均涵盖在权利要求书的范围内。

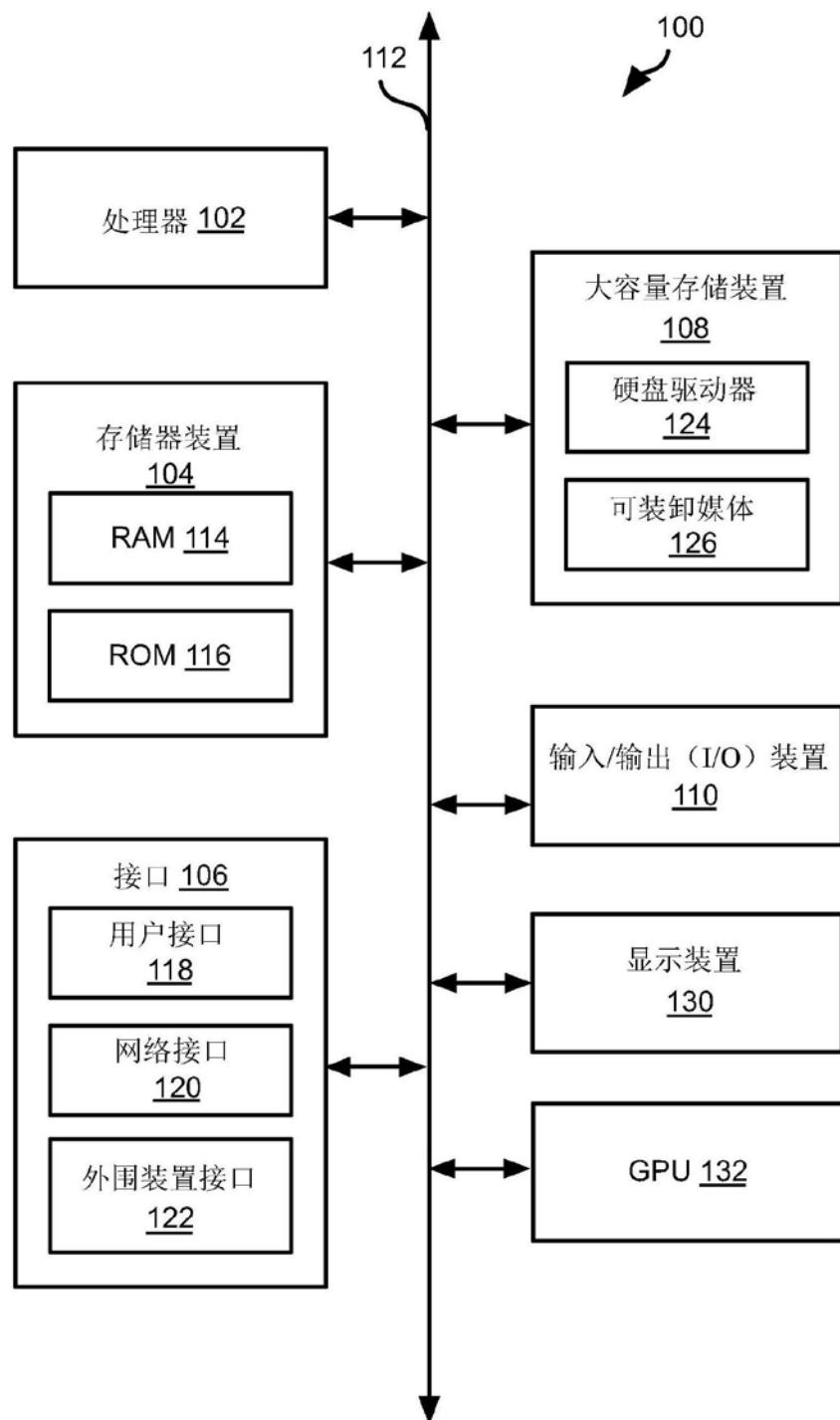


图1

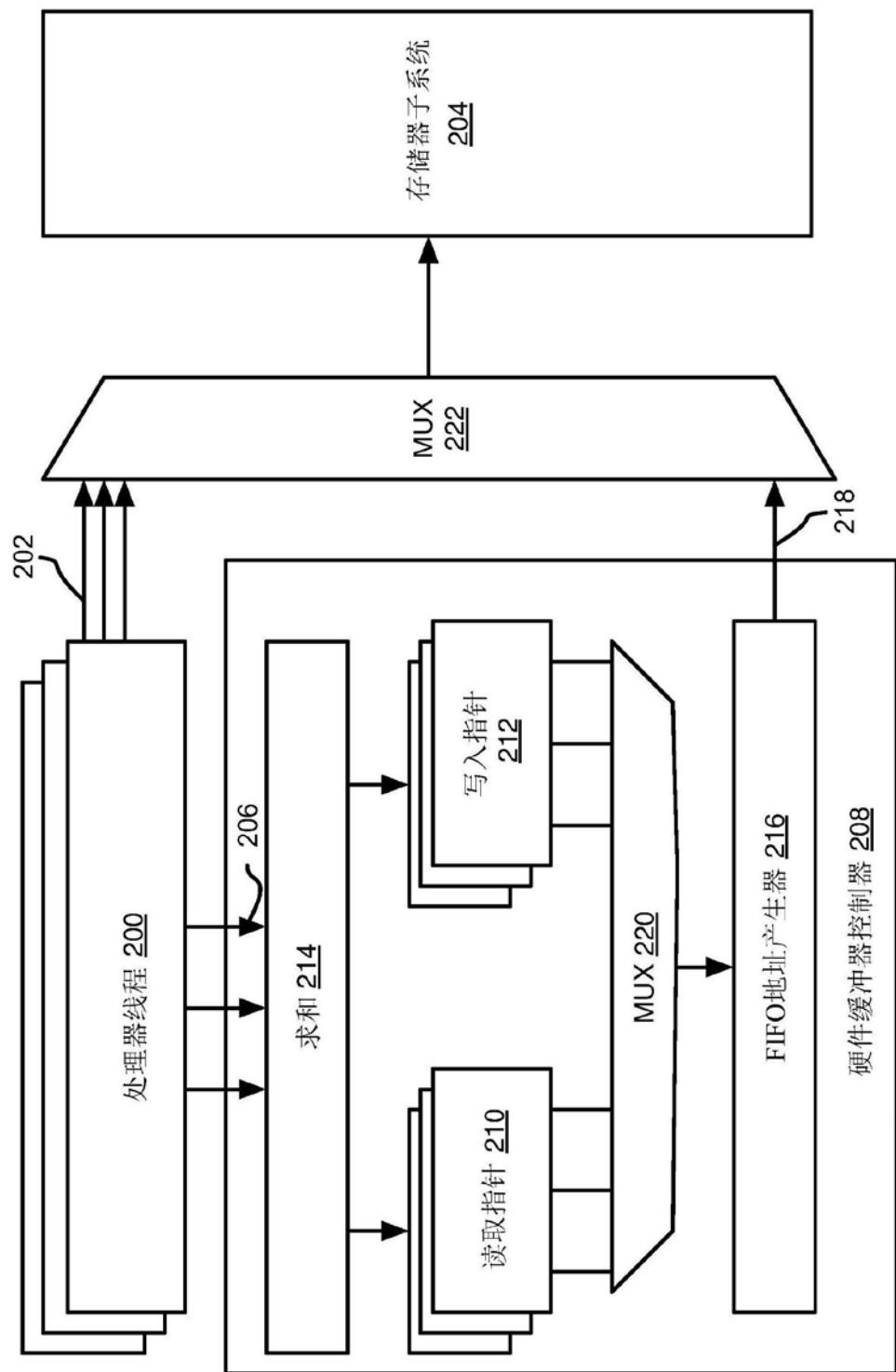


图 2A

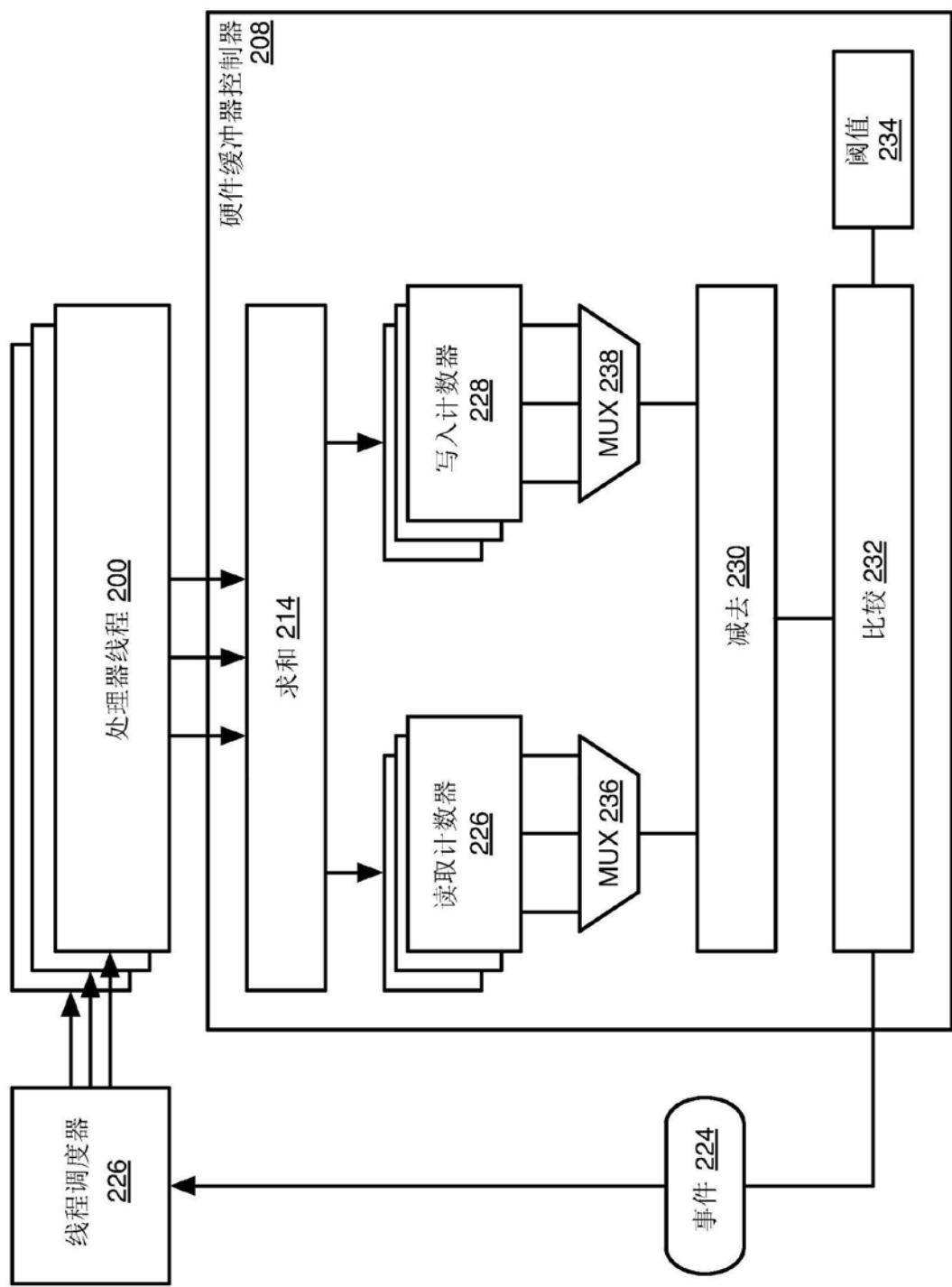


图2B

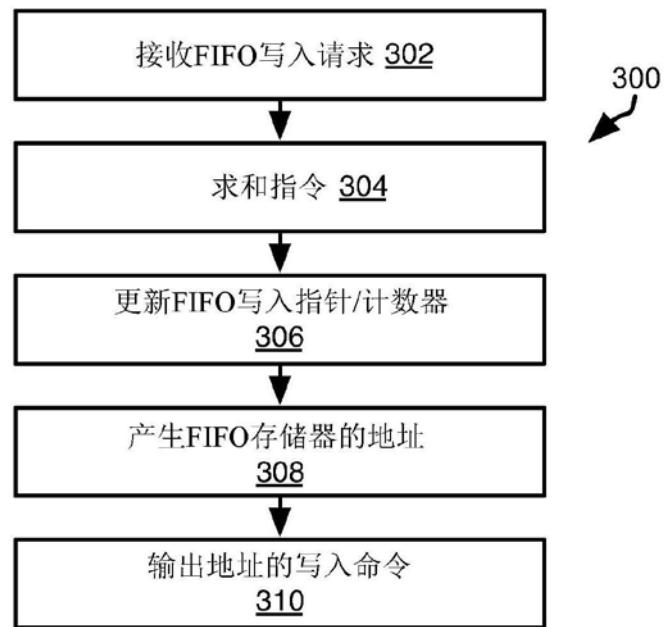


图3A

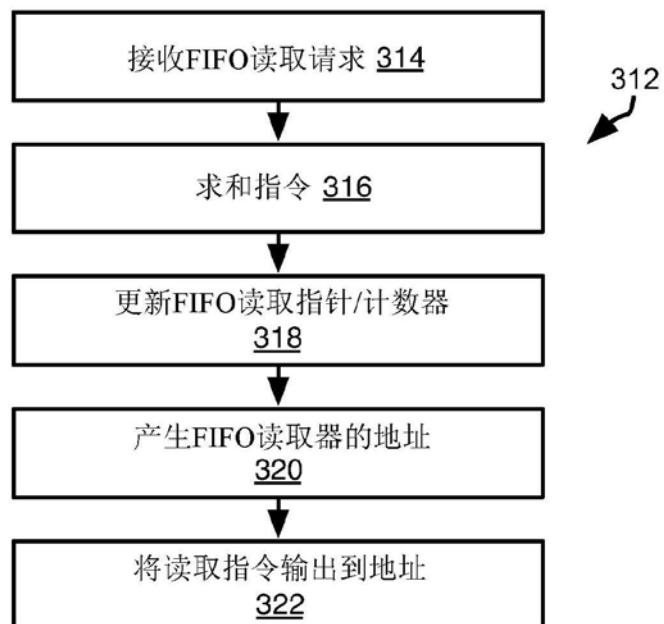


图3B

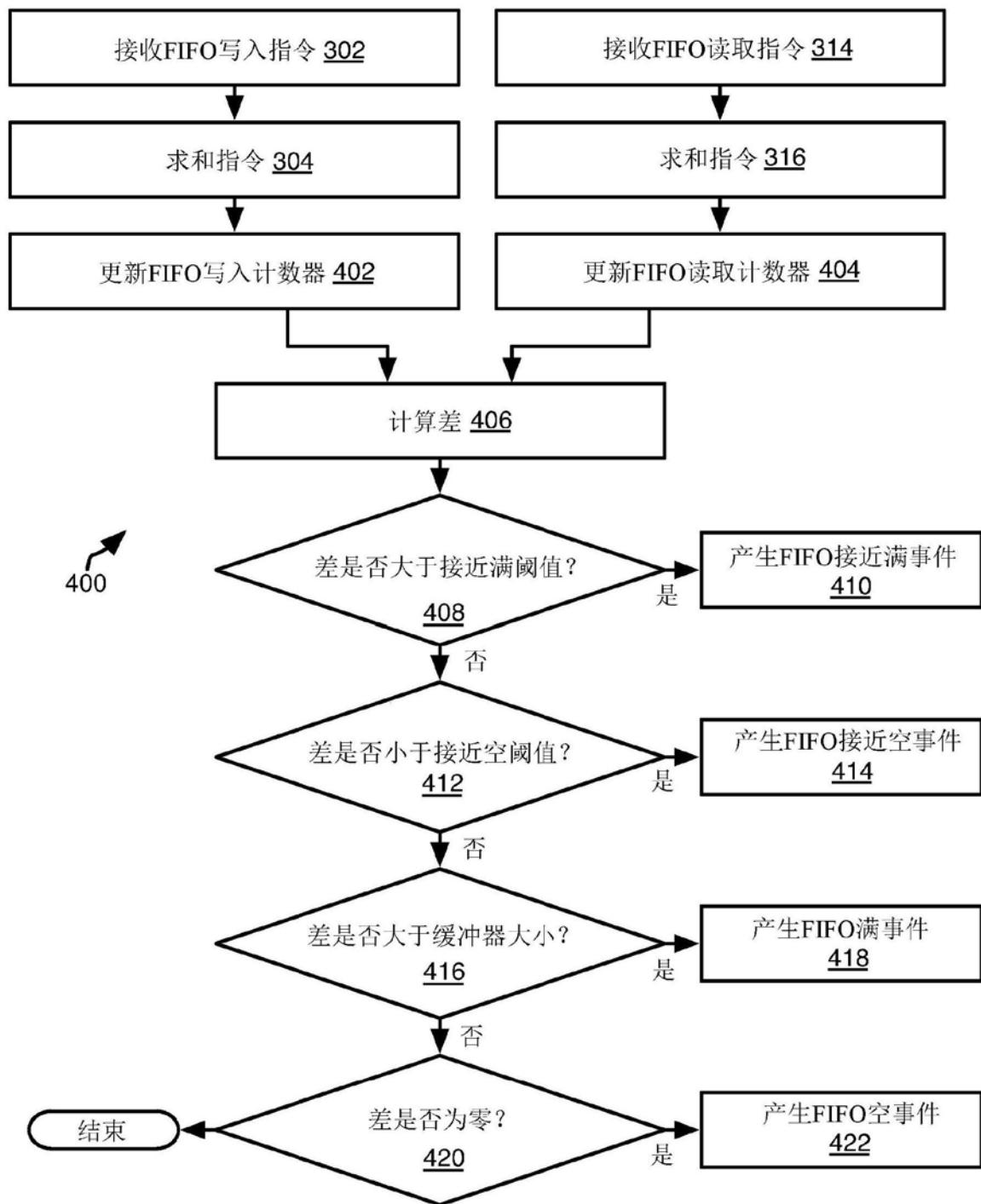


图4