



(12)发明专利

(10)授权公告号 CN 104871246 B

(45)授权公告日 2017.09.29

(21)申请号 201380066624.7

(22)申请日 2013.11.21

(65)同一申请的已公布的文献号

申请公布号 CN 104871246 A

(43)申请公布日 2015.08.26

(30)优先权数据

13/725,393 2012.12.21 US

(85)PCT国际申请进入国家阶段日

2015.06.18

(86)PCT国际申请的申请数据

PCT/US2013/071283 2013.11.21

(87)PCT国际申请的公布数据

W02014/099249 EN 2014.06.26

(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 A·E·格鲁贝尔 汪涛

沙姆巴富·坎德瓦勒

(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.

G11C 7/10(2006.01)

G09G 5/393(2006.01)

G09G 3/36(2006.01)

G09G 5/395(2006.01)

G09F 3/14(2006.01)

G06F 12/02(2006.01)

G06F 12/06(2006.01)

G06F 13/16(2006.01)

(56)对比文件

US 2003/0231176 A1,2003.12.18,

CN 102591783 A,2012.07.18,

EP 1191445 A2,2002.03.27,

审查员 邢白灵

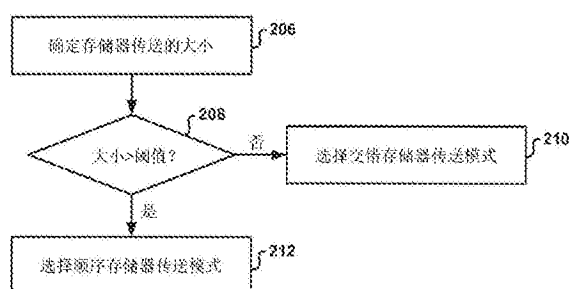
权利要求书8页 说明书38页 附图24页

(54)发明名称

用于执行基于图形处理单元的存储器传送操作的多模式存储器存取技术

(57)摘要

本发明描述用于基于可选存储器传送模式以图形处理单元GPU执行存储器传送操作的技术,以及用于选择用于以GPU执行存储器传送操作的全部或部分的存储器传送模式的技术。在一些实例中,本发明的技术可包含选择用于执行存储器传送操作的至少部分的存储器传送模式,且以GPU基于所述选定存储器传送模式执行所述存储器传送操作。所述存储器传送模式可选自包含交错存储器传送模式和顺序存储器传送模式的至少两个不同存储器传送模式的集合。本发明的技术可用以改善GPU辅助的存储器传送操作的性能。



1. 一种用于执行存储器的存储器传送操作的方法,其包括:

以一或多个处理器选择用于执行所述存储器传送操作的至少部分的存储器传送模式,所述存储器传送模式选自包含交错存储器传送模式和顺序存储器传送模式的至少两个不同存储器传送模式的集合,其中所述顺序存储器传送模式包括在单独的时间间隔期间交替发出多个连续读取请求的群组 and 多个连续写入请求的群组;以及

以图形处理单元GPU基于所述选定存储器传送模式执行所述存储器传送操作。

2. 根据权利要求1所述的方法,其中以所述一或多个处理器选择所述存储器传送模式包括:

以所述一或多个处理器基于所述存储器传送操作将传送的数据量选择所述存储器传送模式。

3. 根据权利要求2所述的方法,其中以所述一或多个处理器基于所述存储器传送操作将传送的所述数据量选择所述存储器传送模式包括:

以所述一或多个处理器在所述存储器传送操作将传送的所述数据量大于阈值的情况下将所述顺序存储器传送模式选择为所述存储器传送模式;以及

以所述一或多个处理器在所述存储器传送操作将传送的所述数据量不大于所述阈值的情况下将所述交错存储器传送模式选择为所述存储器传送模式。

4. 根据权利要求1所述的方法,其中以所述一或多个处理器选择所述存储器传送模式包括:

以所述一或多个处理器基于所述存储器传送操作的目的地区与固定屏幕对准缓冲区具有的重叠量而选择用于相对于所述固定屏幕对准缓冲区执行存储器传送操作的存储器传送模式。

5. 根据权利要求1所述的方法,

其中执行所述存储器传送操作包括执行着色器程序,且

其中以所述一或多个处理器选择所述存储器传送模式包括基于所述着色器程序的执行是否可能为所述存储器传送操作的性能瓶颈的确定而选择所述存储器传送模式。

6. 根据权利要求5所述的方法,其中以所述一或多个处理器基于所述着色器程序的所述执行是否可能为所述存储器传送操作的所述性能瓶颈的所述确定而选择所述存储器传送模式包括:

响应于确定所述着色器程序的所述执行不大可能为所述存储器传送操作的所述性能瓶颈而将所述顺序存储器传送模式选择为所述存储器传送模式;以及

响应于确定所述着色器程序的所述执行可能为所述存储器传送操作的所述性能瓶颈而将所述交错存储器传送模式选择为所述存储器传送模式。

7. 根据权利要求1所述的方法,

其中执行所述存储器传送操作包括执行着色器程序,且

其中以所述一或多个处理器选择所述存储器传送模式包括基于用于执行所述存储器传送操作的着色器程序的大小和用于执行所述存储器传送操作的所述着色器程序的复杂性中的至少一者而选择所述存储器传送模式。

8. 根据权利要求7所述的方法,其中以所述一或多个处理器基于用于执行所述存储器传送操作的所述着色器程序的所述大小和用于执行所述存储器传送操作的所述着色器程

序的所述复杂性中的所述至少一者而选择所述存储器传送模式包括：

以所述一或多个处理器在所述着色器程序的所述大小和所述着色器程序的所述复杂性中的所述至少一者小于阈值的情况下将所述顺序存储器传送模式选择为所述存储器传送模式；以及

以所述一或多个处理器在所述着色器程序的所述大小和所述着色器程序的所述复杂性中的所述至少一者不小于所述阈值的情况下将所述交错存储器传送模式选择为所述存储器传送模式。

9. 根据权利要求1所述的方法，其中以所述GPU执行所述存储器传送操作包括：

以所述GPU响应于交错存储器传送模式被选择为用于执行所述存储器传送操作的所述存储器传送模式而基于所述交错存储器传送模式执行所述存储器传送操作；以及

以所述GPU响应于顺序存储器传送模式被选择为用于执行所述存储器传送操作的所述存储器传送模式而基于所述顺序存储器传送模式执行所述存储器传送操作。

10. 根据权利要求9所述的方法，

其中以所述GPU基于所述交错存储器传送模式执行所述存储器传送操作包括基于生产者-消费者范例致使读取引擎发出读取请求且致使写入引擎发出写入请求，且

其中以所述GPU基于所述顺序存储器传送模式执行所述存储器传送操作包括基于相互排斥范例致使所述读取引擎发出读取请求且致使所述写入引擎发出写入请求。

11. 根据权利要求9所述的方法，

其中以所述GPU基于所述交错存储器传送模式执行所述存储器传送操作包括每次由所述GPU从存储器接收到足以发出用于所述存储器传送操作的写入请求的源数据时，以写入引擎消耗相应源数据而无需等待针对至少一个其它写入请求随后接收额外源数据，且

其中以所述GPU基于所述顺序存储器传送模式执行所述存储器传送操作包括针对所述存储器传送操作的至少部分，响应于所述GPU从所述存储器接收到足以发出用于所述存储器传送操作的写入请求的源数据而以所述写入引擎推迟所述源数据的消耗直到随后接收到足以发出至少一个其它写入请求的额外源数据。

12. 根据权利要求9所述的方法，

其中以所述GPU基于所述交错存储器传送模式执行所述存储器传送操作包括每次写入引擎消耗足以发出用于所述存储器传送操作的写入请求的源数据时，以读取引擎发出读取请求而无需等待由所述写入引擎随后消耗用于至少一个其它写入请求的额外源数据，且

其中以所述GPU基于所述顺序存储器传送模式执行所述存储器传送操作包括针对所述存储器传送操作的至少部分，响应于所述写入引擎消耗足以发出用于所述存储器传送操作的写入请求的源数据，以读取引擎推迟后续读取请求的发出直到所述写入引擎随后消耗足以发出至少一个其它写入请求的额外源数据。

13. 根据权利要求1所述的方法，

其中所述GPU包括缓冲器，所述缓冲器经配置以存储响应于由所述GPU向所述存储器发出的读取请求而接收的源数据，

其中以所述GPU基于所述交错存储器传送模式执行所述存储器传送操作包括：

发出一或多个读取请求；

将响应于所述一或多个读取请求从所述存储器接收的源数据放置到所述缓冲器中；

针对所述多个读取请求中的每一者,响应于针对所述多个读取请求中的相应一者的所述源数据被放置于所述缓冲器中,消耗放置于所述缓冲器中的所述源数据且基于所述消耗的源数据发出写入请求;以及

针对多个写入请求中的每一者,响应于从所述缓冲器消耗针对所述写入请求中的相应一者的源数据而发出后续读取请求,且

其中以所述GPU基于所述顺序存储器传送模式执行所述存储器传送操作包括:

发出多个读取请求;

将响应于所述多个读取请求从所述存储器接收的源数据放置到所述缓冲器中;

响应于检测到已经发出足够读取请求来填充所述缓冲器而停止发出读取请求;

响应于检测到所述缓冲器是满的而消耗所述缓冲器中的源数据且基于所述消耗的源数据发出多个写入请求;

响应于检测到所述缓冲器是空的而停止消耗所述缓冲器中的源数据且停止发出写入请求;以及

响应于检测到所述缓冲器是空的而恢复读取请求的发出。

14. 根据权利要求1所述的方法,其进一步包括:

使用选自至少两个不同存储器传送模式的所述集合的第一存储器传送模式执行所述存储器传送操作的第一部分;以及

使用选自至少两个不同存储器传送模式的所述集合的第二存储器传送模式执行所述存储器传送操作的第二部分。

15. 根据权利要求1所述的方法,其中以所述GPU执行所述存储器传送操作包括:

选择用于缓冲在基于所述选定存储器传送模式的所述存储器传送操作期间从存储器检索的源数据的高速缓冲存储器的高速缓存行大小;以及

基于所述选定高速缓存行大小配置用于所述存储器传送操作的所述高速缓冲存储器。

16. 根据权利要求15所述的方法,其中选择所述高速缓存行大小包括:

响应于所述顺序存储器传送模式被选择为用于执行所述存储器传送操作的所述存储器传送模式而选择等于所述高速缓冲存储器的整个容量的高速缓存行大小;以及

响应于所述交错存储器传送模式被选择为用于执行所述存储器传送操作的所述存储器传送模式而选择小于所述高速缓冲存储器的所述整个容量的高速缓存行大小。

17. 根据权利要求1所述的方法,其中所述一或多个处理器包括所述GPU,且其中以所述一或多个处理器选择所述存储器传送模式包括以所述GPU选择用于执行所述存储器传送操作的所述至少部分的所述存储器传送模式。

18. 根据权利要求1所述的方法,其中所述一或多个处理器包括CPU,其中以所述一或多个处理器选择所述存储器传送模式包括以所述CPU选择用于执行存储器传送操作的所述至少部分的所述存储器传送模式,且其中所述方法进一步包括致使所述GPU基于所述选定存储器传送模式执行所述存储器传送操作。

19. 一种用于执行存储器传送操作的方法,其包括:

以一或多个处理器选择用于执行所述存储器传送操作的至少部分的存储器传送模式,所述存储器传送模式选自包含交错存储器传送模式和顺序存储器传送模式的至少两个不同存储器传送模式的集合;以及

以图形处理单元GPU基于所述选定存储器传送模式执行所述存储器传送操作,其中以所述一或多个处理器基于所述存储器传送操作的目的地区与固定屏幕对准缓冲区具有的重叠量而选择所述存储器传送模式包括:

以所述一或多个处理器基于所述存储器传送操作的所述目的地区与所述固定屏幕对准缓冲区具有的重叠量而选择用于相对于所述固定屏幕对准缓冲区执行存储器传送操作的存储器传送模式;

在所述目的地区完全覆盖所述固定屏幕对准缓冲区的情况下将所述顺序存储器传送模式选择为所述存储器传送模式;以及

在所述目的地区并不完全覆盖所述固定屏幕对准缓冲区的情况下将所述交错存储器传送模式选择为所述存储器传送模式。

20. 一种用于执行存储器传送模式的方法,其包括:

以一或多个处理器选择用于执行所述存储器传送操作的至少部分的存储器传送模式,所述存储器传送模式选自包含交错存储器传送模式和顺序存储器传送模式的至少两个不同存储器传送模式的集合;以及

以图形处理单元GPU基于所述选定存储器传送模式执行所述存储器传送操作,其中以所述GPU执行所述存储器传送操作包括:

以所述GPU响应于交错存储器传送模式被选择为用于执行所述存储器传送操作的所述存储器传送模式而基于所述交错存储器传送模式执行所述存储器传送操作,其中基于所述交错存储器传送模式的所述存储器传送操作包括响应于所述GPU从存储器接收到足以用于发出写入请求的源数据而致使写入引擎发出所述写入请求且响应于所述写入引擎消耗足以用于发出所述写入请求的所述源数据而致使读取引擎发出读取请求,且

其中以所述GPU响应于顺序存储器传送模式被选择为用于执行所述存储器传送操作的所述存储器传送模式而基于所述顺序存储器传送模式执行所述存储器传送操作,以及其中基于所述顺序存储器传送模式的所述存储器传送操作包括致使所述读取引擎和所述写入引擎在单独的时间间隔期间交替发出多个连续读取请求的群组 and 多个连续写入请求的群组。

21. 一种用于执行存储器传送操作的装置,其包括:

至少一个存储器;

一或多个处理器,其经配置以选择用于执行所述至少一个存储器的所述存储器传送操作的至少部分的存储器传送模式,所述存储器传送模式选自包含交错存储器传送模式和顺序存储器传送模式的至少两个不同存储器传送模式的集合,其中所述顺序存储器传送模式包括在单独的时间间隔期间交替发出多个连续读取请求的群组 and 多个连续写入请求的群组;以及

图形处理单元GPU,其经配置以基于所述选定存储器传送模式执行所述存储器传送操作。

22. 根据权利要求21所述的装置,其中所述一或多个处理器进一步经配置以基于所述存储器传送操作将传送的数据量选择所述存储器传送模式。

23. 根据权利要求22所述的装置,其中所述一或多个处理器进一步经配置以在所述存储器传送操作将传送的所述数据量大于阈值的情况下将所述顺序存储器传送模式选择为

所述存储器传送模式,且在所述存储器传送操作将传送的所述数据量不大于所述阈值的情况下将所述交错存储器传送模式选择为所述存储器传送模式。

24.根据权利要求21所述的装置,其中所述一或多个处理器进一步经配置以基于所述存储器传送操作的目的地区与固定屏幕对准缓冲区具有的重叠量而选择用于相对于所述固定屏幕对准缓冲区执行存储器传送操作的存储器传送模式。

25.根据权利要求24所述的装置,其中所述一或多个处理器进一步经配置以在所述目的地区完全覆盖所述固定屏幕对准缓冲区的情况下将所述顺序存储器传送模式选择为所述存储器传送模式,且在所述目的地区并不完全覆盖所述固定屏幕对准缓冲区的情况下将所述交错存储器传送模式选择为所述存储器传送模式。

26.根据权利要求21所述的装置,其中所述GPU进一步经配置以执行着色器程序以执行所述存储器传送操作,且其中所述一或多个处理器进一步经配置以基于所述着色器程序的执行是否可能为所述存储器传送操作的性能瓶颈的确定而选择所述存储器传送模式。

27.根据权利要求26所述的装置,其中所述一或多个处理器进一步经配置以响应于确定所述着色器程序的所述执行不大可能为所述存储器传送操作的性能瓶颈而将所述顺序存储器传送模式选择为所述存储器传送模式,且响应于确定所述着色器程序的所述执行可能为所述存储器传送操作的性能瓶颈而将所述交错存储器传送模式选择为所述存储器传送模式。

28.根据权利要求21所述的装置,其中所述GPU进一步经配置以执行着色器程序以执行所述存储器传送操作,且其中所述一或多个处理器进一步经配置以基于用于执行所述存储器传送操作的着色器程序的大小和用于执行所述存储器传送操作的所述着色器程序的复杂性中的至少一者而选择所述存储器传送模式。

29.根据权利要求28所述的装置,其中所述一或多个处理器进一步经配置以在所述着色器程序的所述大小和所述着色器程序的所述复杂性中的所述至少一者小于阈值的情况下将所述顺序存储器传送模式选择为所述存储器传送模式,且在所述着色器程序的所述大小和所述着色器程序的所述复杂性中的所述至少一者不小于所述阈值的情况下将所述交错存储器传送模式选择为所述存储器传送模式。

30.根据权利要求21所述的装置,其中所述GPU进一步经配置以响应于交错存储器传送模式被选择为用于执行所述存储器传送操作的所述存储器传送模式而基于所述交错存储器传送模式执行所述存储器传送操作,且响应于顺序存储器传送模式被选择为用于执行所述存储器传送操作的所述存储器传送模式而基于所述顺序存储器传送模式执行所述存储器传送操作。

31.根据权利要求30所述的装置,

其中所述GPU进一步经配置以当基于所述交错存储器传送模式执行所述存储器传送操作时基于生产者-消费者范例致使读取引擎发出读取请求且致使写入引擎发出写入请求,且

其中所述GPU进一步经配置以当基于所述顺序存储器传送模式执行所述存储器传送操作时基于相互排斥范例致使所述读取引擎发出读取请求且致使所述写入引擎发出写入请求。

32.根据权利要求30所述的装置,

其中所述GPU进一步经配置以当基于所述交错存储器传送模式执行所述存储器传送操作时响应于所述GPU从存储器接收到足以用于发出写入请求的源数据而致使所述写入引擎发出所述写入请求且响应于所述写入引擎消耗足以用于发出写入请求的源数据而致使读取引擎发出读取请求,且

其中所述GPU进一步经配置以当基于所述顺序存储器传送模式执行所述存储器传送操作时致使所述读取引擎和所述写入引擎在单独的时间间隔期间交替发出多个连续读取请求的群组或多个连续读取请求写入请求的群组。

33. 根据权利要求30所述的装置,

其中所述GPU进一步经配置以每次由所述GPU从存储器接收到足以发出用于所述存储器传送操作的写入请求的源数据时,以写入引擎消耗相应源数据而无需当基于所述交错存储器传送模式执行所述存储器传送操作时等待针对至少一个其它写入请求随后接收额外源数据,且

其中所述GPU进一步经配置以针对所述存储器传送操作的至少部分,响应于所述GPU从所述存储器接收到足以发出用于所述存储器传送操作的写入请求的源数据而以所述写入引擎推迟所述源数据的消耗直到当基于所述顺序存储器传送模式执行所述存储器传送操作时随后接收到足以发出至少一个其它写入请求的额外源数据。

34. 根据权利要求30所述的装置,

其中所述GPU进一步经配置以每次写入引擎消耗足以发出用于所述存储器传送操作的写入请求的源数据时,以读取引擎发出读取请求而无需当基于所述交错存储器传送模式执行所述存储器传送操作时等待由所述写入引擎随后消耗用于至少一个其它写入请求的额外源数据,且

其中所述GPU进一步经配置以针对所述存储器传送操作的至少部分,响应于所述写入引擎消耗足以发出用于所述存储器传送操作的写入请求的源数据,以所述读取引擎推迟后续读取请求的发出直到当基于所述顺序存储器传送模式执行所述存储器传送操作时所述写入引擎随后消耗足以发出至少一个其它写入请求的源数据。

35. 根据权利要求21所述的装置,其中所述GPU包括缓冲器,所述缓冲器经配置以存储响应于由所述GPU向所述存储器发出的读取请求而接收的源数据,

其中所述GPU进一步经配置以当基于所述交错存储器传送模式执行所述存储器传送操作时:

发出一或多个读取请求;

将响应于所述一或多个读取请求从所述存储器接收的源数据放置到所述缓冲器中;

针对所述多个读取请求中的每一者,响应于针对所述多个读取请求中的相应一者的所述源数据被放置于所述缓冲器中,消耗放置于所述缓冲器中的所述源数据且基于所述消耗的源数据发出写入请求;以及

针对多个写入请求中的每一者,响应于从所述缓冲器消耗针对所述写入请求中的相应一者的源数据而发出后续读取请求,且

其中所述GPU进一步经配置以当基于所述顺序存储器传送模式执行所述存储器传送操作时:

发出多个读取请求;

将响应于所述多个读取请求从所述存储器接收的源数据放置到所述缓冲器中；

响应于检测到已经发出足够读取请求来填充所述缓冲器而停止发出读取请求；

响应于检测到所述缓冲器是满的而消耗所述缓冲器中的源数据且基于所述消耗的源数据发出多个写入请求；

响应于检测到所述缓冲器是空的而停止消耗所述缓冲器中的源数据且停止发出写入请求；以及

响应于检测到所述缓冲器是空的而恢复读取请求的发出。

36. 根据权利要求21所述的装置，其中所述GPU进一步经配置以使用选自至少两个不同存储器传送模式的所述集合的第一存储器传送模式执行所述存储器传送操作的第一部分，且使用选自至少两个不同存储器传送模式的所述集合的第二存储器传送模式执行所述存储器传送操作的第二部分。

37. 根据权利要求21所述的装置，其中所述GPU进一步经配置以选择用于缓冲在基于所述选定存储器传送模式的所述存储器传送操作期间从存储器检索的源数据的高速缓冲存储器的高速缓存行大小，且基于所述选定高速缓存行大小配置用于所述存储器传送操作的所述高速缓冲存储器。

38. 根据权利要求37所述的装置，其中所述GPU进一步经配置以响应于所述顺序存储器传送模式被选择为用于执行所述存储器传送操作的所述存储器传送模式而选择等于所述高速缓冲存储器的整个容量的高速缓存行大小，且响应于所述交错存储器传送模式被选择为用于执行所述存储器传送操作的所述存储器传送模式而选择小于所述高速缓冲存储器的所述整个容量的高速缓存行大小。

39. 根据权利要求21所述的装置，其中所述一或多个处理器包括所述GPU，且其中所述GPU进一步经配置以选择用于执行存储器传送操作的所述至少部分的所述存储器传送模式。

40. 根据权利要求21所述的装置，其中所述一或多个处理器包括CPU，所述CPU经配置以选择用于执行所述存储器传送操作的所述至少部分的所述存储器传送模式，且致使所述GPU基于所述存储器传送模式执行所述存储器传送操作。

41. 根据权利要求21所述的装置，其中所述装置包括无线通信装置。

42. 根据权利要求21所述的装置，其中所述装置包括移动电话手持机。

43. 一种用于执行存储器传送操作的设备，其包括：

用于选择用于执行所述存储器传送操作的至少部分的存储器传送模式的装置，所述存储器传送模式选自包含交错存储器传送模式和顺序存储器传送模式的至少两个不同存储器传送模式的集合，其中所述顺序存储器传送模式包括在单独的时间间隔期间交替发出多个连续读取请求的群组 and 多个连续写入请求的群组；以及

用于基于所述选定存储器传送模式执行所述存储器传送操作的装置。

44. 根据权利要求43所述的设备，其中所述用于选择所述存储器传送模式的装置包括：

用于基于所述存储器传送操作将传送的数据量、用于执行所述存储器传送操作的着色器程序的大小以及用于执行所述存储器传送操作的所述着色器程序的复杂性中的至少一者选择所述存储器传送模式的装置。

45. 根据权利要求43所述的设备，其中所述用于选择所述存储器传送模式的装置包括：

用于基于所述存储器传送操作的目的地区与固定屏幕对准缓冲区具有的重叠量而选择用于相对于所述固定屏幕对准缓冲区执行存储器传送操作的存储器传送模式的装置。

46. 根据权利要求43所述的设备，

其中所述用于执行所述存储器传送操作的装置包括图形处理单元GPU，且

其中所述用于选择所述存储器传送模式的装置包括CPU和所述GPU中的至少一者。

用于执行基于图形处理单元的存储器传送操作的多模式存储器存取技术

技术领域

[0001] 本发明涉及图形处理,且更具体地说,涉及用于以图形处理器执行存储器传送操作的技术。

背景技术

[0002] 图形处理单元 (GPU) 常常在计算装置中使用以处置和/或提供用于显示图形图像所需的操作中的一些或多者的硬件加速。GPU通常处置的一类操作是存储器传送操作。存储器传送操作可包含存储器复制操作和组构操作。在一些实例中,存储器传送操作可被称为传图 (blit) 操作或BLT操作 (即,位块传送操作)。执行这些存储器传送操作可涉及相对于存储器用于存储器传送操作的源和目的地数据的芯片外存储器执行大量读取和写入操作。

发明内容

[0003] 本发明描述用于基于可选存储器传送模式以图形处理单元 (GPU) 执行存储器传送操作的技术,以及用于选择用于以GPU执行存储器传送操作的全部或部分的存储器传送模式的技术。在一些实例中,用于执行存储器传送操作的技术可使用经配置以在使用交错存储器传送模式与顺序存储器传送模式之间选择性切换以用于执行存储器传送操作的全部或部分的GPU。交错存储器传送模式可致使GPU基于生产者-消费者范例执行存储器传送操作,其中读取请求和写入请求彼此交错。顺序存储器传送模式可致使GPU执行存储器传送操作以使得GPU在发出多个连续读取请求的群组与多个连续写入请求的群组之间交替。在额外实例中,用于选择存储器传送模式的技术可使用主机处理器 (例如,CPU) 和/或GPU来基于一或多个准则选择存储器传送模式以用于存储器传送操作的全部或部分。

[0004] 在一个实例中,本发明描述一种方法,其包含以一或多个处理器选择用于执行存储器传送操作的至少部分的存储器传送模式。所述存储器传送模式可选自包含交错存储器传送模式和顺序存储器传送模式的至少两个不同存储器传送模式的集合。所述方法进一步包含以GPU基于所述选定存储器传送模式执行所述存储器传送操作。

[0005] 在另一个实例中,本发明描述一种装置,其包含经配置以选择用于执行存储器传送操作的至少部分的存储器传送模式的一或多个处理器。所述存储器传送模式可选自包含交错存储器传送模式和顺序存储器传送模式的至少两个不同存储器传送模式的集合。所述装置进一步包含经配置以基于所述选定存储器传送模式执行所述存储器传送操作的GPU。

[0006] 在另一个实例中,本发明描述一种设备,其包含用于选择用于执行存储器传送操作的至少部分的存储器传送模式的装置。所述存储器传送模式可选自包含交错存储器传送模式和顺序存储器传送模式的至少两个不同存储器传送模式的集合。所述设备进一步包含用于基于所述选定存储器传送模式执行所述存储器传送操作的装置。

[0007] 在另一个实例中,本发明描述一种计算机可读存储媒体,其存储当被执行时致使一或多个处理器选择用于执行存储器传送操作的至少部分的存储器传送模式的指令。所述

存储器传送模式可选自包含交错存储器传送模式和顺序存储器传送模式的至少两个不同存储器传送模式的集合。所述计算机可读存储媒体进一步存储当被执行时致使一或多个处理器基于所述选定存储器传送模式执行所述存储器传送操作的指令。

[0008] 在附图和下文描述中阐述本发明的一或多个实例的细节。本发明的其它特征、目标和优点将从所述描述和图式以及权利要求书而显而易见。

附图说明

[0009] 图1是说明可用以实施本发明的多模式存储器传送技术和本发明的存储器传送模式选择技术的实例计算装置的框图。

[0010] 图2是进一步详细说明图1中的计算装置的CPU、GPU和存储器子系统的框图。

[0011] 图3是说明图2中所示的实例配置中可使用的实例存储器子系统的框图。

[0012] 图4是说明图2中所示的实例配置中可使用的另一实例存储器子系统的框图。

[0013] 图5是说明根据本发明的实例单个源存储器传送操作的概念图。

[0014] 图6是说明根据本发明的实例多源存储器传送操作的概念图。

[0015] 图7是说明根据本发明的实例GPU辅助的单个源存储器传送操作的概念图。

[0016] 图8是说明根据本发明的实例GPU辅助的多源存储器传送操作的概念图。

[0017] 图9A和9B是说明根据本发明的交错存储器传送模式执行的单个源存储器传送操作的实例时序特性的时序图。

[0018] 图10是说明根据本发明的顺序存储器传送模式执行的单个源存储器传送操作的实例时序特性的时序图。

[0019] 图11A和11B是说明根据本发明的交错存储器传送模式执行的多源存储器传送操作的实例时序特性的时序图。

[0020] 图12是说明根据本发明的顺序存储器传送模式执行的多源存储器传送操作的实例时序特性的时序图。

[0021] 图13是说明用于根据本发明的交错存储器传送技术执行存储器传送操作的实例技术的流程图。

[0022] 图14是说明用于根据本发明的交错同步模式发出读取请求的实例技术的流程图。

[0023] 图15是说明用于根据本发明的交错同步模式发出写入请求的实例技术的流程图。

[0024] 图16是说明用于根据本发明的顺序存储器传送技术执行存储器传送操作的实例技术的流程图。

[0025] 图17是说明用于根据本发明的顺序同步模式发出读取请求的实例技术的流程图。

[0026] 图18是说明用于根据本发明的顺序同步模式发出写入请求的实例技术的流程图。

[0027] 图19是说明用于根据本发明的顺序存储器传送模式执行存储器传送操作的实例技术的流程图。

[0028] 图20是说明根据本发明的用于执行存储器传送操作的实例技术的流程图。

[0029] 图21和22是说明根据本发明的用于选择用于执行存储器传送操作的存储器传送模式的实例技术的流程图。

[0030] 图23是说明根据本发明的具有多个屏幕对准缓冲区和一目的地表面的屏幕的概念图。

[0031] 图24到26是说明根据本发明的用于选择用于执行存储器传送操作的存储器传送模式的实例技术的流程图。

[0032] 图27是说明根据本发明的用于执行存储器传送操作的实例技术的流程图。

具体实施方式

[0033] 本发明描述用于基于可选存储器传送模式以图形处理单元 (GPU) 执行存储器传送操作的技术, 以及用于选择用于以GPU执行存储器传送操作的全部或部分的存储器传送模式的技术。在一些实例中, 用于执行存储器传送操作的技术可使用经配置以在使用交错存储器传送模式与顺序存储器传送模式之间选择性切换以用于执行存储器传送操作的全部或部分的GPU。交错存储器传送模式可致使GPU基于生产者-消费者范例执行存储器传送操作, 其中读取请求和写入请求彼此交错。顺序存储器传送模式可致使GPU执行存储器传送操作以使得GPU在发出多个连续读取请求的群组与多个连续写入请求的群组之间交替。在额外实例中, 用于选择存储器传送模式的技术可使用主机处理器 (例如, CPU) 和/或GPU来基于一或多个准则选择存储器传送模式以用于存储器传送操作的全部或部分。

[0034] 不同存储器传送模式对于不同类型的存储器传送操作可比其它存储器传送模式更有效。举例来说, 顺序存储器传送模式对于较大存储器传送操作可比交错存储器传送模式大体上更有效, 而交错存储器传送模式在一些情况下对于较小存储器传送操作可比顺序存储器传送模式更有效, 具体来说是在其中在顺序存储器传送模式期间每读取请求存取的数据的最小量较大的情况下。本发明的技术可用以选择性使用较适合于特定类型的存储器传送操作的不同存储器传送模式, 进而改善GPU的存储器传送操作的性能。

[0035] 存储器传送操作可包含存储器复制操作和组构操作。在一些实例中, 存储器传送操作可被称为传图 (即, 位块传送) 操作。存储器复制操作可涉及将数据从源表面的源区复制到目的地表面的目的地区。也就是说, GPU可从源表面的源区读取数据且将所述源表面的源区中含有的数据的精确复制品写入到目的地表面的目的地区。

[0036] 组构操作可涉及以某种方式组合相同或不同源表面的至少两个不同源区, 且将源区的组合写入到目的地表面的目的地区。在一些情况下, 源表面中的一或多者可为与用于组构操作的目的地表面相同的表面。举例来说, 组构操作可涉及读数单个源区和单个目的地区, 将所述两者掺合在一起, 且将所掺合结果写入回到目的地区。

[0037] 在一些实例中, 当执行组构操作时, 可根据光栅操作组合不同源区, 其对于组构操作的不同实例可不同。光栅操作可为指定针对特定组构操作应如何组合位的操作。在一些实例中, 光栅操作可包含逐位运算, 例如布尔型逻辑运算。布尔型逻辑运算可包含例如逐位AND、OR、NOT和/或XOR运算。

[0038] 在额外实例中, 光栅操作可包含掺合操作 (例如, α 掺合操作)。在一些实例中, 掺合操作可指定在掺合操作的执行期间将使用的源像素乘数和目的地像素乘数。在此些实例中, 在掺合操作的执行期间, 源像素值 (例如, 源像素颜色值和/或源像素 α 值) 可乘以源像素乘数以产生第一乘积, 对应目的地像素值 (例如, 目的地像素颜色值和/或目的地像素 α 值) 可乘以目的地像素乘数以产生第二乘积, 且可将两个乘积的总和相加在一起以产生像素位置的目的地值。其它类型的掺合操作也是可能的。在其它实例中, 不同源区可根据在GPU上执行的着色器程序而组合。

[0039] 存储器传送操作中使用的源和目的地表面可为相同或不同表面,且可各自对应于像素数据的二维阵列。像素数据可包含红、绿、蓝(RGB)颜色值,且在一些情况下,包含用于表面中的每一像素的 α 值(A)。因为所述表面对应于像素数据的二维阵列,所以表面中的个别像素位置的像素数据可通过行和列坐标来寻址。存储器传送操作中使用的源区中的每一者可对应于相应源表面的全部或子集。类似地,存储器传送操作中使用的目的地区可对应于目的地表面的全部或子集。源和目的地区可为相同的大小和形状,且通常但不一定是矩形形状区。

[0040] 当GPU执行存储器传送操作时,用于存储器传送操作的源表面和目的地表面通常存储在芯片外存储器中。换句话说,存储器可形成于、定位于和/或安置于微芯片、集成电路和/或不同于所述微芯片、集成电路的裸片和/或其上形成、定位和/或安置GPU的裸片上。存储器可经由一或多个总线且在一些情况下经由存储器控制器以通信方式耦合到GPU。

[0041] 当执行存储器复制操作时,GPU可从源区中的像素位置中的每一者读取数据且将从源区读取的数据写入到目的地区中的对应像素位置。类似地,组构操作的执行可涉及从多个源区中的位置中的每一者读取数据,组合所述数据,且将组合数据写入到目的地区。因此单个存储器传送操作可涉及由GPU经由一或多个总线将大量读取请求和大量写入请求发出到芯片外存储器。

[0042] 为了执行用于存储器传送操作的存储器读取和写入操作,在一些实例中,GPU可使用读取引擎和写入引擎。读取引擎可处置用于存储器传送操作的读取请求的发出,且写入引擎可处置用于存储器传送操作的写入请求的发出。当使用交错存储器传送模式执行存储器传送操作时,读取和写入引擎可利用生产者-消费者类型的同步,其中读取客户端发出初始量的读取请求,且在响应于读取请求接收到源数据之后写入客户端可即刻消耗所述数据且发出对应写入请求到存储器。在消耗源数据之后,读取引擎可即刻继续发出后续读取请求。读取与写入引擎之间的此类型的宽松同步可致使读取和写入流变为彼此混合和/或交错的。因此,当基于交错存储器传送模式执行存储器传送操作时,存储器和总线在服务于读取请求与服务于写入请求之间频繁地切换。

[0043] 对于一些类型的存储器,例如双数据速率同步动态随机存取存储器(DDR SDRAM),一定量的处理时间开销可与存储器和/或总线需要在服务于读取操作与服务于写入操作之间切换的每一时间相关联。举例来说,每次在服务于读取与写入请求之间发生切换时,在总线改变方向之前可能需要特定量的空闲时间来使总线稳定。此处理时间开销可被称为读取-写入周转时间。作为另一实例,如果读取和写入流存取存储器的不同页,那么每次在服务于读取与写入请求之间发生切换时存储器可能需要打开和关闭存储器的不同页和/或组,进而对读取-写入切换添加额外处理时间开销。

[0044] 存储器传送操作,例如组构和传图操作,常常是存储器受限的。也就是说,执行这些操作需要的存储器读取和写入请求常常变成GPU的性能瓶颈。因此,导致读取与写入请求的服务之间的大量切换的读取-写入引擎同步方案可不仅减少GPU的存储器存取效率,而且还可减少总存储器传送操作的性能。

[0045] 本发明中描述的顺序存储器传送模式可用以例如通过在单独的时间间隔期间将多个连续读取请求的序列和多个连续写入请求的序列发出到存储器而减少当在GPU中执行存储器传送操作时发生的读取-写入周转开销。因为读取和写入请求是在单独的时间间隔

期间在多个连续请求的群组中发出,所以系统存储器和/或总线可需要在服务于读取与写入请求之间切换的次数可减少,进而减少存储器传送操作的读取-写入周转开销。

[0046] 对于相对大的存储器传送操作,由顺序存储器传送模式提供的减少的读取-写入周转开销可致使顺序存储器传送模式在执行存储器传送操作时比交错存储器传送模式更有效。然而,对于较小存储器传送操作,执行顺序存储器传送模式中的同步所需要的额外开销在一些情况下可能超越了减少读取-写入周转开销的益处。此外,在其中在顺序存储器传送模式期间针对读取请求从存储器传送到GPU的数据的最小量与存储器传送操作(例如,BLT操作)的大小相比相对大的情况下,顺序存储器传送模式可致使从存储器读取额外的数据,进而减少存储器传送操作的性能。

[0047] 根据本发明的技术提供在用于特定存储器传送操作或其部分的顺序存储器传送模式与交错存储器传送模式之间选择性切换的能力可允许GPU选择性地在此模式的益处不被其它因素超过和/或减少的情况下使用顺序存储器传送模式来增加存储器效率,且在其中顺序存储器传送模式的益处被其它因素超过和/或减少的情况下使用交错存储器传送模式。通过允许GPU选择性使用较好地适合于特定类型(例如,大小)的存储器传送操作的不同存储器传送模式,本发明的技术可能改善执行多种不同类型的存储器传送操作的GPU的总性能。

[0048] 在一些实例中,在应用程序处理器上执行的GPU和/或GPU驱动器可基于存储器传送操作的大小选择用于执行存储器传送操作的全部或部分的存储器传送模式。存储器传送操作的大小可指代存储器传送操作将传送的数据的量。举例来说,如果存储器传送操作的大小大于阈值,那么GPU和/或GPU驱动器可将顺序存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式。作为另一实例,如果存储器传送操作的大小不大于阈值,那么GPU和/或GPU驱动器可将交错存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式。

[0049] 在额外实例中,在应用程序处理器上执行的GPU和/或GPU驱动器可基于针对存储器传送操作界定的目的地区与固定屏幕对准缓冲区具有的重叠的量而选择用于执行存储器传送操作的全部或部分的存储器传送模式。在一些情况下,表面可对应于显示监视器或屏幕上含有的像素。屏幕可经细分为多个区,其各自具有对应于用以存储所读取数据的缓冲器的大小的固定大小。这些经细分区可被称为固定屏幕对准缓冲区。在此些实例中,当在顺序存储器传送模式中操作时,GPU可经配置以在对应于固定屏幕对准缓冲区的大小的固定大小的数据单元中在存储器与GPU之间传送数据。

[0050] 如果目的地区部分地但不完全地覆盖特定的固定屏幕对准缓冲区,那么GPU和/或GPU驱动器可将交错存储器传送模式选择为用于屏幕的所述区的存储器传送模式,因为其中将存储读取数据的缓冲器将不会完全利用。如果在此情况下选定顺序存储器模式,那么固定大小存储器读取将致使从存储器读取额外的数据,这可超越通过使用顺序存储器传送模式将发生的任何读取-写入周转节省。另一方面,如果目的地区完全覆盖特定的固定屏幕对准缓冲区,那么GPU和/或GPU驱动器可将顺序存储器传送模式选择为用于屏幕的所述区的存储器传送模式,因为其中将存储读取数据的缓冲器将完全利用且将不浪费时间检索额外的数据。

[0051] 在其它实例中,在应用程序处理器上执行的GPU和/或GPU驱动器可基于用以执行

存储器传送操作(例如,组构操作)的着色器程序的大小和/或复杂性而选择用于执行存储器传送操作的全部或部分的存储器传送模式。着色器程序的大小和/或复杂性可指示用于特定存储器传送操作的着色器操作的性能是否可能为存储器传送操作的性能瓶颈。

[0052] 在此些实例中,如果着色器程序的大小和/或复杂性小于阈值,那么GPU和/或GPU驱动器可确定着色器操作的性能不大可能为存储器传送操作的性能瓶颈。在此情况下,GPU和/或GPU驱动器可选择用于存储器传送操作的顺序存储器传送模式,因为有可能存储器存取效率可为存储器传送操作的性能瓶颈。另一方面,如果着色器程序的大小和/或复杂性大于阈值,那么GPU和/或GPU驱动器可确定着色器操作的性能可能为存储器传送操作的性能瓶颈。在此情况下,GPU和/或GPU驱动器可选择用于存储器传送操作的交错存储器传送模式,因为存储器效率的增加是不需要的且因为在顺序存储器传送模式中发生的读取操作的延期可能具有使着色器程序耗尽数据的可能,这在着色器程序是性能瓶颈的情况下通常是不合意的。

[0053] 在额外实例中,GPU可基于选定存储器传送模式选择和配置用于高速缓冲存储器的高速缓冲存储线大小。高速缓冲存储器可对应于GPU中的其中存储读取数据的缓冲器。在此些实例中,如果存储器传送模式是顺序存储器传送模式,那么GPU可选择等于高速缓冲存储器的整个容量的高速缓冲存储线大小以便允许读取和写入操作的尽可能大的分组。另一方面,如果存储器传送模式是交错存储器传送模式,那么GPU可选择小于高速缓冲存储器的整个容量的高速缓冲存储线大小以便允许当服务于交错请求时同时高速缓冲存储多个数据区。

[0054] 根据本发明的一些方面,描述用于用信号表示将由GPU使用用于执行存储器传送操作的存储器传送模式的技术。所述信令技术可允许在主机应用程序处理器上执行的GPU驱动器和/或用户应用程序控制用于执行存储器传送操作的存储器传送模式。

[0055] 在一些实例中,指定存储器传送模式的信息字段可添加到经配置以由GPU执行的一或多个现有GPU命令。所述信息字段在一些情况下可为指定将用于与指令相关联的存储器传送操作的存储器传送模式的单个模式位。举例来说,图形驱动器可通过对GPU指定源和目的地存储器区且通过发出(三维)3D绘制调用指令到GPU而服务于用户层级传图命令。3D绘制调用指令可包含指示在3D绘制调用指令的执行期间将使用的存储器传送模式的模式位。GPU可经配置以基于指令中指定的存储器传送模式执行存储器传送操作。

[0056] 在额外实例中,GPU可包含指定将用于执行存储器传送操作的存储器传送模式的存储器传送模式状态变量。在此些实例中,GPU命令(例如,从GPU驱动器发出到GPU的命令)可用以将存储器传送模式状态变量的状态编程到指示将用于执行存储器传送操作的存储器传送模式的值。GPU可经配置以基于存储器传送模式状态变量的状态执行存储器传送操作。

[0057] 在其它实例中,存储器传送模式可为作为GPU命令流的部分而经修改的GPU状态的部分。GPU命令流可包含到由一或多个GPU着色器处理器执行的代码的状态寄存器设定命令以及指令(或指令指针)。

[0058] 在额外实例中,指令可添加到用于GPU驱动器的应用程序编程接口(API)。指令可包含存储器传送模式指示符,其指示将用于执行存储器传送操作的存储器传送模式。GPU驱动器可经配置以致使GPU基于存储器传送模式指示符执行存储器传送操作。以此方式,用户

应用程序可能指定当执行存储器传送操作时由GPU使用的存储器传送模式。

[0059] 图1是说明可用以实施本发明的多模式存储器传送技术和本发明的存储器传送模式选择技术的实例计算装置2的框图。计算装置2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、无线通信装置(例如,移动电话、蜂窝式电话、卫星电话和/或移动电话手持机)、陆线电话、因特网电话、手持式装置(例如,便携式视频游戏装置或个人数字助理(PDA))、个人音乐播放器、视频播放器、显示装置、电视、电视机顶盒、服务器、中间网络装置、主机计算机、或处理及/或显示图形数据的任何其它类型的装置。

[0060] 如图1的实例中所示,计算装置2包含用户接口4、CPU 6、存储器子系统8、图形处理单元(GPU) 10、GPU高速缓冲存储器12、显示器接口14、显示器16和总线18。用户接口4、CPU 6、存储器子系统8、GPU 10和显示器接口14可使用总线18彼此通信。应注意图1中所示的不同组件之间的总线和通信接口的特定配置仅是示范性的,且具有相同或不同组件的计算装置和/或其它图形处理系统的其它配置可用以实施本发明的技术。

[0061] CPU 6可包括控制计算装置2的操作的通用或专用处理器。用户可将输入提供到计算装置2以致使CPU 6执行一或多个软件应用程序。在CPU 6上执行的软件应用可包含(例如)操作系统、字处理器应用、电子邮件应用、电子数据表应用、媒体播放器应用、视频游戏应用、图形用户接口应用或另一程序。用户可经由一或多个输入装置(未图示)(例如,键盘、鼠标、麦克风、触摸垫或经由用户接口4耦合到计算装置2的另一输入装置)将输入提供到计算装置2。

[0062] 在CPU 6上执行的软件应用程序可包含指示GPU 10致使图形数据到显示器16的再现的一或多个图形再现指令。在一些实例中,软件指令可符合图形应用程序编程接口(API),例如开放图形库(OpenGL[®]) API、开放图形库嵌入系统(OpenGL ES) API、Direct3D API、DirectX API、RenderMan API、WebGL API或任何其它公众或专有标准图形API。为了处理图形再现指令,CPU 6可将一或多个图形再现命令发出到GPU 10以致使GPU 10执行图形数据的再现的一些或全部。在一些实例中,将再现的图形数据可包含例如点、线、三角形、四边形、三角形带、片等图形基元的列表。

[0063] 存储器子系统8可经配置以服务于从计算装置2内的其它组件接收的存储器存取请求。举例来说,存储器子系统8可服务于来自CPU 6、GPU 10和/或显示器接口14的存储器存取请求。存储器存取请求可包含读取存取请求和写入存取请求。存储器子系统8经由总线18以通信方式耦合到CPU 6、GPU 10、用户接口4和显示器接口14。

[0064] 存储器子系统8包含存储器控制器20和存储器22。存储器控制器20促进进出存储器22的数据的传送。举例来说,存储器控制器20可接收来自CPU 6和/或GPU 10的存储器读取请求和存储器写入请求,且相对于存储器22服务于此些请求以便为计算装置2中的组件提供存储器服务。

[0065] 读取请求可指定返回到请求者的数据应来自的存储器地址或存储地址范围。写入请求可指定指定数据应写入到的存储器地址或存储地址范围。待写入的数据可结合写入请求由请求者(例如,CPU 6和/或GPU 10)经由一或多个数据总线供应到存储器控制器20。在一些实例中,待写入数据可形成写入请求的部分。响应于接收到存储器存取请求,存储器控制器20可发出一或多个读取命令到存储器22和/或发出一或多个写入命令到存储器22以服

务于读取和写入请求。

[0066] 存储器控制器20经由专用存储器总线以通信方式耦合到存储器22,且经由总线18耦合到其它组件(例如,CPU 6和GPU 10)。虽然存储器控制器20在图1的实例计算装置2中说明为与CPU 6和存储器22两者分开的处理模块,但在其它实例中,存储器控制器20的功能性中的一些或全部可在CPU 6、GPU 10和存储器22中的一或多个者中实施。

[0067] 存储器22可存储可由CPU 6存取以用于执行的程序模块及/或指令及/或由在CPU6上执行的程序使用的数据。举例来说,存储器22可存储用户应用程序及与所述应用程序相关联的图形数据。存储器22还可存储用于由计算装置2的其它组件使用和/或由其产生的信息。举例来说,存储器22可充当用于GPU 10的装置存储器且可存储将在GPU 10上操作的数据以及由GPU 10执行的操作而产生的数据。举例来说,存储器22可存储表面、源表面、目的地表面、纹理缓冲器、深度缓冲器、模板缓冲器、顶点缓冲器、帧缓冲器或类似物的任何组合。另外,存储器22可存储用于由GPU 10处理的命令流。存储器22可包含一或多个易失性或非易失性存储器或存储装置,例如随机存取存储器(RAM)、静态RAM(SRAM)、动态RAM(DRAM)、同步动态随机存取存储器(SDRAM)、只读存储器(ROM)、可擦除可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)、快闪存储器、磁性数据媒体或光学存储媒体。在一些实例中,存储器22可为双数据速率(DDR) SDRAM,例如DDR1 SDRAM、DDR2 SDRAM、DDR3 SDRAM、DDR4 SDRAM等。

[0068] GPU 10可经配置以执行由CPU 6发出到GPU 10的命令。由GPU 10执行的命令可包含图形命令、绘制调用命令、GPU状态编程命令、存储器传送命令、通用计算命令、核心执行命令等。存储器传送命令可包含例如存储器复制命令、存储器组构命令和传图命令。

[0069] 在一些实例中,GPU 10可经配置以执行图形操作以将一或多个图形基元再现到显示器16。在此等实例中,当在CPU 6上执行的软件应用程序中的一者需要图形处理时,CPU 6可将图形数据提供到GPU 10用于向显示器16再现且将一或多个图形命令发出到GPU 10。所述图形命令可包含例如绘制调用命令、GPU状态编程命令、存储器传送命令、传图命令等。所述图形数据可包含顶点缓冲器、纹理数据、表面数据等。在一些实例中,CPU 6可通过将命令和图形数据写入到可由GPU 10存取的存储器22而将命令和图形数据提供到GPU 10。

[0070] 在其它实例中,GPU 10可经配置以执行用于CPU 6上执行的应用程序的通用计算。在此等实例中,当CPU 6上执行的软件应用程序中的一者决定将计算任务卸载到GPU 10时,CPU 6可将通用计算数据提供到GPU 10,且将一或多个通用计算命令发出到GPU 10。所述通用计算命令可包含例如核心执行命令、存储器传送命令等。在一些实例中,CPU 6可通过将命令和图形数据写入到可由GPU 10存取的存储器22而将命令和通用计算数据提供到GPU 10。

[0071] 在一些情况下,GPU 10可内置有高度并行的结构,其提供比CPU 6对向量操作的更有效的处理。举例来说,GPU 10可包含经配置而以并行方式对多个顶点、控制点、像素和/或其它数据操作的多个处理元件。在一些情况下,GPU 10的高度并行本质可允许GPU 10比使用CPU 6再现图像更快地将图形图像(例如,GUI和二维(2D)和/或三维(3D)图形场景)再现到显示器16上。另外,GPU 10的高度并行本质可允许GPU 10比CPU 6更快地处理某些类型的向量和矩阵运算以用于通用计算应用程序。

[0072] 在一些实例中,GPU 10可集成到计算装置2的母板中。在其它情况下,GPU 10可存在于安装在计算装置2的母板中的端口中的图形卡上或否则可并入在经配置以与计算装置

2交互操作的外围装置内。在进一步例子中, GPU 10可与CPU 6位于同一微芯片上从而形成芯片上系统(SoC)。GPU 10可包含一或多个处理器,例如一或多个微处理器、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、数字信号处理器(DSP)或其它等效的集成或离散逻辑电路。

[0073] 在一些实例中, GPU 10可直接耦合到GPU高速缓冲存储器12。因此, GPU 10可从GPU高速缓冲存储器12读取数据且写入数据到GPU高速缓冲存储器12而不一定使用总线18。换句话说, GPU 10可使用局部存储装置而非芯片外存储器来局部地处理数据。此允许GPU 10通过消除GPU 10经由总线18读取及写入数据的需要而以更有效的方式操作,其中经由总线操作可经历繁重的总线业务。然而,在一些情况下, GPU 10可不包含单独的高速缓冲存储器,而是经由总线18利用存储器22。GPU高速缓冲存储器12可包含一或多个易失性或非易失性存储器或存储装置,例如随机存取存储器(RAM)、静态RAM(SRAM)、动态RAM(DRAM)、可擦除可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)、快闪存储器、磁性数据媒体或光学存储媒体。

[0074] CPU 6及/或GPU 10可将再现的图像数据存储在存储器22内所分配的帧缓冲器中。显示器接口14可从帧缓冲器检索数据且配置显示器16以显示由经再现图像数据表示的图像。在一些实例中,显示器接口14可包含数/模转换器(DAC),其经配置以将从帧缓冲器检索的数字值转换为可由显示器16消耗的模拟信号。在其它实例中,显示器接口14可将数字值直接传递到显示器16以进行处理。显示器16可包含监视器、电视、投影装置、液晶显示器(LCD)、等离子显示器面板、发光二极管(LED)阵列、阴极射线管(CRT)显示器、电子纸、表面传导电子发射显示器(SED)、激光电视显示器、纳米晶体显示器或另一类型的显示单元。显示器16可集成在计算装置2内。举例来说,显示器16可为移动电话手持机或平板计算机的屏幕。或者,显示器16可为经由有线或无线通信链路而耦合到计算机装置2的独立装置。举例来说,显示器16可为经由电缆或无线链路而连接到个人计算机的计算机监视器或平板显示器。

[0075] 总线18可使用总线结构和总线协议的任何组合来实施,包含第一、第二和第三代总线结构和协议、共享总线结构和协议、点对点总线结构和协议、单向总线结构和协议以及双向总线结构和协议。可用以实施总线18的不同总线结构和协议的实例包含例如超传输总线、InfiniBand总线、高级图形端口总线、外围组件互连(PCI)总线、PCI高速总线、高级微控制器总线架构(AMBA)、高级高性能总线(AHB)、AMBA高级外围总线(APB),和AMBA高级eXtensible接口(AXI)总线。也可以使用其它类型的总线结构和协议。

[0076] 根据本发明, CPU 6和GPU 10中的一者或两者可经配置以选择用于执行存储器传送操作的至少部分的存储器传送模式,且GPU 10可经配置以基于所述选定存储器传送模式执行存储器传送操作。存储器传送模式可选自至少两个不同存储器传送模式的集合。在一些实例中,至少两个不同存储器传送模式的集合可包含交错存储器传送模式和顺序存储器传送模式。

[0077] 在一些实例中,存储器传送模式可指示在存储器传送操作的执行期间读取请求的发出和写入请求的发出如何经同步。在此些实例中,本发明的技术可允许GPU 10在用于读取请求和写入请求的发出的同步的至少两个不同模式之间选择性切换。举例来说,如果将交错存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式,

那么GPU 10可致使基于生产者-消费者同步范例发出读取请求和写入请求。举例来说,写入引擎可响应于GPU 10接收到源数据(例如,响应于正产生源数据)而消耗源数据,且读取引擎可响应于源数据正由写入引擎消耗而发出读取请求。另一方面,如果将顺序存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式,那么GPU 10可致使基于相互排斥同步范例发出读取请求和写入请求。举例来说,GPU 10可在单独的时间间隔期间在启用读取引擎以发出多个连续读取请求与启用写入引擎以发出多个连续写入请求之间交替。

[0078] 在其它实例中,存储器传送模式可指示在存储器传送操作的执行期间读取请求和写入请求将交错的粒度级别。在此些实例中,本发明的技术可允许GPU 10在用于存储器传送操作的执行的读取请求和写入请求的至少两个不同交错粒度之间选择性切换。举例来说,如果将交错存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式,那么GPU 10可致使读取请求和写入请求以相对精细粒度级别交错。举例来说,单个读取请求和单个写入请求可彼此交错。另一方面,如果将顺序存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式,那么GPU 10可致使读取请求和写入请求以相对粗糙粒度级别交错。举例来说,GPU 10可在发出多个连续读取请求的群组与多个连续写入请求的群组之间交替。

[0079] 在额外实例中,存储器传送模式可指示读取请求和写入请求是否需要由GPU 10以多个连续读取请求的群组和多个连续写入请求的群组发出。在此些实例中,本发明的技术可允许GPU 10在以下各项之间选择性切换:(1) 要求读取请求和写入请求以多个连续读取请求的群组和多个连续写入请求的群组发出;以及(2) 不要求读取请求和写入请求以多个连续读取请求的群组和多个连续写入请求的群组发出。

[0080] 举例来说,如果将交错存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式,那么GPU 10可不需要以多个连续读取请求的群组和多个连续写入请求的群组发出读取请求和写入请求。举例来说,GPU 10可允许读取引擎和写入引擎基于生产者-消费者范例独立地发出个别存储器请求。另一方面,如果将顺序存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式,那么GPU 10可需要以多个连续读取请求的群组和多个连续写入请求的群组发出读取请求和写入请求。举例来说,GPU 10可在对读取引擎准予独占式存储器存取以便发出多个连续读取请求到存储器22与对写入引擎准予独占式存储器存取以便发出多个连续写入请求到存储器22之间交替。

[0081] 在一些实例中,GPU 10和/或CPU 6(例如,在CPU 6上执行的GPU驱动器)可基于存储器传送操作的大小选择用于执行存储器传送操作的全部或部分的存储器传送模式。存储器传送操作的大小可指代存储器传送操作将传送的数据的量。举例来说,如果存储器传送操作的大小大于阈值,那么GPU 10和/或CPU 6可将顺序存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式。另一方面,如果存储器传送操作的大小不大于阈值,那么GPU 10和/或CPU 6可将交错存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式。

[0082] 在额外实例中,GPU 10和/或CPU 6(例如,在CPU 6上执行的GPU驱动器)可基于针对存储器传送操作界定的目的地区与固定屏幕对准缓冲区具有的重叠的量而选择用于执行存储器传送操作的全部或部分的存储器传送模式。在一些情况下,表面可对应于显示监

视器或屏幕上含有的像素。屏幕可经细分为多个区,其各自具有对应于用以存储所读取数据的缓冲器的大小的固定大小。这些经细分区可被称为固定屏幕对准缓冲区。在此些实例中,当在顺序存储器传送模式中操作时,GPU 10可经配置以在对应于固定屏幕对准缓冲区的大小的固定大小的数据单元中在存储器22与GPU 10之间传送数据。

[0083] 如果目的地区部分地但不完全地覆盖特定的固定屏幕对准缓冲区,那么GPU 10和/或CPU 6可将交错存储器传送模式选择为用于屏幕的所述区的存储器传送模式,因为其中将存储读取数据的缓冲器将不会完全利用。如果在此情况下选定顺序存储器模式,那么固定大小存储器读取将致使从存储器读取额外的数据,这可超越通过使用顺序存储器传送模式将发生的任何读取-写入周转节省。另一方面,如果目的地区完全覆盖特定的固定屏幕对准缓冲区,那么GPU 10和/或CPU 6可将顺序存储器传送模式选择为用于屏幕的所述区的存储器传送模式,因为其中将存储读取数据的缓冲器将完全利用且将不浪费时间检索额外的数据。

[0084] 在其它实例中,GPU 10和/或CPU 6(例如,在CPU 6上执行的GPU驱动器)可基于用以执行存储器传送操作(例如,组构操作)的着色器程序的大小和/或复杂性而选择用于执行存储器传送操作的存储器传送模式。着色器程序的大小和/或复杂性可指示用于特定存储器传送操作的着色器操作的性能是否可能为存储器传送操作的性能瓶颈。

[0085] 在此些实例中,如果着色器程序的大小和/或复杂性小于阈值,那么GPU 10和/或CPU 6可确定着色器操作的性能不大可能为存储器传送操作的性能瓶颈。在此情况下,GPU 10和/或CPU 6可选择用于存储器传送操作的顺序存储器传送模式,因为有可能存储器存取效率可为存储器传送操作的性能瓶颈。另一方面,如果着色器程序的大小和/或复杂性大于阈值,那么GPU 10和/或CPU 6可确定着色器操作的性能可能为存储器传送操作的性能瓶颈。在此情况下,GPU 10和/或CPU 6可选择用于存储器传送操作的交错存储器传送模式,因为存储器效率的增加是不需要的且因为在顺序存储器传送模式中发生的读取操作的延期可能具有使着色器程序耗尽数据的可能,这在着色器程序是性能瓶颈的情况下通常是不合意的。

[0086] 在额外实例中,GPU 10和/或CPU 6(例如,在CPU 6上执行的GPU驱动器)可基于着色器程序的执行是否可能为存储器传送操作的性能瓶颈的确定而选择存储器传送模式。举例来说,GPU 10和/或CPU 6可响应于确定着色器程序的执行不大可能为存储器传送操作的性能瓶颈而将顺序存储器传送模式选择为存储器传送模式。另一方面,GPU 10和/或CPU 6响应于确定着色器程序的执行可能为存储器传送操作的性能瓶颈而交错存储器传送模式选择为存储器传送模式。

[0087] 在额外实例中,GPU 10可基于选定存储器传送模式选择和配置用于高速缓冲存储器的高速缓冲存储线大小。高速缓冲存储器可对应于GPU中的其中存储读取数据的缓冲器。在此些实例中,如果存储器传送模式是顺序存储器传送模式,那么GPU 10可选择等于高速缓冲存储器的整个容量的高速缓冲存储线大小以便允许读取和写入操作的尽可能大的分组。另一方面,如果存储器传送模式是交错存储器传送模式,那么GPU10可选择小于高速缓冲存储器的整个容量的高速缓冲存储线大小以便允许当服务于交错请求时同时高速缓冲存储多个数据区。

[0088] 在一些实例中,计算装置2内的一或多个其它组件(例如,CPU 6)可用信号表示将

由GPU 10用于执行存储器传送操作的存储器传送模式。所述信令技术可允许在应用程序处理器上执行的GPU驱动器和/或用户应用程序控制用于执行存储器传送操作的存储器传送模式。

[0089] 不同存储器传送模式对于不同类型的存储器传送操作可比其它存储器传送模式更有效。举例来说,顺序存储器传送模式对于较大存储器传送操作可比交错存储器传送模式大体上更有效,而交错存储器传送模式对于较小存储器传送操作可比顺序存储器传送模式更有效,具体来说是在其中在顺序存储器传送模式期间每读取请求存取的数据的最小量较大的情况下。本发明的技术可用以选择性使用较适合于特定类型的存储器传送操作的不同存储器传送模式,进而改善GPU的存储器传送操作的性能。

[0090] 图2是进一步详细说明图1中计算装置2的CPU 6、GPU 10和存储器子系统8的框图。如图2中所示,CPU 6以通信方式耦合到GPU 10和存储器子系统8,且GPU10以通信方式耦合到CPU 6和存储器子系统8。在一些实例中,GPU 10可与CPU 6集成到母板上。在额外实例中,GPU 10可实施在安装于包含CPU 6的母板的端口中的图形卡上。在其它实例中,GPU 10可并入在经配置以与CPU 6交互操作的外围装置内。在额外实例中,GPU 10可与CPU 6位于同一微芯片上从而形成芯片上系统(SoC)。

[0091] CPU 6可经配置以执行软件应用程序24、图形API 26、GPU驱动器28和操作系统30。软件应用程序24可包含致使显示图形图像的一或多个指令和/或致使在GPU 10上执行非图形任务(例如,通用计算任务)的一或多个指令。软件应用程序24可发出指令到图形API 26。图形API 26可为运行时间服务,其将从软件应用程序24接收的指令转换为可由GPU驱动器28消耗的格式。

[0092] GPU驱动器28经由图形API 26从软件应用24接收指令,且控制GPU 10的操作以服务所述指令。举例来说,GPU驱动器28可调配一或多个命令,将命令放置到存储器22中,且指示GPU 10执行所述命令。在一些实例中,GPU驱动器28可将命令放置到存储器22中,且经由操作系统30(例如经由一或多个系统调用)与GPU 10通信。

[0093] 在一些实例中,GPU驱动器28(或CPU 6上的另一软件组件)可经配置以选择将用于执行存储器传送操作的全部或部分的存储器传送模式,且致使GPU 10基于所述选定存储器传送模式执行存储器传送操作。在一些实例中,存储器传送模式可选自包含交错存储器传送模式和顺序存储器传送模式的至少两个存储器传送模式的集合。

[0094] 在额外实例中,GPU驱动器28可经配置以根据本发明中描述的技术中的任一者选择将用于存储器传送操作的全部或部分的存储器传送模式。举例来说,GPU驱动器28可基于存储器传送操作的大小(例如,存储器传送操作将传送的数据的量)而选择存储器传送模式。作为另一实例,GPU驱动器28可基于针对存储器传送操作界定的目的地区与固定屏幕对准缓冲区具有的重叠的量而选择存储器传送模式。作为另一个实例,GPU驱动器28可基于着色器程序的执行是否可能为存储器传送操作的性能瓶颈的确定而选择存储器传送模式。在额外实例中,GPU 10可使用一或多个着色器程序执行存储器传送操作,且GPU驱动器28可基于用于执行存储器传送操作的着色器程序的大小和/或用于执行存储器传送操作的着色器程序的复杂性而选择存储器传送模式。

[0095] GPU 10包含命令引擎32、光栅操作单元34、缓冲器36、读取引擎38和写入引擎40。在一些实例中,GPU 10可包含3D图形再现管线(未图示)。在此些实例中,光栅操作单元34可

形成3D图形再现管线的部分。

[0096] 在一些实例中, GPU 10可包含多个处理单元(未图示), 其中的每一者可为可编程处理单元或固定功能处理单元。在其中GPU 10包含3D图形再现管线的情况下, 所述多个处理单元可一起操作以实施3D图形再现管线。在一些实例中, 3D图形再现管线可符合由一或多个公众和/或专有3D图形标准API指定的3D图形管线, 例如开放图形库 (OpenGL[®]) API、开放图形库嵌入系统 (OpenGL ES) API、Direct3D API、DirectX API等。

[0097] 可编程处理单元可包含(例如) 可编程着色器单元, 其经配置以执行从CPU 6下载到GPU 10上的一或多个着色器程序。在一些实例中, 着色器程序可为以高级着色语言编写的程序的经编译版本, 例如OpenGL着色语言 (GLSL)、高级着色语言 (HLSL)、用于图形的C (Cg) 着色语言等。在一些实例中, 可编程着色器单元可包含经配置以并行操作的多个处理单元, 例如SIMD管线。可编程着色器单元可具有存储着色器程序指令的程序存储器和执行状态寄存器, 例如指示程序存储器中正执行的当前指令或待提取的下一指令的程序计数器寄存器。所述可编程着色器单元可包含(例如) 顶点着色器单元、像素着色器单元、几何着色器单元、壳着色器单元、域着色器单元、计算着色器单元等。

[0098] 固定功能处理单元可包含经硬连线以执行某些功能的硬件。虽然固定功能硬件可经由(例如) 一或多个控制信号而配置以执行不同功能, 但所述固定功能硬件通常并不包含能够接收用户编译程序的程序存储器。GPU 10中包含的固定功能处理单元可包含(例如) 执行例如光栅操作、深度测试、剪刀测试、 α 掺杂等的处理单元。在其中光栅操作单元34包含在3D再现管线中的实例中, 在一些实例中光栅操作单元34可为可配置固定功能处理单元。在一些实例中, 光栅操作单元34可实施为3D再现管线的输出合并器处理单元。

[0099] 命令引擎32经配置以从CPU 6接收命令且致使GPU 10执行所述命令。在一些实例中, CPU 6可将命令放置于存储器22中, 且命令引擎32可从存储器22检索所述命令。所述命令可包含(例如) 改变GPU 10的状态(例如, 再现状态)的状态命令、指示GPU 10将特定着色器程序加载到3D再现管线中的可编程处理单元中的一或多个者中的着色器程序绑定命令、指示GPU 10再现存储在存储器22中的特定数据的绘制命令、指示GPU 10相对于存储在存储器22中的特定数据执行一或多个通用计算内核的通用计算命令, 以及指示GPU 10执行一或多个存储器传送操作的存储器传送命令。

[0100] 响应于接收到状态命令, 命令引擎32可经配置以基于状态命令将GPU 10中的一或多个状态寄存器设定于特定值, 和/或基于状态命令配置用于GPU 10的3D再现管线(未图示)中的固定功能处理单元中的一或多个者。响应于接收到绘制调用命令, 命令引擎32可经配置以致使GPU 10中的3D再现管线(未图示)从存储器22检索3D几何形状, 且将所述3D几何形状数据再现为2D图形图像。响应于接收到着色器程序绑定命令, 命令引擎32可将一或多个特定着色器程序加载到3D再现管线中的可编程处理单元中的一或多个者中。

[0101] 在一些实例中, 响应于接收到指令GPU 10执行存储器传送操作的存储器传送命令, 命令引擎32可选择将用于执行存储器传送操作的全部或部分的存储器传送模式, 且致使GPU 10基于所述选定存储器传送模式执行存储器传送操作。在一些实例中, 存储器传送模式可选自包含交错存储器传送模式和顺序存储器传送模式的至少两个存储器传送模式的集合。

[0102] 在额外实例中, 命令引擎32可经配置以根据本发明中描述的技术中的任一者选择

将用于存储器传送操作的全部或部分的存储器传送模式。举例来说,命令引擎32可基于存储器传送操作的大小(例如,存储器传送操作将传送的数据的量)而选择存储器传送模式。作为另一实例,命令引擎32可基于针对存储器传送操作界定的目的地区与固定屏幕对准缓冲区具有的重叠的量而选择存储器传送模式。作为另一个实例,命令引擎32可基于着色器程序的执行是否可能为存储器传送操作的性能瓶颈的确定而选择存储器传送模式。在额外实例中,GPU 10可使用一或多个着色器程序执行存储器传送操作,且命令引擎32可基于用于执行存储器传送操作的着色器程序的大小和/或用于执行存储器传送操作的着色器程序的复杂性而选择存储器传送模式。

[0103] 在其它实例中,命令引擎32可基于从CPU 6接收的指示命令引擎32使用特定存储器传送模式的一或多个命令选择存储器传送模式。在一些情况下,所述一或多个命令可包含指示GPU 10设定GPU 10中的存储器传送模式状态变量的状态的状态命令。在此些情况下,命令引擎32可基于存储器传送模式状态变量的当前状态选择存储器传送模式。存储器传送模式状态变量可(例如)存储在GPU 10中的寄存器中。在额外情况中,所述一或多个命令可为指示GPU 10执行存储器传送操作或某个其它操作的命令,且所述命令可包含指示当执行存储器传送操作或其它操作时将使用的存储器传送操作的旗标或其它信息。

[0104] 为了基于选定存储器传送操作模式执行存储器传送操作,命令引擎32可基于所述选定存储器传送模式配置GPU 10中用于执行存储器传送操作的一或多个组件。举例来说,命令引擎32可如下文进一步详细描述基于所述选定存储器传送模式配置缓冲器36、读取引擎38和写入引擎40中的一或多个者。

[0105] 光栅操作单元34经配置以基于源数据执行一或多个光栅操作以产生目的地数据。所述源数据可对应于存储器传送操作的一或多个源区。在一些实例中,光栅操作单元34可从缓冲器36和/或从写入引擎40接收源数据。目的地数据可写入到存储器传送操作的目的地区。

[0106] 由光栅操作单元34执行的光栅操作的类型可例如由命令引擎32配置。光栅操作类型可指定如何基于用于特定存储器传送操作的源数据产生目的地数据。在其中光栅操作单元34组合两个不同源区以产生目的地数据(例如,组构操作)的情况下,光栅化操作类型可指定如何组合所述两个不同源区以产生目的地数据。在一些实例中,光栅操作类型可为逐位运算,其指定如何基于源区中的对应像素位置产生用于目的地区中的特定像素位置的目的地数据。源区中的对应像素位置可为在其相应源区的相对位置与在目的地区中相同的像素位置。逐位运算可包含(例如)布尔型运算,例如逐位AND、OR、NOT和/或XOR运算。逐位运算还可包含位复制操作、位反转操作、实心填充操作等。光栅操作可为指定针对特定组构操作应如何组合位的操作。在一些实例中,光栅操作可包含逐位运算,例如布尔型逻辑运算。布尔型逻辑运算可包含例如逐位AND、OR、NOT和/或XOR运算。

[0107] 在额外实例中,光栅操作类型可包含掺合操作(例如, α 掺合操作)。在一些实例中,掺合操作可指定在掺合操作的执行期间将使用的源像素乘数和目的地像素乘数。在此些实例中,在掺合操作的执行期间,源像素值(例如,源像素颜色和/或源 α 值)可乘以源像素乘数以产生第一乘积,对应目的地像素值(例如,目的地像素颜色和/或目的地 α 值)可乘以目的地像素乘数以产生第二乘积,且所述两个乘积的总和可加在一起以产生像素位置的目的地值。其它类型的掺合操作也是可能的。

[0108] 在一些实例中,命令引擎32可基于由GPU 10从CPU 6接收的存储器传送操作命令中包含的数据配置光栅操作单元34。举例来说,命令引擎32可基于存储器传送操作命令中指定的光栅操作的类型选择将由光栅操作单元34执行的光栅操作的类型,且使用所述选定光栅化操作的类型配置光栅操作单元34以执行存储器传送操作。

[0109] 虽然本文中已将光栅操作单元34描述为组合不同源区以产生用于存储器传送操作的目的地数据,但在其它实例中,除了光栅操作单元34之外或替代于光栅操作单元34可使用GPU 10中的一或多个其它组件执行存储器传送操作。举例来说,在GPU 10的着色器单元(未图示)上执行的着色器程序可根据着色器程序中指定的组合技术组合不同源区以产生用于存储器传送操作的目的地数据。

[0110] 缓冲器36经配置以存储在存储器传送操作的执行期间从存储器22检索的源数据。在一些实例中,缓冲器36可实施为GPU 10内的多个寄存器。缓冲器36可利用任何多种缓冲方案,包含(例如)先进先出(FIFO)缓冲方案。根据FIFO缓冲方案,存储在缓冲器36中的源数据以源数据放置到缓冲器36中的次序由消费者(例如,写入引擎40)消耗。

[0111] 在一些实例中,缓冲器36可产生指示缓冲器36是否完整的信号和指示缓冲器36是否为空的信号。在此些实例中,这些信号可用以在存储器传送操作的执行期间同步读取引擎38和/或写入引擎40。在额外实例中,缓冲器36可产生指示当前存储在缓冲器36中的源数据是否大于或等于一或多个阈值或小于或等于一或多个阈值的信号,且除了完整和空的信号之外或替代于完整和空的信号可使用这些信号来在存储器传送操作的执行期间同步读取引擎38和/或写入引擎40。

[0112] 在一些实例中,缓冲器36可为根据相对于下伏存储器22的命中/未中高速缓冲存储范例操作的高速缓冲存储器。在此些实例中,缓冲器36在一些实例中在高速缓冲存储线大小可为可配置的意义可为可配置高速缓冲存储器。高速缓冲存储线可指代响应于高速缓冲存储器未中和/或响应于高速缓冲存储器回写而在下伏存储器(例如,存储器22)与高速缓冲存储器(例如,缓冲器36)之间传送的数据单元。高速缓冲存储线的大小可指代在下伏存储器与高速缓冲存储器之间传送的数据单元的大小。当高速缓冲存储线从下伏存储器传送到高速缓冲存储器中时,高速缓冲存储线可放置于高速缓冲存储器条目中且与其对应。因此,高速缓冲存储线的大小还可指代高速缓冲存储器中的单个高速缓冲存储器条目的大小。

[0113] 在其中缓冲器36是可配置高速缓冲存储器的实例中,命令引擎32可基于选定存储器传送模式选择且配置缓冲器36的高速缓冲存储线大小。举例来说,命令引擎32可基于选定存储器传送模式选择用于缓冲器36的高速缓冲存储线大小,且根据所述选定高速缓冲存储线大小配置缓冲器36以操作。

[0114] 在一些实例中,命令引擎32可响应于顺序存储器传送模式被选择为用于执行存储器传送操作的存储器传送模式而选择等于缓冲器36的整个容量的用于缓冲器36的高速缓冲存储线大小。在一些情况下,这可允许连续读取操作的分组和连续写入操作的分组在顺序存储器传送模式期间尽可能大。在此些实例中,缓冲器36可响应于交错存储器传送模式被选择为用于执行存储器传送操作的存储器传送模式而选择小于高速缓冲存储器的整个容量的用于缓冲器36的高速缓冲存储线大小。这可允许当服务于交错请求时同时高速缓冲存储下伏存储器中的数据的数据的多个区,这在一些情况下可减少高速缓冲存储器未中率且改善

交错模式的效率。

[0115] 读取引擎38经配置以发出读取请求到存储器子系统8且将响应于读取请求所接收的源数据放置到缓冲器36中。读取引擎38可经配置以根据多个同步模式中的一者而操作。用以控制读取引擎38的操作的同步模式可界定读取请求的发出与写入引擎40的写入请求的发出经同步的方式和/或读取请求的发出与缓冲器36的状态经同步的方式。读取引擎38使用的同步模式可包含对应于交错存储器传送操作的交错同步模式 and 对应于顺序存储器传送操作的顺序同步模式。

[0116] 在一些实例中,命令引擎32可基于选定存储器传送模式配置将由读取引擎38使用的同步模式以用于存储器传送操作的全部或部分的执行。举例来说,命令引擎32可基于选定存储器传送模式选择同步模式,且配置读取引擎38以根据所述选定同步模式而操作。作为另一实例,命令引擎32可响应于交错存储器传送模式被选择为用于执行存储器传送操作的存储器传送模式而选择将由读取引擎38使用的交错同步模式,且响应于顺序存储器传送模式被选择为用于执行存储器传送操作的存储器传送模式而选择将由读取引擎38使用的顺序同步模式。

[0117] 当根据交错同步模式操作时读取引擎38可响应于存储器传送操作的起始而发出初始数目的读取请求。在已经发出初始数目的读取请求之后,读取引擎38可响应于缓冲器36正消耗足以发出用于存储器传送操作的写入请求的源数据而发出额外读取请求。读取引擎38可以此方式继续发出读取请求直到已经发出所有读取请求用于存储器传送操作。响应于响应于特定读取请求接收到源数据单元,读取引擎38可将源数据单元放置到缓冲器36中和/或使源数据单元直接通过写入引擎40和/或光栅操作单元34用于进一步处理。

[0118] 足以发出用于存储器传送操作的写入请求的数据可指代对应于计算目的地中的单个位置的目的地数据需要的所有源操作数的数据。举例来说,对于存储器复制操作,足以发出用于存储器传送操作的写入请求的数据可指代对应于存储器复制操作的源区中的单个位置的数据。作为另一实例,对于组构操作,足以发出用于存储器传送操作的写入请求的数据可指代对应于用于存储器复制操作的第一源区中的单个位置的数据以及对应于用于存储器复制操作的第二源区中的单个对应位置的数据。对应位置可指代在不同区中具有相同的相对位置的位置。举例来说,两个不同区中的第二行的第三列即使所述区位于其相应表面的不同部分中也具有相同的相对位置。

[0119] 当根据顺序同步模式操作时,读取引擎38可开始响应于存储器传送操作的起始而发出读取请求。读取引擎38可继续发出读取请求直到发出的读取请求的数目大于或等于第一阈值。一旦发出的读取请求的数目大于或等于第一阈值,读取引擎38便可停止发出读取请求直到存储在缓冲器36中的未消耗源数据(例如,尚未由写入引擎40消耗的数据)的量小于或等于第二阈值。一旦存储在缓冲器36中的未消耗源数据的量小于或等于第二阈值,读取引擎38便可恢复发出读取请求。读取引擎38可以此方式继续停止和恢复发出读取请求直到已经针对存储器传送操作发出所有读取请求。第一阈值可大于第二阈值。

[0120] 在一些实例中,第一阈值可基于缓冲器36的大小。举例来说,在一些实例中第一阈值可等于缓冲器36每次能够存储的读取数据单元的最大数目,其中读取数据单元对应于响应于单个读取请求返回的源数据。在此些实例中,读取引擎38可发出“恰好足够的”读取请求以用响应于所述请求返回的源数据填满缓冲器36。在一些实例中,读取引擎38可跟踪针

对读取请求的给定突发已经发出多少读取请求以确定是否已经发出阈值数目的读取请求。在额外实例中,读取引擎38可监视当前存储在缓冲器36中的源数据的量以确定是否已经发出阈值数目的读取请求。举例来说,读取引擎38可利用由缓冲器36产生的指示缓冲器36是否为满的信号或由缓冲器36产生的指示存储在缓冲器36中的数据的量是否大于或等于阈值的信号来确定是否已经发出阈值数目的读取请求。

[0121] 在其它实例中,第二阈值可等于0。也就是说,读取引擎38可停止发出读取请求直到缓冲器36中不存在更多未消耗数据。换句话说,读取引擎38可停止发出读取请求直到缓冲器36为空的。一旦缓冲器36为空的,则读取引擎38可恢复发出读取请求以填满缓冲器。在一些实例中,读取引擎38可利用由缓冲器36产生的指示缓冲器36是否为空的信号或由缓冲器36产生的指示缓冲器36中未消耗源数据的量是否小于或等于阈值的信号来确定未消耗数据是否小于或等于第二阈值。

[0122] 由缓冲器36发出的读取请求可指定返回到GPU 10的数据应来自的存储器22内的存储器地址或存储器地址范围。由存储器子系统8响应于单个读取请求而返回的数据可被称为源数据单元。在一些实例中,由读取引擎38发出的每一读取请求可对应于用于存储器传送操作的源区中的多个位置中的相应一者,且响应于读取请求返回的源数据单元可对应于与源区中所述多个位置中的所述相应一者相关联的数据。在此些实例中,对于以单个源表面的存储器复制操作,读取引擎38可针对目的地表面中的每一位置发出单个读取请求。对于以多个源表面的组构操作,读取引擎38可针对目的地表面中的位置中的每一者发出多个读取请求,例如每一源表面中的对应位置一个读取请求。

[0123] 写入引擎40经配置以从缓冲器36消耗源数据,基于消耗的源数据产生目的地数据,且发出写入请求到存储器子系统8以便将目的地数据写入到存储器22中的目的地。在一些实例中,写入引擎40可产生目的地数据以使得目的地数据等同于源数据。此些实例可用以执行(例如)存储器复制操作。在其它实例中,写入引擎40可组合来自至少两个不同源区的源数据以产生目的地数据。此些实例可用以执行(例如)组构操作。在此些实例中,写入引擎40可在一些实例中使用光栅操作单元34执行光栅操作以便组合源数据且产生目的地数据。

[0124] 写入引擎40可经配置以根据多个同步模式中的一者操作。用以控制写入引擎40的操作的同步模式可界定写入请求的发出与读取引擎38的读取请求的发出经同步的方式和/或写入请求的发出与缓冲器36的状态经同步的方式。写入引擎40使用的同步模式可包含对应于交错存储器传送操作的交错同步模式 and 对应于顺序存储器传送操作的顺序同步模式。

[0125] 在一些实例中,命令引擎32可基于选定存储器传送模式配置将由写入引擎40使用的同步模式以用于存储器传送操作的全部或部分的执行。举例来说,命令引擎32可基于选定存储器传送模式选择同步模式,且配置写入引擎40以根据所述选定同步模式而操作。作为另一实例,命令引擎32可响应于交错存储器传送模式被选择为用于执行存储器传送操作的存储器传送模式而选择将由写入引擎40使用的交错同步模式,且响应于顺序存储器传送模式被选择为用于执行存储器传送操作的存储器传送模式而选择将由写入引擎40使用的顺序同步模式。

[0126] 当根据交错同步模式操作时,响应于存储器传送操作的起始,写入引擎40可初始地等待直到足以发出用于存储器传送操作的写入请求的源数据已由GPU 10接收和/或放置

到缓冲器36中。一旦源数据已由GPU 10接收和/或放置到缓冲器36中,写入引擎40便可消耗源数据,基于源数据产生目的地数据,且发出写入请求到存储器子系统8以将目的地数据写入到存储器22。在已发出写入请求之后,写入引擎40可确定是否任何额外源数据已由GPU 10接收和/或放置到缓冲器36中。如果是,那么写入引擎40可消耗源数据且发出针对额外源数据的另一写入请求。否则,写入引擎40可等待额外数据由GPU 10接收和/或放置到缓冲器36中。写入引擎40可以此方式继续消耗源数据直到存储器传送操作完成。

[0127] 在一些实例中,消耗源数据可指代写入引擎40直接从读取引擎38接收源数据且处理源数据。在其它实例中,消耗源数据可指代写入引擎40从缓冲器36检索源数据且处理源数据。处理源数据可包含基于源数据产生目的地数据。在已经处理数据之后,写入引擎40可发出写入请求以写入目的地数据。未消耗源数据可指代已放置到缓冲器36中但尚未由写入引擎40消耗的源数据。未消耗源数据的量可指代当前存储在缓冲器36中的未消耗源数据单元的数目,其中每一源数据单元对应于响应于单个读取请求而返回的源数据。源数据单元可替代地被称作读取数据字。

[0128] 当根据顺序同步模式操作时,响应于存储器传送操作的起始,写入引擎40可初始地等待直到缓冲器36中的未消耗数据的量大于或等于第一阈值。在等待周期期间,写入引擎40可不消耗任何源数据或发出针对存储器传送操作的任何写入请求。一旦缓冲器36中未消耗数据的量大于或等于第一阈值,写入引擎40便可开始消耗源数据且发出针对消耗的源数据的写入请求。写入引擎40可继续发出写入请求直到缓冲器36中未消耗数据的量小于或等于第二阈值。一旦缓冲器36中的未消耗数据的量小于或等于第二阈值,写入引擎40便可停止消耗源数据且发出针对消耗的源数据的写入请求直到缓冲器36中未消耗数据的量大于或等于第一阈值。在缓冲器36中未消耗数据的量大于或等于第一阈值之后,写入引擎40可再次开始消耗源数据且发出写入请求。写入引擎40可以此方式继续停止和恢复消耗源数据且发出写入请求直到针对存储器传送操作已经发出所有写入请求。第一阈值可大于第二阈值。

[0129] 在一些实例中,第一阈值可基于缓冲器36的大小。举例来说,在一些实例中第一阈值可等于缓冲器36每次能够存储的读取数据单元的最大数目,其中读取数据单元对应于响应于单个读取请求返回的源数据。在一些实例中,写入引擎40可监视当前存储在缓冲器36中的未消耗源数据的量以确定存储在缓冲器36中的消耗的数据的量是否大于或等于第一阈值。举例来说,写入引擎40可利用由缓冲器36产生的指示缓冲器36是否为满的信号或由缓冲器36产生的指示存储在缓冲器36中的数据的数据的量是否大于或等于阈值的信号来确定存储在缓冲器36中的消耗的数据的量是否大于或等于第一阈值。

[0130] 在其它实例中,第二阈值可等于0。也就是说,写入引擎40可当缓冲器36中不存在更多未消耗数据时停止消耗源数据和发出写入请求。换句话说,写入引擎40可在一旦缓冲器36为空时便停止消耗源数据和发出写入请求。在一些实例中,写入引擎40可利用由缓冲器36产生的指示缓冲器36是否为空的信号或由缓冲器36产生的指示缓冲器36中未消耗源数据的量是否小于或等于阈值的信号来确定未消耗数据是否小于或等于第二阈值。

[0131] 由写入引擎40发出的写入请求可指定指定数据将写入到的存储器22内的存储器地址或存储器地址范围。待写入的数据可结合写入请求由写入引擎40经由一或多个数据总线供应到存储器控制器20。在一些实例中,待写入数据可形成写入请求的部分。写入请求的

存储器地址或存储器地址范围可对应于目的地区中的相对位置,所述相对位置与用以产生目的地数据的源数据从其获得的一或多个源区中的相对位置相同。

[0132] 存储器子系统8包含存储器22。存储器22可存储一或多个源表面42和一目的地表面44。源表面42和目的地表面44中的每一者对应于像素数据的二维阵列。像素数据可包含红、绿、蓝(RGB)颜色值,且在一些情况下可包含用于表面中的每一像素位置的 α 值(A)。因为源表面42和目的地表面44对应于像素数据的二维阵列,所以表面中的个别像素位置的像素数据可通过行和列坐标而寻址。在存储器传送操作中使用的源区中的每一者可对应于相应源表面的全部或子集,且目的地区可对应于目的地表面的全部或子集。源和目的地区可为相同的大小和形状,且通常但不一定是矩形形状区。虽然图2描绘单独的源表面42和目的地表面44。在一些实例中,目的地表面44可用作用于存储器传送操作的源表面和/或源表面42中的一或多者可充当用于存储器传送操作的目的地表面。存储器子系统8经由总线18以通信方式耦合到CPU 6、GPU 10、读取引擎38和写入引擎40。

[0133] 响应于接收到存储器存取请求,存储器子系统8可致使相对于存储器22执行所请求存储器操作。举例来说,响应于接收到读取请求,存储器子系统8可从存储器22中对应于读取请求中指定的一或多个存储器地址的一或多个存储器位置获得(即,读取)存储在存储器22中的数据,且将获得的数据提供到GPU 10(例如,读取引擎38)。作为另一实例,响应于接收到写入请求和对应于写入请求的写入数据,存储器子系统8可致使写入数据存储在存储器22中对应于所述写入请求中指定的一或多个存储器地址的一或多个位置处。在一些实例中,存储器子系统8可对写入引擎40提供写入请求成功地完成的确认。

[0134] 在一些实例中,由GPU 10发出且由存储器子系统8接收的读取和写入请求可呈可由存储器22消耗的格式。在此些实例中,存储器子系统8可将由GPU 10接收的请求直接传递到存储器22以便从存储器22读取数据和/或写入数据到存储器22。在其它实例中,由GPU 10发出且由存储器子系统8接收的读取和写入请求可不呈可由存储器22消耗的格式。在此些实例中,存储器子系统8可基于读取和写入请求产生可由存储器22消耗的一或多个存储器命令,且发出此些命令到存储器22以便从存储器22读取数据和/或写入数据到存储器22。在一些实例中,所述一或多个命令可采取用于读取操作的命令的读取序列和用于写入操作的命令的写入序列的形式。

[0135] 总线18可使用总线结构和总线协议的任何组合来实施,包含共享总线结构和协议、点对点总线结构和协议、单向总线结构和协议以及双向总线结构和协议。在一些实例中,总线18可经配置以包含GPU 10与存储器子系统8之间的双向数据总线,以及从GPU 10到存储器子系统8的单向命令和地址总线。在其它实例中,总线18可经配置以包含从存储器子系统8到GPU 10的单向读取总线、从GPU 10到存储器子系统8的单向写入总线,以及从GPU 10到存储器子系统8的单向命令和地址总线。这些总线配置仅是实例,且也可以使用具有更多或更少类型的总线且呈相同或不同配置的总线结构的其它配置。

[0136] 如上文所论述,为了基于选定存储器传送模式执行存储器传送操作,命令引擎32可基于所述选定存储器传送模式配置缓冲器36、读取引擎38和写入引擎40中的一或多者。在一些实例中,如果所述选定存储器传送模式是交错存储器传送模式,那么命令引擎32可配置读取引擎38和写入引擎40以根据交错同步模式而操作。在此些实例中,如果所述选定存储器传送模式是顺序存储器传送模式,那么命令引擎32可配置读取引擎38和写入引擎40

以根据顺序同步模式而操作。

[0137] 在其中缓冲器36是具有可配置高速缓冲存储器大小的高速缓冲存储器的额外实例中,如果所述选定存储器传送模式是交错存储器传送模式,那么命令引擎32可配置读取引擎38和写入引擎40以根据交错同步模式而操作且配置缓冲器36以使得缓冲器36的高速缓冲存储线大小小于缓冲器36的最大高速缓冲存储线大小。在此些实例中如果所述选定存储器传送模式是顺序存储器传送模式,那么命令引擎32可配置读取引擎38和写入引擎40以根据顺序同步模式而操作且配置缓冲器36以使得缓冲器36的高速缓冲存储线大小等于缓冲器36的最大高速缓冲存储线大小。

[0138] 在一些实例中,当将交错存储器传送模式选择为用于执行存储器传送操作的全部或部分的存储器传送模式时,命令引擎32可配置读取引擎38和写入引擎40以在交错同步模式中操作以便基于生产者-消费者范例而致使读取引擎38发出读取请求且写入引擎40发出写入请求。基于生产者-消费者范例致使读取引擎38发出读取请求且写入引擎40发出写入请求可包含响应于GPU 10从存储器子系统8和/或存储器22接收到足以用于发出写入请求的源数据而致使写入引擎40发出写入请求且响应于写入引擎40从缓冲器36消耗足以用于发出写入请求的数据而致使读取引擎38发出读取请求。

[0139] 在其它实例中,当将顺序存储器传送模式选择为用于执行存储器传送操作的全部或部分的存储器传送模式时,命令引擎32可配置读取引擎38和写入引擎40以在顺序同步模式中操作以便基于相互排斥范例而致使读取引擎38发出读取请求且写入引擎40发出写入请求。基于相互排斥范例致使读取引擎38发出读取请求且写入引擎40发出写入请求可包含在单独的时间间隔期间致使读取引擎38和写入引擎40交替发出多个连续读取请求的群组或多个连续写入请求的群组到存储器子系统8和/或存储器22。

[0140] 在额外实例中,当命令引擎32配置读取引擎38和写入引擎40以在交错同步模式中操作时,每次足以发出用于存储器传送操作的写入请求的数据由GPU 10(例如,读取引擎38)从存储器子系统8(例如,存储器22)接收时,写入引擎40可消耗相应数据而无需等待针对至少一个其它写入请求随后接收额外数据。另一方面,当命令引擎32配置读取引擎38和写入引擎40以在顺序同步模式中操作时,对于存储器传送操作的至少部分,响应于GPU 10(例如,读取引擎38)从存储器子系统8(例如,存储器22)接收到足以发出用于存储器传送操作的写入请求的源数据,写入引擎40可推迟源数据的消耗直到随后接收到足以发出至少一个其它写入请求的额外数据。举例来说,写入引擎40可推迟源数据的消耗直到未消耗数据的量大于第一阈值。

[0141] 在其它实例中,当命令引擎32配置读取引擎38和写入引擎40以在交错同步模式中操作时,每次写入引擎40消耗足以发出用于存储器传送操作的写入请求的数据时,读取引擎38可发出读取请求而无需等待写入引擎40针对至少一个其它写入请求随后消耗额外数据。另一方面,当命令引擎32配置读取引擎38和写入引擎40以在顺序同步模式中操作时,对于存储器传送操作的至少部分,响应于写入引擎40消耗足以发出用于存储器传送操作的写入请求的数据,读取引擎38可推迟后续读取请求的发出直到写入引擎40随后消耗足以发出至少一个其它写入请求的数据。举例来说,读取引擎38可推迟后续读取请求的发出直到未消耗数据的量小于或等于第二阈值。

[0142] 在更多实例中,当将交错存储器传送模式选择为用于执行存储器传送操作的全部

或部分的存储器传送模式时,命令引擎32可配置读取引擎38和写入引擎40以在交错同步模式中操作,且发出命令到读取引擎38和/或写入引擎40以起始存储器传送操作的执行。响应于接收到起始存储器传送操作的执行的命令,读取引擎38可发出一或多个读取请求。读取引擎38可将响应于所述多个读取请求从存储器子系统8(例如,存储器22)接收的源数据放置到缓冲器36中。对于所述多个读取请求中的每一者,响应于针对所述多个读取请求中的相应一者将源数据放置于缓冲器36中,写入引擎40可消耗放置于缓冲器36中的源数据且基于消耗的源数据而发出写入请求。举例来说,写入引擎40可基于消耗的源数据而产生目的地数据且发出写入请求以将目的地数据写入到用于存储器传送操作的目的地地区中的对应位置。对于多个写入请求中的每一者,响应于针对写入请求中的相应一者从缓冲器36消耗源数据,读取引擎38可发出后续读取请求。

[0143] 在一些情况下,读取引擎38可监视缓冲器36以确定何时已由写入引擎40从缓冲器36消耗数据。在额外情况中,读取引擎38可从缓冲器36和/或从写入引擎40接收通知以确定何时已由写入引擎40从缓冲器36消耗数据。在一些情况下,写入引擎40可监视缓冲器36以确定读取引擎38何时已将数据放置到缓冲器36中。在额外情况中,写入引擎40可从缓冲器36和/或从读取引擎38接收通知以确定读取引擎38何时已将数据放置到缓冲器36中。

[0144] 在额外实例中,当将顺序存储器传送模式选择为用于执行存储器传送操作的全部或部分的存储器传送模式时,命令引擎32可配置读取引擎38和写入引擎40以在顺序同步模式中操作,且发出命令到读取引擎38和/或写入引擎40以起始存储器传送操作的执行。响应于接收到起始存储器传送操作的执行的命令,读取引擎38可发出多个读取请求。读取引擎38可将响应于所述多个读取请求而从存储器子系统8(例如,存储器22)接收的源数据放置到缓冲器36中。读取引擎38可响应于检测到缓冲器36是满的而停止发出读取请求。响应于检测到缓冲器36是满的,写入引擎40可消耗缓冲器中的源数据且基于消耗的数据而发出多个写入请求。响应于检测到缓冲器36是空的,写入引擎40可停止消耗缓冲器36中的源数据且发出写入请求。读取引擎38可响应于检测到缓冲器36是空的而恢复读取请求的发出。

[0145] 在一些情况下,读取引擎38和/或写入引擎40可监视缓冲器36以确定何时缓冲器36是满的或空的。在额外情况中,读取引擎38和/或写入引擎40可从缓冲器36和/或写入引擎40接收通知以确定何时缓冲器36是满的或空的。

[0146] 虽然本文中命令引擎32描述为执行存储器传送模式选择和/或基于存储器传送模式选择配置GPU 10中的其它处理组件,但在其它实例中,GPU 10中的一或多个其它组件可除命令引擎32之外或替代于命令引擎32而执行这些功能。举例来说,专用存储器传送模式选择和配置单元可用以执行上述功能性,或者读取引擎38和写入引擎40可经配置以执行此功能性。

[0147] 在一些实例中,在CPU 6上执行的GPU驱动器28(或在CPU 6上执行的其它程序)可执行本发明的存储器传送模式选择技术中的任一者且将所述选定存储器传送模式提供到GPU 10。GPU 10可随后基于由CPU 6提供到GPU 10的所述选定存储器传送模式而选择用于执行存储器传送的存储器传送模式。

[0148] 在额外实例中,GPU驱动器28可从软件应用程序24(例如,图形应用程序)接收指定将用于执行存储器传送操作的存储器传送模式的存储器传送模式命令。响应于接收到存储器传送模式命令,GPU驱动器28可致使GPU 10基于存储器传送模式命令执行存储器传送操

作。举例来说, GPU驱动器28可发出一或多个命令到GPU 10, 所述命令指定当执行存储器传送操作时将使用的存储器传送模式。

[0149] 图3是说明可在图2中所示的实例配置中使用的实例存储器子系统8的框图。如图3中所示, 存储器子系统8包含存储器控制器20、存储器22和总线46。存储器控制器20可经配置以从GPU 10接收存储器存取请求, 且相对于存储器22服务于所接收的存储器存取请求。所述存储器存取请求可包含读取请求和/或写入请求。由GPU 10发出且由存储器控制器20接收的存储器存取请求可呈可由存储器控制器20消耗的格式。举例来说, 存储器存取请求可符合AMBA AXI规范、AMBA AHB规范或任何其它存储器控制器接口。

[0150] 在一些实例中, 由存储器控制器20接收的存储器存取请求可不呈可由存储器22消耗的格式。在此些实例中, 存储器控制器20可产生可由存储器22消耗的一或多个存储器命令且发出此些命令到存储器22以便从存储器22读取数据和/或写入数据到存储器22。在一些实例中, 如果存储器22是DDR SDRAM, 那么可由存储器22消耗的所述一或多个存储器命令可为与DDR SDRAM接口兼容的命令。

[0151] 在一些实例中, 由存储器控制器20产生的所述一或多个命令可采取用于读取操作的命令的读取序列和用于写入操作的命令的写入序列的形式。举例来说, 如果存储器22是DDR SDRAM, 那么由存储器控制器20产生的所述一或多个存储器命令可采取符合DDR SDRAM接口的命令的读取序列和/或命令的写入序列的形式。命令的读取和写入序列可采取符合DDR SDRAM接口的一或多个控制信号的断言的形式。举例来说, 响应于接收到存储器存取请求, 存储器控制器20可产生用以激活对应于存储器存取请求中指定的地址的存储器22的特定行和组(如果尚未激活)的信号。存储器控制器20还可产生用以选择对应于存储器存取请求中指定的地址的经激活行和组中的特定列的信号。存储器控制器20还可产生用以向存储器22指示是否应从存储器22读取数据或写入数据到存储器22的信号。如果存储器存取请求是写入请求, 那么存储器控制器20可经由总线46内的双向数据总线将写入数据供应到存储器22。如果存储器存取请求是读取请求, 那么存储器控制器20可经由总线46内的双向数据总线从存储器22接收数据且经由总线18将读取数据提供到CPU 14。

[0152] 存储器22经配置以接收读取命令和写入命令, 且相对于存储器22内含有的数据存储空间服务于所述命令。响应于接收到读取命令, 存储器22可从存储器22内含有的数据存储空间在由读取命令中包含的地址指定的位置处获得数据, 且经由总线46内的双向总线将获得的数据提供到存储器控制器20。响应于接收到写入命令, 存储器22可将经由总线46内的双向总线从存储器控制器20接收的数据存储到存储器22内含有的数据存储空间中由写入命令中包含的地址指定的位置处。

[0153] 源表面42和目的地表面44可实质上类似于图2中所示的对应组件。因此, 为了简洁且避免冗余, 将不进一步详细描述这些组件。

[0154] 总线46可实施为任何类型的总线结构且根据任何类型的总线协议操作。在一些实例中, 总线46可包含对应于用以控制存储器22的控制信号的位线。举例来说, 如果存储器22是DDR SDRAM, 那么总线46可包含时钟信号、片选信号、存储器组选择信号、行地址选通、列地址选通、写入启用信号、双向数据总线等中的一或多个。

[0155] 如上文所论述, 总线46可包含双向数据总线。换句话说, 总线46中的相同位线可当执行读取请求时从存储器22传送读取数据到存储器控制器20且可当执行写入请求时从存

存储器控制器20传送写入数据到存储器22。每次存储器控制器20从发出读取请求改变到发出写入请求或从发出写入请求改变到发出读取请求时,在改变方向之前可能需要空闲时间来允许总线46中的双向数据总线稳定。此稳定时间可被称为读取-写入周转时间。

[0156] 本发明中描述的顺序存储器传送模式可用以减少当执行存储器传送操作时发生的读取到写入转变的数目,进而减少在存储器传送操作期间由于读取-写入周转时间而发生的总延迟。虽然顺序存储器传送模式可减少在存储器传送操作期间由于读取-写入周转时间而发生的延迟,但是对于某些类型的存储器传送操作,顺序存储器传送模式可不一定比交错存储器传送模式更有效。举例来说,如果在顺序存储器传送模式期间针对读取操作始终检索相对大的固定量的数据,那么对于较小存储器传送操作,顺序模式可能需要获取额外的数据,这可能超过通过减少读取-写入周转实现的益处。

[0157] 由于这些和其它考虑,一种类型的存储器传送模式可能不会通用地为用于执行存储器传送操作的最高效存储器传送模式。因此,本发明的技术可用以选择较好地适合于特定类型的存储器传送操作的适当存储器传送模式,进而改善可执行不同类型和大小的存储器传送操作的GPU的性能。

[0158] 图4是说明可在图2中所示的实例配置中使用的另一实例存储器子系统8的框图。如图4中所示,存储器子系统8包含直接耦合到总线18而无介入存储器控制器的存储器22。在此实例中,存储器22可经配置以接收来自GPU 10的存储器存取请求,且相对于存储器22内含有的数据存储空间服务于所接收的存储器存取请求。存储器存取请求可包含读取请求和/或写入请求。

[0159] 在一些实例中,存储器22可包含内建式存储器控制器功能性。在此些实例中,由GPU 10发出且由存储器22接收的存储器存取请求可呈可由存储器控制器消耗的格式。举例来说,存储器存取请求可符合AMBA AXI规范、AMBA AHB规范或任何其它存储器控制器接口。所述内建式存储器控制器功能性可将所接收的存储器存取请求转换为可由存储器22消耗的一或多个命令。举例来说,如果存储器22是DDR SDRAM,那么可由存储器22消耗的所述一或多个存储器命令可为与DDR SDRAM接口兼容的命令。

[0160] 在其它实例中,GPU 10中的读取引擎38和写入引擎40可将可由存储器22消耗的命令发出到存储器子系统8和存储器22。举例来说,如果存储器22是DDR SDRAM,那么可由存储器22消耗的所述一或多个存储器命令可为与DDR SDRAM接口兼容的命令。

[0161] 在额外实例中,GPU 10可包含内建式存储器控制器功能性。在此些实例中,由读取引擎38和写入引擎40发出的存储器存取请求可不一定呈可由存储器22消耗的格式。举例来说,存储器存取请求可符合AMBA AXI规范、AMBA AHB规范或任何其它存储器控制器接口。GPU 10中的所述内建式存储器控制器功能性可将所接收的存储器存取请求转换为可由存储器22消耗的一或多个命令。举例来说,如果存储器22是DDR SDRAM,那么可由存储器22消耗的所述一或多个存储器命令可为与DDR SDRAM接口兼容的命令。

[0162] 在一些实例中,总线18可包含双向数据总线。本发明中描述的顺序存储器传送模式可用以减少当执行存储器传送操作时发生的读取到写入转变的数目,进而减少在存储器传送操作期间由于读取-写入周转时间而发生的总延迟。然而,如上文已经论述,顺序存储器传送模式可能不是用于所有类型和大小的存储器传送操作的最高效存储器传送模式。因此,本发明的技术可用以选择较好地适合于特定类型的存储器传送操作的适当存储器传送

模式,进而改善可执行不同类型和大小的存储器传送操作的GPU的性能。

[0163] 图5是说明根据本发明的实例单个源存储器传送操作的概念图。图5描绘源表面50和目的地表面52,其中的每一者对应于像素数据的二维阵列。像素数据可包含红、绿、蓝(RGB)颜色值,且在一些情况下,包含用于表面中的每一像素的 α 值(A)。因为表面50、52中的每一者对应于像素数据的二维阵列,所以表面中的个别像素位置的像素数据可通过行和列坐标来编索引。在一些实例中,源表面50和目的地表面52可为位图。在其它实例中,源表面50和目的地表面52可为缓冲器资源,例如纹理缓冲器。图5中的实例源表面50和目的地表面52是相同的形状和大小,且是矩形形状的。然而,在其它实例中,源表面50和目的地表面52可能不一定是相同的形状和大小和/或可为其它形状。

[0164] 源表面50包含源区54,且目的地表面52包含目的地地区56。源区54包含源表面50中包含的像素数据的子集,且目的地地区56包含目的地表面52中包含的像素数据的子集。表面中包含的像素数据的子集可指代表面中包含的像素数据的全部或少于全部。如图5中示出,源区54和目的地地区56是相同的大小和形状,且是矩形形状的。然而在其它实例中,源区54和目的地地区56可不是矩形形状的。

[0165] 在图5中所示的实例存储器传送操作中,数据从源表面50的源区54传送到目的地表面52的目的地地区56。一般来说,源区54中的每一像素位置的数据传送到目的地地区56中的对应像素位置。如果像素位置中的每一者在其相应区中的相对位置是相同的,那么不同区中的不同像素位置可称为对应像素位置。举例来说,如果源区54中的像素位置在源区54中具有与目的地地区56中的像素位置在目的地地区56中的相对位置相同的相对位置,那么源区54中的像素位置可对应于目的地地区56中的像素位置。举例来说,源区54和目的地地区56中的第二行的第三列具有相同的相对位置。

[0166] 图5中所示的存储器传送操作是单个源存储器传送操作,因为单个源区用作用于存储器传送操作的源操作数。图5中所示的单个源存储器传送操作可表示存储器复制操作,其中源区54的相同数据复制到目的地地区56中。虽然图5中所示的存储器传送操作展示单独的源表面50和目的地表面52,但在其它实例中,源表面50和目的地表面52可为相同表面。在此些实例中,源区54和目的地地区56在一些实例中可为同一表面内通过存储器复制操作在其之间传送数据的不同区。

[0167] 图6是说明根据本发明的实例多个源存储器传送操作的概念图。图6描绘第一源表面58、第二源表面60和目的地表面62。表面58、60、62中的每一者可实质上类似于上文相对于图5所论述的表面50、52。第一源表面58包含第一源区64,第二源表面60包含第二源区66,且目的地表面62包含目的地地区68。第一源区64包含第一源表面58中包含的像素数据的子集,第二源区66包含第二源表面60中包含的像素数据的子集,且目的地地区68包含目的地表面62中包含的像素数据的子集。如图5中示出,第一源区64、第二源区66和目的地地区68是相同的大小和形状,且是矩形形状的。然而在其它实例中,区64、66和68可不是矩形形状的。

[0168] 在图6的实例存储器传送操作中,来自第一源区64的源数据和来自第二源区66的源数据经由光栅操作70组合以产生目的地数据,且将目的地数据放置到目的地表面62的目的地地区68中。一般来说,第一源区64中的每一像素位置的数据与来自第二源区66中的对应像素位置的数据组合,且写入到目的地地区68中的对应像素位置。如果像素位置中的每一者在其相应区中的相对位置是相同的,那么不同区中的不同像素位置可称为对应像素位置。

[0169] 图6中所示的存储器传送操作是多源存储器传送操作,因为多个源区作用于存储器传送操作的源操作数。图6中所示的多源存储器传送操作可表示组构操作,其中来自两个不同源区的数据组合以产生写入到目的地区的目的地数据。虽然图6中所示的存储器传送操作展示单独的第一源表面58、第二源表面60和目的地表面62,但在其它实例中,表面58、60、62中的一或多者可为相同表面。举例来说,第二源表面60和目的地表面62可为相同表面,且第二源区66和目的地区68可为所述表面内的相同区。在此实例中,组构操作可使用特定光栅算子合并第一源区64中的像素数据与目的地表面62中的现有像素数据。

[0170] 图7是说明根据本发明的实例GPU辅助的单个源存储器传送操作的概念图。图7中所示的GPU辅助的存储器传送操作可为与图5中所描绘的存储器传送操作相同的存储器传送操作。如图7中示出,源表面50和目的地表面52存储在存储器22中。

[0171] 为了执行存储器传送操作,GPU 10从源区54读取源数据,基于从源区54读取的源数据产生目的地数据,且将目的地数据写入到目的地区56。目的地数据可与源数据相同。

[0172] 对于源区54中的每一像素位置,GPU 10可发出读取请求以从对应于相应像素位置的源区54读取源数据。响应于接收到读取请求,存储器22可将对应于相应像素位置的源数据提供到GPU 10。在一些实例中,GPU 10可将源数据存储在GPU 10中的缓冲器中。GPU 10可基于相应像素位置的源数据产生相应像素位置的目的地数据。对于存储器复制操作,相应像素位置的目的地数据可与相应像素位置的源数据相同。GPU 10可发出写入请求以将目的地数据写入到目的地区56中对应于源区54的相应像素位置的像素位置。响应于接收到写入请求,存储器22可将目的地数据存储在目的地区56中对应于源区54的相应像素位置的像素位置。GPU 10可针对源区54中的每一像素位置和目的地区56中的对应像素位置执行这些操作。

[0173] 图8是说明根据本发明的实例GPU辅助的多源存储器传送操作的概念图。图8中所示的实例GPU辅助的存储器传送操作可为与图6中所描绘的存储器传送操作相同的存储器传送操作。如图8中示出,第一源表面58、第二源表面60和目的地表面62存储在存储器22中。

[0174] 为了执行存储器传送操作,GPU 10从第一源表面58和第二源表面60读取源数据,基于从第一源表面58和第二源表面60读取的源数据产生目的地数据,且将目的地数据写入到目的地表面62。目的地数据可基于源数据的组合。举例来说,GPU 10可基于指定来自第一源表面58和第二源表面60的源数据将组合的方式的光栅算子来产生目的地数据。GPU 10的光栅操作单元(例如,图2中所示的光栅操作单元34)可用以组合所述数据。

[0175] 对于第一源区64中的每一像素位置,GPU 10可发出读取请求以从第一源区64读取对应于相应像素位置的第一源数据,且发出读取请求以从第二源区66中对应于第一源区64中的相应像素位置的像素位置(即,第二源区66中具有与第一源区64中的相应像素位置相同的相对位置的像素位置)读取第二源数据。响应于接收到读取请求,存储器22可将对应于第一源区64中的相应像素位置的第一源数据提供到GPU 10,且将对应于第二源区66中的对应像素位置的第二源数据提供到GPU 10。在一些实例中,GPU 10可将源数据存储在GPU 10中的缓冲器中。GPU 10可基于相应像素位置的第一源数据和第二源数据产生相应像素位置的目的地数据。对于组构操作,源数据可根据光栅操作组合以产生目的地数据。GPU 10可发出写入请求以将目的地数据写入到目的地区68中对应于第一源区64的相应像素位置的像素位置。GPU 10可针对第一源区64、第二源区66和目的地区68中的对应像素位置的每一集

合执行这些操作。

[0176] 图9到12是说明根据本发明的各种存储器传送模式执行的各种存储器传送操作的实例时序特性的时序图。出于示范性目的,图9到12中的时序图是相对于图3中所示的计算装置2的配置而描述。然而应理解,在相同或不同配置中具有相同或不同组件的其它装置可用以执行本发明的技术。

[0177] 应注意,仅提供图9到12中呈现的时序图以描绘涉及在各种存储器传送模式中发出读取请求和写入请求以及当使用不同存储器传送模式时可发生的读取-写入周转延迟的变化本发明的概念。因此应理解,这些时序图并不希望为当存取存储器时可发生的所有时序考虑的物理上严格或穷尽性的模型。类似地,不同存储器控制器和/或存储器可以不同于图9到12的时序图中所描绘的方式的方式响应于读取请求和写入请求。因此应理解,本发明的技术不限于图9到12中所描绘的特定存储器控制器行为和/或存储器行为。

[0178] 图9A和9B是说明根据本发明的交错存储器传送模式执行的单个源存储器传送操作的实例时序特性的时序图。如上文所论述,图9A和9B中所描绘的存储器传送操作是以图3中说明的计算装置2的配置执行。如图3中所展示,所述配置包含存储器控制器20,其经由总线18以通信方式耦合到GPU 10且经由总线46以通信方式耦合到存储器22。为了根据交错存储器传送模式执行存储器传送操作,命令引擎32可配置读取引擎38和写入引擎40以在交错同步模式中操作。图9A和9B中所描绘的存储器传送操作是相对于各自具有十二个像素的源区和目的地区执行。因此,在存储器传送操作期间,相对于存储器22执行十二个读取请求和十二个写入请求。

[0179] 在图9A和9B中,每一行表示在存储器传送操作的执行期间的特定时隙或时间间隔,其中第一行是最早的时隙且最后行是最晚的时隙。“时间”列表示时隙编号。

[0180] “读取引擎”列包含“读取”事件,其表示由读取引擎38发出的读取请求。“写入引擎”列包含“写入”事件,其表示由写入引擎40发出的写入请求。读取和写入请求可经由总线18内的一或多个命令和/或地址总线通道从GPU 10发送到存储器控制器20。

[0181] 跟随读取请求的数字列举与源区中的不同像素位置相关联的不同读取请求。跟随写入请求的数字列举与目的地区中的不同像素位置相关联的不同写入请求。具有相同数字的读取请求和写入请求指示读取请求和写入请求是相对于源和目的地区中的对应像素位置(即,具有相同相对位置的像素)而执行。换句话说,用于具有特定数字的写入请求的写入数据是基于响应于具有相同特定数字的读取请求而接收的源数据而产生。

[0182] “读取总线”和“写入总线”列表示用于在GPU 10与存储器控制器20之间传送数据的独立数据通道。“读取总线”和“写入总线”可对应于图3中所示的总线18。“读取总线”可为总线18内将读取数据从存储器控制器20输送到GPU 10的单向总线通道。“写入总线”可为总线18内将写入数据从GPU 10输送到存储器控制器20的单向总线通道。

[0183] “读取总线”列包含“读取数据”事件,其表示响应于由读取引擎38发出的特定读取请求从存储器控制器20传送到GPU 10的源数据。跟随“读取数据”事件的数字指示对应于由“读取数据”事件表示的源数据的读取请求。“写入总线”列包含“写入数据”事件,其表示结合由写入引擎40发出的写入请求从GPU 10传送到存储器控制器20的目的地区数据。跟随“写入数据”事件的数字指示对应于由“写入数据”事件表示的目的地区数据的写入请求。

[0184] “存储器控制器”列包含“读取命令”事件和“写入命令”事件,其分别表示从存储器

控制器20发送到存储器22的读取命令和写入命令。所述命令可经由总线46内的一或多个命令和/或地址总线通道发送。跟随“读取命令”事件的数字指示对应于由“读取命令”事件表示的读取命令的读取请求。类似地,跟随“写入命令”事件的数字指示对应于由“写入命令”事件表示的写入命令的写入请求。

[0185] “存储器总线”列表示存储器控制器20与存储器22之间的双向数据总线。所述双向数据总线可为总线46的部分。“存储器总线”列包含“读取数据”事件,其表示响应于由存储器控制器20发出的特定读取命令从存储器22传送到存储器控制器20的源数据,以及“写入数据”事件,其表示结合由存储器控制器20发出的特定写入命令从存储器控制器20传送到存储器22的目的地数据。跟随“读取数据”事件的数字指示对应于由“读取数据”事件表示的源数据的读取命令。跟随“写入数据”事件的数字指示对应于由“写入数据”事件表示的目的地数据的写入命令。

[0186] 图9A和9B中所描绘的存储器传送操作是以支持管线化读取存取和管线化写入存取的存储器架构执行。管线化读取存取可允许GPU 10和/或存储器控制器20响应于所有先前发出的读取请求在接收数据之前发出后续读取请求。换句话说,管线化读取存取可允许GPU 10和/或存储器控制器20发出第一读取请求,且响应于第一读取请求在发出第一读取请求之后且在接收源数据之前发出第二读取请求。

[0187] 管线化写入存取可允许GPU 10和/或存储器控制器20在接收所有先前发出的写入请求已完成的确认之前发出后续写入请求。换句话说,管线化写入存取可允许GPU 10和/或存储器控制器20发出第一写入请求,且在发出第一写入请求之后且在接收到第一写入请求已完成的确认之前发出第二写入请求。

[0188] 存储器传送操作的执行开始于时隙1-4,其中读取引擎38开始发出读取请求。存储器控制器20将每一读取请求转换为读取命令。在时隙2-5期间,存储器控制器20经由总线46(即,存储器总线)将读取命令发出到存储器22。在此实例中,存储器22经配置以在接收到读取命令之后在第二时隙处响应于读取请求而返回读取数据。举例来说,响应于在时隙2接收到“读取命令1”,存储器22返回“读取数据1”,其对应于在时隙4的“读取命令1”。在时隙4-7期间,存储器22将前四个读取请求的读取数据传送到存储器控制器20。在时隙5-8期间,存储器控制器20经由“读取总线”将读取数据重定向到GPU 10。

[0189] 在时隙5,写入引擎40检测足以发出写入请求的源数据已响应于读取请求而由GPU 10接收。在此实例中,足以发出写入请求的源数据对应于响应于单个读取请求而接收的一个源数据单元。在时隙6期间,响应于检测到足以发出写入请求的源数据已由GPU 10接收,写入引擎40消耗所接收的源数据,基于源数据产生目的地数据,发出写入请求到存储器控制器20,且经由“写入总线”将目的地数据传送到存储器控制器20。在一些实例中,目的地数据可等同于源数据。

[0190] 在时隙6,存储器控制器20接收写入请求,但出于至少两个原因而并不发出写入命令到存储器22直到时隙11。首先,存储器控制器20仍在等待响应于先前发出的读取请求而从存储器22接收源数据。第二,如上文所论述,每次服务于读取与写入请求之间发生切换时,在总线改变方向(例如,从读取到写入)之前可能需要特定量的空闲时间来使总线稳定。此空闲时间可被称为读取-写入周转延迟。在图9A和9B的实例中,读取-写入周转延迟是两个空闲循环(即,时隙)。因此,在“读取数据4”事件之后,存储器控制器20将“写入命令1”的发

出延迟两个时隙。结合发出“写入命令1”，存储器控制器20在存储器总线上将“写入数据1”传送到存储器22。

[0191] 返回到间隙6，读取引擎38检测足以发出写入请求的源数据已由写入引擎40消耗。在间隙7期间，响应于检测到足以发出写入请求的源数据已由写入引擎40消耗，读取引擎38发出后续读取请求。也在间隙7期间，写入引擎40响应于GPU 10接收到“读取数据2”而发出“写入2”写入请求。

[0192] 此时，存储器控制器20具有多个不同类型的存储器存取请求要仲裁和区分优先级。对于图9A和9B中所描绘的实例存储器传送操作，存储器控制器20处理在较晚接收的存储器存取请求之前较早接收的存储器存取请求。如果读取请求和写入请求在同一间隙期间接收，那么存储器控制器20在写入请求之前处理读取请求。因此，即使读取请求和写入请求两者在间隙7期间由存储器控制器20接收，存储器控制器20也在处理写入请求之前处理读取请求。这在图9A中展示，其中存储器控制器20在间隙11期间发出“读取命令5”且在间隙16期间发出“写入命令2”。应注意以上规则仅是存储器控制器20中可实施的仲裁和区分优先级方案的一个实例，且在其它实例中也可以使用其它类型的仲裁和区分优先级方案。

[0193] 在间隙11之后，读取引擎38和写入引擎40根据生产者-消费者范例继续发出读取请求和写入请求。在源区中的全部十二个像素已经读取且目的地中的全部十二个像素已经写入之后，存储器传送操作完成。

[0194] 如图9A和9B中示出，交错存储器传送模式可致使读取请求和写入请求交错。举例来说，在间隙7-9期间，GPU 10在同一间隙期间发出读取请求和写入请求到存储器控制器20。这致使存储器控制器20如间隙11-28中的“存储器总线”列中示出交错读取请求和写入请求。作为另一实例，在间隙10-39期间，GPU 10以交错方式发出读取请求和写入请求到存储器控制器20，这又致使存储器控制器20在间隙29-55期间以交错方式处理所述请求。由于交错，大量间隙在“存储器总线”上空闲，因为总线频繁地改变方向。

[0195] 图10是说明根据本发明的顺序存储器传送模式执行的单个源存储器传送操作的实例时序特性的时序图。类似于图9A和9B中所描绘的存储器传送操作，图10中的存储器传送操作是以图3中说明的计算装置2的配置执行。同样类似于图9A和9B中所描绘的存储器传送操作，图10中所描绘的存储器传送操作支持管线化读取存取和管线化写入存取。图10中所描绘的表的格式实质上类似于图9A和9B中所示的表，并且因此将不进一步详细描述。

[0196] 为了根据顺序存储器传送模式执行存储器传送操作，命令引擎32可配置读取引擎38和写入引擎40以在顺序同步模式中操作。图10中所描绘的存储器传送操作是相对于各自具有十二个像素的源区和目的地执行。因此，在存储器传送操作期间，相对于存储器22执行十二个读取请求和十二个写入请求。

[0197] 存储器传送操作的执行开始于间隙1-4，其中读取引擎38开始发出读取请求。在间隙5，读取引擎38响应于发出的读取请求的数目大于或等于第一阈值而停止发出读取请求。在此实例中，第一阈值数目是四个读取请求。

[0198] 在间隙8，写入引擎40检测存储在缓冲器36中的未消耗数据的量大于或等于第一阈值。在间隙9，响应于检测到存储在缓冲器36中的未消耗数据的量大于或等于第一阈值，写入引擎40开始从缓冲器36消耗源数据，基于消耗的源数据产生目的地数据，且发出写入请求以将目的地数据写入到存储器22。在间隙13，写入引擎40检测缓冲器36中的未消耗数

据的量小于或等于第二阈值。响应于检测到缓冲器36中的未消耗数据的量小于或等于第二阈值,写入引擎40停止消耗源数据且发出针对消耗的源数据的写入请求。在此实例中,第二阈值是0。

[0199] 在时隙12,读取引擎38检测缓冲器36中的未消耗数据的量小于或等于第二阈值,且在时隙13恢复读取请求的发出。读取引擎38和写入引擎40可根据顺序存储器传送模式继续交替发出连续读取请求的序列和连续写入请求的序列。在源区中的全部十二个像素已经读取且目的地中的全部十二个像素已经写入之后,存储器传送操作完成。

[0200] 如图10中示出,顺序存储器传送模式可致使在读取请求和写入请求的单独群组中发出读取请求和写入请求。举例来说,在时隙1-4期间,GPU 10发出四个连续读取请求的群组,且在时隙9-12期间,GPU 10发出四个连续写入请求的群组。如图10中所展示,发出读取和写入请求的单独群组致使“存储器总线”改变方向的次数相对于图9A和9B中所示的交错存储器传送模式减少。因此,顺序存储器模式可相对于交错存储器传送模式减少当执行存储器传送操作时发生的总读取-写入周转延迟。此情形可通过以下方式见到:观测同一存储器传送操作在图9A和9B中当根据交错传送模式执行时采取61个时隙来完成且在图10中当根据顺序存储器传送模式执行时采取37个时隙来完成。

[0201] 虽然顺序存储器传送模式可减少在存储器传送操作期间由于读取-写入周转时间而发生的延迟,但是对于某些类型的存储器传送操作,顺序存储器传送模式可不一定比交错存储器传送模式更有效。举例来说,如果在顺序存储器传送模式期间针对读取操作始终检索相对大的固定量的数据,那么对于较小存储器传送操作,顺序模式可需要获取额外的数据,这可能超过通过减少读取-写入周转实现的益处。

[0202] 由于这些和其它考虑,一种类型的存储器传送模式可能不会通用地为用于执行存储器传送操作的最高效存储器传送模式。因此,本发明的技术可用以选择较好地适合于特定类型的存储器传送操作的适当存储器传送模式,进而改善可执行不同类型和大小的存储器传送操作的GPU的性能。

[0203] 图11A和11B是说明根据本发明的交错存储器传送模式执行的多源存储器传送操作的实例时序特性的时序图。类似于图9A和9B中所描绘的存储器传送操作,图11A和11B中的存储器传送操作是以图3中说明的计算装置2的配置执行。同样类似于图9A和9B中所描绘的存储器传送操作,图11A和11B中所描绘的存储器传送操作支持管线化读取存取和管线化写入存取。

[0204] 为了根据交错存储器传送模式执行存储器传送操作,命令引擎32可配置读取引擎38和写入引擎40以在交错同步模式中操作。图11A和11B中所描绘的存储器传送操作是相对于两个不同源区和一目的地执行,所述区中的每一者具有八个像素。因此,在存储器传送操作期间,相对于存储器22执行十六个读取请求和八个写入请求。

[0205] 图11A和11B中所描绘的表的格式类似于图9A和9B中所示的表,不同的是读取请求、读取命令和读取数据是通过数字和字母指定(例如,“1A”、“1B”、“2A”、“2B”等)。在此情况下,不同字母表示不同源区,且不同数字表示不同区中的对应像素位置(即,具有相同相对位置的像素)。举例来说,“读取1A”是用于第一源区中的特定像素位置的读取请求,且“读取1B”是用于第二源区中的对应像素位置的读取请求。在此实例中,“写入1”可为将目的地数据写入到目的地中对应于用于“读取1A”和“读取1B”两者的像素位置的像素位置的写

入请求。像素位置的目的地数据可基于与两个源区中的对应像素位置相关联的源数据而产生。举例来说, GPU 10可基于“读取数据1A”和“读取数据1B”而产生“写入数据1”。光栅操作可用以产生目的地数据。

[0206] 读取引擎38和写入引擎40以实质上类似于相对于图9A和9B描述的方式的方式操作,不同的是足以发出用于存储器传送操作的写入请求的数据可为响应于两个不同读取请求返回的两个源数据单元而不是响应于单个读取请求返回的单个源数据单元。举例来说,在时隙6,写入引擎40检测足以发出用于存储器传送操作的写入请求(即,“读取数据1A”和“读取数据1B”)的源数据已由GPU 10接收,且在时隙7期间,响应于检测到足以发出用于存储器传送操作的写入请求的源数据已由GPU 10接收,写入引擎40消耗所接收的源数据,基于消耗的源数据产生目的地数据,发出写入请求到存储器控制器20,且经由“写入总线”将目的地数据传送到存储器控制器20。

[0207] 作为另一实例,在时隙7,读取引擎38检测足以发出用于存储器传送操作的写入请求(即,“读取数据1A”和“读取数据1B”)的源数据已由写入引擎40消耗。在时隙8期间,响应于检测到足以发出用于存储器传送操作的写入请求的源数据已由写入引擎40消耗,读取引擎38发出后续读取请求。

[0208] 图12是说明根据本发明的顺序存储器传送模式执行的多源存储器传送操作的实例时序特性的时序图。类似于图11A和11B中所描绘的存储器传送操作,图12中的存储器传送操作是以图3中说明的计算装置2的配置执行。同样类似于图11A和11B中所描绘的存储器传送操作,图12中所描绘的存储器传送操作支持管线化读取存取和管线化写入存取。图12中所描绘的表的格式实质上类似于图11A和11B中所示的表,并且因此将不进一步详细描述。

[0209] 为了根据顺序存储器传送模式执行存储器传送操作,命令引擎32可配置读取引擎38和写入引擎40以在顺序同步模式中操作。图11A和11B中所描绘的存储器传送操作是相对于两个不同源区和一目的地区执行,所述区中的每一者具有八个像素。因此,在存储器传送操作期间,相对于存储器22执行十六个读取请求和八个写入请求。

[0210] 如图12中所展示,发出读取和写入请求的单独群组致使“存储器总线”改变方向的次数相对于图11A和11B中所示的交错存储器传送模式减少。因此,顺序存储器模式可相对于交错存储器传送模式减少当执行存储器传送操作时发生的总读取-写入周转延迟。此情形可通过以下方式见到:观测同一存储器传送操作在图11A和11B中当根据交错传送模式执行时采取53个时隙来完成且在图12中当根据顺序存储器传送模式执行时采取33个时隙来完成。

[0211] 再次,一种类型的存储器传送模式可能不会通用地为用于执行存储器传送操作的最高效存储器传送模式。因此,本发明的技术可用以选择较好地适合于特定类型的存储器传送操作的适当存储器传送模式,进而改善可执行不同类型和大小的存储器传送操作的GPU的性能。

[0212] 图13到19说明用于根据本发明的交错和顺序存储器传送模式执行存储器传送操作的不同技术。出于示范性目的,图13到19中所示的技术描述为由图1和2中所示的GPU 10执行。在其它实例中,图13到19中说明的技术可在以相同或不同配置具有相同或不同组件的其它系统中实施。

[0213] 图13是说明用于根据本发明的交错存储器传送模式执行存储器传送操作的实例技术的流程图。GPU 10起始存储器传送操作(100)。GPU 10发出读取请求到存储器子系统8(例如,存储器控制器20或存储器22)以读取与源区内的像素位置相关联的源数据(102)。

[0214] GPU 10等待直到已响应于读取请求从存储器子系统8(例如存储器控制器20或存储器22)接收源数据。举例来说,GPU 10确定是否已响应于读取请求从存储器子系统8接收源数据(104)。如果尚未响应于读取请求从存储器子系统8接收源数据,那么GPU10环回到决策框104。响应于从存储器子系统8接收到源数据,GPU 10消耗所述数据,基于所述源数据产生目的地数据,且发出写入请求以将目的地数据写入到目的地区中的对应像素位置(106)。

[0215] 响应于源数据被消耗,GPU 10发出后续读取请求,前提是需要读取更多数据用于存储器传送操作。更具体来说,GPU 10确定是否存在更多数据要读取(108)。如果存在更多数据要读取,那么GPU 10前进到过程框102且发出后续读取请求。如果不存在更多数据要读取,那么GPU 10完成存储器传送操作(110)。

[0216] 图14是说明用于根据本发明的交错同步模式发出读取请求的实例技术的流程图。读取引擎38起始存储器传送操作(112)。读取引擎38发出读取请求到存储器子系统8(例如,存储器控制器20或存储器22)以读取与源区内的像素位置相关联的源数据(114)。

[0217] 读取引擎38等待直到源数据已由写入引擎40从缓冲器36消耗。举例来说,读取引擎38确定源数据是否已由写入引擎40从缓冲器36消耗(116)。如果源数据尚未从缓冲器36消耗,那么GPU 10环回到决策框116。

[0218] 响应于源数据正由写入引擎40从缓冲器36消耗,读取引擎38发出后续读取请求,前提是需要读取更多数据用于存储器传送操作。举例来说,读取引擎38确定是否存在更多数据要读取(118)。如果存在更多数据要读取,那么读取引擎38前进到过程框114且发出后续读取请求。如果不存在更多数据要读取,那么读取引擎38完成存储器传送操作(120)。

[0219] 图15是说明用于根据本发明的交错同步模式发出写入请求的实例技术的流程图。写入引擎40起始存储器传送操作(122)。

[0220] 写入引擎40等待直到已响应于读取请求从存储器子系统8(例如存储器控制器20或存储器22)接收源数据。举例来说,写入引擎40确定源数据是否已响应于读取请求由GPU 10从存储器子系统8接收(124)。举例来说,写入引擎40可确定读取引擎38是否将源数据放置到缓冲器36中。如果源数据尚未响应于读取请求从存储器子系统8接收,那么写入引擎40环回到决策框124。响应于确定源数据已由GPU 10接收,存储器22消耗所述数据,基于源数据产生目的地数据,且发出写入请求以将目的地数据写入到目的地区中的对应像素位置(126)。

[0221] 作为存储器传送操作的部分,写入引擎40确定是否存在更多数据要消耗(128)。如果存在更多数据要消耗,那么写入引擎40前进到决策框124且等待接收更多源数据。如果作为存储器传送操作的部分不存在更多数据要消耗,那么写入引擎40完成存储器传送操作(130)。

[0222] 图16是说明用于根据本发明的顺序存储器传送模式执行存储器传送操作的实例技术的流程图。GPU 10起始存储器传送操作(132)。GPU 10发出读取请求到存储器子系统8(例如,存储器控制器20或存储器22)以读取与源区内的像素位置相关联的源数据(134)。

[0223] GPU 10确定已经发出的读取请求的数目是否大于或等于第一阈值(136)。响应于

确定已经发出的读取请求的数目不大于或等于第一阈值, GPU 10发出另一读取请求(134)。GPU 10持续发出读取请求直到已经发出的读取请求的数目大于或等于第一阈值。响应于确定已经发出的读取请求的数目大于或等于第一阈值, GPU 10停止发出读取请求(138)。

[0224] GPU 10确定未消耗源数据的量是否大于或等于第一阈值(140)。响应于确定未消耗源数据的量不大于或等于第一阈值, GPU 10继续停止发出读取请求(138)。响应于确定未消耗源数据的量大于或等于第一阈值, GPU 10开始消耗源数据, 基于源数据产生目的地数据, 且发出写入请求(142)。

[0225] GPU 10确定未消耗源数据的量是否小于或等于第二阈值(144)。第二阈值可小于第一阈值。响应于确定未消耗源数据的量不小于或等于第二阈值, GPU 10继续消耗源数据, 产生目的地数据, 且发出写入请求(142)。GPU 10继续消耗源数据, 产生目的地数据, 且发出写入请求直到未消耗源数据的量小于或等于第二阈值。响应于确定未消耗源数据的量小于或等于第二阈值, GPU 10停止消耗源数据, 产生目的地数据, 和发出写入请求(146)。

[0226] 并且, 响应于确定未消耗源数据的量小于或等于第二阈值, GPU 10可恢复发出读取请求, 前提是作为存储器传送操作的部分存在更多数据要读取。举例来说, GPU 10确定是否存在更多数据要读取(148)。如果存在更多数据要读取, 那么GPU 10前进到过程框134且发出后续读取请求。如果不存在更多数据要读取, 那么GPU 10完成存储器传送操作(150)。

[0227] 图17是说明用于根据本发明的顺序同步模式发出读取请求的实例技术的流程图。读取引擎38起始存储器传送操作(152)。读取引擎38发出读取请求到存储器子系统8(例如, 存储器控制器20或存储器22)以读取与源区内的像素位置相关联的源数据(154)。

[0228] 读取引擎38确定已经发出的读取请求的数目是否大于或等于第一阈值(156)。响应于确定已经发出的读取请求的数目不大于或等于第一阈值, 读取引擎38发出另一读取请求(154)。读取引擎38继续发出读取请求直到已经发出的读取请求的数目大于或等于第一阈值。响应于确定已经发出的读取请求的数目大于或等于第一阈值, 读取引擎38停止发出读取请求(158)。

[0229] 读取引擎38确定未消耗源数据的量是否小于或等于第二阈值(160)。第二阈值可小于第一阈值。响应于确定未消耗源数据的量不小于或等于第二阈值, 读取引擎38继续停止发出读取请求(158)。读取引擎38继续停止发出读取请求直到未消耗源数据的量小于或等于第二阈值。

[0230] 响应于确定未消耗源数据的量小于或等于第二阈值, GPU 10可恢复发出读取请求, 前提是作为存储器传送操作的部分存在更多数据要读取。举例来说, GPU 10确定是否存在更多数据要读取(162)。如果存在更多数据要读取, 那么GPU 10前进到过程框154且发出后续读取请求。如果不存在更多数据要读取, 那么GPU 10完成存储器传送操作(164)。

[0231] 图18是说明用于根据本发明的顺序同步模式发出写入请求的实例技术的流程图。写入引擎40起始存储器传送操作(166)。

[0232] 写入引擎40等待直到未消耗源数据的量大于或等于第一阈值。举例来说, 写入引擎40确定未消耗源数据的量是否大于或等于第一阈值(168)。响应于确定未消耗源数据的量不大于或等于第一阈值, 写入引擎40继续等待。响应于确定未消耗源数据的量大于或等于第一阈值, 写入引擎40开始消耗源数据, 基于源数据产生目的地数据, 且发出写入请求(170)。

[0233] 写入引擎40确定未消耗源数据的量是否小于或等于第二阈值(172)。第二阈值可小于第一阈值。响应于确定未消耗源数据的量不小于或等于第二阈值,写入引擎40继续消耗源数据,产生目的地数据,且发出写入请求(170)。写入引擎40继续消耗源数据,产生目的地数据,且发出写入请求直到未消耗源数据的量小于或等于第二阈值。响应于确定未消耗源数据的量小于或等于第二阈值,写入引擎40停止消耗源数据,产生目的地数据,和发出写入请求(174)。

[0234] 写入引擎40可停止消耗源数据,产生目的地数据和发出写入请求直到未消耗源数据的量大于或等于第一阈值。只要作为存储器传送操作的部分存在更多数据要消耗,写入引擎40可响应于确定未消耗源数据的量大于或等于第一阈值而恢复消耗源数据,产生目的地数据且发出写入请求。举例来说,写入引擎40确定作为存储器传送操作的部分是否存在更多数据要消耗(176)。如果存在更多数据要消耗,那么写入引擎40前进到决策框168且等待未消耗源数据的量大于或等于第一阈值。如果作为存储器传送操作的部分不存在更多数据要消耗,那么写入引擎40完成存储器传送操作(178)。

[0235] 图19是说明用于根据本发明的顺序存储器传送模式执行存储器传送操作的实例技术的流程图。GPU 10起始存储器传送操作(180)。GPU 10激活读取引擎38(182)。当读取引擎38激活时,读取引擎38发出读取请求直到读取引擎38经去激活。

[0236] 读取引擎38保持激活直到已经发出的读取请求的数目大于或等于第一阈值。举例来说,GPU 10确定已经发出的读取请求的数目是否大于或等于第一阈值(184)。响应于确定已经发出的读取请求的数目不大于或等于第一阈值,GPU 10环回到决策框184。响应于确定已经发出的读取请求的数目大于或等于第一阈值,GPU 10去激活读取引擎38(186)。当读取引擎38经去激活时,读取引擎38可停止发出读取请求。

[0237] GPU 10等待直到未消耗源数据的量大于或等于第一阈值。举例来说,GPU 10确定未消耗源数据的量是否大于或等于第一阈值(188)。响应于确定未消耗源数据的量不大于或等于第一阈值,GPU 10环回到决策框188。响应于确定未消耗源数据的量大于或等于第一阈值,GPU 10激活写入引擎40(190)。当写入引擎40经激活时,写入引擎40可消耗源数据,基于源数据产生目的地数据,且发出写入请求以将目的地数据写入到存储器22。

[0238] 写入引擎40保持激活直到未消耗源数据的量小于或等于第二阈值。举例来说,GPU 10确定未消耗源数据的量是否小于或等于第二阈值(192)。第二阈值可小于第一阈值。响应于确定未消耗源数据的量不小于或等于第二阈值,GPU 10环回到决策框192。响应于确定未消耗源数据的量小于或等于第二阈值,GPU 10去激活写入引擎40(194)。当写入引擎40经减活时,写入引擎40可停止消耗源数据,产生目的地数据和发出写入请求。

[0239] 并且,响应于确定未消耗源数据的量小于或等于第二阈值,GPU 10可再激活读取引擎38,前提是作为存储器传送操作的部分存在更多数据要读取。举例来说,GPU 10确定是否存在更多数据要读取(196)。如果存在更多数据要读取,那么GPU 10前进到过程框182且激活读取引擎38。如果不存在更多数据要读取,那么GPU 10完成存储器传送操作(198)。

[0240] 图20是说明根据本发明的用于执行存储器传送操作的实例技术的流程图。出于示范性目的,图20中的技术描述为由图1和2中所示的计算装置2执行。在其它实例中,图20中说明的技术可在以相同或不同配置具有相同或不同组件的其它系统中实施。

[0241] CPU 6和/或GPU 10选择用于执行存储器传送操作的全部或部分的存储器传送模

式(202)。在一些实例中,存储器传送模式可选自包含交错存储器传送模式和顺序存储器传送模式的至少两个不同存储器传送模式的集合。

[0242] GPU 10基于所述选定存储器传送模式执行存储器传送操作(204)。举例来说, GPU10可响应于将交错存储器传送模式选择为用于执行存储器传送操作的存储器传送模式而基于交错存储器传送模式执行存储器传送操作。作为另一实例, GPU 10可响应于将顺序存储器传送模式选择为用于执行存储器传送操作的存储器传送模式而基于顺序存储器传送模式执行存储器传送操作。

[0243] 在一些实例中, GPU 10可使用选自至少两个不同存储器传送模式的集合的第一存储器传送模式执行存储器传送操作的第一部分, 且使用选自至少两个不同存储器传送模式的集合的第二存储器传送模式执行存储器传送操作的第二部分。

[0244] 图21-22和24-26说明根据本发明的用于选择用于执行存储器传送操作的存储器传送模式的不同技术。出于示范性目的, 图21-22和24-26中所示的技术描述为由图1和2中所示的计算装置2执行。在其它实例中, 图21-22和24-26中说明的技术可在以相同或不同配置具有相同或不同组件的其它系统中实施。

[0245] 图21是说明根据本发明的用于基于将由存储器传送操作传送的数据量而选择用于执行存储器传送操作的存储器传送模式的实例技术的流程图。在一些实例中, 图21中所示的技术可用以实施图20中所示的过程框202。

[0246] CPU 6和/或GPU 10确定存储器传送操作的大小(206)。存储器传送操作的大小可对应于存储器传送操作将传送的数据的量。在一些实例中, 存储器传送操作的大小可在存储器传送命令中指定。在其它实例中, CPU 6和/或GPU 10可基于源区和目的地区的尺寸且基于在存储器传送操作中使用的源区的数目而确定存储器传送操作的大小。

[0247] CPU 6和/或GPU 10确定存储器传送操作的大小是否大于阈值(208)。如果存储器传送操作的大小不大于阈值, 那么CPU 6和/或GPU 10将交错存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式(210)。如果存储器传送操作的大小大于阈值, 那么CPU 6和/或GPU 10将顺序存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式(212)。

[0248] 图22是说明根据本发明的用于基于存储器传送操作的目的地区与固定屏幕对准缓冲区具有的重叠量而选择用于执行存储器传送操作的存储器传送模式的实例技术的流程图。在一些实例中, 图22中所示的技术可用以实施图20中所示的过程框202。

[0249] CPU 6和/或GPU 10确定针对存储器传送操作界定的目的地区的屏幕位置(214)。所述屏幕位置可指代屏幕空间中的目的地区的位置。在一些实例中, CPU 6和/或GPU10可基于存储器传送命令中接收的数据确定屏幕位置。

[0250] CPU 6和/或GPU 10选择至少部分与目的地区重叠的屏幕对准缓冲区(216)。图23是说明具有多个屏幕对准缓冲区(即, 瓦片1-20)和目的地表面218的屏幕的概念图。如图23中所展示, 屏幕对准缓冲区1-4、6-9和11-14至少部分与目的地表面218重叠, 而屏幕对准缓冲区5、10、15和16-20完全不与目的地表面218重叠。

[0251] CPU 6和/或GPU 10确定目的地区是否完全重叠选定屏幕对准缓冲区(220)。换句话说, CPU 6和/或GPU 10可确定目的地区是否完全覆盖选定固定屏幕对准缓冲区。如果目的地区并不完全重叠选定屏幕对准缓冲区(例如, 如果目的地区并不完全覆盖选定屏幕对

准缓冲区),那么CPU 6和/或GPU 10将交错存储器传送模式选择为存储器传送模式以相对于所述选定屏幕对准缓冲区执行存储器传送操作(222)。否则,如果目的地区完全重叠所述选定屏幕对准缓冲区(例如,如果目的地区完全覆盖所述选定屏幕对准缓冲区),那么CPU 6和/或GPU 10将顺序存储器传送模式选择为存储器传送模式以相对于所述选定屏幕对准缓冲区执行存储器传送操作(224)。

[0252] 举例来说,相对于图23,目的地区218完全覆盖固定屏幕对准缓冲区7、8和9。因此,CPU 6和/或GPU 10可选择用于相对于屏幕对准缓冲区7、8和9执行存储器传送操作的顺序存储器传送模式。类似地,目的地区218部分地覆盖但并不完全覆盖屏幕对准缓冲区1-4、6和11-14。因此,CPU 6和/或GPU 10可选择用于相对于屏幕对准缓冲区1-4、6和11-14执行存储器传送操作的交错存储器传送模式。

[0253] 在一些情况下,表面可对应于显示监视器或屏幕上含有的像素。屏幕可经细分为多个区,其各自具有对应于用以存储所读取数据的缓冲器的大小的固定大小。这些经细分区可被称为固定屏幕对准缓冲区。在此些实例中,当在顺序存储器传送模式中操作时,GPU 10可经配置以在对应于固定屏幕对准缓冲区的大小的固定大小的数据单元中在存储器22与GPU 10之间传送数据。

[0254] 如果目的地区部分地但不完全地覆盖特定的固定屏幕对准缓冲区,那么CPU 6和/或GPU 10可将交错存储器传送模式选择为用于屏幕的所述区的存储器传送模式,因为其中将存储读取数据的缓冲器将不会完全利用。如果在此情况下选定顺序存储器模式,那么固定大小存储器读取将致使从存储器读取额外的数据,这可超越通过使用顺序存储器传送模式将发生的任何读取-写入周转节省。另一方面,如果目的地区完全覆盖特定的固定屏幕对准缓冲区,那么CPU 6和/或GPU 10可将顺序存储器传送模式选择为用于屏幕的所述区的存储器传送模式,因为其中将存储读取数据的缓冲器将完全利用且将不浪费时间检索额外的数据。

[0255] 在一些实例中,存储器传送操作可以GPU 10通过以下方式实施:再现在空间上对应于存储器传送操作的目的地区的一或多个基元,将对应于源区数据的纹理施加到经再现的一或多个基元,且将所述一或多个基元与已经存储在帧缓冲器中的任何目的地数据合并。在此些实例中,CPU 6和/或GPU 10可基于待再现的基元与固定屏幕对准缓冲区具有的重叠量而选择用于相对于固定屏幕对准缓冲区执行存储器传送操作的存储器传送模式。举例来说,CPU 6和/或GPU 10可在待再现基元完全覆盖固定屏幕对准缓冲区的情况下将顺序存储器传送模式选择为存储器传送模式,且在待再现基元并不完全覆盖固定屏幕对准缓冲区的情况下将交错存储器传送模式选择为存储器传送模式。

[0256] 图24是说明根据本发明的用于基于用以执行存储器传送操作的着色器程序的复杂性选择用于执行存储器传送操作的存储器传送模式的实例技术的流程图。在一些实例中,图24中所示的技术可用以实施图20中所示的过程框202。

[0257] CPU 6和/或GPU 10确定用以执行存储器传送操作的着色器程序的复杂性(226)。所述着色器程序在一些实例中可为像素着色器程序。在一些实例中,着色器程序的复杂性可对应于着色器程序的大小。在其它实例中,着色器程序的复杂性对应于着色器程序将执行的计算的数目。在其它实例中,着色器程序的复杂性可对应于着色器程序将执行的循环或迭代的数目。

[0258] CPU 6和/或GPU 10确定着色器程序的复杂性是否大于阈值(228)。如果着色器程序的复杂性不大于阈值,那么CPU 6和/或GPU 10可将顺序存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式(230)。如果着色器程序的复杂性大于阈值,那么CPU 6和/或GPU 10可将交错存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式(232)。

[0259] 图25是说明根据本发明的用于基于用以执行存储器传送操作的着色器程序的大小选择用于执行存储器传送操作的存储器传送模式的实例技术的流程图。在一些实例中,图25中所示的技术可用以实施图20中所示的过程框202。

[0260] CPU 6和/或GPU 10确定用以执行存储器传送操作的着色器程序的大小(234)。所述着色器程序在一些实例中可为像素着色器程序。在一些实例中,着色器程序的大小可对应于着色器程序中的指令的数目。在其它实例中,着色器程序的大小可对应于着色器程序中的字节的数目。

[0261] CPU 6和/或GPU 10确定着色器程序的大小是否大于阈值(236)。如果着色器程序的大小不大于阈值,那么CPU 6和/或GPU 10可将顺序存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式(238)。如果着色器程序的大小大于阈值,那么CPU 6和/或GPU 10可将交错存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式(240)。

[0262] 图26是说明根据本发明的用于基于着色器程序的执行是否可能为存储器传送操作的性能瓶颈的确定而选择用于执行存储器传送操作的存储器传送模式的实例技术的流程图。在一些实例中,图26中所示的技术可用以实施图20中所示的过程框202。

[0263] CPU 6和/或GPU 10做出着色器程序的执行是否可能为存储器传送操作的性能瓶颈的确定(242)。着色器程序在一些实例中可为像素着色器程序。在一些实例中,CPU 6和/或GPU 10可基于着色器程序的复杂性和/或着色器程序的大小做出所述确定。

[0264] 如果着色器程序的执行不大可能为存储器传送操作的性能瓶颈(例如,着色器程序的大小和/或复杂性小于阈值),那么CPU 6和/或GPU 10可将顺序存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式(244)。如果着色器程序的执行可能为存储器传送操作的性能瓶颈(例如,着色器程序的大小和/或复杂性大于阈值),那么CPU 6和/或GPU 10可将交错存储器传送模式选择为将用于执行存储器传送操作的全部或部分的存储器传送模式(246)。

[0265] 图27是说明根据本发明的用于执行存储器传送操作的实例技术的流程图。在一些实例中,图27中所示的技术可用以实施图20中所示的过程框204。

[0266] GPU 10确定是否已选定顺序存储器传送模式或交错存储器传送模式(248)。如果尚未选定顺序存储器传送模式(即,已选定交错存储器传送模式),那么GPU 10可配置将用以执行存储器传送操作的高速缓冲存储器的高速缓冲存储线大小为第一大小(250)。如果已选定顺序存储器传送模式(即,已选定交错存储器传送模式),那么GPU 10可配置将用以执行存储器传送操作的高速缓冲存储器的高速缓冲存储线大小为第二大小(252)。第二大小可大于第一大小。在一些实例中,第二大小可对应于等于高速缓冲存储器的整个容量的高速缓冲存储线大小,且第一大小可对应于等于小于高速缓冲存储器的整个容量的高速缓冲存储线大小。

[0267] 根据本发明的一些方面,描述用于用信号表示将由GPU使用用于执行存储器传送操作的存储器传送模式的技术。所述信令技术可允许在应用程序处理器上执行的GPU驱动器和/或用户应用程序控制用于执行存储器传送操作的存储器传送模式。

[0268] 在一些实例中,可将指定存储器传送模式的信息字段添加到GPU指令集架构 (ISA) 中包含的一或多个现有GPU命令。所述信息字段在一些情况下可为指定将用于与指令相关联的存储器传送操作的存储器传送模式的单个模式位。举例来说,图形驱动器可通过对GPU指定源和目的地存储器区且发出(三维) 3D绘制调用指令到GPU而服务于用户层级传图命令。3D绘制调用指令可包含指示在3D绘制调用指令的执行期间将使用的存储器传送模式的模式位。GPU可经配置以基于指令中指定的存储器传送模式执行存储器传送操作。

[0269] 在额外实例中,GPU可包含指定将用于执行存储器传送操作的存储器传送模式的存储器传送模式状态变量。在此些实例中,可将额外指令添加到GPU ISA中包含的指令。所述额外指令可用以将存储器传送模式状态变量的状态编程为指示将用于执行存储器传送操作的存储器传送模式的值。GPU可经配置以基于存储器传送模式状态变量的状态执行存储器传送操作。

[0270] 在其它实例中,指令可添加到用于GPU驱动器的应用程序编程接口 (API)。所述指令可包含存储器传送模式指示符,其指示将用于执行存储器传送操作的存储器传送模式。GPU驱动器可经配置以致使GPU基于存储器传送模式指示符执行存储器传送操作。以此方式,用户应用程序可能指定当执行存储器传送操作时由GPU使用的存储器传送模式。

[0271] 本发明的实例方面包含使用内部存储器缓冲器来允许按顺序首先读入数据且随后写出数据而不是交错数据。本发明的实例方面进一步包含用以基于与光栅化图案的基元相交点使用内部缓冲器自动进行光栅化的选择算法。因此,基元三角形在切换到用于“宽”区段的“顺序模式”的同时可不使用用于“窄”区段的内部缓冲器。本发明的实例方面进一步包含指示是否允许使用“顺序模式”的模式位。所述模式位可基于操作是否估计为存储器效率受限的而设定。实例实施方案可涉及使用现有高速缓冲存储器,但在“顺序模式”期间改变其操作以使用较大高速缓存行。

[0272] 在一些实例中,GPU驱动器和/或用户应用程序可实现不同存储器传送模式的选择(而不是迫使GPU进行一个或另一个)。在此些实例中,GPU在一些实例中可基于在运行时间期间由GPU发现的信息而做出将用于存储器传送操作的存储器传送模式的最终决策。

[0273] 本发明中所描述的技术可至少部分实施于硬件、软件、固件或其任何组合中。举例来说,所描述技术的各方面可实施于一或多个处理器中,包含一或多个微处理器、数字信号处理器 (DSP)、专用集成电路 (ASIC)、现场可编程门阵列 (FPGA),或任何其它等效集成或离散逻辑电路,以及此类组件的任何组合。术语“处理器”或“处理电路”可大体上指前述逻辑电路中的任一者(单独或结合其它逻辑电路)或例如执行处理的离散硬件的任何其它等效电路。

[0274] 此硬件、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,所描述单元、模块或组件中的任一者可一起或单独作为离散但可互操作逻辑装置而实施。将不同特征描述为模块或单元意图强调不同功能方面且未必暗示此等模块或单元必须由单独硬件或软件组件实现。替代地,与一或多个模块或单元相关联的功能性可由单独硬件、固件和/或软件组件执行,或整合到共用或单独硬件或软件组件内。

[0275] 本发明中所描述的技术也可存储、体现或编码于计算机可读媒体(例如,存储指令的计算机可读存储媒体)中。嵌入或编码于计算机可读媒体中的指令可导致一或多个处理器执行本文中所描述的技术(例如,当由一或多个处理器执行指令时)。计算机可读存储媒体可包含随机存取存储器(RAM)、只读存储器(ROM)、可编程只读存储(PROM)、可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)、快闪存储器、硬盘、CD-ROM、软盘、卡盒、磁性媒体、光学媒体或其它有形计算机可读存储媒体。

[0276] 计算机可读媒体可包含计算机可读存储媒体,其对应于例如上文所列的有形存储媒体的有形存储媒体。计算机可读媒体还可包括通信媒体,其包含促进计算机程序从一个地点到另一地点的传送(例如,根据通信协议)的任何媒体。以此方式,短语“计算机可读媒体”大体上可对应于(1)非暂时性有形计算机可读存储媒体和(2)例如暂时性信号或载波的非有形计算机可读通信媒体。

[0277] 已描述各种方面和实例。然而,可在不脱离以下权利要求的范围的情况下对本发明的结构或技术作出修改。

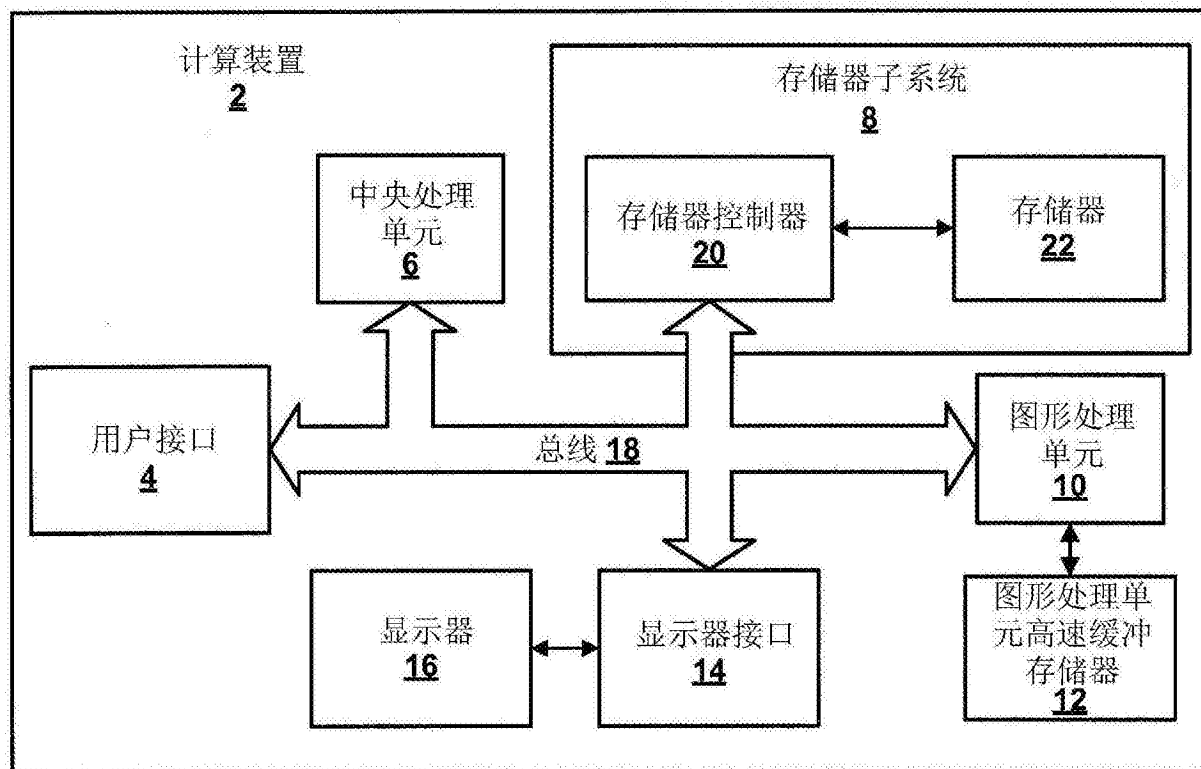


图1

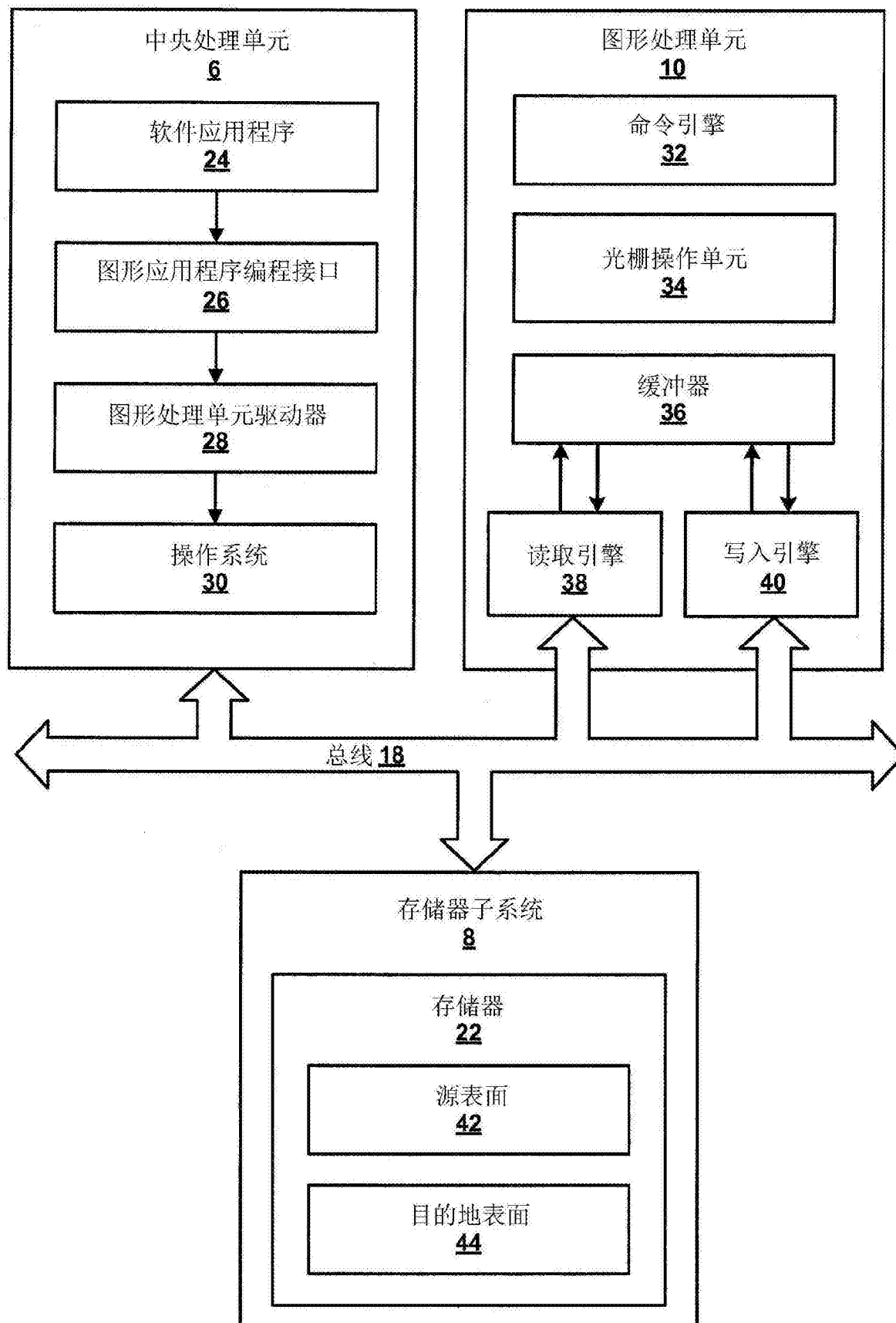


图2

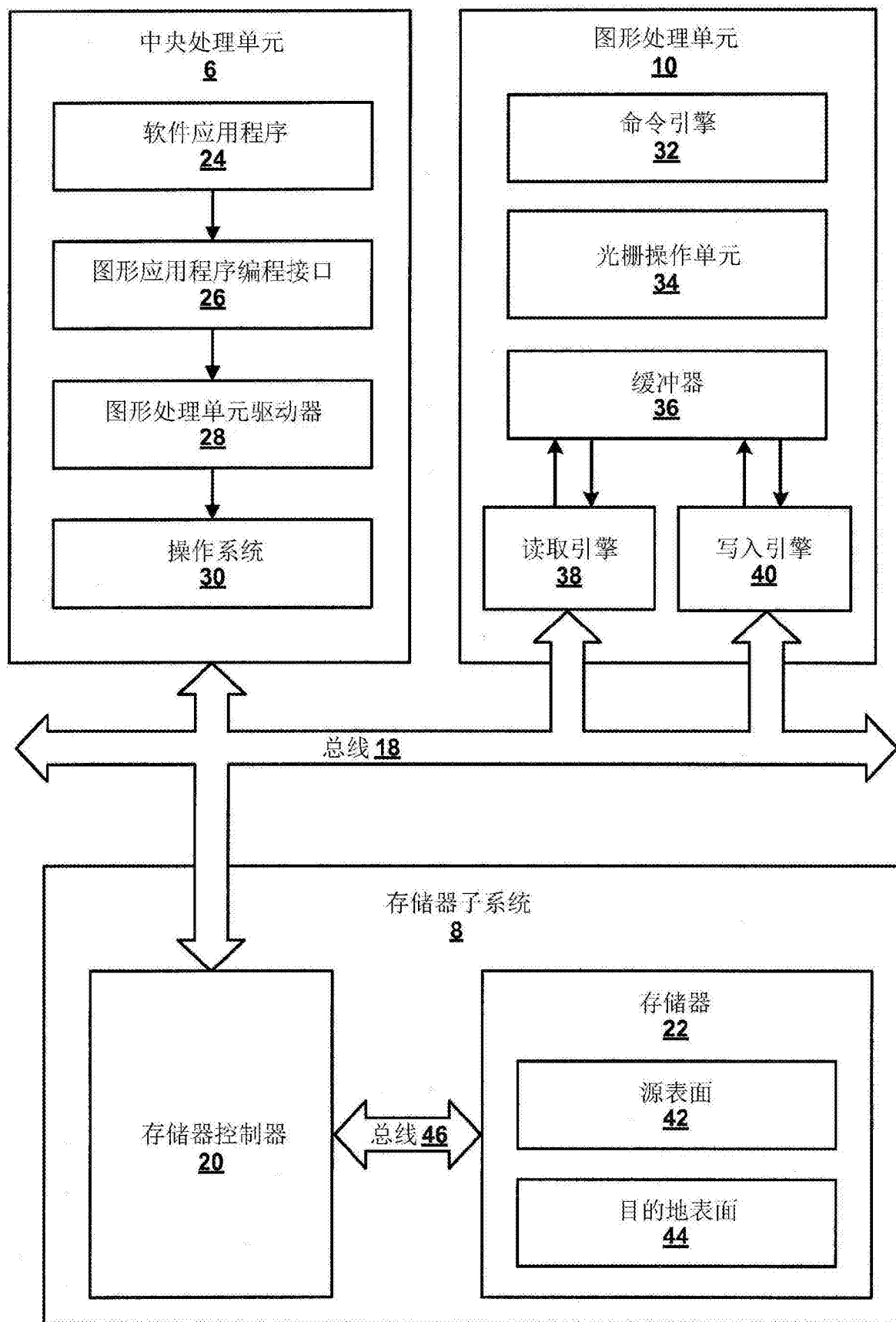


图3

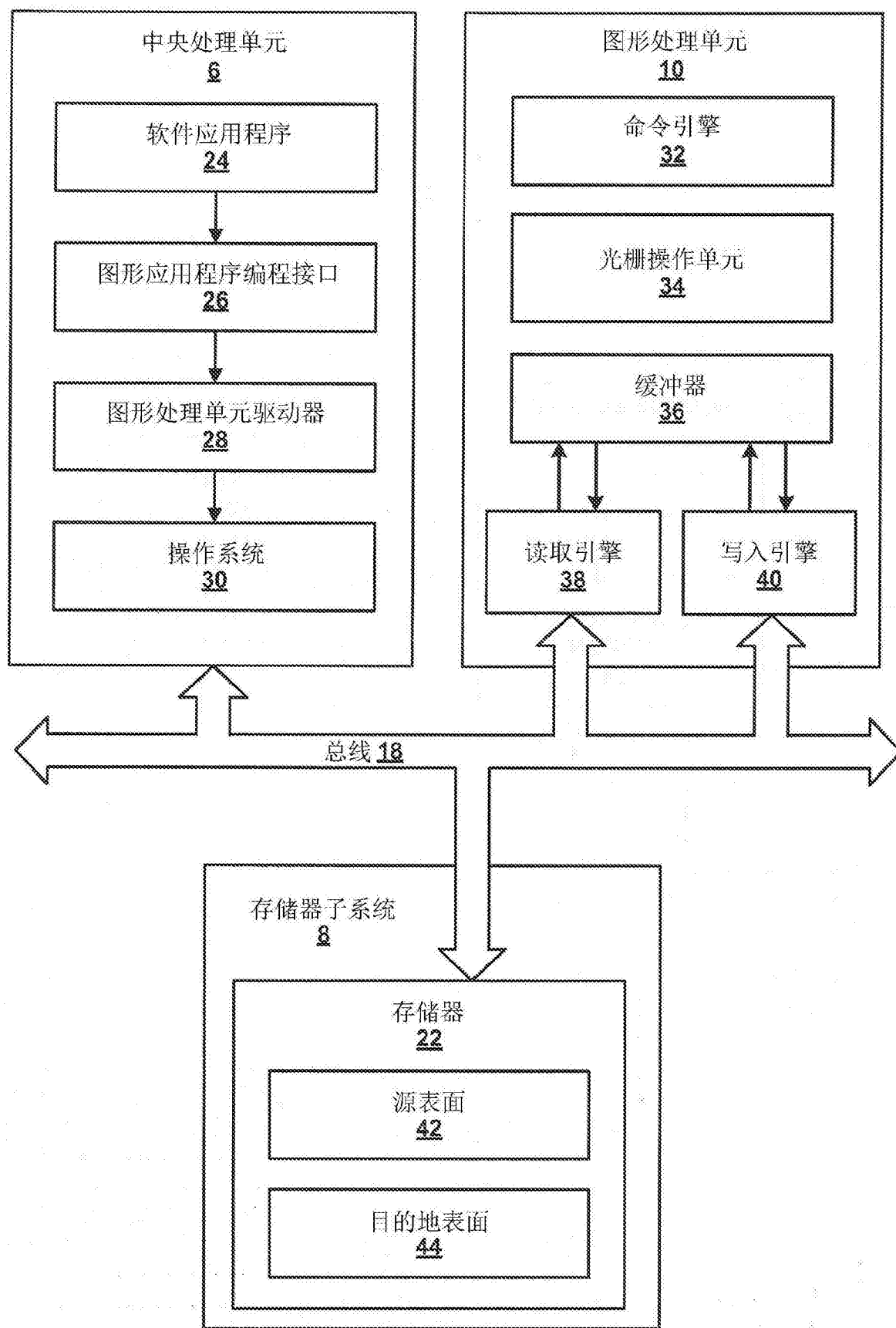


图4

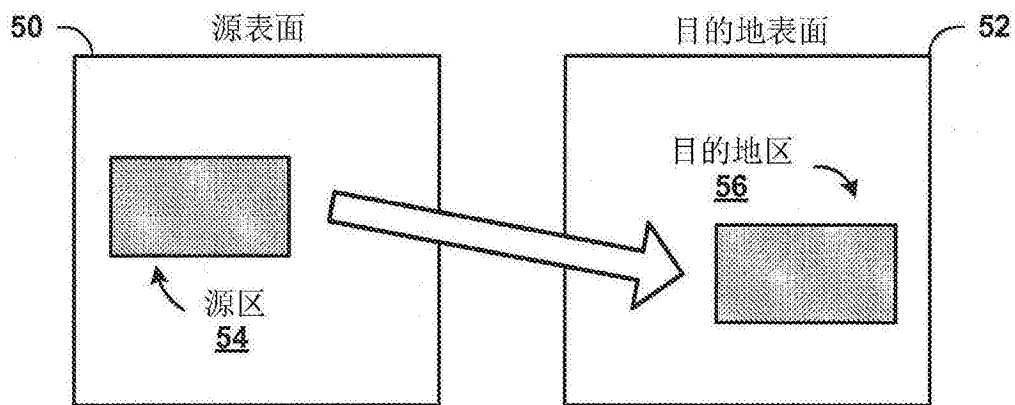


图5

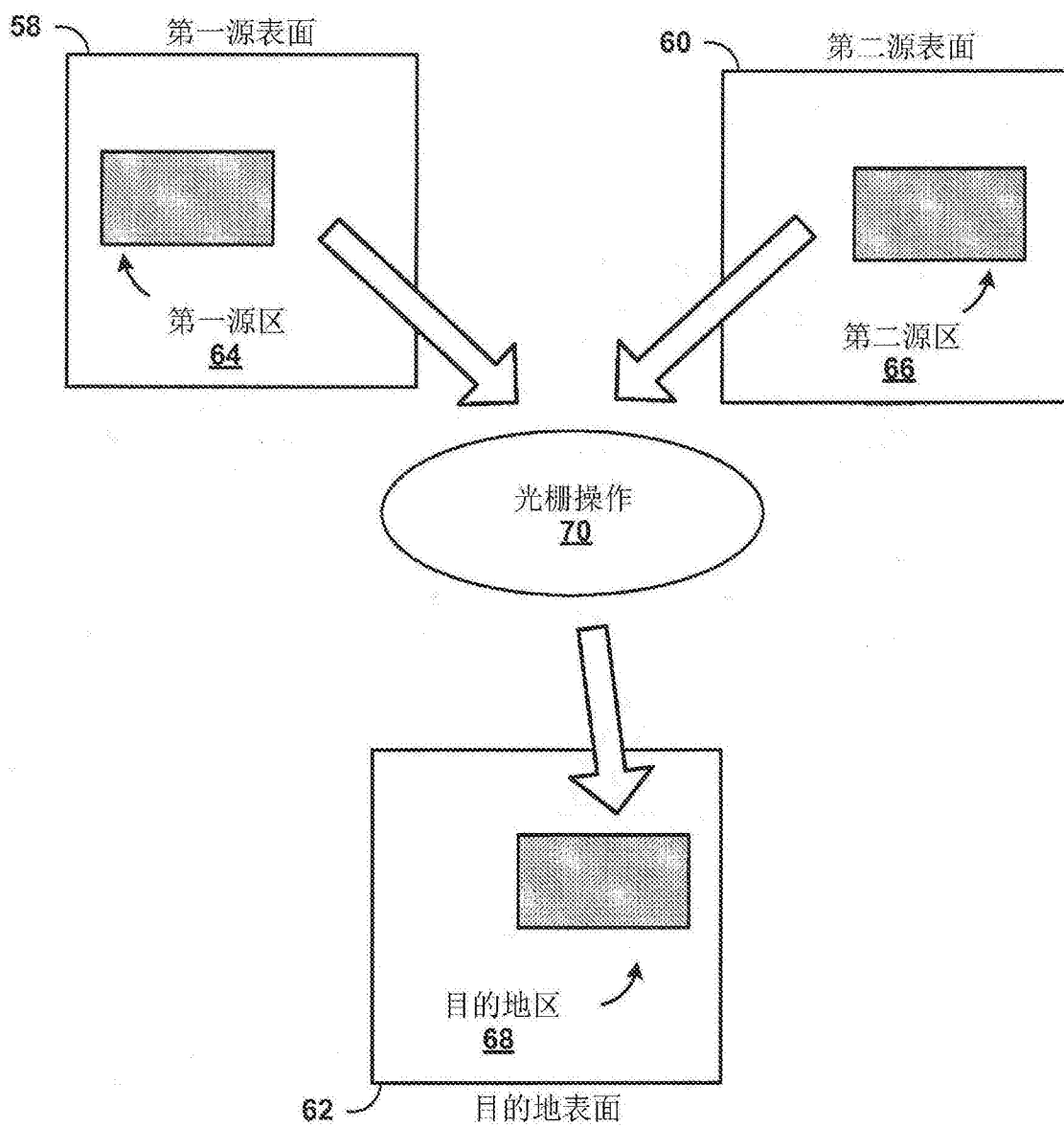


图6

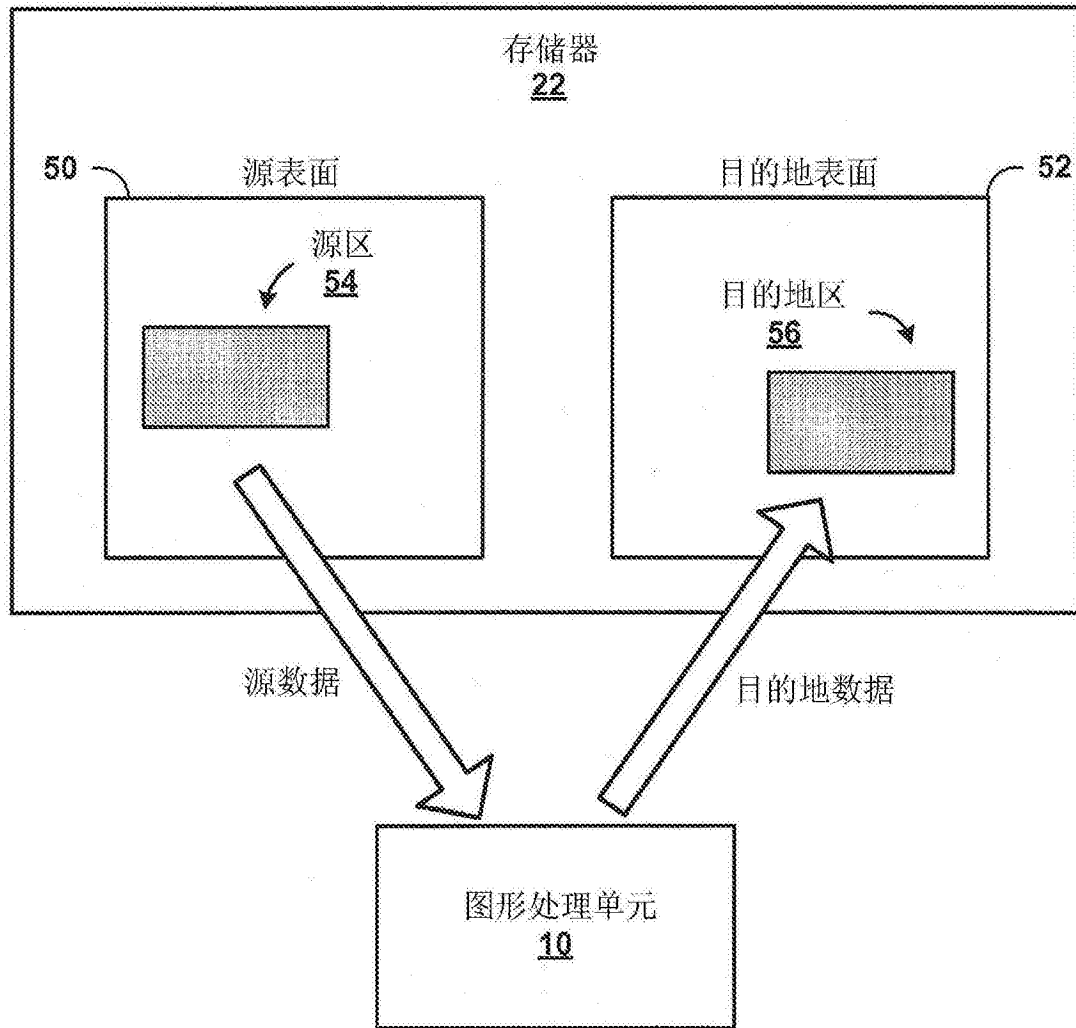


图7

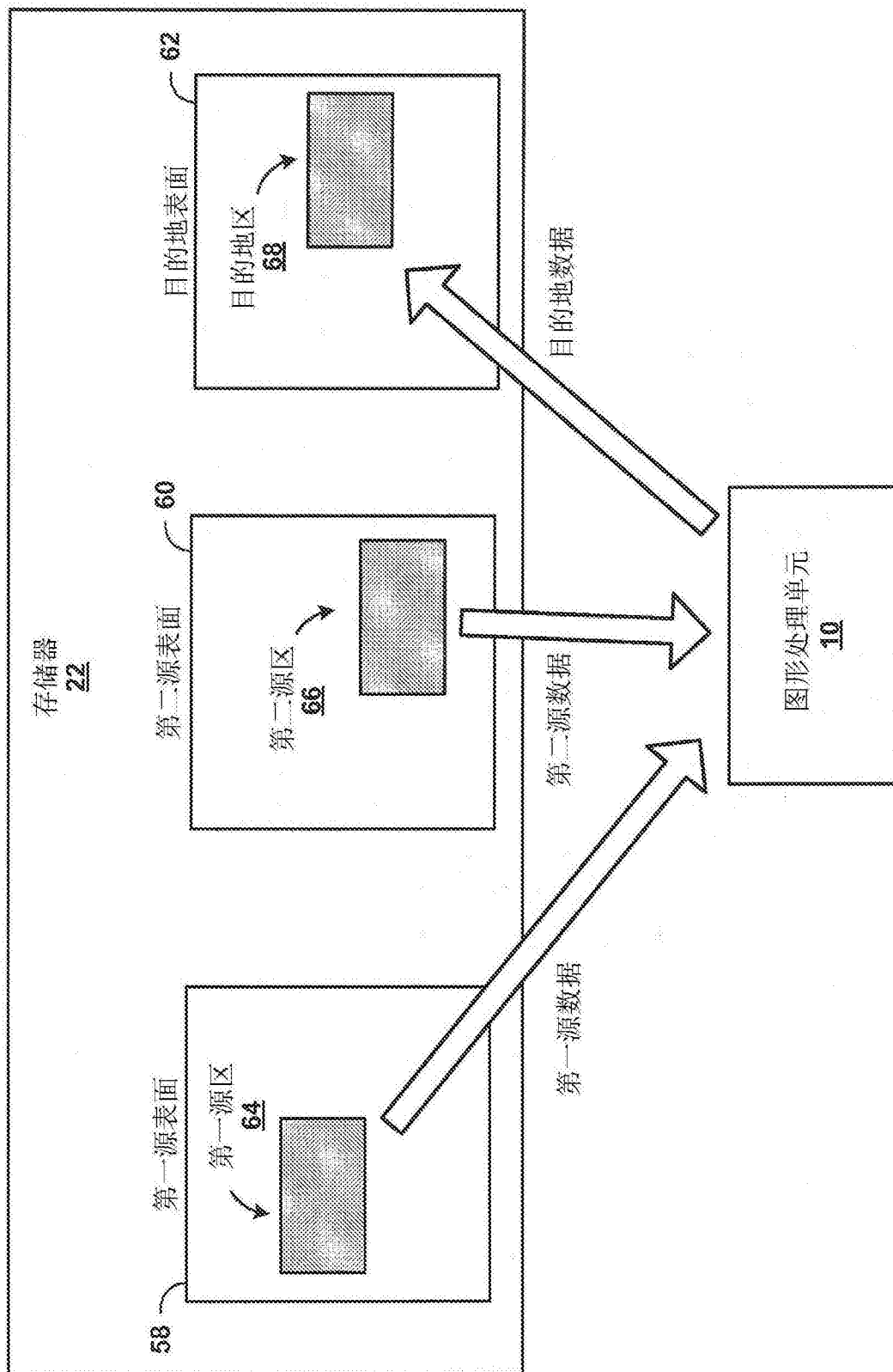


图8

时间	读取引擎	写入引擎	读取总线	写入总线	存储器控制器	存储器总线
1	读取1					
2	读取2				读取命令1	
3	读取3				读取命令2	
4	读取4				读取命令3	读取数据1
5			读取数据1		读取命令4	读取数据2
6		写入1	读取数据2	写入数据1		读取数据3
7	读取5	写入2	读取数据3	写入数据2		读取数据4
8	读取6	写入3	读取数据4	写入数据3		
9	读取7	写入4		写入数据4		
10	读取8				写入命令1	写入数据1
11					读取命令5	
12						
13						读取数据5
14			读取数据5			
15		写入5		写入数据5		
16	读取9				写入命令2	写入数据2
17					读取命令6	
18						
19						读取数据6
20			读取数据6			
21		写入6		写入数据6		
22	读取10				写入命令3	写入数据3
23					读取命令7	
24						
25						读取数据7
26			读取数据7			
27		写入7		写入数据7		
28	读取11				写入命令4	写入数据4
29					读取命令8	
30						
31						读取数据8
32			读取数据8			
33		写入8		写入数据8		
34	读取12				写入命令5	写入数据5
35					读取命令9	
36						
37						读取数据9
38			读取数据9			

图9A

时间	读取引擎	写入引擎	读取总线	写入总线	存储器控制器	存储器总线
39		写入9		写入数据9		
40					写入命令6	写入数据6
41					读取命令10	
42						
43						读取数据10
44			读取数据10			
45		写入10		写入数据10		
46					写入命令7	写入数据7
47					读取命令11	
48						
49						读取数据11
50			读取数据11			
51		写入11		写入数据11		
52					写入命令8	写入数据8
53					读取命令12	
54						
55						读取数据12
56			读取数据12			
57		写入12		写入数据12		
58					写入命令9	写入数据9
59					写入命令10	写入数据10
60					写入命令11	写入数据11
61					写入命令12	写入数据12

图9B

时间	读取引擎	写入引擎	读取总线	写入总线	存储器控制器	存储器总线
1	读取1					
2	读取2				读取命令1	
3	读取3				读取命令2	
4	读取4				读取命令3	读取数据1
5			读取数据1		读取命令4	读取数据2
6			读取数据2			读取数据3
7			读取数据3			读取数据4
8			读取数据4			
9		写入1		写入数据1		
10		写入2		写入数据2	写入命令1	写入数据1
11		写入3		写入数据3	写入命令2	写入数据2
12		写入4		写入数据4	写入命令3	写入数据3
13	读取5				写入命令4	写入数据4
14	读取6				读取命令5	
15	读取7				读取命令6	
16	读取8				读取命令7	读取数据5
17			读取数据5		读取命令8	读取数据6
18			读取数据6			读取数据7
19			读取数据7			读取数据8
20			读取数据8			
21		写入5		写入数据5		
22		写入6		写入数据6	写入命令5	写入数据5
23		写入7		写入数据7	写入命令6	写入数据6
24		写入8		写入数据8	写入命令7	写入数据7
25	读取9				写入命令8	写入数据8
26	读取10				读取命令9	
27	读取11				读取命令10	
28	读取12				读取命令11	读取数据9
29			读取数据9		读取命令12	读取数据10
30			读取数据10			读取数据11
31			读取数据11			读取数据12
32			读取数据12			
33		写入9		写入数据9		
34		写入10		写入数据10	写入命令9	写入数据9
35		写入11		写入数据11	写入命令10	写入数据10
36		写入12		写入数据12	写入命令11	写入数据11
37					写入命令12	写入数据12

图10

时间	读取引擎	写入引擎	读取总线	写入总线	存储器控制器	存储器总线
1	读取1A					
2	读取1B				读取命令1A	
3	读取2A				读取命令1B	
4	读取2B				读取命令2A	读取数据1A
5			读取数据1A		读取命令2B	读取数据1B
6			读取数据1B			读取数据2A
7		写入1	读取数据2A	写入数据1		读取数据2B
8	读取3A		读取数据2B			
9	读取3B	写入2		写入数据2		
10	读取4A				写入命令1	写入数据1
11	读取4B				读取命令3A	
12					读取命令3B	
13						读取数据3A
14			读取数据3A			读取数据3B
15			读取数据3B			
16		写入3		写入数据3		
17	读取5A				写入命令2	写入数据2
18	读取5B				读取命令4A	
19					读取命令4B	
20						读取数据4A
21			读取数据4A			读取数据4B
22			读取数据4B			
23		写入4		写入数据4		
24	读取6A				写入命令3	写入数据3
25	读取6B				读取命令5A	
26					读取命令5B	
27						读取数据5A
28			读取数据5A			读取数据5B
29			读取数据5B			
30		写入5		写入数据5		
31	读取7A				写入命令4	写入数据4
32	读取7B				读取命令6A	
33					读取命令6B	
34						读取数据6A
35			读取数据6A			读取数据6B
36			读取数据6B			
37		写入6		写入数据6		

图11A

时间	读取引擎	写入引擎	读取总线	写入总线	存储器控制器	存储器总线
38	读取8A				写入命令5	写入数据5
39	读取8B				读取命令7A	
40					读取命令7B	
41						读取数据7A
42			读取数据7A			读取数据7B
43			读取数据7B			
44		写入7		写入数据7		
45					写入命令6	写入数据6
46					读取命令8A	
47					读取命令8B	
48						读取数据8A
49			读取数据8A			读取数据8B
50			读取数据8B			
51		写入8		写入数据8		
52					写入命令7	写入数据7
53					写入命令8	写入数据8

图11B

时间	读取引擎	写入引擎	读取总线	写入总线	存储器控制器	存储器总线
1	读取1A					
2	读取1B				读取命令1A	
3	读取2A				读取命令1B	
4	读取2B				读取命令2A	读取数据1A
5	读取3A		读取数据1A		读取命令2B	读取数据1B
6	读取3B		读取数据1B		读取命令3A	读取数据2A
7	读取4A		读取数据2A		读取命令3B	读取数据2B
8	读取4B		读取数据2B		读取命令4A	读取数据3A
9			读取数据3A		读取命令4B	读取数据3B
10			读取数据3B			读取数据4A
11			读取数据4A			读取数据4B
12			读取数据4B			
13		写入1		写入数据1		
14		写入2		写入数据2	写入命令1	写入数据1
15		写入3		写入数据3	写入命令2	写入数据2
16		写入4		写入数据4	写入命令3	写入数据3
17	读取5A				写入命令4	写入数据4
18	读取5B				读取命令5A	
19	读取6A				读取命令5B	
20	读取6B				读取命令6A	读取数据5A
21	读取7A		读取数据5A		读取命令6B	读取数据5B
22	读取7B		读取数据5B		读取命令7A	读取数据6A
23	读取8A		读取数据6A		读取命令7B	读取数据6B
24	读取8B		读取数据6B		读取命令8A	读取数据7A
25			读取数据7A		读取命令8B	读取数据7B
26			读取数据7B			读取数据8A
27			读取数据8A			读取数据8B
28			读取数据8B			
29		写入5		写入数据5		
30		写入6		写入数据6	写入命令5	写入数据5
31		写入7		写入数据7	写入命令6	写入数据6
32		写入8		写入数据8	写入命令7	写入数据7
33					写入命令8	写入数据8

图12

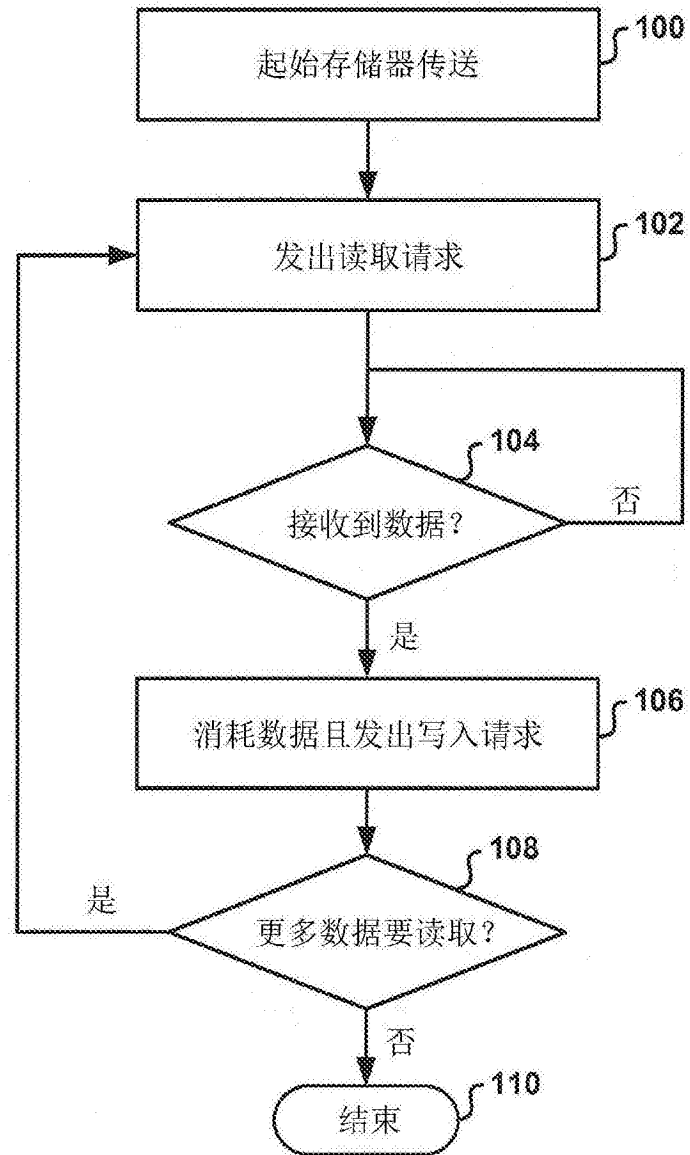


图13

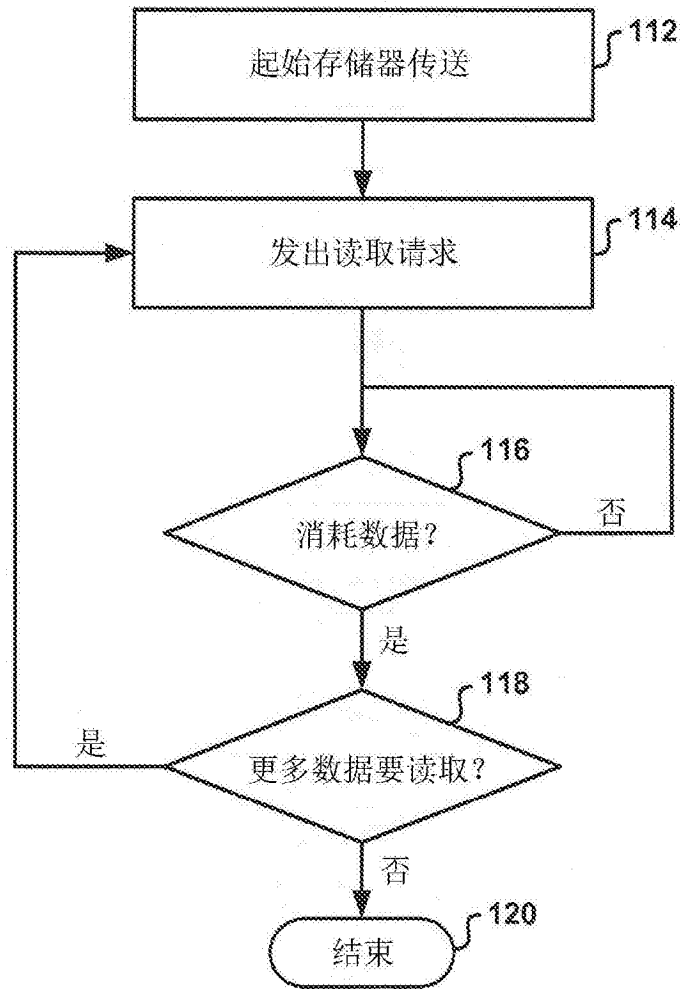


图14

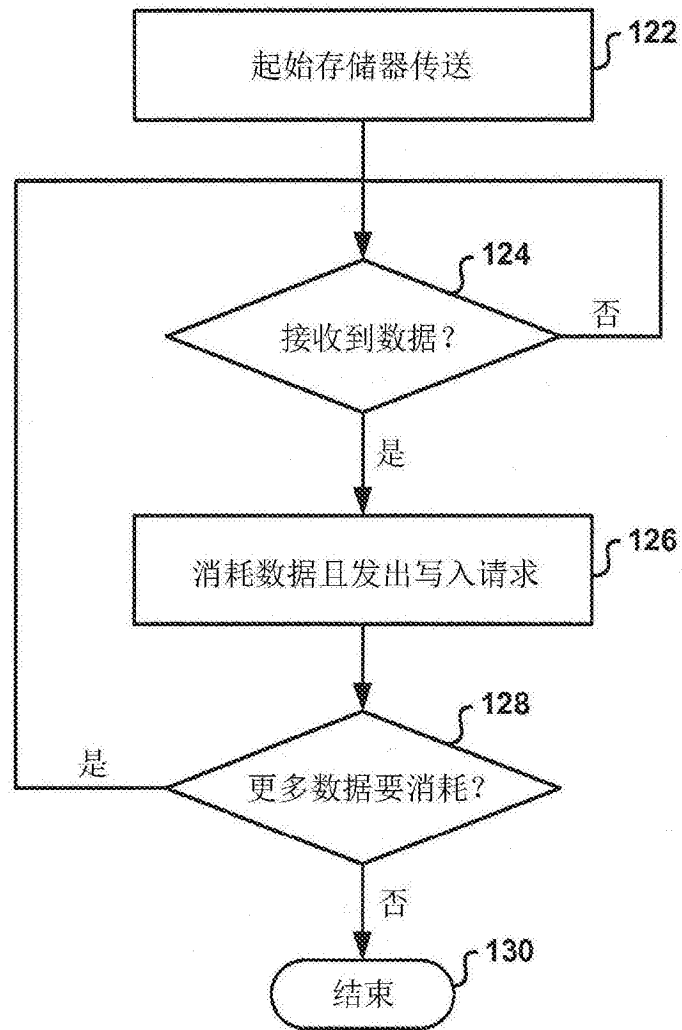


图15

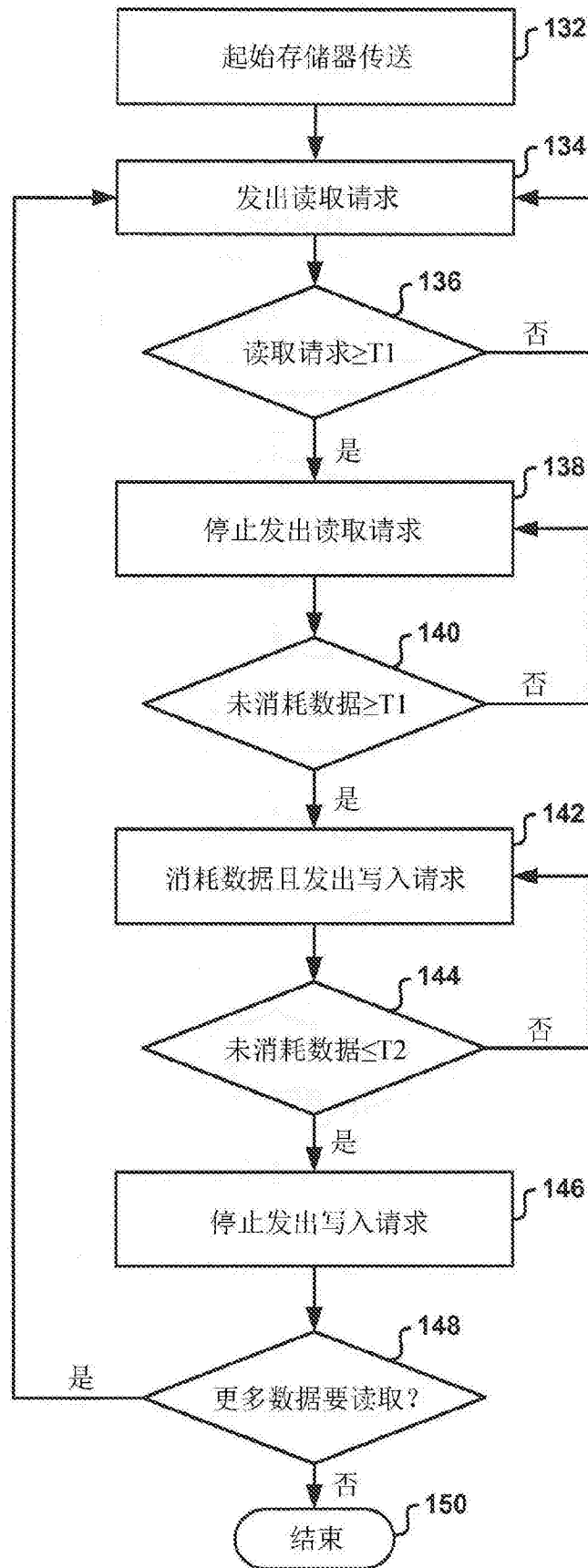


图16

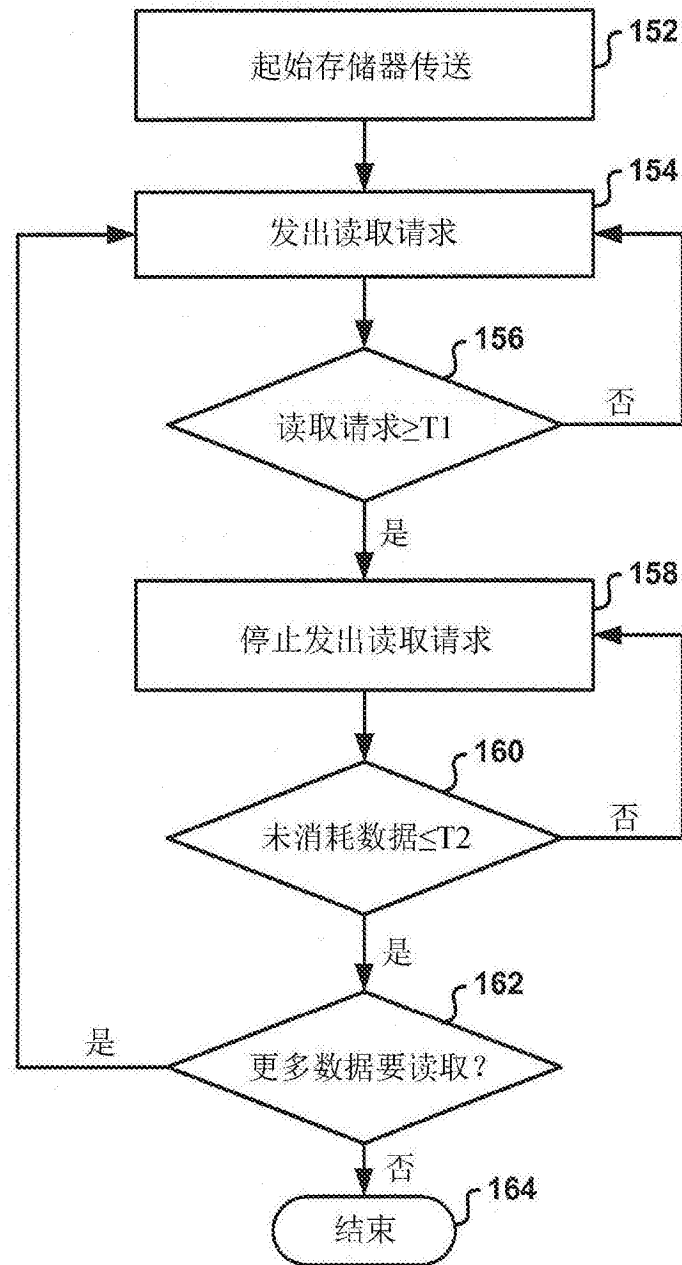


图17

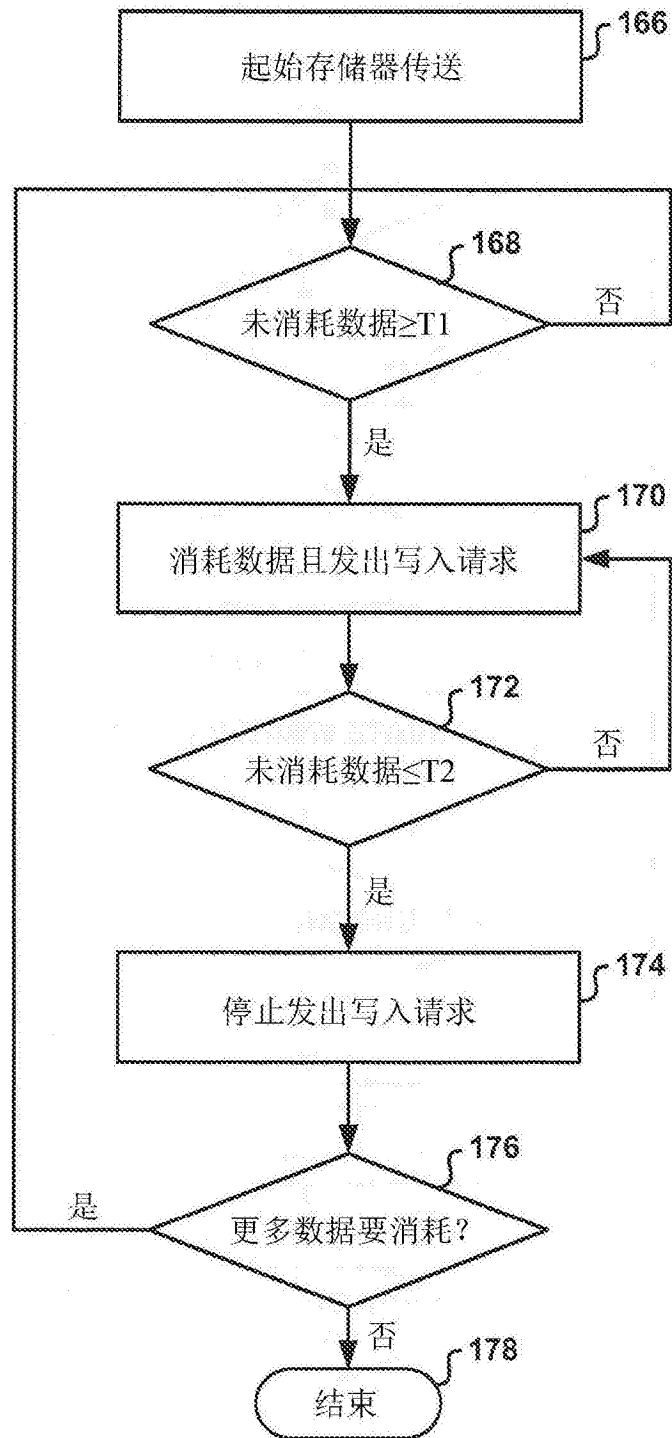


图18

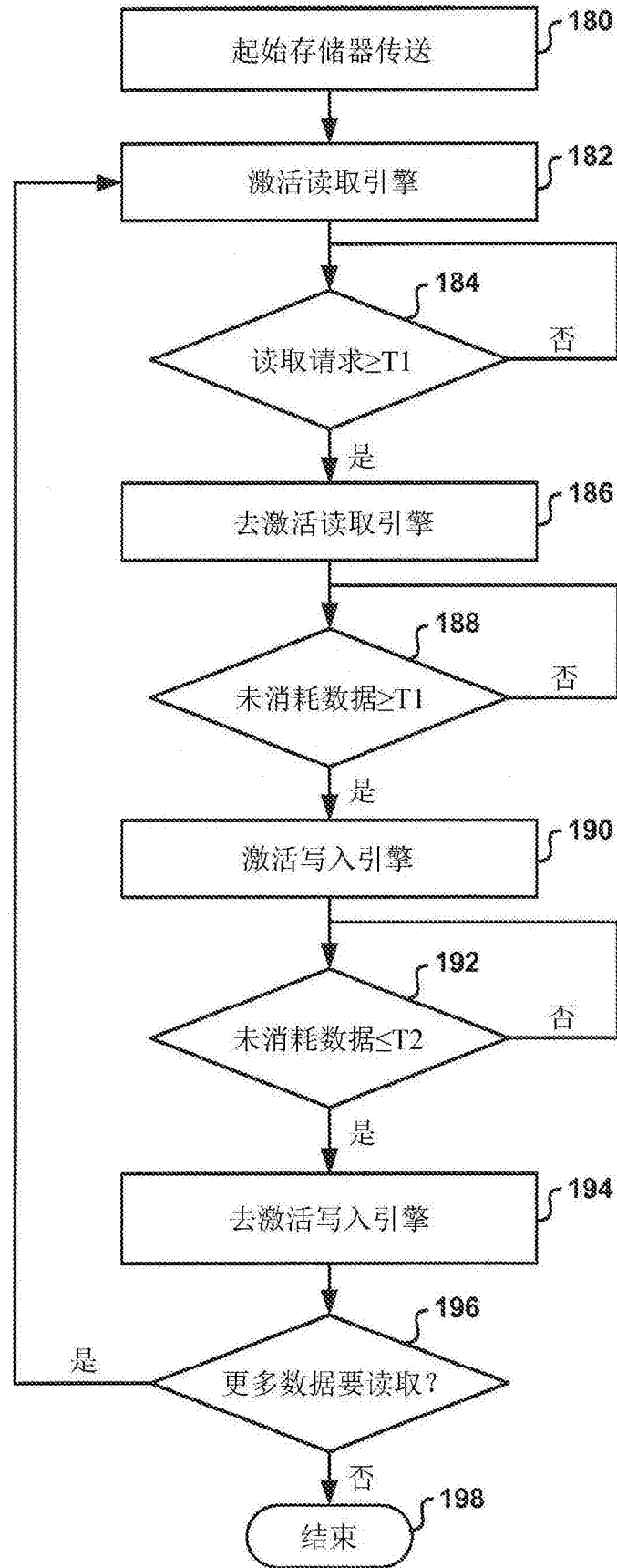


图19

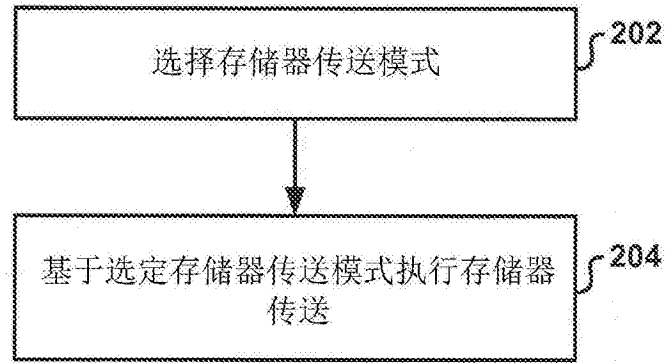


图20

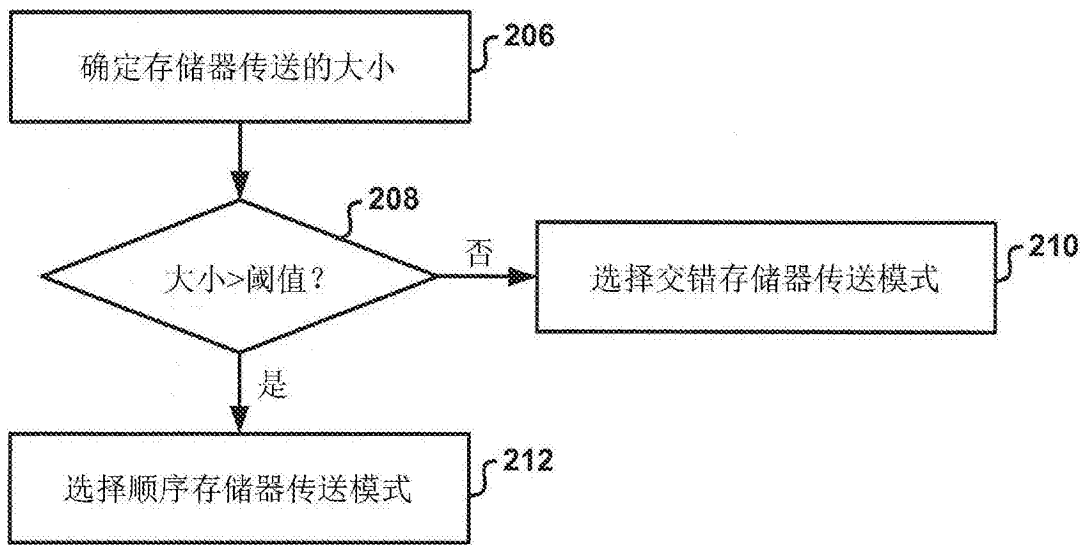


图21

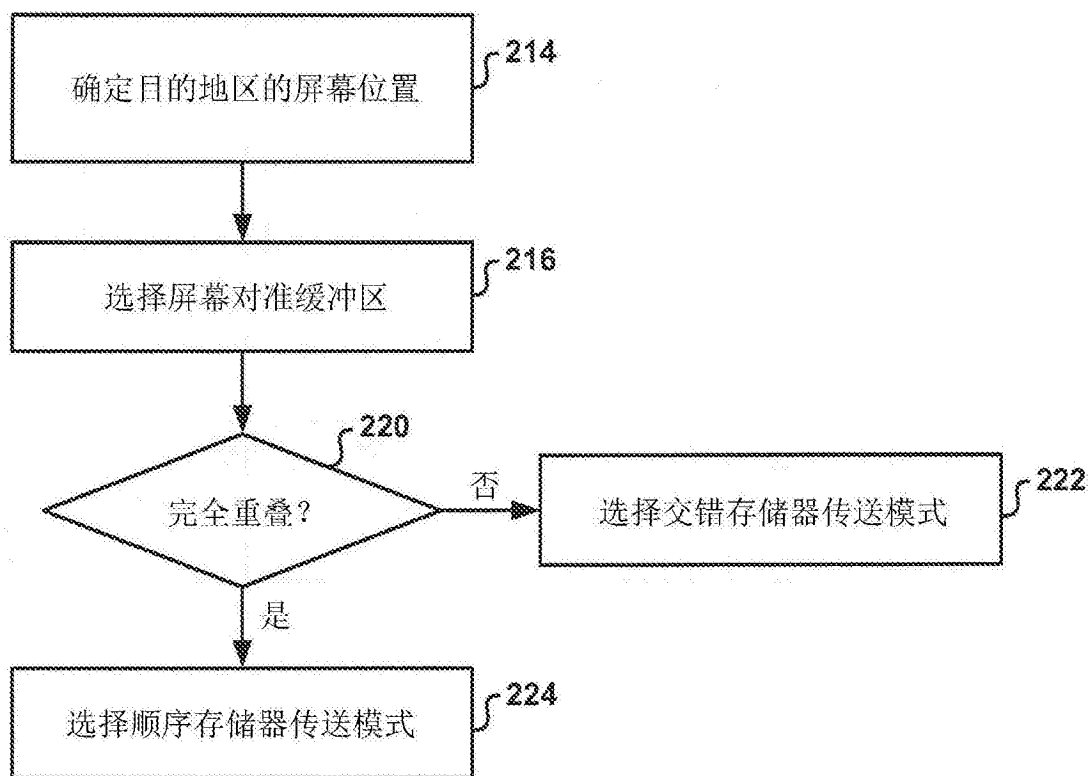


图22

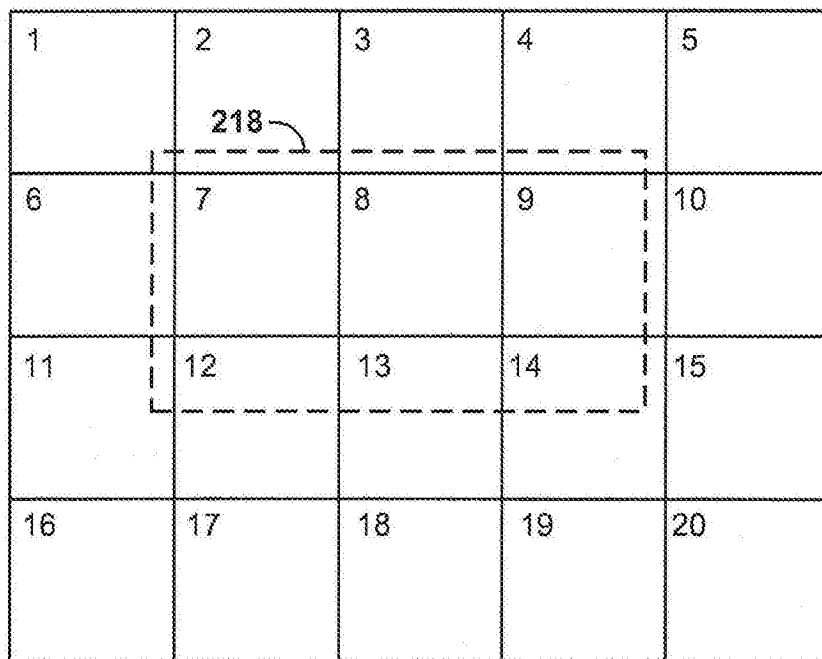


图23

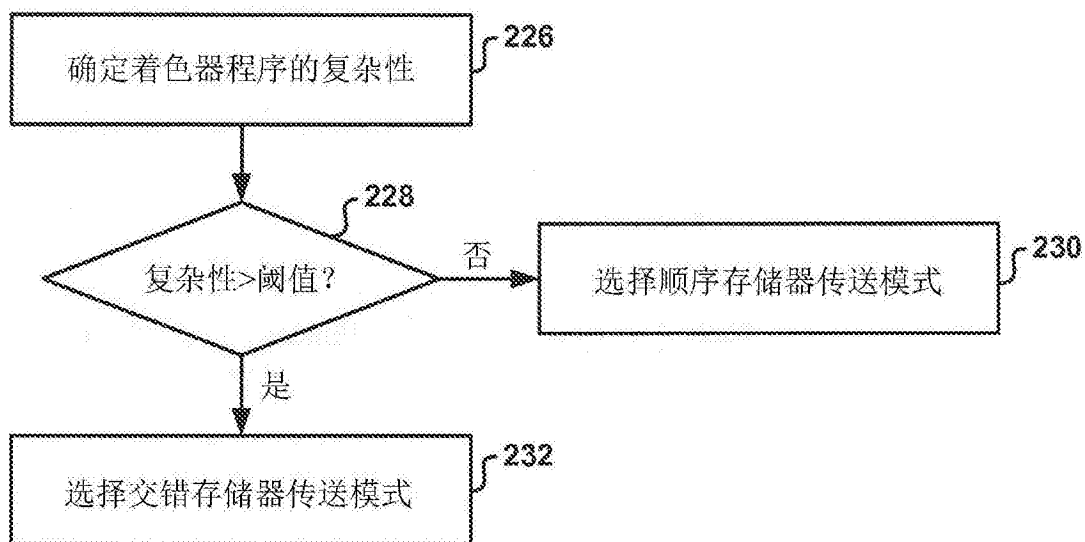


图24

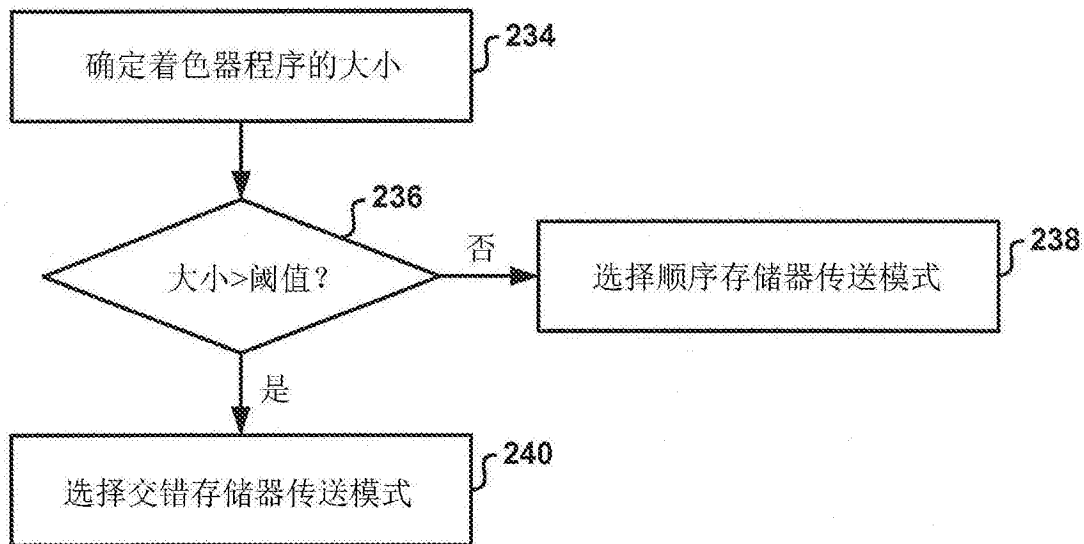


图25

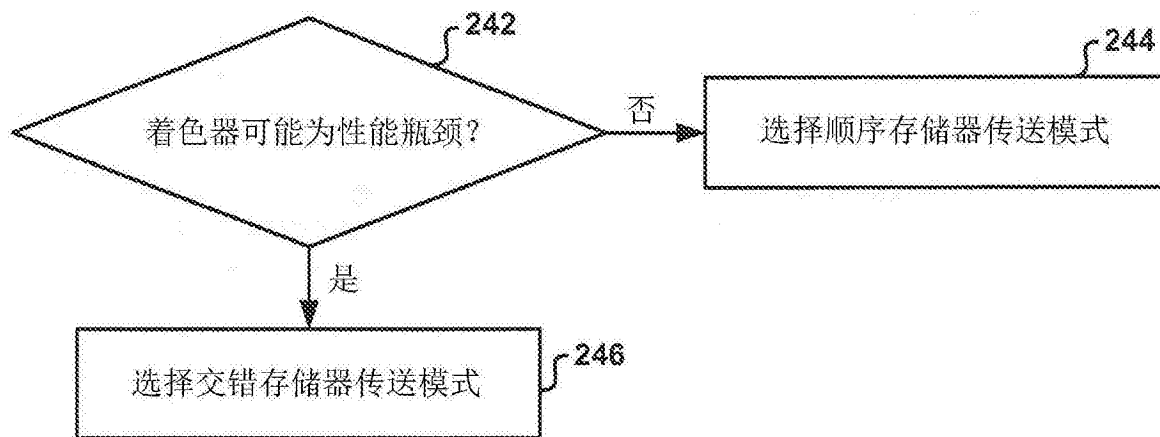


图26

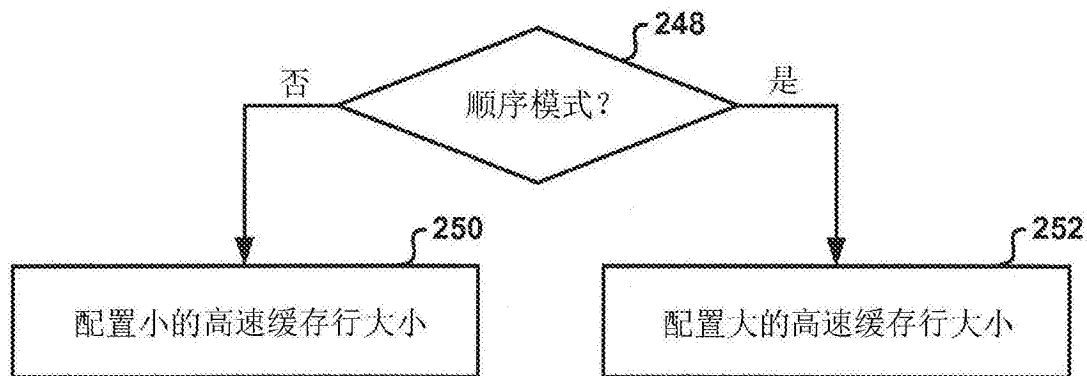


图27