



(12) 发明专利申请

(10) 申请公布号 CN 116028118 A

(43) 申请公布日 2023.04.28

(21) 申请号 202310077710.3

(22) 申请日 2023.01.31

(71) 申请人 南京砺算科技有限公司

地址 210031 江苏省南京市(江苏)自由贸易
易试验区南京片区团结路99号孵鹰大
厦2794室

申请人 砺算科技(上海)有限公司

(72) 发明人 阙恒 朱康挺 王斐 周义满

(74) 专利代理机构 北京集佳知识产权代理有限
公司 11227

专利代理师 李笑笑

(51) Int. Cl.

G06F 9/30 (2018.01)

G06T 1/20 (2006.01)

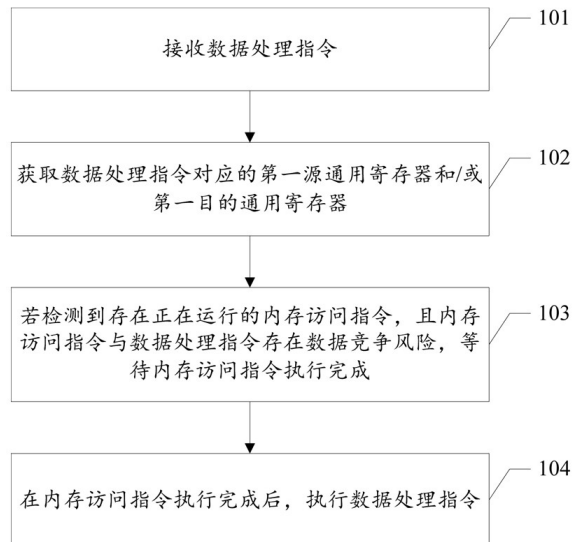
权利要求书2页 说明书11页 附图1页

(54) 发明名称

保障数据一致性的指令执行方法及图形处
理器、介质

(57) 摘要

一种保障数据一致性的指令执行方法及图
形处理器、介质,所述方法包括:接收数据处
理指令;获取所述数据处理指令对应的第一源通用寄
存器和/或第一目的通用寄存器;若检测到存在
正在运行的内存访问指令,且所述内存访问指令
与所述数据处理指令存在数据竞争风险,等待所
述内存访问指令执行完成;所述内存访问指令与
所述数据处理指令存在数据竞争风险包括如下
任一情形:所述内存访问指令对应的第二目的通
用寄存器与所述第一源通用寄存器存在交集;所
述内存访问指令对应的第二源通用寄存器与所
述第一目的通用寄存器存在交集;在所述内存访
问指令执行完成后,执行所述数据处理指令。采
用上述方案,能够有效解决数据竞争风险,避免
指令执行出现错误。



1. 一种保障数据一致性的指令执行方法,其特征在于,包括:

接收数据处理指令;

获取所述数据处理指令对应的第一源通用寄存器和/或第一目的通用寄存器;

若检测到存在正在运行的内存访问指令,且所述内存访问指令与所述数据处理指令存在数据竞争风险,等待所述内存访问指令执行完成;所述内存访问指令与所述数据处理指令存在数据竞争风险包括如下任一情形:所述内存访问指令对应的第二目的通用寄存器与所述第一源通用寄存器存在交集;所述内存访问指令对应的第二源通用寄存器与所述第一目的通用寄存器存在交集;

在所述内存访问指令执行完成后,执行所述数据处理指令。

2. 如权利要求1所述的保障数据一致性的指令执行方法,其特征在于,检测所述内存访问指令与所述数据处理指令存在数据竞争风险,包括:

从第一数据竞争状态表中,获取所述第二源通用寄存器对应比特域的取值,以及所述第二目的通用寄存器对应比特域的取值;

若所述第二源通用寄存器对应比特域的取值为第一值,且所述第二源通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二源通用寄存器对应比特域的取值为第二值,或所述第二源通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;

若所述第二目的通用寄存器对应比特域的取值为第一值,且所述第二目的通用寄存器与所述第一源通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二目的通用寄存器对应比特域的取值为第二值,或所述第二目的通用寄存器与所述第一源通用寄存器不存在交集,则确定未检测到所述数据竞争风险;

其中,所述第一数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

3. 如权利要求2所述的保障数据一致性的指令执行方法,其特征在于,所述在所述内存访问指令执行完成后,执行所述数据处理指令,包括:

在所述内存访问指令执行完成后,对所述第一数据竞争状态表进行更新,将所述第二目的通用寄存器对应比特域的取值置为第二值,以及,将所述第二源通用寄存器对应比特域的取值置为第二值。

4. 如权利要求1所述的保障数据一致性的指令执行方法,其特征在于,所述数据处理指令对应写操作,检测所述内存访问指令与所述数据处理指令存在数据竞争风险,包括:

从第二数据竞争状态表中,获取所述第二源通用寄存器对应比特域的取值;

若所述第二源通用寄存器对应比特域的取值为第一值,且所述第二源通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二源通用寄存器对应比特域的取值为第二值,或所述第二源通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;

所述第二数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

5. 如权利要求4所述的保障数据一致性的指令执行方法,其特征在于,所述在所述内存访问指令执行完成后,执行所述数据处理指令,包括:

在所述内存访问指令执行完成后,对所述第二数据竞争状态表进行更新,将所述第二源通用寄存器对应比特域的取值置为第二值。

6. 如权利要求1所述的保障数据一致性的指令执行方法,其特征在于,所述数据处理指令对应读操作,检测所述内存访问指令与所述数据处理指令存在数据竞争风险,包括:

从第三数据竞争状态表中,获取所述第二目的通用寄存器对应比特域的取值;

若所述第二目的通用寄存器对应比特域的取值为第一值,且所述第二目的通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二目的通用寄存器对应比特域的取值为第二值,或所述第二目的通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;

所述第三数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

7. 如权利要求6所述的保障数据一致性的指令执行方法,其特征在于,所述在所述内存访问指令执行完成后,执行所述数据处理指令,包括:

在所述内存访问指令执行完成后,对所述第三数据竞争状态表进行更新,将所述第二目的通用寄存器对应比特域的取值置为第二值。

8. 如权利要求1所述的保障数据一致性的指令执行方法,其特征在于,还包括:若检测到存在正在运行的内存访问指令,且所述内存访问指令与所述数据处理指令不存在所述数据竞争风险时,执行所述数据处理指令。

9. 一种图形处理器,其特征在于,包括:调度执行核心、内存访问控制器以及通用寄存器,其中:

所述内存访问控制器,用于接收内存访问指令;

所述调度执行核心,用于将所述内存访问指令转发至所述内存访问控制器;以及,接收数据处理指令,获取所述数据处理指令对应的第一源通用寄存器和/或第一目的通用寄存器;若检测到存在正在运行的内存访问指令,且所述内存访问指令与所述数据处理指令存在数据竞争风险,等待所述内存访问指令执行完成;所述内存访问指令与所述数据处理指令存在数据竞争风险包括如下任一情形:所述内存访问指令对应的第二目的通用寄存器与所述第一源通用寄存器存在交集;所述内存访问指令对应的第二源通用寄存器与所述第一目的通用寄存器存在交集;在所述内存访问指令执行完成后,执行所述数据处理指令。

10. 一种计算机可读存储介质,所述计算机可读存储介质为非易失性存储介质或非瞬态存储介质,其上存储有计算机程序,所述计算机程序被处理器运行时执行权利要求1~8任一项所述的指令执行方法的步骤。

保障数据一致性的指令执行方法及图形处理器、介质

技术领域

[0001] 本发明涉及图形处理器技术领域,尤其涉及一种保障数据一致性的指令执行方法及图形处理器、介质。

背景技术

[0002] 图形处理器(Graphics Processing Unit,GPU)在进行图形渲染等处理时,需要调用内存(memory)数据,将memory数据载入到GPU的调度执行核心。后续的执行指令可以针对载入的memory数据进行特定的运算。

[0003] GPU的硬件执行memory访问指令的延迟(latency)是不固定的,而调度执行核心内的算术逻辑单元(Arithmetic&Logic Unit,ALU)指令的延迟固定。若某一执行尚未执行完成,但后续指令确过早执行,则会导致后续指令未能够使用正确的memory数据,导致执行错误。

[0004] 例如,memory访问指令对应的数据写入与ALU指令对应的数据读出之间存在依赖关系,存在写后读(Read After Write data hazard,RAW)数据竞争风险。又如,memory访问指令上存放坐标或地址的通用寄存器(General Purpose Register file,GPR)在被后续指令用做写入数据的目的通用寄存器时,存在读后写(Write After Read data hazard,WAR)数据竞争风险。上述的RAW数据竞争风险与WAR数据竞争风险可以统称为数据竞争风险。

发明内容

[0005] 本发明实施例解决的是无法解决数据竞争风险,导致指令执行错误的技术问题。

[0006] 为解决上述技术问题,本发明实施例提供一种保障数据一致性的指令执行方法,包括:接收数据处理指令;获取所述数据处理指令对应的第一源通用寄存器和/或第一目的通用寄存器;若检测到存在正在运行的内存访问指令,且所述内存访问指令与所述数据处理指令存在数据竞争风险,等待所述内存访问指令执行完成;所述内存访问指令与所述数据处理指令存在数据竞争风险包括如下任一情形:所述内存访问指令对应的第二目的通用寄存器与所述第一源通用寄存器存在交集;所述内存访问指令对应的第二源通用寄存器与所述第一目的通用寄存器存在交集;在所述内存访问指令执行完成后,执行所述数据处理指令。

[0007] 可选的,检测所述内存访问指令与所述数据处理指令存在数据竞争风险,包括:从第一数据竞争状态表中,获取所述第二源通用寄存器对应比特域的取值,以及所述第二目的通用寄存器对应比特域的取值;若所述第二源通用寄存器对应比特域的取值为第一值,且所述第二源通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二源通用寄存器对应比特域的取值为第二值,或所述第二源通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;若所述第二目的通用寄存器对应比特域的取值为第一值,且所述第二目的通用寄存器与所述第一源通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二目的通用寄存

器对应比特域的取值为第二值,或所述第二目的通用寄存器与所述第一源通用寄存器不存在交集,则确定未检测到所述数据竞争风险;其中,所述第一数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

[0008] 可选的,所述在所述内存访问指令执行完成后,执行所述数据处理指令,包括:在所述内存访问指令执行完成后,对所述第一数据竞争状态表进行更新,将所述第二目的通用寄存器对应比特域的取值置为第二值,以及,将所述第二源通用寄存器对应比特域的取值置为第二值。

[0009] 可选的,所述数据处理指令对应写操作,检测所述内存访问指令与所述数据处理指令存在数据竞争风险,包括:从第二数据竞争状态表中,获取所述第二源通用寄存器对应比特域的取值;若所述第二源通用寄存器对应比特域的取值为第一值,且所述第二源通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二源通用寄存器对应比特域的取值为第二值,或所述第二源通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;所述第二数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

[0010] 可选的,所述在所述内存访问指令执行完成后,执行所述数据处理指令,包括:在所述内存访问指令执行完成后,对所述第二数据竞争状态表进行更新,将所述第二源通用寄存器对应比特域的取值置为第二值。

[0011] 可选的,所述数据处理指令对应读操作,检测所述内存访问指令与所述数据处理指令存在数据竞争风险,包括:从第三数据竞争状态表中,获取所述第二目的通用寄存器对应比特域的取值;若所述第二目的通用寄存器对应比特域的取值为第一值,且所述第二目的通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二目的通用寄存器对应比特域的取值为第二值,或所述第二目的通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;所述第三数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

[0012] 可选的,所述在所述内存访问指令执行完成后,执行所述数据处理指令,包括:在所述内存访问指令执行完成后,对所述第三数据竞争状态表进行更新,将所述第二目的通用寄存器对应比特域的取值置为第二值。

[0013] 可选的,所述指令执行方法还包括:若检测到存在正在运行的内存访问指令,且所述内存访问指令与所述数据处理指令不存在所述数据竞争风险时,执行所述数据处理指令。

[0014] 本发明实施例还提供了一种图形处理器,包括:调度执行核心、内存访问控制器以及通用寄存器,其中:所述内存访问控制器,用于接收内存访问指令;所述调度执行核心,用于将所述内存访问指令转发至所述内存访问控制器;以及,接收数据处理指令,获取所述数

据处理指令对应的第一源通用寄存器和/或第一目的通用寄存器;若检测到存在正在运行的内存访问指令,且所述内存访问指令与所述数据处理指令存在数据竞争风险,等待所述内存访问指令执行完成;所述内存访问指令与所述数据处理指令存在数据竞争风险包括如下任一情形:所述内存访问指令对应的第二目的通用寄存器与所述第一源通用寄存器存在交集;所述内存访问指令对应的第二源通用寄存器与所述第一目的通用寄存器存在交集;在所述内存访问指令执行完成后,执行所述数据处理指令。

[0015] 可选的,所述调度执行核心,用于从第一数据竞争状态表中,获取所述第二源通用寄存器对应比特域的取值,以及所述第二目的通用寄存器对应比特域的取值;若所述第二源通用寄存器对应比特域的取值为第一值,且所述第二源通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二源通用寄存器对应比特域的取值为第二值,或所述第二源通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;若所述第二目的通用寄存器对应比特域的取值为第一值,且所述第二目的通用寄存器与所述第一源通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二目的通用寄存器对应比特域的取值为第二值,或所述第二目的通用寄存器与所述第一源通用寄存器不存在交集,则确定未检测到所述数据竞争风险;其中,所述第一数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

[0016] 可选的,所述调度执行核心,用于在所述内存访问指令执行完成后,对所述第一数据竞争状态表进行更新,将所述第二目的通用寄存器对应比特域的取值置为第二值,以及,将所述第二源通用寄存器对应比特域的取值置为第二值。

[0017] 可选的,所述调度执行核心,用于从第二数据竞争状态表中,获取所述第二源通用寄存器对应比特域的取值;若所述第二源通用寄存器对应比特域的取值为第一值,且所述第二源通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二源通用寄存器对应比特域的取值为第二值,或所述第二源通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;所述第二数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

[0018] 可选的,所述调度执行核心,用于在所述内存访问指令执行完成后,对所述第二数据竞争状态表进行更新,将所述第二源通用寄存器对应比特域的取值置为第二值。

[0019] 可选的,所述调度执行核心,用于从第三数据竞争状态表中,获取所述第二目的通用寄存器对应比特域的取值;若所述第二目的通用寄存器对应比特域的取值为第一值,且所述第二目的通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二目的通用寄存器对应比特域的取值为第二值,或所述第二目的通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;所述第三数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

[0020] 可选的,所述调度执行核心,用于在所述内存访问指令执行完成后,对所述第三数据竞争状态表进行更新,将所述第二目的通用寄存器对应比特域的取值置为第二值。

[0021] 可选的,所述调度执行核心,还用于在检测到存在正在运行的内存访问指令,且所述内存访问指令与所述数据处理指令不存在所述数据竞争风险时,执行所述数据处理指令。

[0022] 本发明实施例还提供了一种计算机可读存储介质,所述计算机可读存储介质为非易失性存储介质或非瞬态存储介质,其上存储有计算机程序,所述计算机程序被处理器运行时执行上述任一种所述的保障数据一致性的指令执行方法的步骤。

[0023] 与现有技术相比,本发明实施例的技术方案具有以下有益效果:

在接收到数据处理指令之后,获取数据处理指令对应的第一源通用寄存器和/或第一目的通用寄存器。若检测到第一源通用寄存器和/或第一目的通用寄存器存在数据竞争风险,则暂停执行数据处理指令。在内存访问指令完成之后,调度执行核心再去执行数据处理指令,故能够有效避免数据竞争风险。

[0024] 进一步,通过设置第一数据竞争状态表来表征第一源通用寄存器和/或第一目的通用寄存器的使用状态,可以减少电路的数量,降低图形处理器的功耗。

附图说明

[0025] 图1是本发明实施例中的一种保障数据一致性的指令执行方法的流程图;

图2是本发明实施例中的一种图形处理器的结构示意图。

具体实施方式

[0026] 现有技术中,为解决数据竞争风险,由软件编译器维护分配栅栏计数器(fence counter)及1bit的标记(flag)来表征指令之间存在数据依赖关系,存在数据竞争风险的指令组合会被分配一组栅栏计数器和标记。当硬件执行到该指令组时,便会将其等待挂起,直到目的数据返回或者源数据被读走,方可解除释放栅栏(fence)资源。

[0027] 然而,栅栏计数器及标记的资源是有限的,若存在较多的存在数据竞争风险的指令组合,则可能无法为所有存在数据竞争风险的指令组合分配相应的栅栏计数器和标记,可能仍会存在指令执行错误的情况。

[0028] 在本发明实施例中,在接收到数据处理指令之后,获取数据处理指令对应的第一源通用寄存器和/或第一目的通用寄存器。若检测到第一源通用寄存器和/或第一目的通用寄存器存在数据竞争风险,则暂停执行数据处理指令。在内存访问指令完成之后,调度执行核心再去执行数据处理指令,故能够有效避免数据竞争风险。

[0029] 为使本发明的上述目的、特征和有益效果能够更为明显易懂,下面结合附图对本发明的具体实施例做详细的说明。

[0030] 本发明实施例提供了一种保障数据一致性的指令执行方法,参照图1,以下通过具体步骤进行详细说明。

[0031] 步骤101,接收数据处理指令。

[0032] 在具体实施中,图形处理器的调度执行核心可以接收数据处理指令,数据处理指令可以为ALU指令,或者为对内存(memory)的通用寄存器进行访问的指令。具体地,数据处

理指令可以包括smp指令、mov指令、fmul指令、fadd指令等。数据处理指令的具体类型并不仅限于上述示例，且数据处理指令的具体类型并不会对本发明实施例的保护范围造成影响。

[0033] 在本发明实施例中，调度执行核心在接收到数据处理指令之后，并不是立即执行数据处理指令，而是需要执行步骤102。

[0034] 步骤102，获取数据处理指令对应的第一源通用寄存器和/或第一目的通用寄存器。

[0035] 在本发明实施例中，数据处理指令可以为读操作指令，读操作指令中可以携带有读取操作对应的第一源通用寄存器。数据处理指令可以为写操作指令，写操作指令中可以携带有写操作对应的第一目的通用寄存器。数据处理指令还可以为其他类型的操作指令，该操作指令可以携带有第一源通用寄存器以及第一目的通用寄存器。

[0036] 调度执行核心在接收到数据处理指令之后，可以获取数据处理指令对应的第一源通用寄存器和/或第一目的通用寄存器。

[0037] 步骤103，若检测到存在正在运行的内存访问指令，且所述内存访问指令与所述数据处理指令存在数据竞争风险，等待所述内存访问指令执行完成。

[0038] 在具体实施中，数据竞争风险包括WAR数据竞争风险以及RAW数据竞争风险。

[0039] 在本发明实施例中，WAR数据竞争风险可以是指：内存访问指令对应的第二源通用寄存器与数据访问指令对应的第一目的通用寄存器存在交集；RAW数据竞争风险可以是指：内存访问指令对应的第二目的通用寄存器与数据访问指令对应的第一源通用寄存器存在交集。

[0040] 在具体实施中，针对WAR数据竞争风险、RAW数据竞争风险，可以分别独立设置相应的数据竞争状态表，也可以设置一个总的的数据竞争状态表。

[0041] 在本发明实施例中，针对WAR数据竞争风险（即内存访问指令对应读操作且数据处理指令对应写操作），可以设置第二数据竞争状态表。第二数据竞争状态表中，可以包括每一个通用寄存器一一对应的比特域，每一个比特域的长度可以相同，具体长度可以为1bit或者其他bit。通过对比特域赋不同的值，来表征相应的通用寄存器是否处于使用状态。

[0042] 在具体实施中，内存访问指令存在对应的第二源通用寄存器。调度执行核心可以在执行内存访问指令时，可以先将第二数据竞争状态表中第二源通用寄存器对应的比特域的取值置为第一值。在内存访问指令执行完成后，调度执行核心可以将第二数据竞争状态表中第二源通用寄存器对应的比特域的取值进行更新，将第二源通用寄存器对应的比特域的取值更新为第二值，第一值与第二值不等。

[0043] 换言之，当第二数据竞争状态表中，某一通用寄存器对应的比特域的取值为第一值时，表征该通用寄存器为内存访问指令对应的第二目的通用寄存器，且该通用寄存器正在被内存访问指令使用。当第二数据竞争状态表中，某一通用寄存器对应的比特域的取值为第二值时，表征该通用寄存器处于未被使用状态。

[0044] 例如，第二数据竞争状态表中，每一个通用寄存器对应的比特域的长度均为1bit，第一值为1，第二值为0。当某一通用寄存器比特域的取值为1时，表征该通用寄存器作为内存访问指令对应的第二源通用寄存器，且正在被内存访问指令正在使用；当比特域的取值为0时，表征该通用寄存器未被使用。

[0045] 在具体实施中,调度执行核心在执行数据处理指令之前,获知数据处理指令为写操作指令,且写操作指令对应的第一目的通用寄存器与正在执行的内存访问指令对应的第二源通用寄存器存在交集,则调度执行核心暂时不去执行数据处理指令。

[0046] 也就是说,针对WAR数据竞争风险,当检测到数据处理指令对应的第一目的通用寄存器对应的比特域的取值为1时,调度执行核心不会真正执行数据处理指令。

[0047] 在本发明实施例中,针对RAW数据竞争风险(即内存访问指令对应写操作且数据处理指令对应读操作),可以设置第三数据竞争状态表。第三数据竞争状态表中,可以包括每一个通用寄存器一一对应的比特域,每一个比特域的长度可以为1bit或者其他bit。通过对比比特域赋不同的值,来表征相应的通用寄存器是否处于使用状态。

[0048] 在具体实施中,内存访问指令存在对应的第二目的通用寄存器。调度执行核心可以在执行内存访问指令时,可以先将第三数据竞争状态表中第二目的通用寄存器对应的比特域的取值置为第一值。在内存访问指令执行完成后,调度执行核心可以将第三数据竞争状态表中第二目的通用寄存器对应的比特域的取值进行更新,将第二目的通用寄存器对应的比特域的取值更新为第二值,第一值与第二值不等。

[0049] 换言之,当第三数据竞争状态表中,某一通用寄存器对应的比特域的取值为第一值时,表征该通用寄存器为内存访问指令对应的第二目的通用寄存器,且该通用寄存器正在被内存访问指令使用。当第三数据竞争状态表中,某一通用寄存器对应的比特域的取值为第二值时,表征该通用寄存器处于未被使用状态。

[0050] 例如,第三数据竞争状态表中,每一个通用寄存器对应的比特域的长度为1bit,第一值为1,第二值为0。当某一通用寄存器比特域的取值为1时,表征该通用寄存器作为内存访问指令对应的第二目的通用寄存器,且正在被内存访问指令正在使用;当比特域的取值为0时,表征该通用寄存器未被使用。

[0051] 针对RAW数据竞争风险,当检测到数据处理指令对应的第一源通用寄存器对应的比特域的取值为1时,调度执行核心不会真正执行数据处理指令。

[0052] 在本发明实施例中,针对WAR数据竞争风险与RAW数据竞争风险,也可以设置一个总的的数据竞争状态表,以下简称为第一数据竞争状态表。

[0053] 在具体实施中,调度执行核心可以在执行内存访问指令时,可以先将第一数据竞争状态表中,内存访问指令关联的第二源通用寄存器、第二目的通用寄存器对应的比特域的取值均置为第一值。在内存访问指令执行完成后,调度执行核心可以将第二数据竞争状态表中第二源通用寄存器、第二目的通用寄存器对应的比特域的取值进行更新,将第二源通用寄存器对应的比特域的取值更新为第二值,第一值与第二值不等。

[0054] 在第一数据竞争状态表中,每一个通用寄存器均存在与之一一对应的比特域,每一个比特域的长度可以为1bit或者其他bit。通过对比比特域赋不同的值,来表征相应的通用寄存器是否处于使用状态。

[0055] 当数据处理指令对应的第一目的通用寄存器的比特域的取值为第一值时,表征该通用寄存器被内存访问指令正在使用;当数据处理指令对应的第一目的通用寄存器的比特域的取值为第二值时,表征该通用寄存器未被使用;第一值与第二值不等。

[0056] 例如,比特域的长度为1bit,第一值为1,第二值为0。当比特域的取值为1时,表征该通用寄存器被内存访问指令正在使用;当比特域的取值为0时,表征该通用寄存器未被使

用。

[0057] 可以理解的是,上述的第一数据竞争状态表还可以其他映射形式呈现。第二数据竞争状态表也可以其他映射形式呈现。第三数据竞争状态表也可以其他映射形式呈现。

[0058] 步骤104,在内存访问指令执行完成后,执行数据处理指令。

[0059] 在具体实施中,若调度执行核心检测到存在数据竞争风险,也即第一源通用寄存器和/或第一目的通用寄存器被内存访问指令正在使用,则不去真正执行数据处理指令,而是在内存访问指令完成使用第一源通用寄存器和/或第一目的通用寄存器之后,再去执行数据处理指令。

[0060] 在本发明实施例中,调度执行核心在接收到第一源通用寄存器和/或第一目的通用寄存器输出的状态清零请求之后,确定检测到内存访问指令执行完成。

[0061] 在具体实施中,存在WAR数据竞争风险时,若内存访问指令对应的第二源通用寄存器的个数为多个,则内存访问指令对应的多个第二通用寄存器中的至少一部分,为数据处理指令对应的第一目的通用寄存器。

[0062] 针对WAR数据竞争风险,在内存访问指令的运行过程中,调度执行核心可以使用第二数据竞争状态表。在第二数据竞争状态表中,将内存访问指令对应的第二源通用寄存器的比特域的取值置第一值。在内存访问指令完成读操作后,第二源通用寄存器可以向调度执行核心发送状态清零请求,以通知调度执行核心对第二数据竞争状态表进行更新。调度执行核心在接收到状态清零请求之后,将第二数据竞争状态表中,内存访问指令对应的第二源通用寄存器的比特域的取值更新为第二值。

[0063] 例如,内存访问指令对应的第二数据竞争状态表中,内存访问指令对应的第二源通用寄存器为r5,数据处理指令对应的第一目的通用寄存器为r5。在内存访问指令开始执行时,第二数据竞争状态表中r5对应的比特域的取值为1。在完成读操作之后,r5向调度执行核心发送状态清零请求。调度执行核心接收到状态清零请求之后,确定内存访问指令完成使用第二源通用寄存器,故执行数据处理指令。

[0064] 在具体实施中,存在RAW数据竞争风险时,内存访问指令对应的第二目的通用寄存器中的至少部分,为数据处理指令对应的第一源通用寄存器。

[0065] 针对RAW数据竞争风险,在内存访问指令的运行过程中,调度执行核心可以使用第三数据竞争状态表。在第三数据竞争状态表中,将内存访问指令对应的第二目的通用寄存器的比特域的取值置第一值。在内存访问指令完成读操作后,第二目的通用寄存器可以向调度执行核心发送状态清零请求,以通知调度执行核心对第三数据竞争状态表进行更新。调度执行核心在接收到状态清零请求之后,将第二数据竞争状态表中,内存访问指令对应的第二目的通用寄存器的比特域的取值更新为第二值。

[0066] 例如,内存访问指令对应的第三数据竞争状态表中,内存访问指令对应的第二目的通用寄存器为r5,数据处理指令对应的第一源通用寄存器为r5。在内存访问指令开始执行时,第三数据竞争状态表中r5对应的比特域的取值为1。在完成写操作之后,r5向调度执行核心发送状态清零请求。调度执行核心接收到状态清零请求之后,确定内存访问指令完成使用第一源通用寄存器,故执行数据处理指令。

[0067] 在具体实施中,在内存访问指令的运行过程中,调度执行核心可以在第一数据竞争状态表中,将与内存访问指令对应的第二源通用寄存器以及第二目的通用寄存器的比特

域的取值均设置为第一值。若第二源通用寄存器与第二目的通用寄存器不相同,在执行完读操作/写操作后,相应的第二源通用寄存器和/或第二目的通用寄存器即可向调度执行核心发送状态清零请求,可以无需等到内存访问指令执行完成再发送。调度执行核心接收到状态清零请求之后,将第一数据竞争状态表中相应的通用寄存器的比特域的取值置为第二值。

[0068] 若内存访问指令对应的第二源通用寄存器与第二目的通用寄存器存在相同,则需要等待第二源通用寄存器与第二目的通用寄存器均输出状态清零请求之后,再将第一数据竞争状态表中第二源通用寄存器的比特域的取值置为第二值。

[0069] 例如,第二源通用寄存器与第二目的通用寄存器均为r5。内存访问指令先从r5中读取数据,再向r5中写入数据。调度执行核心将r5对应的比特域的取值置1。在完成数据读取后,r5并没有向调度执行核心发送状态清零请求。在向r5中写入数据之后,r5再次向调度执行核心发送状态清零请求。调度执行核心在再次接收到状态清零请求之后,将r5对应的比特域的取值置0。

[0070] 下面通过几个示例对本发明上述实施例中提供的保障数据一致性的指令执行方法进行说明。

[0071] 示例一

参见如下伪指令:

```
smp r8, [r5],t[0],s[0]
mov r5,cb2
```

上述smp指令为内存访问指令,mov指令为ALU指令(即上述的数据处理指令)。上述smp指令所指示的操作为:将r5中存储的数据作为地址或坐标,从外部memory中抓取相应的数据,并将抓取的数据存储在r8中。r5中存储的数据为纹理坐标,t[0]表示纹理的属性类型,s[0]表示滤波算法的属性类型。

[0072] 上述mov指令所指示的操作为:将统一变量(uniform)值cb2赋给r5。

[0073] 可见,上述smp指令与mov指令均需要对r5进行操作,存在WAR场景下的数据竞争风险。因此,调度执行核心在执行smp指令时,将第一数据竞争状态表中r5对应的bit的取值置1,同时将第一数据竞争状态表中r8对应的bit的取值也置1(因为在执行smp指令时,尚未获知mov指令对应的目的通用寄存器是否包含r5,r8)。

[0074] 当纹理采样控制器将r5中的纹理坐标读取出来之后,调度执行核心即可清除第一数据竞争状态表中r5对应的bit(也即将第一数据竞争状态表中r5对应的bit置0)。之后,调度执行核心对mov指令放行,mov指令被执行,r5中的数据被cb2的取值所取代。在外部memory的数据被抓取并存储到r8之后,再将第一数据竞争状态表中r8对应的bit置0。

[0075] 示例二

参见如下伪指令:

```
smp r8, [r5],t[0],s[0]
fmul r2,r1,r8
```

上述smp指令为内存访问指令,fmul指令为ALU指令。上述fmul指令为浮点数相乘指令。

[0076] fmul指令对应两个源通用寄存器,分别为r1、r8,目的通用寄存器为r2。smp指令对

应的目的通用寄存器为r8。可见，fmul指令对应的源通用寄存器与smp指令对应的目的通用寄存器都包括r8，存在RAW场景下的数据竞争风险。当调度执行核心执行到smp指令时，将第一数据竞争状态表中r5对应的bit的取值置1，同时将第一数据竞争状态表中r8对应的bit的取值也置1。当调度执行核心需要执行fmul指令时，检测到第一数据竞争状态表中r8对应的bit为1，也即r8当前处于被smp指令使用状态，故等待，直到smp指令执行完成，调度执行核心才开始执行smp指令。

[0077] 示例三

参见如下伪指令：

```
smp r5, [r5],t[0],s[0]
fadd r2,r1,r5
```

上述smp指令为内存访问指令，fadd指令为ALU指令。上述fadd指令为浮点数加法指令。smp指令对应的源通用寄存器与目的通用寄存器相同，均为r5。fadd指令的一个源通用寄存器为r5，故存在RAW场景下的数据竞争风险。

[0078] 当调度执行核心执行到smp指令时，将第一数据竞争状态表中r5对应的bit的取值置1，但是，由于源通用寄存器与目的通用寄存器相同，故调度执行核心在完成从r5中读取数据之后，并不会将第一数据竞争状态表中r5对应的bit的取值置0，而是等待内存数据返回进r5之后，才将第一数据竞争状态表中r5对应的bit从置1修改为置0。之后，调度执行核心才开始执行fadd指令。

[0079] 若从r5中读取数据之后，将第一数据竞争状态表中r5对应的bit的取值置0，则fadd指令会被执行结束，导致fadd指令使用的r5的数据并不是内存(memory)的返回数据，会导致错误情况出现。

[0080] 示例四

参见如下伪指令：

```
smp r5, [r5],t[0],s[0]
mov r5,r1
```

上述smp指令为内存访问指令，mov指令为ALU指令。smp指令对应的源通用寄存器与目的通用寄存器相同，均为r5。mov指令的目的通用寄存器为r5，故存在WAR场景下的数据竞争风险。

[0081] 当调度执行核心执行到smp指令时，将第一数据竞争状态表中r5对应的bit的取值置1，但是，由于源通用寄存器与目的通用寄存器相同，故调度执行核心在完成从r5中读取数据之后，并不会将第一数据竞争状态表中r5对应的bit的取值置0，而是等待内存数据返回进r5之后，才将第一数据竞争状态表中r5对应的bit从置1修改为置0。之后，调度执行核心才开始执行mov指令。

[0082] 若从r5中读取数据之后，将第一数据竞争状态表中r5对应的bit的取值置0，则mov指令会被执行结束，导致mov指令使用的r5的数据并不是r1的数据，而是smp指令从内存(memory)中抓取返回的数据，会导致错误情况出现。

[0083] 综上所述，采用本发明上述实施例中提供的保障数据一致性的指令执行方法，在执行数据处理指令时，数据处理指令对应的第一源通用寄存器和/或第一目的通用寄存器均未被其他内存访问指令使用，故能够有效避免数据竞争风险。

[0084] 参照图2,给出了本发明实施例中的一种图形处理器,包括:调度执行核心201、内存访问控制器202以及通用寄存器203,其中:

所述内存访问控制器202,用于接收内存访问指令;

所述调度执行核心201,用于将所述内存访问指令转发至所述内存访问控制器;以及,接收数据处理指令,获取所述数据处理指令对应的第一源通用寄存器和/或第一目的通用寄存器;若检测到存在正在运行的内存访问指令,且所述内存访问指令与所述数据处理指令存在数据竞争风险,等待所述内存访问指令执行完成;所述内存访问指令与所述数据处理指令存在数据竞争风险包括如下任一情形:所述内存访问指令对应的第二目的通用寄存器与所述第一源通用寄存器存在交集;所述内存访问指令对应的第二源通用寄存器与所述第一目的通用寄存器存在交集;在所述内存访问指令执行完成后,执行所述数据处理指令;

所述通用寄存器,包括所述第一源通用寄存器、所述第一目的通用寄存器、所述第二源通用寄存器以及所述第二目的通用寄存器。

[0085] 在具体实施中,所述调度执行核心201,用于从第一数据竞争状态表中,获取所述第二源通用寄存器对应比特域的取值,以及所述第二目的通用寄存器对应比特域的取值;若所述第二源通用寄存器对应比特域的取值为第一值,且所述第二源通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二源通用寄存器对应比特域的取值为第二值,或所述第二源通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;若所述第二目的通用寄存器对应比特域的取值为第一值,且所述第二目的通用寄存器与所述第一源通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二目的通用寄存器对应比特域的取值为第二值,或所述第二目的通用寄存器与所述第一源通用寄存器不存在交集,则确定未检测到所述数据竞争风险;其中,所述第一数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

[0086] 在具体实施中,所述调度执行核心201,用于在所述内存访问指令执行完成后,对所述第一数据竞争状态表进行更新,将所述第二目的通用寄存器对应比特域的取值置为第二值,以及,将所述第二源通用寄存器对应比特域的取值置为第二值。

[0087] 在具体实施中,所述调度执行核心201,用于从第二数据竞争状态表中,获取所述第二源通用寄存器对应比特域的取值;若所述第二源通用寄存器对应比特域的取值为第一值,且所述第二源通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二源通用寄存器对应比特域的取值为第二值,或所述第二源通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;所述第二数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

[0088] 在具体实施中,所述调度执行核心201,用于在所述内存访问指令执行完成后,对所述第二数据竞争状态表进行更新,将所述第二源通用寄存器对应比特域的取值置为第二值。

[0089] 在具体实施中,所述调度执行核心201,用于从第三数据竞争状态表中,获取所述第二目的通用寄存器对应比特域的取值;若所述第二目的通用寄存器对应比特域的取值为第一值,且所述第二目的通用寄存器与所述第一目的通用寄存器存在交集,则确定检测到存在所述数据竞争风险;若所述第二目的通用寄存器对应比特域的取值为第二值,或所述第二目的通用寄存器与所述第一目的通用寄存器不存在交集,则确定未检测到所述数据竞争风险;所述第三数据竞争状态表中,任一通用寄存器对应比特域的取值为第一值时,表征所述通用寄存器处于被使用状态;任一通用寄存器对应比特域的取值为第二值时,表征所述通用寄存器处于未被使用状态;所述第一值不等于所述第二值。

[0090] 在具体实施中,所述调度执行核心201,用于在所述内存访问指令执行完成后,对所述第三数据竞争状态表进行更新,将所述第二目的通用寄存器对应比特域的取值置为第二值。

[0091] 在具体实施中,所述调度执行核心201,还用于在检测到存在正在运行的内存访问指令,且所述内存访问指令与所述数据处理指令不存在所述数据竞争风险时,执行所述数据处理指令。

[0092] 本发明实施例还提供了一种计算机可读存储介质,所述计算机可读存储介质为非易失性存储介质或非瞬态存储介质,其上存储有计算机程序,所述计算机程序被处理器运行时执行上述任一实施例提供的保障数据一致性的指令执行方法的步骤。

[0093] 本领域普通技术人员可以理解上述实施例的各种方法中的全部或部分步骤是可以通程序来指示相关的硬件来完成,该程序可以存储于一计算机可读存储介质中,存储介质可以包括:ROM、RAM、磁盘或光盘等。

[0094] 虽然本发明披露如上,但本发明并非限于此。任何本领域技术人员,在不脱离本发明的精神和范围内,均可作各种更动与修改,因此本发明的保护范围应当以权利要求所限定的范围为准。

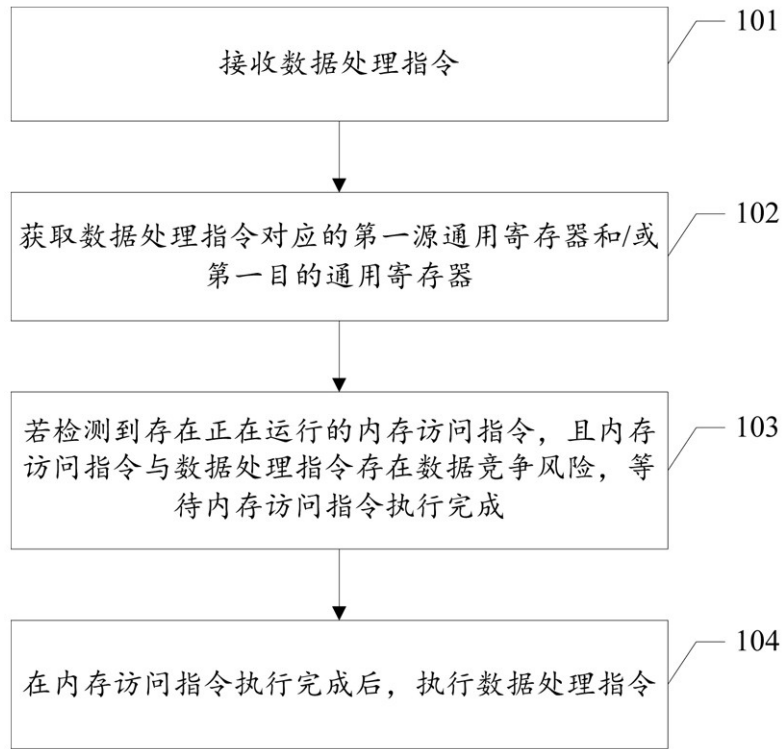


图 1

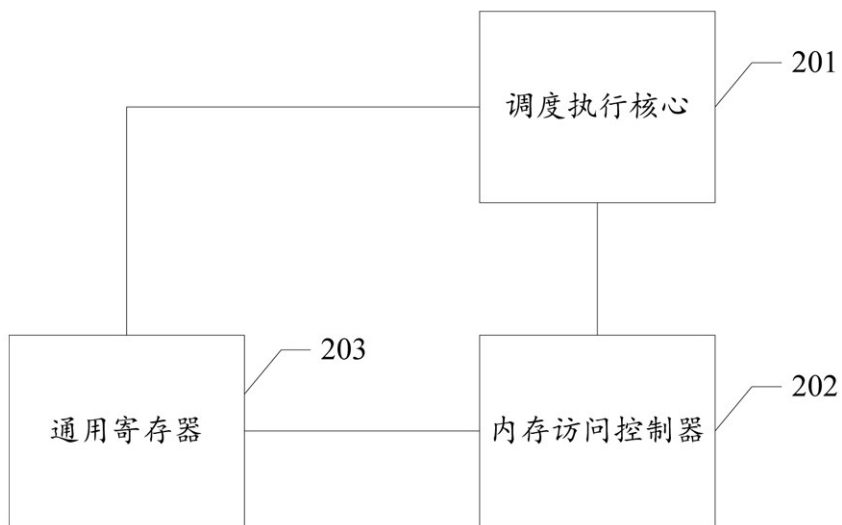


图 2