



(12)发明专利

(10)授权公告号 CN 104641396 B

(45)授权公告日 2018.04.10

(21)申请号 201380046499.3

(22)申请日 2013.06.20

(65)同一申请的已公布的文献号
申请公布号 CN 104641396 A

(43)申请公布日 2015.05.20

(30)优先权数据
13/554,805 2012.07.20 US

(85)PCT国际申请进入国家阶段日
2015.03.06

(86)PCT国际申请的申请数据
PCT/US2013/046844 2013.06.20

(87)PCT国际申请的公布数据
W02014/014614 EN 2014.01.23

(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 埃杜瓦杜斯·A·梅茨
奈杰尔·特伦斯·普尔

科林·克里斯托弗·夏普
安德鲁·格鲁伯

(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.
G06T 1/20(2006.01)
G06F 9/48(2006.01)

(56)对比文件
CN 1916962 A, 2007.02.21,
US 2011072211 A1, 2011.03.24,
US 6809735 B1, 2004.10.26,
US 2011050713 A1, 2011.03.03,
SHINPEI KATO等, RGEM: A Responsive
GPGPU Execution Model for Runtime
Engines.《2011 IEEE 32ND REAL-TIME SYSTEMS
SYMPOSIUM》. 2011, 57-66.

审查员 刘晓丹

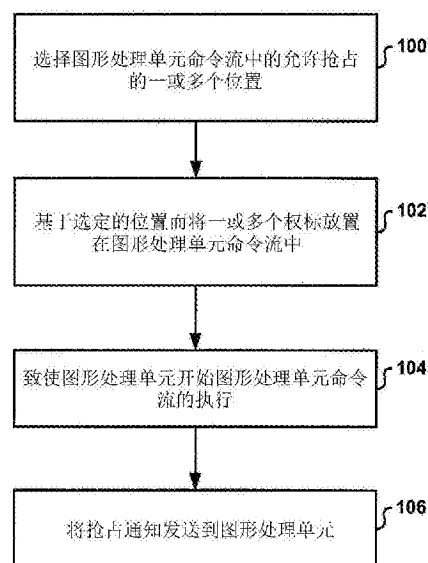
权利要求书10页 说明书21页 附图9页

(54)发明名称

用于调度图形处理单元命令流的延迟抢占技术

(57)摘要

本发明针对用于调度用于在图形处理单元GPU上执行的GPU命令流的延迟抢占技术。描述经配置以控制GPU来执行延迟抢占调度的主机CPU。举例来说,主机CPU可响应于接收到抢占通知而选择GPU命令流中的一或多个位置以作为在其处允许发生抢占的一或多个位置,且可基于所述选定的一或多个位置将一或多个权标放置在所述GPU命令流中。所述权标可向所述GPU指示允许在所述选定的一或多个位置处发生抢占。本发明进一步描述经配置以基于放置在GPU命令流中的一或多个权标而抢占GPU命令流的执行的GPU。



1. 一种用于图形处理的方法,其包括:

使用处理器选择第一图形处理单元GPU命令流中的一或多个位置以作为在其处允许处理所述第一GPU命令流的GPU响应于从所述处理器接收到抢占所述第一GPU命令流的执行的通知而抢占所述第一GPU命令流的执行且响应于从所述处理器接收到抢占所述第一GPU命令流的执行的所述通知而从执行所述第一GPU命令流切换到执行第二GPU命令流的一或多个位置;及

使用所述处理器基于所述选定的一或多个位置而将一或多个权标放置在所述第一GPU命令流中,所述一或多个权标向所述GPU指示允许在所述第一GPU命令流中的所述选定的一或多个位置处发生抢占。

2. 根据权利要求1所述的方法,其中选择所述一或多个位置包括:

基于用于在所述第一GPU命令流中的所述一或多个位置中的每一者处执行GPU背景切换的所估计成本而选择所述第一GPU命令流中的所述一或多个位置。

3. 根据权利要求2所述的方法,其中用于执行所述GPU背景切换的所述所估计成本是以下各者中的至少一者的函数:执行所述GPU背景切换所需的处理时间量、执行所述GPU背景切换所需的存储器带宽量及执行所述GPU背景切换所需的存储器量。

4. 根据权利要求1所述的方法,其中选择所述一或多个位置包括:

基于指示所述第一GPU命令流中的命令的分组的信息而选择所述第一GPU命令流中的所述一或多个位置,命令的所述分组中的每一者与多个图形帧中的相应一者的再现、多个GPU内核中的相应一者的执行及多个GPU任务中的相应一者的执行中的至少一者相关联。

5. 根据权利要求1所述的方法,其中选择所述一或多个位置包括:

基于从与所述第一GPU命令流相关联的软件应用接收的指示其中应发生抢占的特定命令及其中不应发生抢占的特定命令中的至少一者的信息而选择所述第一GPU命令流中的所述一或多个位置。

6. 根据权利要求1所述的方法,其中选择所述一或多个位置包括:

选择所述第一GPU命令流中的多个位置以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置。

7. 根据权利要求1所述的方法,其中选择所述一或多个位置包括:

选择一或多个命令边界以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置。

8. 根据权利要求1所述的方法,其中选择所述一或多个位置包括:

选择一或多个命令范围以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置。

9. 根据权利要求1所述的方法,其中选择所述一或多个位置包括:

选择一或多个命令范围以作为在其处不允许所述GPU抢占所述第一GPU命令流的执行的位置;及

基于所述选定的一或多个命令范围而选择所述一或多个位置以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置。

10. 根据权利要求1所述的方法,其进一步包括:

将所述第一GPU命令流放置到可由所述GPU存取的存储器中。

11. 根据权利要求1所述的方法,其进一步包括:

致使所述GPU开始执行所述第一GPU命令流;及

在所述第一GPU命令流在所述GPU上执行时,基于放置在所述第一GPU命令流中的所述一或多个权标向所述GPU发送通知,所述通知指示所述第二GPU命令流即将在所述GPU上执行且所述GPU应抢占在所述GPU上执行的具有比所述第二GPU命令流的调度优先级低的调度优先级的任何命令流,所述第二GPU命令流具有比所述第一GPU命令流的调度优先级大的调度优先级。

12. 根据权利要求1所述的方法,其进一步包括:

使用GPU基于放置在所述第一GPU命令流中的所述一或多个权标而抢占所述第一GPU命令流的执行。

13. 根据权利要求1所述的方法,其进一步包括:

使用GPU分派所述第一GPU命令流中的一或多个命令以用于在所述GPU的一或多个处理单元上执行;

使用所述GPU接收指示所述第二GPU命令流即将在所述GPU上执行的通知,所述第二GPU命令流具有比所述第一GPU命令流的调度优先级大的调度优先级;

响应于接收到所述通知,使用所述GPU分派所述第一GPU命令流中的命令以用于在所述一或多个处理单元上执行,直到在所述第一GPU命令流中遇到所述一或多个权标中的一权标为止;

使用所述GPU响应于遇到所述第一GPU命令流中的所述权标而保存与所述第一GPU命令流的执行相关联的所述GPU的状态;及

使用所述GPU响应于遇到所述第一GPU命令流中的所述权标而分派所述第二GPU命令流中的一或多个命令以用于在所述一或多个处理单元上执行。

14. 根据权利要求13所述的方法,其进一步包括:

在完成所述第二GPU命令流的执行之后恢复所述GPU的所述保存的状态。

15. 一种用于图形处理的装置,其包括:

处理器,其经配置以:

选择第一图形处理单元GPU命令流中的一或多个位置以作为在其处允许处理所述第一GPU命令流的GPU响应于从所述处理器接收到抢占所述第一GPU命令流的执行的通知而抢占所述第一GPU命令流的执行且响应于从所述处理器接收到抢占所述第一GPU命令流的执行的通知而从执行所述第一GPU命令流切换到执行第二GPU命令流的一或多个位置;及

基于所述选定的一或多个位置而将一或多个权标放置在所述第一GPU命令流中,所述一或多个权标向所述GPU指示允许在所述第一GPU命令流中的所述选定的一或多个位置处发生抢占。

16. 根据权利要求15所述的装置,其中所述处理器进一步经配置以基于在所述第一GPU命令流中的所述一或多个位置中的每一者处执行GPU背景切换的所估计成本而选择所述第一GPU命令流中的所述一或多个位置。

17. 根据权利要求16所述的装置,其中用于执行所述GPU背景切换的所述所估计成本是以下各者中的至少一者的函数:执行所述GPU背景切换所需的处理时间量、执行所述GPU背景切换所需的存储器带宽量及执行所述GPU背景切换所需的存储器量。

18. 根据权利要求15所述的装置, 其中所述处理器进一步经配置以基于指示所述第一GPU命令流中的命令的分组的信息而选择所述第一GPU命令流中的所述一或多个位置, 命令的所述分组中的每一者与多个图形帧中的相应一者的再现、多个GPU内核中的相应一者的执行及多个GPU任务中的相应一者的执行中的至少一者相关联。

19. 根据权利要求15所述的装置, 其中所述处理器进一步经配置以基于从与所述第一GPU命令流相关联的软件应用接收的指示其中应发生抢占的特定命令及其中不应发生抢占的特定命令中的至少一者的信息而选择所述第一GPU命令流中的所述一或多个位置。

20. 根据权利要求15所述的装置, 其中所述处理器进一步经配置以选择所述第一GPU命令流中的多个位置以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置。

21. 根据权利要求15所述的装置, 其中所述处理器进一步经配置以选择一或多个命令边界以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置。

22. 根据权利要求15所述的装置, 其中所述处理器进一步经配置以选择一或多个命令范围以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置。

23. 根据权利要求15所述的装置, 其中所述处理器进一步经配置以: 选择一或多个命令范围以作为在其处不允许所述GPU抢占所述第一GPU命令流的执行的位置; 及基于所述选定的一或多个命令范围而选择所述一或多个位置以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置。

24. 根据权利要求15所述的装置, 其中所述处理器进一步经配置以将所述第一GPU命令流放置到可由所述GPU存取的存储器中。

25. 根据权利要求15所述的装置, 其中所述处理器进一步经配置以:

致使所述GPU开始执行所述第一GPU命令流; 及

在所述第一GPU命令流在所述GPU上执行时, 基于放置在所述第一GPU命令流中的所述一或多个权标向所述GPU发送通知, 所述通知指示所述第二GPU命令流即将在所述GPU上执行且所述GPU应抢占在所述GPU上执行的具有比所述第二GPU命令流的调度优先级低的调度优先级的任何命令流, 所述第二GPU命令流具有比所述第一GPU命令流的调度优先级大的调度优先级。

26. 根据权利要求15所述的装置, 其进一步包括:

GPU, 其经配置以基于放置在所述第一GPU命令流中的所述一或多个权标而抢占所述第一GPU命令流的执行。

27. 根据权利要求15所述的装置, 其中所述GPU进一步经配置以:

分派所述第一GPU命令流中的一或多个命令以用于在所述GPU的一或多个处理单元上执行;

接收指示所述第二GPU命令流即将在所述GPU上执行的通知, 所述第二GPU命令流具有比所述第一GPU命令流的调度优先级大的调度优先级;

响应于接收到所述通知, 分派所述第一GPU命令流中的命令以用于在所述一或多个处理单元上执行, 直到在所述第一GPU命令流中遇到所述一或多个权标中的一权标为止;

响应于遇到所述第一GPU命令流中的所述权标而保存与所述第一GPU命令流的执行相关联的所述GPU的状态; 及

响应于遇到所述第一GPU命令流中的所述权标而分派所述第二GPU命令流中的一或多

个命令以用于在所述一或多个处理单元上执行。

28. 根据权利要求27所述的装置, 其中所述GPU进一步经配置以在完成所述第二GPU命令流的执行之后恢复所述GPU的所述保存的状态。

29. 根据权利要求15所述的装置, 其中所述装置包括无线通信装置。

30. 根据权利要求15所述的装置, 其中所述装置包括移动电话手持机。

31. 一种用于图形处理的设备, 其包括:

用于选择第一图形处理单元GPU命令流中的一或多个位置以作为在其处允许处理所述第一GPU命令流的GPU响应于接收到抢占所述第一GPU命令流的执行的通知而抢占所述第一GPU命令流的执行的一或多个位置且响应于从所述处理器接收到抢占所述第一GPU命令流的执行的通知而从执行所述第一GPU命令流切换到执行第二GPU命令流的装置; 及

用于基于所述选定的一或多个位置而将一或多个权标放置在所述第一GPU命令流中的装置, 所述一或多个权标向所述GPU指示允许在所述第一GPU命令流中的所述选定的一或多个位置处发生抢占。

32. 根据权利要求31所述的设备, 其中所述用于选择所述一或多个位置的装置包括:

用于基于在所述第一GPU命令流中的所述一或多个位置中的每一者处执行GPU背景切换的所估计成本而选择所述第一GPU命令流中的所述一或多个位置的装置。

33. 根据权利要求32所述的设备, 其中用于执行所述GPU背景切换的所述所估计成本是以下各者中的至少一者的函数: 执行所述GPU背景切换所需的处理时间量、执行所述GPU背景切换所需的存储器带宽量及执行所述GPU背景切换所需的存储器量。

34. 根据权利要求31所述的设备, 其中所述用于选择所述一或多个位置的装置包括:

用于基于指示所述第一GPU命令流中的命令的分组的信息而选择所述第一GPU命令流中的所述一或多个位置的装置, 命令的所述分组中的每一者与多个图形帧中的相应一者的再现、多个GPU内核中的相应一者的执行及多个GPU任务中的相应一者的执行中的至少一者相关联。

35. 根据权利要求31所述的设备, 其中所述用于选择所述一或多个位置的装置包括:

用于选择一或多个命令边界以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置的装置。

36. 根据权利要求31所述的设备, 其中所述用于选择所述一或多个位置的装置包括:

用于选择一或多个命令范围以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置的装置。

37. 根据权利要求31所述的设备, 其进一步包括:

用于致使所述GPU开始执行所述第一GPU命令流的装置; 及

用于在所述第一GPU命令流在所述GPU上执行时基于放置在所述第一GPU命令流中的所述一或多个权标向所述GPU发送通知的装置, 所述通知指示所述第二GPU命令流即将在所述GPU上执行且所述GPU应抢占在所述GPU上执行的具有比所述第二GPU命令流的调度优先级低的调度优先级的任何命令流, 所述第二GPU命令流具有比所述第一GPU命令流的调度优先级大的调度优先级。

38. 根据权利要求31所述的设备, 其进一步包括:

用于基于放置在所述第一GPU命令流中的所述一或多个权标而抢占所述第一GPU命令

流的执行的装置。

39. 根据权利要求31所述的设备,其进一步包括:

用于分派所述第一GPU命令流中的一或多个命令以用于在所述GPU的一或多个处理单元上执行的装置;

用于接收指示所述第二GPU命令流即将在所述GPU上执行的通知的装置,所述第二GPU命令流具有比所述第一GPU命令流的调度优先级大的调度优先级;

用于以下操作的装置:响应于接收到所述通知,分派所述第一GPU命令流中的命令以用于在所述一或多个处理单元上执行,直到在所述第一GPU命令流中遇到所述一或多个权标中的一权标为止;

用于响应于遇到所述第一GPU命令流中的所述权标而保存与所述第一GPU命令流的执行相关联的所述GPU的状态的装置;及

用于响应于遇到所述第一GPU命令流中的所述权标而分派所述第二GPU命令流中的一或多个命令以用于在所述一或多个处理单元上执行的装置。

40. 一种存储指令的计算机可读存储媒体,所述指令在被执行时致使一或多个处理器进行以下操作:

选择第一图形处理单元GPU命令流中的一或多个位置以作为在其处允许处理所述第一GPU命令流的GPU响应于从所述一或多个处理器接收到抢占所述第一GPU命令流的执行的通知而抢占所述第一GPU命令流的执行且响应于从所述处理器接收到抢占所述第一GPU命令流的执行的通知而从执行所述第一GPU命令流切换到执行第二GPU命令流的一或多个位置;及

基于所述选定的一或多个位置而将一或多个权标放置在所述第一GPU命令流中,所述一或多个权标向所述GPU指示允许在所述第一GPU命令流中的所述选定的一或多个位置处发生抢占。

41. 根据权利要求40所述的计算机可读存储媒体,其中所述致使所述一或多个处理器选择所述一或多个位置的指令包括在被执行时致使所述一或多个处理器进行以下操作的指令:

基于在所述第一GPU命令流中的所述一或多个位置中的每一者处执行GPU背景切换的所估计成本而选择所述第一GPU命令流中的所述一或多个位置。

42. 根据权利要求41所述的计算机可读存储媒体,其中用于执行所述GPU背景切换的所述所估计成本是以下各者中的至少一者的函数:执行所述GPU背景切换所需的处理时间量、执行所述GPU背景切换所需的存储器带宽量及执行所述GPU背景切换所需的存储器量。

43. 根据权利要求40所述的计算机可读存储媒体,其中所述致使所述一或多个处理器选择所述一或多个位置的指令包括在被执行时致使所述一或多个处理器进行以下操作的指令:

基于指示所述第一GPU命令流中的命令的分组的信息而选择所述第一GPU命令流中的所述一或多个位置,命令的所述分组中的每一者与多个图形帧中的相应一者的再现、多个GPU内核中的相应一者的执行及多个GPU任务中的相应一者的执行中的至少一者相关联。

44. 根据权利要求40所述的计算机可读存储媒体,其中所述致使所述一或多个处理器选择所述一或多个位置的指令包括在被执行时致使所述一或多个处理器进行以下操作的

指令：

选择一或多个命令边界以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置。

45. 根据权利要求40所述的计算机可读存储媒体，其中所述致使所述一或多个处理器选择所述一或多个位置的指令包括在被执行时致使所述一或多个处理器进行以下操作的指令：

选择一或多个命令范围以作为在其处允许所述GPU抢占所述第一GPU命令流的执行的位置。

46. 根据权利要求40所述的计算机可读存储媒体，其中所述计算机可读存储媒体进一步存储在被执行时致使一或多个处理器进行以下操作的指令：

致使所述GPU开始执行所述第一GPU命令流；及

在所述第一GPU命令流在所述GPU上执行时，基于放置在所述第一GPU命令流中的所述一或多个权标向所述GPU发送通知，所述通知指示所述第二GPU命令流即将在所述GPU上执行且所述GPU应抢占在所述GPU上执行的具有比所述第二GPU命令流的调度优先级低的调度优先级的任何命令流，所述第二GPU命令流具有比所述第一GPU命令流的调度优先级大的调度优先级。

47. 一种用于图形处理的方法，其包括：

使用图形处理单元GPU确定GPU命令流中的命令时隙是否含有包含在所述GPU命令流中的一或多个权标中的一权标，所述一或多个权标指示允许在所述GPU命令流中的一或多个位置处发生抢占；

使用所述GPU确定是否已从主机装置接收到抢占通知；及

使用所述GPU响应于确定已接收到所述抢占通知且所述命令时隙含有所述一或多个权标中的所述权标而抢占所述GPU命令流的执行。

48. 根据权利要求47所述的方法，

其中所述GPU命令流是第一GPU命令流，

其中所述方法进一步包括：

使用所述GPU分派所述第一GPU命令流中的一或多个命令以用于在所述GPU的一或多个处理单元上执行；

使用所述GPU从所述主机装置接收指示第二GPU命令流即将在所述GPU上执行的抢占通知，所述第二GPU命令流具有比所述第一GPU命令流的调度优先级大的调度优先级；及

响应于接收到所述抢占通知，使用所述GPU分派所述第一GPU命令流中的命令以用于在所述一或多个处理单元上执行，直到在所述第一GPU命令流中遇到所述一或多个权标中的所述权标为止，

其中抢占所述GPU命令流的所述执行包括：

使用所述GPU响应于遇到所述第一GPU命令流中的所述权标而保存与所述第一GPU命令流的执行相关联的所述GPU的状态；及

使用所述GPU响应于遇到所述第一GPU命令流中的所述权标而分派所述第二GPU命令流中的一或多个命令以用于在所述一或多个处理单元上执行。

49. 根据权利要求47所述的方法，其进一步包括：

响应于确定尚未接收到抢占通知或所述命令时隙不含有所述一或多个权标中的一权标,分派所述命令时隙中的命令以用于在所述GPU的一或多个处理单元上执行而不在当前命令处理循环期间抢占所述GPU命令流的执行。

50. 根据权利要求47所述的方法,其进一步包括:

基于指示在当前命令处理循环内是否允许抢占的抢占允许状态变量而确定在所述当前命令处理循环内是否允许抢占,所述抢占允许状态变量的值响应于处理所述GPU命令流中的所述一或多个权标中的一或多者而改变;及

确定是否已从所述主机装置接收到抢占通知,

其中抢占所述GPU命令流的所述执行包括响应于确定已接收到抢占通知且在所述当前命令处理循环内允许抢占而抢占所述GPU命令流的执行。

51. 根据权利要求50所述的方法,其进一步包括:

响应于确定尚未接收到抢占通知或在当前命令处理循环内不允许抢占,分派所述命令时隙中的命令以用于在所述GPU的一或多个处理单元上执行而不在所述当前命令处理循环期间抢占所述GPU命令流的执行。

52. 一种用于图形处理的设备,其包括:

图形处理单元GPU,其经配置以:

确定GPU命令流中的命令时隙是否含有包含在所述GPU命令流中的一或多个权标中的一权标,所述一或多个权标指示允许在所述GPU命令流中的一或多个位置处发生抢占;

确定是否已从主机装置接收到抢占通知;及

响应于确定已接收到所述抢占通知且所述命令时隙含有所述一或多个权标中的所述权标而基于所述一或多个权标抢占所述GPU命令流的执行。

53. 根据权利要求52所述的设备,其中所述GPU命令流是第一GPU命令流,且其中所述GPU进一步经配置以:

分派所述第一GPU命令流中的一或多个命令以用于在所述GPU的一或多个处理单元上执行;

从所述主机装置接收指示第二GPU命令流即将在所述GPU上执行的所述抢占通知,所述第二GPU命令流具有比所述第一GPU命令流的调度优先级大的调度优先级;

响应于接收到所述抢占通知,分派所述第一GPU命令流中的命令以用于在所述一或多个处理单元上执行,直到在所述第一GPU命令流中遇到所述一或多个权标中的所述权标为止;

响应于遇到所述第一GPU命令流中的所述权标而保存与所述第一GPU命令流的执行相关联的所述GPU的状态;及

响应于遇到所述第一GPU命令流中的所述权标而分派所述第二GPU命令流中的一或多个命令以用于在所述一或多个处理单元上执行。

54. 根据权利要求52所述的设备,其中所述GPU进一步经配置以响应于确定尚未接收到抢占通知或所述命令时隙不含有所述一或多个权标中的一权标,分派所述命令时隙中的命令以用于在所述GPU的一或多个处理单元上执行而不在当前命令处理循环期间抢占所述GPU命令流的执行。

55. 根据权利要求52所述的设备,其中所述GPU进一步经配置以:

基于指示在当前命令处理循环内是否允许抢占的抢占允许状态变量而确定在所述当前命令处理循环内是否允许抢占,所述抢占允许状态变量的值响应于处理所述GPU命令流中的所述一或多个权标中的一或多者而改变;

确定是否已从所述主机装置接收到抢占通知;及

响应于确定已接收到抢占通知且在所述当前命令处理循环内允许抢占而抢占所述GPU命令流的执行。

56. 根据权利要求55所述的设备,其中所述GPU进一步经配置以响应于确定尚未接收到抢占通知或在当前命令处理循环内不允许抢占,分派所述命令时隙中的命令以用于在所述GPU的一或多个处理单元上执行而不在所述当前命令处理循环期间抢占所述GPU命令流的执行。

57. 根据权利要求52所述的设备,其中所述设备包括无线通信装置。

58. 根据权利要求52所述的设备,其中所述设备包括移动电话手持机。

59. 一种用于图形处理的设备,其包括:

用于执行图形处理单元GPU命令流的装置,所述GPU命令流包含指示允许在所述GPU命令流中的一或多个位置处发生抢占的一或多个权标;

用于确定所述GPU命令流中的命令时隙是否含有包含在所述GPU命令流中的所述一或多个权标中的一权标的装置;

用于确定是否已从主机装置接收到抢占通知的装置;及

用于响应于确定已接收到所述抢占通知且所述命令时隙含有所述一或多个权标中的所述权标而抢占所述GPU命令流的执行的装置。

60. 根据权利要求59所述的设备,

其中所述GPU命令流是第一GPU命令流,

其中所述设备进一步包括:

用于分派所述第一GPU命令流中的一或多个命令以用于在所述GPU的一或多个处理单元上执行的装置;

用于使用所述GPU从所述主机装置接收指示第二GPU命令流即将在所述GPU上执行的所述抢占通知的装置,所述第二GPU命令流具有比所述第一GPU命令流的调度优先级大的调度优先级;及

用于以下操作的装置:响应于接收到所述抢占通知,分派所述第一GPU命令流中的命令以用于在所述一或多个处理单元上执行,直到在所述第一GPU命令流中遇到所述一或多个权标中的所述权标为止,

其中所述用于抢占所述GPU命令流的所述执行的装置包括:

用于响应于遇到所述第一GPU命令流中的所述权标而保存与所述第一GPU命令流的执行相关联的所述GPU的状态的装置;及

用于响应于遇到所述第一GPU命令流中的所述权标而分派所述第二GPU命令流中的一或多个命令以用于在所述一或多个处理单元上执行的装置。

61. 根据权利要求59所述的设备,其进一步包括:

用于以下操作的装置:响应于确定尚未接收到抢占通知或所述命令时隙不含有所述一或多个权标中的一权标,分派所述命令时隙中的命令以用于在所述GPU的一或多个处理单

元上执行而不在当前命令处理循环期间抢占所述GPU命令流的执行。

62. 根据权利要求59所述的设备,其进一步包括:

用于基于指示在当前命令处理循环内是否允许抢占的抢占允许状态变量而确定在所述当前命令处理循环内是否允许抢占的装置,所述抢占允许状态变量的值响应于处理所述GPU命令流中的所述一或多个权标中的一或多者而改变;及

用于确定是否已从所述主机装置接收到抢占通知的装置,

其中所述用于抢占所述GPU命令流的所述执行的装置包括用于响应于确定已接收到抢占通知且在所述当前命令处理循环内允许抢占而抢占所述GPU命令流的执行的装置。

63. 根据权利要求62所述的设备,其进一步包括:

用于以下操作的装置:响应于确定尚未接收到抢占通知或在当前命令处理循环内不允许抢占,分派所述命令时隙中的命令以用于在所述GPU的一或多个处理单元上执行而不在所述当前命令处理循环期间抢占所述GPU命令流的执行。

64. 一种存储指令的计算机可读存储媒体,所述指令在被执行时致使图形处理单元GPU进行以下操作:

确定GPU命令流中的命令时隙是否含有包含在所述GPU命令流中的一或多个权标中的一权标,所述一或多个权标指示允许在所述GPU命令流中的一或多个位置处发生抢占;

确定是否已从主机装置接收到抢占通知;及

响应于确定已接收到所述抢占通知且所述命令时隙含有所述一或多个权标中的所述权标而抢占所述GPU命令流的执行。

65. 根据权利要求64所述的计算机可读存储媒体,其中所述GPU命令流是第一GPU命令流,且其中所述计算机可读存储媒体进一步包括在被执行时致使所述GPU进行以下操作的指令:

分派所述第一GPU命令流中的一或多个命令以用于在所述GPU的一或多个处理单元上执行;

从所述主机装置接收指示第二GPU命令流即将在所述GPU上执行的所述抢占通知,所述第二GPU命令流具有比所述第一GPU命令流的调度优先级大的调度优先级;

响应于接收到所述抢占通知,分派所述第一GPU命令流中的命令以用于在所述一或多个处理单元上执行,直到在所述第一GPU命令流中遇到所述一或多个权标中的所述权标为止;

响应于遇到所述第一GPU命令流中的所述权标而保存与所述第一GPU命令流的执行相关联的所述GPU的状态;及

响应于遇到所述第一GPU命令流中的所述权标而分派所述第二GPU命令流中的一或多个命令以用于在所述一或多个处理单元上执行。

66. 根据权利要求64所述的计算机可读存储媒体,其进一步包括在被执行时致使所述GPU进行以下操作的指令:

响应于确定尚未接收到抢占通知或所述命令时隙不含有所述一或多个权标中的一权标,分派所述命令时隙中的命令以用于在所述GPU的一或多个处理单元上执行而不在当前命令处理循环期间抢占所述GPU命令流的执行。

67. 根据权利要求64所述的计算机可读存储媒体,其进一步包括在被执行时致使所述

GPU进行以下操作的指令：

基于指示在当前命令处理循环内是否允许抢占的抢占允许状态变量而确定在所述当前命令处理循环内是否允许抢占，所述抢占允许状态变量的值响应于处理所述GPU命令流中的所述一或多个权标中的一或多个者而改变；

确定是否已从所述主机装置接收到抢占通知；及

响应于确定已接收到抢占通知且在所述当前命令处理循环内允许抢占而抢占所述GPU命令流的执行。

68. 根据权利要求67所述的计算机可读存储媒体，其进一步包括在被执行时致使所述GPU进行以下操作的指令：

响应于确定尚未接收到抢占通知或在当前命令处理循环内不允许抢占，分派所述命令时隙中的命令以用于在所述GPU的一或多个处理单元上执行而不在所述当前命令处理循环期间抢占所述GPU命令流的执行。

用于调度图形处理单元命令流的延迟抢占技术

技术领域

[0001] 本发明涉及图形处理系统,且更确切地说,涉及利用命令流的图形处理系统。

背景技术

[0002] 计算装置常常利用图形处理单元 (GPU) 以加速再现用于显示的图形数据。此类计算装置可包含 (例如) 计算机工作站、例如所谓的智能电话等移动电话、嵌入系统、个人计算机、平板计算机及视频游戏控制台。GPU 通常执行图形处理管线,所述图形处理管线包含一起操作以执行图形处理命令的多个处理级。主机中央处理单元 (CPU) 可通过将一或多个图形处理命令发布给 GPU 来控制 GPU 的操作。现代的 CPU 通常能够同时执行多个应用,所述多个应用中的每一者可需要在执行期间利用 GPU。在当前在主机 CPU 上执行的不同应用之间的 GPU 资源的仲裁可向主机 CPU 提出显著挑战,尤其是在特定应用可需要对 GPU 的高优先级存取的情况下。

发明内容

[0003] 本发明针对用于调度用于在图形处理单元 (GPU) 上执行的 GPU 命令流的延迟抢占技术。在一些实例中,本发明的技术可包含将一或多个权标放置在 GPU 命令流中以指示 GPU 命令流中的允许发生抢占的特定位置。在 GPU 处理所述 GPU 命令流且从主机装置接收抢占所述 GPU 命令流的执行以便执行另一 GPU 命令流的通知时, GPU 可继续处理所述 GPU 命令流中的命令,直到所述 GPU 达到所述 GPU 命令流中的允许发生抢占的位置为止。在到达允许发生抢占的位置之后,所述 GPU 可即刻切换到执行其它 GPU 命令流。在一些实例中,所述 GPU 命令流可与不同的图形处理情境相关联。

[0004] 在一个实例中,本发明描述一种方法,其包含:使用处理器选择 GPU 命令流中的一或多个位置以作为在其处允许处理所述 GPU 命令流的 GPU 响应于从所述处理器接收到抢占所述 GPU 命令流的执行的通知而抢占所述 GPU 命令流的执行的一或多个位置。所述方法进一步包含使用所述处理器基于所述选定的一或多个位置而将一或多个权标放置在所述 GPU 命令流中。所述一或多个权标向所述 GPU 指示允许在所述 GPU 命令流中的所述选定的一或多个位置处发生抢占。

[0005] 在另一实例中,本发明描述一种装置,其包含处理器,所述处理器经配置以:选择 GPU 命令流中的一或多个位置以作为在其处允许处理所述 GPU 命令流的 GPU 响应于从所述处理器接收到抢占所述 GPU 命令流的执行的通知而抢占所述 GPU 命令流的执行的一或多个位置。所述处理器进一步经配置以基于所述选定的一或多个位置而将一或多个权标放置在所述 GPU 命令流中。所述一或多个权标向所述 GPU 指示允许在所述 GPU 命令流中的所述选定的一或多个位置处发生抢占。

[0006] 在另一实例中,本发明描述一种设备,其包含:用于选择 GPU 命令流中的一或多个位置以作为在其处允许处理所述 GPU 命令流的 GPU 响应于接收到抢占所述 GPU 命令流的执行的通知而抢占所述 GPU 命令流的执行的一或多个位置的装置。所述设备进一步包含用于基

于所述选定的一或多个位置而将一或多个权标放置在所述GPU命令流中的装置。所述一或多个权标向所述GPU指示允许在所述GPU命令流中的所述选定的一或多个位置处发生抢占。

[0007] 在另一实例中,本发明描述一种存储指令的计算机可读存储媒体,所述指令在被执行时致使一或多个处理器选择GPU命令流中的一或多个位置以作为在其处允许处理所述GPU命令流的GPU响应于从所述一或多个处理器接收到抢占所述GPU命令流的执行的通知而抢占所述GPU命令流的执行的一或多个位置。所述指令进一步致使所述一或多个处理器基于所述选定的一或多个位置而将一或多个权标放置在所述GPU命令流中。所述一或多个权标向所述GPU指示允许在所述GPU命令流中的所述选定的一或多个位置处发生抢占。

[0008] 在另一实例中,本发明描述一种方法,其包含使用GPU基于包含在所述GPU命令流中的一或多个权标而抢占GPU命令流的执行。所述一或多个权标指示允许在所述GPU命令流中的一或多个位置处发生抢占。

[0009] 在另一实例中,本发明描述一种设备,其包含使用GPU,所述GPU经配置以基于包含在所述GPU命令流中的一或多个权标而抢占GPU命令流的执行。所述一或多个权标指示允许在所述GPU命令流中的一或多个位置处发生抢占。

[0010] 在另一实例中,本发明描述一种设备,其包含用于执行GPU命令流的装置。所述设备进一步包含用于基于包含在所述GPU命令流中的一或多个权标而抢占所述GPU命令流的执行的装置。所述一或多个权标指示允许在所述GPU命令流中的一或多个位置处发生抢占。

[0011] 在另一实例中,本发明描述一种存储指令的计算机可读存储媒体,所述指令在被执行时致使GPU基于包含在所述GPU命令流中的一或多个权标而抢占GPU命令流的执行。所述一或多个权标指示允许在所述GPU命令流中的一或多个位置处发生抢占。

[0012] 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

[0013] 图1是说明可用于实施本发明的延迟抢占GPU命令流调度技术的实例计算装置的框图。

[0014] 图2是进一步详细说明图1中的计算装置的CPU、GPU和存储器的框图。

[0015] 图3是说明根据本发明的可将权标放置到其中的GPU命令流的概念图。

[0016] 图4是说明根据本发明的用于将权标放置在图3的GPU命令流中的技术的概念图。

[0017] 图5是说明根据本发明的用于将权标放置在图3的GPU命令流中的另一实例技术的概念图。

[0018] 图6是说明根据本发明的用于控制GPU以执行延迟抢占调度的实例技术的流程图。

[0019] 图7是说明根据本发明的用于基于放置在GPU命令流中的权标而抢占命令流的执行的实例技术的流程图。

[0020] 图8是说明根据本发明的用于处理GPU命令流的实例技术的流程图。

[0021] 图9是说明根据图8中展示的技术的用于在处理命令流时处置抢占通知的实例技术的流程图。

[0022] 图10是说明根据本发明的用于处理GPU命令流的另一实例技术的流程图。

[0023] 图11是说明根据图10中展示的技术的用于在处理命令流时处置抢占通知的实例

技术的流程图。

具体实施方式

[0024] 本发明针对用于调度用于在图形处理单元 (GPU) 上执行的GPU命令流的延迟抢占技术。如今, GPU越来越多地用于用户接口 (UI) 再现。通常需要以及及时的方式服务UI命令流以便实现用户在UI中预期的恰当视觉效果及响应性。在高优先级UI命令流被主机CPU排队在GPU上执行时, GPU可执行与具有较低优先级的不同背景相关联的另一排队的命令流, 例如, 非UI图形背景或使用GPU执行通用计算任务 (即, 图形处理单元通用计算 (GPGPU) 任务) 的背景。在一些情况下, 在执行较高优先级UI命令流之前等待较低优先级背景完成执行可不产生可接受的用户对UI的体验。

[0025] 此问题的一个解决方案是每当主机CPU确定较高优先级的命令流需要执行时允许主机CPU抢占较低优先级命令流的执行且服务较高优先级的命令流。然而, 此解决方案在于GPU命令流中的任意点处在GPU中执行背景切换所需的开销 (例如, 额外处理时间、额外存储器带宽使用及/或额外存储器使用) 方面可为昂贵的。举例来说, 每当较高优先级的命令流需要抢占较低优先级的命令流时, 可需要在执行较高优先级的命令流之前保存较低优先级的命令流的GPU状态, 且随后在执行较高优先级的命令流之后恢复。执行此类背景切换所需的额外开销在GPU命令流中的特定位置可比在GPU命令流中其它位置更昂贵。因此, 允许在GPU命令流中的任意点处发生抢占可导致在其中归因于背景切换而引起的开销相对昂贵的位置处对GPU命令流的抢占, 进而降低GPU的整体性能。

[0026] 根据本发明中描述的各种实例, 针对图形处理系统提供延迟抢占调度技术。延迟抢占调度技术可用于允许较高优先级的命令流在较低优先级的命令流内的特定抢占点处抢占较低优先级的命令流的执行。也就是说, 本发明的技术可允许主机CPU将一或多个权标放置在较低优先级的命令流中以指示较低优先级的命令流中的允许在其处发生抢占的特定位置。在较低优先级的命令流在GPU上执行且GPU命令引擎从主机CPU接收到指示较高优先级的命令流即将执行的信号时, 命令引擎可推迟对所述较低优先级的命令流的抢占, 直到在较低优先级的命令流中遇到下一抢占点为止。

[0027] 在一些情况下, 主机CPU可将抢占点放置在较低优先级的命令流中的其中归因于抢占而引起的开销 (例如, 额外处理时间、额外存储器带宽使用及/或额外存储器使用) 与可归因于较低优先级的命令流中的其它位置处的抢占而发生的开销相比相对较小的位置处。通过将抢占推迟到较低优先级的命令流内的其中抢占开销相对廉价的那些位置, 本发明的技术可能提高图形处理系统的整体性能, 其中在GPU上执行的高优先级的命令流 (例如, UI命令流) 可需要抢占其它较低优先级的命令流。

[0028] 图1是说明可用于实施本发明的延迟抢占GPU命令流调度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。

[0029] 如图1的实例中所说明, 计算装置2包含用户输入接口4、CPU 6、存储器控制器8、系

系统存储器10、图形处理单元 (GPU) 12、GPU高速缓冲存储器14、显示接口16、显示器18及总线20。用户输入接口4、CPU 6、存储器控制器8、GPU 12及显示接口16可使用总线20彼此通信。总线20可为多种总线结构中的任一者,例如第三代总线(例如,HyperTransport总线或InfiniBand总线)、第二代总线(例如,高级图形端口总线、外围组件互连 (PCI) 高速总线,或高级eXtensible接口 (AXI) 总线) 或另一类型的总线或装置互连件。应注意,图1中展示的不同组件之间的总线及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及/或其它图形处理系统的其它配置可用于实施本发明的技术。

[0030] 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。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、触摸垫或经由用户输入接口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。

[0031] 在CPU 6上执行的软件应用可包含指令CPU 6以致使向显示器18再现图形数据的一或多个图形再现指令。在一些实例中,所述软件指令可符合图形应用编程接口(API),例如开放图形库 (OpenGL[®]) API、开放图形库嵌入系统 (OpenGL ES) API、Direct3D API、X3D API、RenderMan API、WebGL API或任何其它公共或专有标准图形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 12以致使GPU 12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、三角形带等图形基元的列表。

[0032] 存储器控制器8促进数据进出系统存储器10的传递。举例来说,存储器控制器8可接收存储器读取及写入命令,且服务关于存储器系统10的此类命令以便为计算装置2中的组件提供存储器服务。存储器控制器8通信地耦合到系统存储器10。虽然存储器控制器8在图1的实例计算装置2中被说明为与CPU 6及系统存储器10两者分开的处理模块,但在其它实例中,存储器控制器8的功能性中的一些或全部可实施在CPU 6及系统存储器10中的一或两者上。

[0033] 系统存储器10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CPU 6上执行的程序使用的数据。举例来说,系统存储器10可存储用户应用及与所述应用相关联的图形数据。系统存储器10可另外存储由计算装置2的其它组件使用及/或产生的信息。举例来说,系统存储器10可充当用于GPU 12的装置存储器且可存储将在GPU 12上操作的数据以及由GPU 12执行的操作而产生的数据。举例来说,系统存储器10可存储纹理缓冲器、深度缓冲器、模板缓冲器、顶点缓冲器、帧缓冲器或其类似者的任何组合。另外,系统存储器10可存储命令流以供GPU 12处理。系统存储器10可包含一或多个易失性或非易失性存储器或存储装置,例如随机存取存储器 (RAM)、静态RAM (SRAM)、动态RAM (DRAM)、只读存储器 (ROM)、可擦除可编程ROM (EPROM)、电可擦除可编程ROM (EEPROM)、快闪存储器、磁性数据媒体或光学存储媒体。

[0034] GPU 12可经配置以执行图形操作以向显示器18再现一或多个图形基元。因此,当在CPU 6上执行的软件应用中的一者需要图形处理时,CPU 6可向GPU 12提供图形命令及图形数据以用于向显示器18再现。图形命令可包含(例如)绘制命令、GPU状态编程命令、存储

器传递命令、通用计算命令、内核执行命令等。在一些实例中,CPU 6可通过将命令及图形数据写入到存储器10而将所述命令及图形数据提供给GPU 12,存储器10可由GPU 12存取。在一些实例中,GPU 12可进一步经配置以针对在CPU 6上执行的应用执行通用计算。

[0035] 在一些情况下,GPU 12可内置有比CPU 6提供对向量操作更有效的处理的高度并行的结构。举例来说,GPU 12可包含经配置而以并行方式对多个顶点或像素操作的多个处理元件。在一些情况下,GPU 12的高度并行性质允许GPU 12比使用CPU 6直接将场景绘制到显示器18更快速地将图形图像(例如,GUI及二维(2D)及/或三维(3D)图形场景)绘制到显示器18上。另外,GPU 12的高度并行性质可允许GPU 12比CPU 6更快速地处理用于通用计算应用的特定类型的向量及矩阵操作。

[0036] 在一些情况下,可将GPU 12集成到计算装置2的母板中。在其它情况下,GPU 12可存在于图形卡上,所述图形卡安装在计算装置2的母板中的端口中或可以其它方式并入经配置以与计算装置2互操作的外围装置内。在其它情况下,GPU 12可位于与CPU 6相同的微芯片上,从而形成系统芯片(SoC)。GPU 12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。

[0037] GPU 12可直接耦合到GPU高速缓冲存储器14。因此,GPU 12可在不必使用总线20的情况下从GPU高速缓冲存储器14读取数据且将数据写入到GPU高速缓冲存储器14。换句话说,GPU 12可使用本地存储装置而不是片外存储器在本地处理数据。此允许GPU 12通过消除GPU 12经由总线20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 12可不包含单独的高速缓冲存储器,而是经由总线20利用系统存储器10。GPU高速缓冲存储器14可包含一或多个易失性或非易失性存储器或储存装置,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。

[0038] CPU 6及/或GPU 12可将再现的图像数据存储在系统存储器10内所分配的帧缓冲器中。显示接口16可从帧缓冲器检索数据且配置显示器18以显示由再现的图像数据表示的图像。在一些实例中,显示接口16可包含经配置以将从帧缓冲器检索的数字值转换为可由显示器18消耗的模拟信号的数/模转换器(DAC)。在其它实例中,显示接口16可将数字值直接传递到显示器18以进行处理。显示器18可包含监视器、电视、投影装置、液晶显示器(LCD)、等离子显示面板、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种类型的显示单元。显示器18可集成在计算装置2内。举例来说,显示器18可为移动电话手持机或平板计算机的屏幕。或者,显示器18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18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。

[0039] 根据本发明,CPU 6可产生供GPU 12执行的命令流,所述命令流包含指示命令流中的其中允许GPU 12抢占命令流的执行的位置的一或多个权标。举例来说,CPU 6可选择GPU命令流中的一或多个位置以作为在其处允许处理GPU命令流的GPU 12响应于从CPU 6接收到抢占GPU命令流的执行的通知而抢占GPU命令流的执行的一或多个位置。CPU 6可基于所

述选定的一或多个位置而将一或多个权标放置在GPU命令流中。所述一或多个权标可向GPU 12指示允许在所述GPU命令流中的所述选定的一或多个位置处发生抢占。举例来说,CPU 6可基于所述选定的位置中的每一者而将一或多个权标放置在GPU命令流中。在此些实例中,对于所述选定的位置中的每一者,与相应的位置相关联的一或多个权标可向GPU 12指示允许在所述相应的位置处发生抢占。在一些实例中,CPU 6可将所得的命令流存储在可由GPU 12存取的存储器10中。GPU命令流可为与在CPU 6上执行的特定应用背景相关联的GPU命令的有序序列。

[0040] 在一些实例中,允许在其处发生抢占的位置可为GPU命令流内的命令边界。命令边界可指GPU命令流中的在完成第一GPU命令的执行之后且在第二GPU命令的执行开始之前出现的点。所述第一和第二GPU命令可被称为相对于特定命令边界的边界命令。第二GPU命令可紧接在形成GPU命令流的GPU命令序列中的第一GPU命令之后出现。在此些实例中,CPU 6可选择GPU命令流中的特定命令边界以用于允许出现抢占,且将向GPU指示所述选定的命令边界的一或多个权标放置在GPU命令流中。由CPU 6选择的用于允许出现抢占的命令边界可在本文中被称作抢占点。

[0041] 在其它实例中,允许在其处发生抢占的位置可为GPU命令流中的一或多个GPU命令的序列。一或多个GPU命令的序列可形成GPU命令的有序序列的部分,所述有序序列形成GPU命令流。在此些实例中,CPU 6可从GPU命令流选择允许发生抢占的特定GPU命令序列,且将指示所述选定的GPU命令序列的一或多个权标放置在GPU命令流中。由CPU 6选择的用于允许发生抢占的命令序列可在本文中被称作抢占范围。在额外实例中,允许在其处发生抢占的位置可包含抢占点与抢占范围的组合。

[0042] 在一些情况下,CPU 6可至少部分通过从GPU命令流选择不允许发生抢占的特定GPU命令序列而从GPU命令流选择允许发生抢占的特定GPU命令序列。由CPU 6选择不允许发生抢占的命令序列可在本文中被称作GPU命令的非可中断范围。在此些实例中,CPU 6可将指示GPU命令的选定的非可中断序列的一或多个权标放置在GPU命令流中。

[0043] CPU 6可产生包含指示允许在其中发生抢占的位置的一或多个权标的所得的GPU命令流。在一些实例中,所得的GPU命令流可包含命令时隙的有序序列,所述命令时隙中的每一者包含GPU命令或抢占权标中的至少一者。在其中CPU 6选择命令边界作为抢占点的情况下,在一些实例中,CPU 6可将权标放置在位于与命令边界的边界命令相关联的两个不同命令时隙之间的命令时隙中。在其中CPU 6选择命令序列作为抢占范围的情况下,在一些实例中,CPU 6可将权标放置在紧接在保持命令序列的第一GPU命令的命令时隙之前出现的命令时隙中,且将另一权标放置在紧接在保持命令序列的最后一个GPU命令的命令时隙之后出现的命令时隙中。在其中CPU 6选择命令序列作为非可中断范围的情况下,在一些实例中,CPU 6可将权标放置在紧接在保持命令的非可中断范围的第一GPU命令的命令时隙之前出现的命令时隙中,且将另一权标放置在紧接在保持命令的非可中断范围的最后一个GPU命令的命令时隙之后出现的命令时隙中。

[0044] CPU 6可基于各种选择准则将GPU命令流中的特定位置选择为抢占点。在一些实例中,CPU 6可基于在GPU命令流中的特定位置处执行背景切换的所估计成本而选择GPU命令流中的允许发生抢占的一或多个位置。在此些实例中,执行GPU背景切换的所估计成本在一些实例中可为以下各者中的一或多者的函数:(1) 执行GPU背景切换所需的处理时间的所估

计的或实际的量(即,处理时间开销);(2)执行GPU背景切换所需的存储器带宽使用的所估计的或实际的量(即,存储器带宽开销);(3)执行GPU背景切换所需的存储器的所估计的或实际的量(即,存储器使用开销);及(4)与GPU背景切换的执行相关联的任何其它开销参数。

[0045] 在其它实例中,CPU 6可基于指示GPU命令流中的命令的分组的信息而选择GPU命令流中的允许发生抢占的一或多个位置。在这些实例中,命令的分组中的每一者可由GPU 12执行的任务或任务集合相关联。在一些情况下,命令的分组中的一或多者可由多个图形帧中的相应一者的再现相关联。在其它情况下,命令的分组中的一或多者可由多个GPU内核中的相应一者的执行相关联。在额外情况下,命令的分组中的一或多者可由多个GPU任务中的相应一者的执行相关联。在一些实例中,GPU任务可为单一基元(例如,三角形)或基元群组的再现。在其中命令的分组与单一基元的再现相关联的情况下,CPU 6可(例如)将权标放置在GPU命令流中的每一绘制调用命令(即,指令GPU再现基元或三角形的命令)之前及/或之后的GPU命令流中。在其中命令的分组与多个基元的再现相关联的情况下,CPU 6可(例如)将权标放置在GPU命令流中的与多个基元的再现相关联的多个绘制调用命令之前及/或之后的GPU命令流中。

[0046] 在额外实例中,CPU 6可基于从与GPU命令流相关联的软件应用接收的指示其中应发生抢占的特定命令及其中不应发生抢占的特定命令中的至少一者的信息而选择所述GPU命令流中的一或多个位置。在又一实例中,CPU 6可将抢占权标放置在GPU命令流中的每一命令之间或GPU命令流的子集中的每一命令之间。进一步的实例可使用这些准则或其它准则的组合以作为上述准则的补充或替代。

[0047] 一般来说,放置在GPU命令流中的权标可为放置在GPU命令流中的指示是否允许在GPU命令流中的一或多个位置处发生抢占的任何类型的信息。在一些实例中,权标可为放置在GPU命令流的命令时隙中且向GPU 12指示是否允许在GPU命令流中的一或多个位置处发生抢占的命令。举例来说,一或多个特定操作代码(即,操作码)可用于指示命令是抢占权标。在这些实例中,CPU 6可至少部分通过将具有指示命令是抢占权标的操作码的命令放置到GPU命令流中而将权标放置在GPU命令流中。在一些情况下,不同的命令或操作码可用于指示不同类型的抢占权标(例如,指示抢占点的权标及/或指示抢占范围或非可中断范围的起点及终点的权标)。

[0048] 在其它实例中,可将权标整合到现有的GPU命令中。举例来说,GPU命令格式可包含指示特定GPU命令是否包含抢占权标的信息字段。在这些实例中,可将抢占权标放置在与不一定与抢占相关的其它GPU命令(例如,GPU图形命令及/或GPU计算命令)相同的命令时隙中。在这些实例中,权标可为放置到信息字段中的指示特定GPU命令包含抢占权标的值,且CPU 6可至少部分通过将特定GPU命令的信息字段设定为指示特定GPU命令含有抢占权标的值而将权标放置在GPU命令流中。在一些情况下,可将不同的值放置到信息字段中且/或一或多个额外信息字段可用于指示不同类型的抢占权标(例如,指示抢占点的权标及/或指示抢占范围或非可中断范围的起点及终点的权标)。信息字段可包含一位旗标、多位代码等。

[0049] 在额外实例中,权标可呈不存在特定代码、不存在特定命令字段、不存在一或多个位、不存在命令或不存在通常包含在GPU命令流中的另一特征的形式。举例来说,位可包含在不包含权标的所有命令中,但可在包含抢占权标的命令中省略此位。作为另一实例,不包含命令的命令时隙(例如,包含否操作(no-op))的命令时隙可充当抢占权标。

[0050] 在其它实例中,权标可包含上述类型的权标的任何组合。举例来说,CPU 6可使用专用命令作为权标以及其它类型的命令内的信息字段来指定权标。

[0051] GPU 12可经配置以基于由CPU 6放置在GPU命令流中的一或多个权标而抢占GPU命令流的执行。举例来说,GPU 12可在第一GPU命令流中分派一或多个命令以用于在GPU 12中的一或多个处理单元上执行,且从CPU 6接收指示第二GPU命令流即将在GPU 12上执行的抢占通知。第二GPU命令流可具有比第一GPU命令流的调度优先级大的调度优先级。响应于接收到抢占通知,GPU 12可基于放置在第一GPU命令流中的一或多个权标而确定何时抢占第一GPU命令流的执行。

[0052] 在一些实例中,响应于接收到抢占通知,GPU 12可在第一GPU命令流中分派一或多个命令以用于在GPU 12中的一或多个处理单元上执行,直到在第一GPU命令流中遇到权标为止。响应于在第一GPU命令流中遇到权标,GPU 12可抢占第一GPU命令流的执行且切换到执行第二GPU命令流。举例来说,GPU 12可保存与第一GPU命令流的执行相关联的GPU 12的状态,且在第二GPU命令流中分派一或多个命令以用于在GPU 12中的一或多个处理单元上执行。

[0053] 在其它实例中,在GPU 12处理第一GPU命令流时,GPU 12可维持抢占允许状态变量,所述抢占允许状态变量在给定时间点处的值可指示在所述给定时间点处是否允许GPU 12抢占第一GPU命令流的执行。GPU 12可响应于第一GPU命令流中包含的权标而更新或改变抢占允许状态变量。在其中第一GPU命令流中的权标指示GPU命令的抢占范围的起点及终点的实例中,GPU 12可响应于处理起点抢占范围权标而将抢占允许状态变量更新为指示允许发生抢占的值,且响应于处理终点抢占范围权标而将抢占允许状态变量更新为指示不允许发生抢占的值。在其中第一GPU命令流中的权标指示GPU命令的非可中断范围的起点及终点的实例中,GPU 12可响应于处理起点非可中断范围权标而将抢占允许状态变量更新为指示允许发生抢占的值,且响应于处理终点非可中断范围权标而将抢占允许状态变量更新为指示不允许发生抢占的值。

[0054] 在一些情况下,CPU 6可选择命令流中的其中归因于抢占而引起的开销(例如,额外处理时间、额外存储器带宽使用及/或额外存储器使用)与可归因于命令流中的其它位置处的抢占而发生的开销相比相对较小的抢占位置。通过将抢占推迟到较低优先级的命令流内的其中抢占开销相对廉价的那些位置,本发明的技术可能够提高图形处理系统的整体性能,其中高优先级的命令流可能需要抢占在GPU上执行的其它命令流。

[0055] 图2是进一步详细说明图1中的计算装置2的CPU 6、GPU 12和存储器10的框图。如图2中所展示,CPU 6通信地耦合到GPU 12及存储器10,且GPU 12通信地耦合到CPU 6及存储器10。在一些实例中,GPU 12可与CPU 6一起集成到母板上。在额外实例中,GPU 12可实施在安装在包含CPU 6的母板的端口中的图形卡上。在其它实例中,GPU 12可并入经配置以与CPU 6交互操作的外围装置内。在额外实例中,GPU 12可位于与CPU 6相同的微芯片上,从而形成系统芯片(SoC)。CPU 6经配置以执行软件应用24A、24B、图形API 26、GPU驱动程序28及操作系统30。GPU 12包含命令引擎32及一或多个处理单元34。存储器10存储命令流36A、36B。

[0056] 软件应用24A、24B可各自包含致使显示图形图像的一或多个指令或致使在GPU 12上执行非图形任务(例如,通用计算任务)的一或多个指令中的至少一者。软件应用24A、24B

可将指令分布到图形API 26。图形API 26可为将从软件应用24A、24B接收的指令转译为可由GPU驱动程序28消耗的格式的运行时服务。

[0057] GPU驱动程序28经由图形API 26从软件应用24A、24B接收指令,且控制GPU 12的操作以服务所述指令。举例来说,GPU驱动程序28可制定一或多个命令流36A、36B,将命令流36A、36B放置到存储器10中,且指令GPU 12执行命令流36A、36B。GPU驱动程序28可将命令流36A、36B放置到存储器10中且经由操作系统30(例如,经由一或多个系统调用)与GPU 12通信。

[0058] GPU驱动程序28可将抢占通知发送到GPU命令引擎32以指示另一命令流(例如,高优先级的命令流)即将执行。在一些实例中,抢占通知还可向GPU 12指示在抢占当前执行的命令流之后即刻执行哪一命令流。GPU驱动程序28可(例如)通过将一或多个值写入(例如,经由操作系统30)到由GPU命令引擎32轮询的一或多个GPU寄存器而将抢占通知发送到GPU命令引擎12。所述一或多个GPU寄存器可包含位于GPU 12中的一或多个硬件GPU寄存器、位于存储器(例如,存储器10)中的一或多个存储器映射的GPU寄存器,或其任何组合。在将值写入到存储器映射的GPU寄存器时,CPU 6可将所述值写入到存储器10中的由GPU命令引擎32轮询的一或多个特定存储器地址。在一些实例中,GPU驱动程序28可将第一值写入到一或多个GPU寄存器以激活抢占通知,且将识别在抢占当前执行的命令流之后即刻执行的命令流的位置的第二值写入到一或多个GPU寄存器。

[0059] 命令引擎32经配置以检索存储于命令流36A、36B中的命令,且分派所述命令以用于在处理单元34中的一或多者上执行。命令引擎32可分派来自命令流的命令以用于在处理单元34的全部或子集上执行。在一些实例中,命令引擎32可经配置以轮询一或多个GPU寄存器,所述一或多个GPU寄存器指示CPU 6是否已将抢占通知发送到GPU 12。在其中GPU寄存器是存储器映射的寄存器的情况下,命令引擎32可经配置以轮询存储器10中的指示CPU 6是否已将抢占通知发送到GPU 12的一或多个特定存储器地址。响应于一或多个GPU寄存器指示CPU 6已将抢占通知发送到GPU 12,命令引擎32可抢占当前执行的GPU命令流的执行且开始执行另一命令流。在一些情况下,命令引擎32可经配置以通过读取存储于一或多个GPU寄存器12中的识别在抢占之后即刻执行的命令流的信息而确定在抢占当前执行的命令流之后即刻执行的命令流。举例来说,CPU 6可将识别较高优先级的命令流的存储器位置的指针存储在一或多个GPU寄存器中。

[0060] 为了抢占命令流,命令引擎32可保存与命令流的执行相关联的GPU 12的状态。与命令流的执行相关联的GPU 12的状态可包含程序计数器、线程激活状态、图形绘制状态变量(例如,色彩、基元类型、变换类型等)、存储器状态及/或用于执行命令流的任何其它状态。在一些实例中,状态变量可为与个别处理单元34或处理单元34的群组相关联的局部状态变量、与GPU 12的全局执行相关联的全局状态变量,或其任何组合。保存GPU 12的状态可涉及(例如)将GPU 12中的一或多个寄存器的当前值存储在另一芯片上或芯片外位置中。保存GPU 12的状态可涉及(例如)将本地GPU存储器的当前状态存储在另一芯片上或芯片外位置中。在一些实例中,命令引擎32可将GPU状态保存到存储器10。

[0061] 在已完成抢占命令流的执行之后,命令引擎32可恢复在抢占通知之前执行的被抢占的命令流的保存的状态。恢复保存的状态可涉及(例如)将状态变量重新加载到GPU 12的一或多个寄存器中及/或将保存的存储器状态重新加载到本地GPU存储器中。在其中将GPU

状态保存到存储器10的实例中,命令引擎32可将存储于存储器10中的保存的状态重新加载到GPU 12上以用于进一步执行被抢占的命令流。

[0062] 处理单元34可包含一或多个处理单元,其中的每一者可为可编程处理单元或固定功能处理单元。可编程处理单元可包含(例如)经配置以执行从CPU 6下载到GPU 12上的一或多个着色器程序的可编程着色器单元。在一些实例中,着色器程序可为以高级着色语言(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)编写的程序的经编译版本。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。处理单元34中的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、几何形状着色器单元、壳着色器单元、域着色器单元、计算着色器单元和/或统一着色器单元。

[0063] 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。在一些实例中,处理单元34中的固定功能处理单元可包含(例如)执行光栅操作的处理单元,例如深度测试、剪刀测试、 α 掺杂等。

[0064] 在图2的实例图形处理系统中,存储器10包含两个命令流36A、36B,每一命令流36A、36B与不同的图形应用背景(即,软件应用24A、24B中的不同一者)相关联。举例来说,命令流36A与用于软件应用24A的图形应用背景相关联,且命令流36B与用于软件应用24B的图形应用背景相关联。为了易于说明,将两个GPU命令流36A、36B描绘为存储于图2的实例图形处理系统的存储器10中。然而,在其它实例中,可使用存储于图形处理系统的相同或不同组件中的相同或不同数目的GPU命令流。

[0065] 如图2中所展示,每一命令流36A、36B具有多个命令时隙,其由跨越所述命令流的顶部的编号识别。每一命令时隙可保持命令包(C)或权标包(T)。命令包可包含用于在GPU上执行的一或多个GPU命令。权标包可不一定包含命令,但可包含用于指示命令流中的其中允许发生抢占的特定位置(即,抢占点)、其中允许发生抢占的特定命令范围(即,抢占范围),或其中不允许发生抢占的特定命令范围(即,非可中断范围)的权标。

[0066] CPU 6的GPU驱动程序28可经配置以将命令流36A、36B写入到存储器10,且GPU 12的命令引擎32可经配置以从存储器10读取命令流36A、36B的一或多个命令。在一些实例中,可将命令流36A、36B中的一或两者存储为存储器10中的环形缓冲器。环形缓冲器可为具有循环寻址方案的缓冲器,其中CPU 6及GPU 12维持与将数据写入到环形缓冲器及从环形缓冲器读取数据相关联的同步状态变量。举例来说,如果命令流36A是环形缓冲器,那么CPU 6及GPU 12中的每一者可存储指示待写入到环形缓冲器中的下一地址的写入指针及指示待从环形缓冲器中读取的下一地址的读取指针。在CPU 6将新命令写入到环形缓冲器时,CPU 6可更新CPU 6中的写入指针且指令GPU 12更新GPU 12中的写入指针。类似地,在GPU 12从环形缓冲器读取新命令时,GPU 12可更新GPU 12中的读取指针且指令CPU 6更新CPU 6中的读取指针。其它同步机制是可能的。在读取及/或写入指针达到针对环形缓冲器所分配的地址范围内的最高地址时,读取及/或写入指针可回绕到最低地址以实施循环寻址方案。

[0067] 现将相对于图2描述根据本发明而设计的实例GPU驱动程序28及实例GPU命令引擎

32的实例操作。在此实例中,软件应用24B具有比软件应用24A的调度优先级高的调度优先级。具体来说,在此实例中,软件应用24B是用户接口(UI)软件应用,其包含致使显示图形图像且需要对GPU 12的高优先级存取以确保UI的及时更新的一或多个指令。同时,在此实例中,软件应用24A是较低优先级的应用,其包含致使显示图形图像的一或多个指令及/或致使在GPU 12上执行非图形任务(例如,GPGPU计算任务)的一或多个指令。

[0068] GPU驱动程序28从软件应用24A接收指定待由GPU 12执行的图形操作及/或通用计算操作的一或多个指令。GPU驱动程序28产生中间命令流以实行由所接收的指令指定的操作、选择中间命令流中的允许发生抢占的一或多个位置,并将权标放置在中间命令流中的其中允许发生抢占的位置中的每一者处以便产生输出命令流36A。GPU驱动程序28将输出命令流36A放置到可由GPU命令引擎32存取的存储器10中。GPU驱动程序28通知GPU命令引擎32对应于软件应用24A的命令流36A可用于处理。举例来说,GPU驱动程序28可将指示命令流36A即将执行的一或多个值写入到GPU寄存器(例如,由GPU 12轮询的GPU硬件寄存器及/或由GPU 12轮询的GPU存储器映射的寄存器)。

[0069] 在通知命令流36A即将执行之后,GPU 12的命令引擎32可即刻确定资源在GPU 12上当前是否可用以开始执行命令流36A。如果资源可用,那么命令引擎32开始基于命令时隙以数值次序分派命令流36A中的命令。更具体来说,命令引擎32分派命令时隙1及2中包含的命令包以用于在处理单元34中的一或多者上执行。响应于在命令时隙3处遇到权标包,命令引擎32确定GPU 12是否已从CPU 6接收到任何抢占通知。在此情况下,命令引擎32尚未接收到任何抢占通知,因此命令引擎32前进到在命令流36A的命令时隙4处分派命令。

[0070] 同时,GPU驱动程序28从UI软件应用24B接收到指定待由GPU 12执行的高优先级图形操作的一或多个指令。GPU驱动程序28产生高优先级的命令流36B以实施由所接收的指令指定的操作,且将所得的命令流36B放置到存储器10中。GPU驱动程序28根据本发明的延迟抢占技术将指示高优先级的命令流36即将处理及命令流36应抢占正在GPU 12上执行的任何其它较低优先级的命令流的抢占通知提供给GPU命令引擎32。在一些实例中,GPU驱动程序28可通过写入到由GPU 12轮询的GPU硬件寄存器及/或通过写入到由GPU 12轮询的存储器映射的寄存器(例如,存储器10中的特定位置)而将抢占通知提供给GPU 12。

[0071] 在于命令流36A中的命令时隙5处分派命令之前,GPU命令引擎32接收到指示较高优先级的命令流36B即将处理且较高优先级的命令流36B应抢占任何其它较低优先级的命令流(即,命令流36A)的处理的抢占通知。然而,因为命令引擎32实施延迟抢占方案,所以命令引擎32继续分派来自命令流36A的命令,直到在命令流36A中遇到权标包(即,抢占点)为止。更具体来说,命令引擎32分派命令时隙5及6中的命令以用于在处理元件34中的一或多者上执行。

[0072] 响应于在命令时隙7处遇到权标包,命令引擎32确定GPU 12是否已从CPU 6接收到任何抢占通知。在此情况下,GPU 12已接收到抢占通知,因此命令引擎32前进到执行从处理与命令流36A相关联的命令到处理与高优先级的命令流36B相关联的命令的背景切换。在一些情况下,为了执行背景切换,命令引擎32可将用于当前处理的背景(即,软件应用24A)的GPU的状态(例如,一或多个寄存器的状态及/或GPU 12中的存储器的状态)保存在存储器10中。另外,如果先前保存了UI软件应用24B的GPU状态,那么背景切换还可恢复UI软件应用24B的先前保存的GPU状态。

[0073] 在执行背景切换之后,命令引擎32前进到分派命令流36B的命令时隙1到6中的命令以用于在GPU 12的处理单元34中的一或多者上执行。在结束命令流36B的处理之后,命令引擎32执行另一背景切换以继续处理命令流36A中的剩余的命令。执行背景切换可涉及将GPU 12中的寄存器及/或存储器的状态恢复到在先前背景切换之前保存的状态。在执行背景切换之后,命令引擎32前进到分派命令流36A的命令时隙8中的命令以用于在处理单元34中的一或多者上执行。

[0074] 在图2中展示的实例命令流36A中,GPU驱动程序28确定命令时隙4到6中的命令形成与特定GPU任务或操作(例如,特定帧的再现或特定GPU计算任务的执行)的执行相关联的命令的逻辑分组,且用于在执行命令的所述逻辑分组时执行背景切换的所估计成本(例如,在处理时间、存储器带宽使用及/或存储器使用方面)与仅一直等到命令时隙7执行背景切换相比将相对昂贵。因此,图形驱动程序28在命令时隙7处放置指示允许GPU 12在命令时隙7处抢占命令流的执行的权标。因此,即使命令引擎32在执行命令时隙5处的命令之前接收到抢占通知,本发明的延迟抢占技术有效地允许将抢占延迟到命令流36A中的其中背景切换在处理时间、存储器带宽使用及/或存储器使用方面将较不昂贵的点。以此方式,本发明的技术可用于提高图形处理系统的整体性能。

[0075] 虽然图2说明其中较高优先级的命令流36B(即,UI命令流)不包含权标包的实例,但在其它实例中,命令流36B可包含权标包,其将致使在CPU 6在命令流36B的执行期间将抢占通知发送到GPU 12的情况下更高优先级的命令流将抢占命令流36B。一般来说,本发明的技术在一些实例中可允许发生任何数目的嵌套式抢占。

[0076] 图3到5是说明根据本发明的在GPU命令流中放置权标的各种技术的概念图。图3说明含有GPU命令序列(即,C1、C2...C8)的实例中间GPU命令流50。所述GPU命令可为能够由GPU执行的任何类型的图形处理命令。举例来说,所述命令可为绘制命令、状态编程命令、通用GPU计算命令等。在命令中的每一者之间界定命令边界52、54、56、58、60、62、64、66及68。在一些实例中,GPU驱动程序28可直接从软件应用24A、24B中的一者接收中间GPU命令流50。在额外实例中,GPU驱动程序28可基于从软件应用24A、24B中的一者接收的指令而产生中间GPU命令流50。

[0077] 图4说明由GPU驱动程序28产生的实例输出GPU命令流70。如图4中所展示,输出GPU命令流70包含命令时隙的有序序列(即,S1、S2...S10)。每一命令时隙包含来自中间GPU命令流50的命令或权标(即,T)中的至少一者。在一些实例中,每一命令可被称为命令包,且每一权标可被称为权标包。为了产生输出GPU命令流70,GPU驱动程序28选择命令边界56及64作为其中允许发生抢占的位置。也就是说,命令边界56及64被选择为响应于GPU 12从CPU 6接收到指令GPU 12抢占输出GPU命令流70的执行的抢占通知而允许GPU 12在其处抢占输出GPU命令流70的执行的执行的位置。命令C2及C3可被称为相对于命令边界56的边界命令,且命令C6及C7可被称为相对于命令边界64的边界命令。

[0078] 为了指示命令边界56是允许在其处发生抢占的选定命令边界,GPU驱动程序28在保持用于命令边界56的边界命令(C2、C3)的命令时隙(S2、S4)之间的命令时隙(S3)中放置权标。类似地,为了指示命令边界64是允许在其处发生抢占的选定的命令边界,GPU驱动程序28在保持用于命令边界64的边界命令(C6、C7)的命令时隙(S7、S9)之间的命令时隙(S8)中放置权标。

[0079] 在图4的实例输出GPU命令流70中,所述权标指示输出GPU命令流70中的其中允许GPU 12响应于从CPU 6接收到抢占通知而抢占输出GPU命令流70的执行的特定抢占点。举例来说,放置在特定命令时隙中的权标指示允许在执行紧接在其中放置权标的命令时隙之前的命令时隙中的命令之后且在执行紧接在其中放置权标的命令时隙之后的命令时隙中的命令之前发生抢占。作为特定实例,放置在S3命令时隙中的权标指示允许在执行命令C2之后且在执行命令C3之前发生抢占。

[0080] 在GPU 12处理输出GPU命令流70时,如果GPU 12遇到权标,随后GPU 12可确定是否已从CPU 6接收到抢占通知。如果已从CPU 6接收到抢占通知,那么GPU 12可抢占输出GPU命令流70的执行以执行由CPU 6指定的另一GPU命令流。否则的话,如果在GPU 12遇到权标时尚未接收到抢占通知,那么GPU 12可继续处理输出GPU命令流70中的下一命令时隙而不抢占输出GPU命令流70的执行。

[0081] GPU驱动程序28可基于各种准则选择允许发生抢占的命令边界56、64。在一些实例中,GPU驱动程序28可基于在GPU命令流中的特定位置处执行GPU背景切换的所估计成本而选择输出GPU命令流70中的命令边界56、64。举例来说,GPU驱动程序28可确定在命令边界56、64中的每一者处执行GPU背景切换的所估计成本与在命令边界52、54、58、60、62、66、68中的一或多者处执行GPU背景切换的所估计成本相比较低。在一些实例中,执行GPU背景切换的所估计成本可为以下各者中的一或多者的函数:(1)执行GPU背景切换所需的处理时间的所估计的或实际的量(即,处理时间开销);(2)执行GPU背景切换所需的存储器带宽使用的所估计的或实际的量(即,存储器带宽开销);(3)执行GPU背景切换所需的存储器的所估计的或实际的量(即,存储器使用开销);及(4)与GPU背景切换的执行相关联的任何其它开销参数。

[0082] 在其它实例中,GPU驱动程序28可基于指示中间GPU命令流50中包含的与特定GPU任务或操作(例如,特定帧的再现或特定GPU任务的执行)的执行相关联的命令的逻辑分组的信息而选择输出GPU命令流70中的命令边界56、64。在一些情况下,GPU驱动程序28可从与中间GPU命令流50相关联的软件应用(例如,软件应用24A、24B)接收指示命令的逻辑分组的信息。举例来说,GPU驱动程序28可从软件应用接收用于执行GPU任务的特定集合的指令,且所接收的指令可识别GPU任务中的每一者的起点及/或终点。在额外情况中,在其中GPU驱动程序28从软件应用接收中间GPU命令流50的情况下,GPU驱动程序28可基于中间GPU命令流50确定指示命令的逻辑分组的信息。

[0083] 在一些实例中,命令的逻辑分组可包含与单一图形帧或图形帧的单一群组的再现相关联的命令。举例来说,命令的不同分组可与不同再现的帧相关联,且可将权标放置在与不同再现的帧中的每一者相关联的命令的不同分组之间的输出GPU命令流70中。作为特定实例,命令C1及C2可与第一再现的帧相关联,命令C3、C4、C5和C6可与第二再现的帧相关联,且命令C7及C8可与第三再现的帧相关联。在此特定实例中,如图4中所展示,将权标放置在命令时隙S3及S8中以允许在用于不同再现的帧的命令的处理之间发生抢占。换句话说,GPU驱动程序28可确定由命令C1及C2界定的命令序列、由命令C3、C4、C5和C6界定的命令序列及由命令C7及C8界定的命令序列是非可中断的命令序列。在额外实例中,命令的逻辑分组可包含与单一GPU内核或单一GPU任务的执行相关联的命令或包含与GPU内核或任务的群组的执行相关联的命令。举例来说,对于GPU计算应用,命令的不同分组可与不同的GPU内核或任

务的执行相关联,且可将权标放置在与GPU内核或任务中的每一者相关联的命令的不同分组之间的输出GPU命令流70中。

[0084] 在额外实例中,GPU驱动程序28可基于从与输出GPU命令流70相关联的软件应用(例如,软件应用24A、24B)接收的信息而选择输出GPU命令流70中的命令边界56、64。举例来说,软件应用可将指示其中应发生抢占的一或多个特定命令及/或命令边界及/或其中不应发生抢占的一或多个特定命令及/或命令边界的信息提供给GPU驱动程序28。GPU驱动程序28可在输出GPU命令流70中放置权标以指示这些位置。

[0085] 图5说明由GPU驱动程序28产生的实例输出GPU命令流80。如图5中所展示,输出GPU命令流80包含命令时隙的有序序列(即,S1、S2...S10)。每一命令时隙包含来自中间GPU命令流50的命令或权标(即,T)中的至少一者。为了产生输出GPU命令流80,GPU驱动程序28将中间GPU命令流50(图3)中的命令序列82(即,命令C2、C3、C4、C5、C6)选择为其中允许发生抢占的位置。也就是说,命令序列82中的命令被选择为允许GPU 12响应于从CPU 6接收到抢占输出GPU命令流80的执行的执行的通知而在其处抢占输出GPU命令流80的执行的执行的位置。命令C1及C7可被称为相对于命令序列82的边界命令。

[0086] 为了指示命令序列82是允许在其内发生抢占的选定命令范围,GPU驱动程序28将权标放置在紧接在保持命令序列82的第一GPU命令的命令时隙(S3)之前出现的命令时隙(S2)中。另外,GPU驱动程序28将另一权标放置在紧接在保持命令序列82的最后一个GPU命令的命令时隙(S6)之后出现的命令时隙(S7)中。

[0087] 在图5的实例输出GPU命令流80中,所述权标指示输出GPU命令流80中的其中允许GPU 12响应于从CPU 6接收到抢占通知而抢占输出GPU命令流80的执行的特定抢占范围。举例来说,放置在第一命令时隙(S2)中的开始抢占范围权标可指示抢占范围的起点,且放置在第二命令时隙(S7)中的终点抢占范围权标可指示抢占范围的终点。开始抢占范围权标可指示允许在处理开始抢占范围权标之后且在处理终点抢占范围权标之前的任何时间发生抢占。终点抢占范围权标可指示不允许在处理终点抢占范围权标之后发生抢占。对于图5中展示的实例抢占范围(C2、C3、C4、C5),应注意,命令时隙S2及S7中的权标指示紧接在命令C2之前但在处理命令时隙S2中的权标之后且紧接在处理命令C5之后但在处理命令时隙S7中的权标之前可在命令C2、C3、C4及C5中的任一者之间发生抢占。

[0088] 在处理输出GPU命令流80期间,在GPU 12遇到命令时隙S2中的权标时,GPU 12将抢占允许状态变量设定为指示允许对当前处理的命令的抢占的值。在GPU 12遇到命令时隙S7中的权标时,GPU 12将抢占允许状态变量重设为指示不允许对当前处理的命令的抢占的值。如果GPU 12从CPU 6接收到抢占通知,那么GPU 12可确定抢占允许状态变量是否指示当前允许抢占。如果当前允许抢占,那么GPU 12可抢占输出GPU命令流80的执行。否则的话,如果当前不允许抢占,那么GPU 12可推迟抢占,直到遇到指示其中允许抢占的命令序列的起点的另一权标为止。

[0089] 在图5中所描绘的实例输出GPU命令流80中,放置在命令时隙S2及S7中的抢占权标分别指示在其内允许发生抢占的命令范围82的起点及终点。在额外实例中,可由GPU驱动程序28将命令范围82选择为命令的非可中断范围。在此些实例中,放置在命令时隙S2及S7中的抢占权标可分别指示不允许在其处发生抢占的命令范围82的起点及终点。对命令范围82的选择可基于类似于上文相对于图4概述的准则的各种准则。

[0090] 虽然图4及5中展示的实例输出GPU命令流70、80将权标描绘为占据个别命令时隙而不在命令时隙中包含额外命令,但在其它实例中,命令时隙可包含权标及命令两者。举例来说,包含在特定命令中的旗标可指示权标是否包含在与命令相同的命令时隙中。

[0091] 图6是说明根据本发明的用于控制GPU以执行延迟抢占调度的实例技术的流程图。出于示范性目的,图6中展示的实例技术被描述为由图1及2中展示的CPU 6执行。在其它实例中,图6中说明的技术可实施在本发明的图1及2中展示的系统或组件中的任一者(例如,在CPU 6上操作的GPU驱动程序28)中或本文中未具体提到的其它类型的图形处理系统中。

[0092] CPU 6选择GPU命令流中的允许发生抢占的一或多个位置(100)。换句话说,CPU 6选择GPU命令流中的一或多个位置以作为在其处允许处理GPU命令流的GPU 12响应于从CPU 6接收到抢占通知而抢占GPU命令流的执行的一或多个位置。CPU 6基于所述选定的一或多个位置而将一或多个权标放置在GPU命令流中(102)。所述一或多个权标可向GPU 12指示允许在所述GPU命令流中的所述选定的一或多个位置处发生抢占。举例来说,CPU 6可基于所述选定的位置中的每一者而将一或多个权标放置在GPU命令流中。在此些实例中,对于所述选定的位置中的每一者,与相应的位置相关联的一或多个权标可向GPU 12指示允许在所述相应的位置处发生抢占。在一些实例中,CPU 6可针对允许在其处发生抢占的位置中的每一者将单一权标放置在GPU命令流中。在此些实例中,在一些情况下,可将所述单一权标放置在GPU命令流中的允许发生抢占的实际位置(例如,命令边界)处,以产生权标与GPU命令流中的在其处允许发生抢占的相应位置之间的一对一映射。在额外实例中,CPU 6可针对在其处允许发生抢占的位置中的每一者将多个权标放置在GPU命令流中。在其它实例中,CPU 6可在不一定是在其处允许发生抢占的位置的位置处将一或多个权标放置在GPU命令流中,但此类权标可仍然指示GPU命令流中的其中允许发生抢占的实际位置。

[0093] CPU 6致使GPU 12开始GPU命令流的执行(104)。在GPU命令流在GPU 12上执行时,CPU 6将指示另一GPU命令流即将在GPU 12上执行的抢占通知发送到GPU 12(106)。所述抢占通知可进一步指示GPU 12应基于放置在GPU命令流中的一或多个权标而抢占在GPU 12上执行的具有比第一GPU命令流的调度优先级低的调度优先级的任何命令流。第二GPU命令流可具有比第一GPU命令流的调度优先级大的调度优先级。

[0094] 在一些实例中,CPU 6可将一或多个命令边界选择为在其处允许GPU 12抢占GPU命令流的执行的位置,且将一或多个权标放置在GPU命令流中以向GPU 12指示选定的边界。举例来说,对于每一选定的命令边界,CPU 6可在与相应的命令边界的边界命令相关联的两个不同的命令时隙之间的命令时隙中放置权标。在此些实例中,权标可在一些实例中指示允许在处理紧接在命令边界之前的命令之后且在处理紧接在命令边界之后的命令之前发生抢占。

[0095] 在其它实例中,CPU 6可将一或多个命令范围选择为在其处允许GPU 12抢占GPU命令流的执行的位置,且将一或多个权标放置在GPU命令流中以向GPU 12指示选定的命令范围。举例来说,对于每一选定命令范围,CPU 6可在紧接在保持命令范围的第一GPU命令的命令时隙之前出现的命令时隙中放置权标,且将另一权标放置在紧接在保持命令范围的最后一个GPU命令的命令时隙之后出现的命令时隙中。在此些实例中,权标可在一些实例中指示允许在处理命令范围中的命令中的任一者之间、在处理紧接在命令范围之前的命令之后、及在处理紧接在命令范围之后的命令之前发生抢占。

[0096] 在额外实例中,CPU 6可基于选定的一或多个命令范围而将一或多个命令范围选择为在其处不允许GPU抢占GPU命令流的执行的位置,且将一或多个位置选择为允许GPU抢占GPU命令流的执行的位置。举例来说,CPU 6可在紧接在命令的非可中断范围的第一GPU命令的命令时隙之前出现的命令时隙中放置权标,且将另一权标放置在紧接在保持命令的非可中断范围的最后一个GPU命令的命令时隙之后出现的命令时隙中。指示命令的非可中断范围的起点及终点的权标还可分别指示其中允许发生抢占的命令范围的终点及起点。

[0097] 在一些实例中,CPU 6可基于在命令流中的多个位置处执行GPU背景切换的所估计成本而选择命令流中的其中允许发生抢占的位置。举例来说,如果用于在特定位置处执行GPU背景切换的所估计成本小于阈值,那么CPU 6可选择所述特定位置作为其中允许发生抢占的位置。在一些实施方案中,所述阈值可为预定阈值。在其它实施方案中,可基于在命令流的各种位置中执行GPU背景切换的所估计的成本的统计分析来确定所述阈值。举例来说,如果在一或多个特定位置处执行GPU背景切换的所估计成本小于在一或多个其它未被选择的位置处执行GPU背景切换的所估计成本,那么CPU 6可选择其中允许发生抢占的那些特定位置。作为另一实例,在所有可能的位置当中,GPU驱动程序可确定具有执行GPU背景切换的最低所估计成本的10%的位置,且选择那些位置作为其中允许发生抢占的位置。在一些实例中,执行GPU背景切换的所估计成本可为由CPU 6确定的实际成本。在一些实例中,所估计的成本或实际成本可为以下各者中的一或多者的函数:(1) 执行GPU背景切换所需的处理时间的所估计的或实际的量(即,处理时间开销);(2) 执行GPU背景切换所需的存储器带宽使用的所估计的或实际的量(即,存储器带宽开销);(3) 执行GPU背景切换所需的存储器的所估计的或实际的量(即,存储器使用开销);及(4) 与GPU背景切换的执行相关联的任何其它开销参数。

[0098] 在其它实例中,CPU 6可基于从与命令流相关联的软件应用接收的指令中包含的命令的逻辑分组而选择命令流中的其中允许发生抢占的位置。命令的逻辑分组内的命令中的每一者可由待由GPU 12执行的任务或任务集合相关联。在一些情况下,命令的分组中的一或多者可由多个GPU内核中的相应一者的执行相关联。举例来说,由软件应用(例如,软件应用24A、24B)提供的指令可包含界定待由GPU 12执行的内核(例如,用于通用计算算法的操作的逻辑分组,其中操作的逻辑分组的若干实例可由GPU 12并行执行)的指令。中断界定内核的指令的执行在此实例中可导致在处理时间、存储器带宽使用及/或存储器使用方面的昂贵(即,高成本)的背景切换。因此,在此实例中,CPU 6可确定不将任何权标放置在与GPU内核的执行相关联的命令群组内,且将一或多个权标放置在与GPU内核的执行相关联的命令群组之前及/或之后。

[0099] 在额外情况下,命令的分组中的一或多者可由多个图形帧中的相应一者的再现相关联。举例来说,由软件应用提供的指令可包含用于待再现的多个图形帧的绘制操作。在此实例中,CPU 6可确定不将任何权标放置在与特定图形帧的再现相关联的命令群组内,且将一或多个权标放置在与图形帧的再现相关联的命令群组之前及/或之后。

[0100] 在其它情况下,命令的分组中的一或多者可由多个GPU任务中的相应一者的执行相关联。举例来说,GPU任务可为绘制调用命令(即,指令GPU再现基元或三角形的命令),且CPU 6可将权标放置在GPU命令流中的一组绘制命令中的每一绘制调用命令之间或与GPU命令流中的多个基元的再现相关联的每一绘制调用命令群组之间。

[0101] 在额外实例中,CPU 6可基于从软件应用接收的关于其中应或不应允许发生抢占的特定命令的信息而选择命令流中的其中允许发生抢占的位置。举例来说,由软件应用提供的指令可识别其中可发生或不可发生抢占的操作的逻辑分组及/或程序中的指令。作为另一实例,由软件应用提供的指令可识别其中可发生或不可发生抢占的程序中的特定指令或命令。在此情况下,CPU 6可基于此信息而将权标放置在命令流中。

[0102] 图7到11是说明根据本发明的用于基于放置在GPU命令流中的权标而抢占命令流的执行的各种技术的流程图。出于示范性目的,图7到11中展示的实例技术被描述为由图1和2中展示的GPU 12执行。在其它实例中,图7到11中说明的技术可实施在本发明的图1到2中展示的系统或组件中的任一者(例如,GPU 12的命令引擎32)中或本文中未具体提到的其它类型的图形处理系统中。

[0103] 图7是说明根据本发明的用于基于放置在GPU命令流中的权标而抢占GPU命令流的执行的实例技术的流程图。GPU 12分派第一GPU命令流中的一或多个命令以用于在GPU 12中的一或多个处理单元34上执行(108)。GPU 12接收抢占通知(110)。所述抢占通知指示第二GPU命令流即将在GPU 12上执行。第二GPU命令流可具有比第一GPU命令流的调度优先级大的调度优先级。在一些实例中,特定GPU命令流的调度优先级可指示以下各者中的一或多个者:(1)是否允许特定GPU命令流抢占其它GPU命令流的执行;(2)允许特定GPU命令流抢占其它GPU命令流中的哪一者(如果存在的话);(3)是否允许特定GPU命令流被其它GPU命令流抢占;及(4)允许其它GPU命令流中的哪一者抢占特定GPU命令流(如果存在的话)。

[0104] 在一些实例中,每一GPU命令流的调度优先级可等于可能的调度优先级值的范围内的值。在此些实例中,具有较高的调度优先级值的GPU命令流可在一些情况下能够抢占具有较低的调度优先级值的GPU命令流的执行。类似地,在一些情况下,具有较低的调度优先级值的GPU命令流可能被具有较高的调度优先级值的GPU命令流抢占。在一些实例中,调度优先级值可由图形驱动程序(例如,图1中的GPU驱动程序28)或由用户应用(例如,图1中的应用24A、24B)指定。

[0105] 响应于接收到抢占通知,GPU 12分派第一GPU命令流中的一或多个命令以用于在处理单元34中的一或多个者上执行,直到在第一GPU命令流中遇到权标为止(112)。响应于遇到第一GPU命令流中的权标,GPU 12保存与第一GPU命令流的执行相关联的GPU 12的状态(114)。如果保存了关于第二GPU命令流的执行的先前状态,那么GPU 12可任选地负载与第二GPU命令流的执行相关联的GPU 12的状态(116)。GPU 12分派第二GPU命令流中的一或多个命令以用于在GPU 12中的处理单元34中的一或多个者上执行(118)。

[0106] 图8是说明根据本发明的用于处理GPU命令流的实例技术的流程图。图8中说明的技术可与指示其中允许发生抢占的特定抢占点的抢占权标(例如,上文关于图4所描述的抢占权标)一起使用。

[0107] GPU 12开始执行第一GPU命令流(120)。GPU 12选择用于处理的下一命令时隙(122)。在其中GPU 12在开始执行第一GPU命令流之后首次执行过程框122的情况下,GPU 12选择第一GPU命令流中的用于处理的第一命令时隙。

[0108] GPU 12确定所述命令时隙是否含有权标(124)。响应于确定所述命令时隙不含权标,GPU 12将命令时隙中的命令分派给处理单元34中的一或多个者而不确定是否已接收抢占通知(126)。在分派用于执行的命令之后,GPU 12可选择第一GPU命令流中的用于处理的下

一命令时隙(122)。所述下一命令时隙可为命令时隙序列中的紧接在先前处理的命令时隙之后出现的命令时隙。

[0109] 返回到决策框124,响应于确定命令时隙确实含有权标,GPU 12确定是否已从主机装置(例如,CPU 6)接收到尚未处理的抢占通知(128)。举例来说,GPU 12可检查指示是否已接收尚未处理的抢占通知的抢占通知接收状态变量的状态。响应于确定尚未接收到抢占通知,GPU 12将命令时隙中的命令分派给处理单元34中的一或多者(126)。在分派用于执行的命令之后,GPU 12可选择第一GPU命令流中的用于处理的下一命令时隙(122)。

[0110] 返回到决策框128,响应于确定已接收到抢占通知,GPU 12抢占第一GPU命令流的执行(130)。对第一GPU命令流的执行的抢占可涉及(例如)保存与第一GPU命令流的执行相关联的GPU 12的状态,并且,在一些情况下,加载GPU 12的先前保存的状态以用于执行第二GPU命令流。在一些实例中,GPU 12可通过读取由CPU 6写入的包含识别第二GPU命令流的位置的信息的GPU寄存器来确定响应于抢占通知而执行哪一命令流。

[0111] GPU 12执行第二GPU命令流(132)。在一些情况下,第二GPU命令流可包含允许对第二GPU命令流的抢占的抢占权标。在这些情况下,GPU 12可(例如)执行图8中展示的技术以用于执行第二GPU命令流。也就是说,GPU 12可开始在过程框120处执行第二GPU命令流。

[0112] 在完成第二GPU命令流的执行之后,GPU 12恢复第一GPU命令流的执行(134)。恢复第一GPU命令流的执行可涉及(例如)恢复与第一GPU命令流的执行相关联的GPU 12的所保存的状态。第一GPU命令流的执行还可涉及将抢占通知接收状态变量复位为指示尚未接收到抢占通知的值。在恢复第一GPU命令流的执行之后,GPU 12可选择第一GPU命令流中的用于处理的下一命令时隙(122)。

[0113] 如图8中所展示,GPU 12可响应于确定已接收到抢占通知(128-是)且用于当前命令处理循环的命令时隙含有权标(124-是)而抢占GPU命令流的执行(130)。另一方面,对于给定命令处理循环,GPU 12可响应于确定尚未接收到抢占通知(128-否)或用于当前命令处理循环的命令时隙不含有权标(124-否)而分派所述命令时隙中的命令以用于在GPU 12中的处理单元34中的一或多者上执行,而不在给定命令处理循环期间抢占GPU命令流的执行。

[0114] 图9是说明根据图8中展示的技术的用于在处理命令流时处置抢占通知的实例技术的流程图。GPU 12接收及/或检测抢占通知(136)。举例来说,GPU 12可轮询含有指示CPU 6是否已将抢占通知发送到GPU 12的值的GPU寄存器(例如,GPU硬件寄存器及/或GPU存储器映射的寄存器)。在CPU 6断言GPU寄存器中的值时,GPU 12检测已接收到抢占通知。在一些实例中,GPU 12可异步地检测抢占通知。举例来说,在GPU寄存器中的值已被CPU 6断言时,中断服务例程可处置所述断言。在额外实例中,GPU 12可同步地检测抢占通知。举例来说,GPU 12可针对每一命令时隙(例如,在处理命令时隙中的命令之前)轮询GPU寄存器以确定是否已接收到抢占通知。响应于接收及/或检测抢占通知,GPU 12改变抢占通知接收状态变量的状态以指示已接收到抢占通知(138)。

[0115] 图10是说明根据本发明的用于处理GPU命令流的另一实例技术的流程图。图10中所说明的技术可与指示其中允许发生或不允许发生抢占的特定抢占范围的起点及终点的抢占权标(例如,上文相对于图5所描述的抢占权标)一起使用。

[0116] GPU 12开始执行第一GPU命令流(140)。GPU 12选择用于处理的下一命令时隙(142)。在其中GPU 12在开始执行第一GPU命令流之后首次执行过程框142的情况下,GPU 12

选择第一GPU命令流中的用于处理的第一命令时隙。

[0117] GPU 12确定所述命令时隙是否含有权标(144)。响应于确定命令时隙不含有权标, GPU 12将命令时隙中的命令分派给处理单元34中的一或多者(146)。在分派用于执行的命令之后, GPU 12可选择第一GPU命令流中的用于处理的下一命令时隙(142)。所述下一命令时隙可为命令时隙序列中的紧接在先前处理的命令时隙之后出现的命令时隙。

[0118] 返回到决策框144, 响应于确定命令时隙确实含有权标, GPU 12基于所述权标而改变抢占允许状态变量(148)。举例来说, 如果所述权标表示代码的可抢占范围(即, 抢占范围)的起点或如果所述权标表示代码的非可中断范围的终点, 那么GPU 12可将抢占允许状态变量设定为指示当前允许抢占以用于处理当前及后续命令时隙的值。作为另一实例, 如果所述权标表示代码的可抢占范围(即, 抢占范围)的终点或如果所述权标表示代码的非可中断范围的起点, 那么GPU 12可将抢占允许状态变量设定为指示当前不允许抢占以用于处理当前及后续命令时隙的值。

[0119] GPU 12确定当前命令处理循环是否允许抢占(150)。举例来说, GPU 12可基于指示当前命令处理循环是否允许抢占的抢占允许状态变量来确定是否允许抢占。如上文相对于过程框148所论述, 可响应于处理GPU命令流中的一或多个权标而改变或设定抢占允许状态变量的值。响应于确定不允许抢占, GPU 12将命令时隙中的命令分派给处理单元34中的一或多者(146)。在分派用于执行的命令之后, GPU 12选择第一GPU命令流中的用于处理的下一命令时隙(142)。

[0120] 另一方面, 响应于确定允许抢占, GPU 12确定是否已从主机装置(例如, CPU 6)接收到尚未处理的抢占通知(152)。举例来说, GPU 12可检查指示是否已接收尚未处理的抢占通知的抢占通知接收状态变量的状态。响应于确定尚未接收到抢占通知, GPU 12将命令时隙中的命令分派给处理单元34中的一或多者而不必针对当前处理循环抢占GPU命令流(146)。在分派用于执行的命令之后, GPU 12选择第一GPU命令流中的用于处理的下一命令时隙(142)。

[0121] 返回到决策框152, 响应于确定已接收到抢占通知, GPU 12抢占第一GPU命令流的执行(154)。对第一GPU命令流的执行的抢占可涉及(例如)保存与第一GPU命令流的执行相关联的GPU 12的状态, 并且, 在一些情况下, 加载GPU 12的先前保存的状态以用于执行第二GPU命令流。在一些实例中, GPU 12可通过读取由CPU 6写入的且包含识别第二GPU命令流的位置的信息的GPU寄存器来确定响应于抢占通知而执行哪一命令流。

[0122] GPU 12执行第二GPU命令流(156)。在一些情况下, 第二GPU命令流可包含允许对第二GPU命令流的抢占的抢占权标。在这些情况下, GPU 12可(例如)执行图10中展示的技术以用于执行第二GPU命令流。也就是说, GPU 12可开始在过程框140处执行第二GPU命令流。

[0123] 在完成第二GPU命令流的执行之后, GPU 12恢复第一GPU命令流的执行(158)。恢复第一GPU命令流的执行可涉及(例如)恢复与第一GPU命令流的执行相关联的GPU 12的所保存的状态。第一GPU命令流的执行还可涉及将抢占通知接收状态变量复位为指示尚未接收到抢占通知的值。在恢复第一GPU命令流的执行之后, GPU 12可选择第一GPU命令流中的用于处理的下一命令时隙(142)。

[0124] 如图10中所展示, GPU 12可响应于确定已接收到抢占通知(152-是)且当前命令处理循环允许抢占(150-是)而抢占GPU命令流的执行(154)。另一方面, 对于给定命令处理循

环, GPU 12可响应于确定尚未接收到抢占通知(152-否)或当前命令处理循环不允许抢占(150-否)而分派所述命令时隙中的命令以用于在GPU 12中的处理单元34中的一或多个上执行,而不在给定命令处理循环期间抢占GPU命令流的执行。

[0125] 图11是说明根据图10中展示的技术的用于在处理命令流时处置抢占通知的实例技术的流程图。GPU 12接收及/或检测抢占通知(160)。举例来说, GPU 12可以上文相对于图9的过程框136所论述的方式类似的方式轮询含有指示CPU 6是否已将抢占通知发送到GPU 12的值的GPU寄存器(例如, GPU硬件寄存器及/或GPU存储器映射的寄存器)。

[0126] 响应于接收及/或检测到抢占通知, GPU 12确定当前命令处理循环是否允许抢占(162)。举例来说, GPU 12可基于指示当前命令处理循环是否允许抢占的抢占允许状态变量来确定是否允许抢占。响应于确定允许抢占, GPU 12抢占第一GPU命令流的执行(164)且执行第二GPU命令流。响应于确定不允许抢占, GPU 12将抢占通知接收状态变量的状态改变为指示已接收到抢占通知的值(166), 且返回到根据图10的处理命令时隙, 直到遇到指示允许抢占的抢占权标为止。在遇到此抢占权标之后, GPU 12可根据图10中概述的技术基于先前抢占通知的接收而抢占第一GPU命令流的执行。

[0127] 本发明的各种方面描述用于中断排队的GPU命令流的解析且服务待决的较高优先级命令流的延迟抢占方案。在一些实例中, 本发明的技术可允许GPU命令流队列中的被权标标记的选定点处的抢占。在一些实例中, 所述权标可将命令流分裂为非可中断的片段, 进而防止抢占发生在命令流的敏感部分中。在一些实例中, 可静态地或动态地选择抢占点。在一些实例中, 在起始抢占请求(即, 抢占通知)时, GPU中的命令解析器(例如, 命令引擎32)可继续解析命令流, 直到到达最近的权标为止, 在此处, 命令解析器可切换到由所述抢占请求指示的命令流, 且在完成时返回到正常的命令流。

[0128] 本发明中所描述的技术可至少部分实施于硬件、软件、固件或其任何组合中。举例来说, 所描述技术的各种方面可实施于一或多个处理器中, 包含一或多个微处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA), 或任何其它等效集成或离散逻辑电路, 以及此类组件的任何组合。术语“处理器”或“处理电路”可大体上指前述逻辑电路中的任一者(单独或结合其他逻辑电路)或例如执行处理的离散硬件的任何其它等效电路。

[0129] 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类模块或单元必须由单独硬件或软件组件实现。而是, 与一或多个模块或单元相关联的功能性可由单独硬件、固件和/或软件组件执行, 或整合到共用或单独硬件或软件组件内。

[0130] 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、盒式磁带、磁性媒体、光学媒体或其它有形计算机可读存储媒体。

[0131] 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。计算机可读媒体也可包括通信媒体,其包含促进计算机程序从一个地点到另一地点的传送(例如,根据通信协议)的任何媒体。以此方式,短语“计算机可读媒体”大体上可对应于(1)非暂时性有形计算机可读存储媒体和(2)例如暂时性信号或载波的非有形计算机可读通信媒体。

[0132] 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。

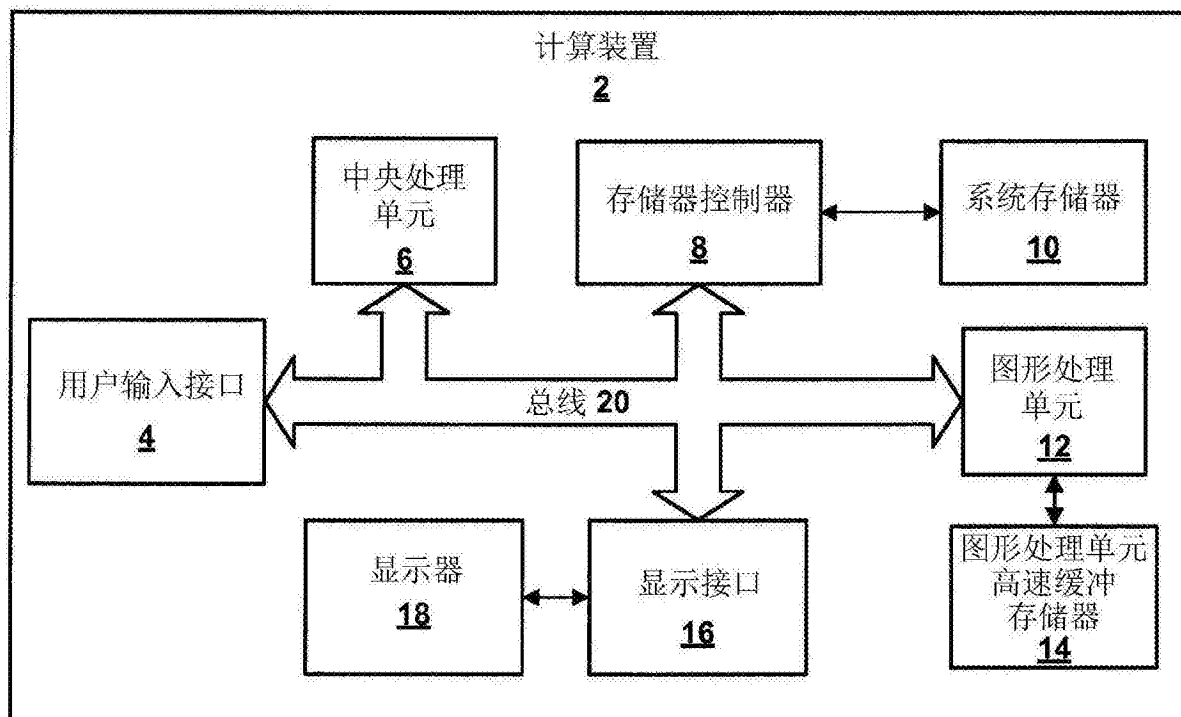


图1

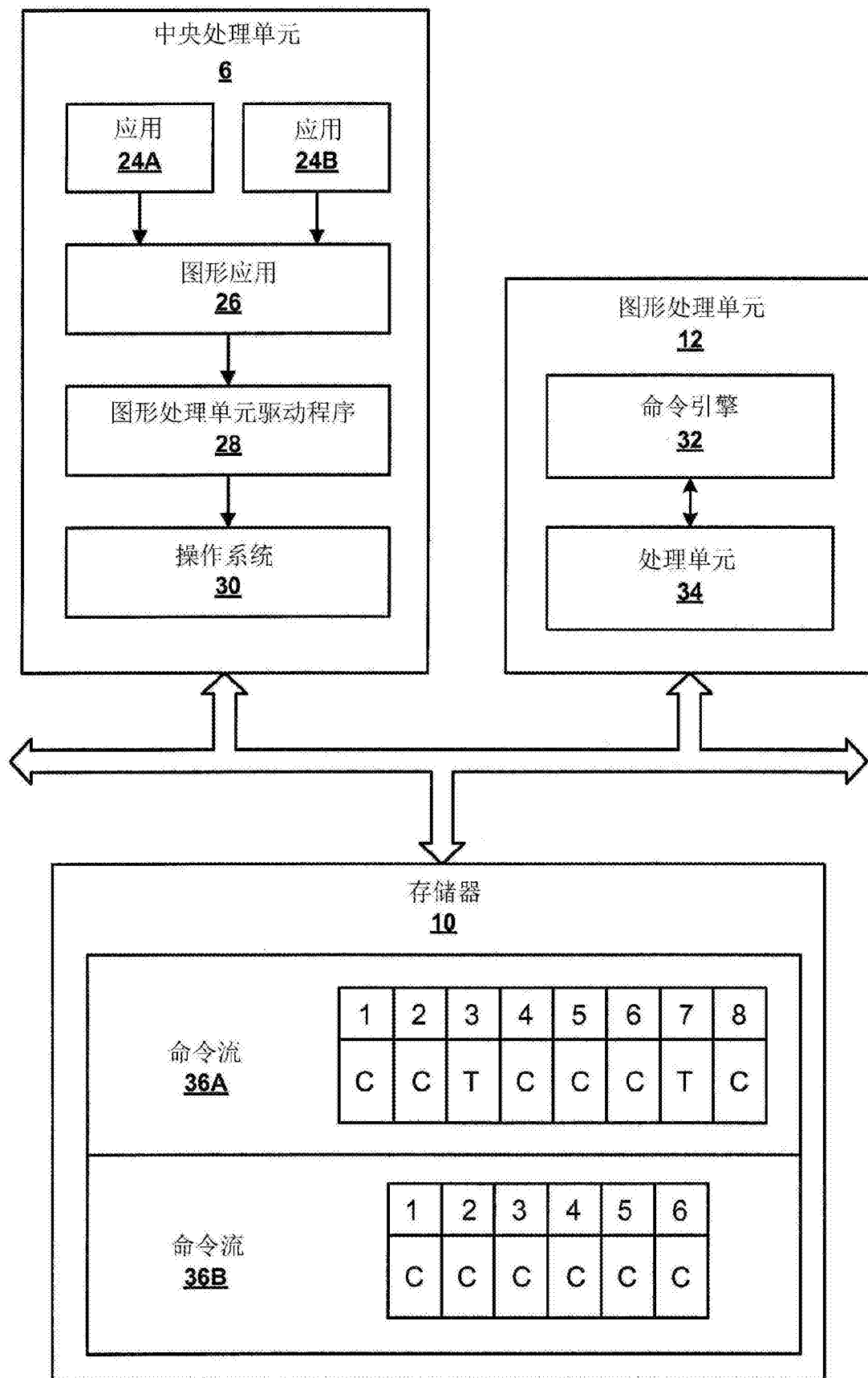


图2

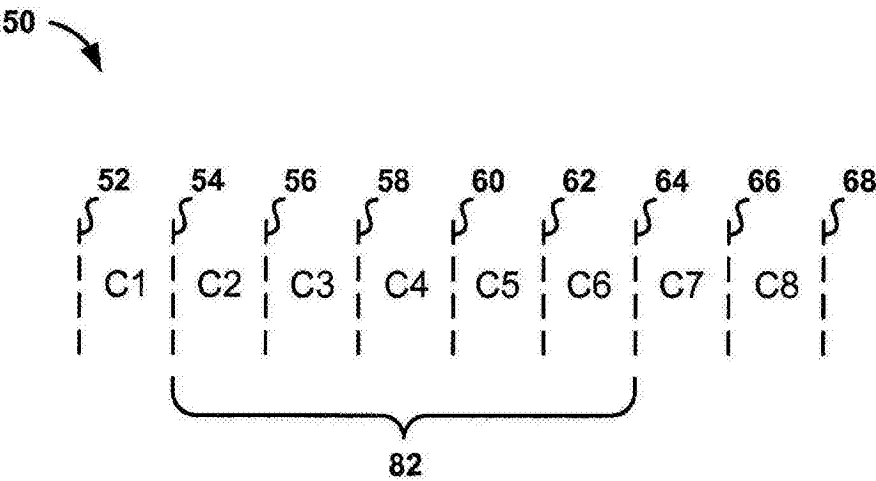


图3

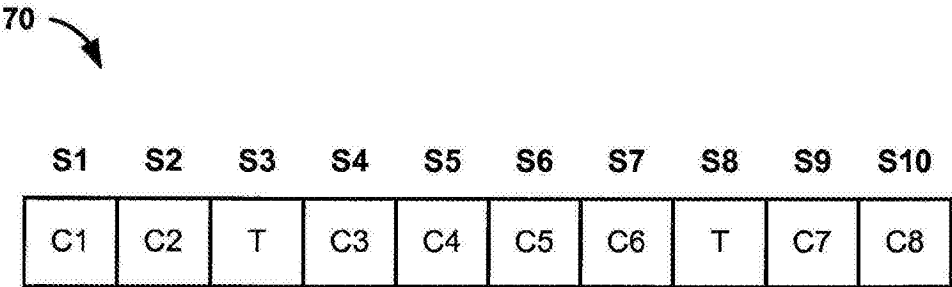


图4

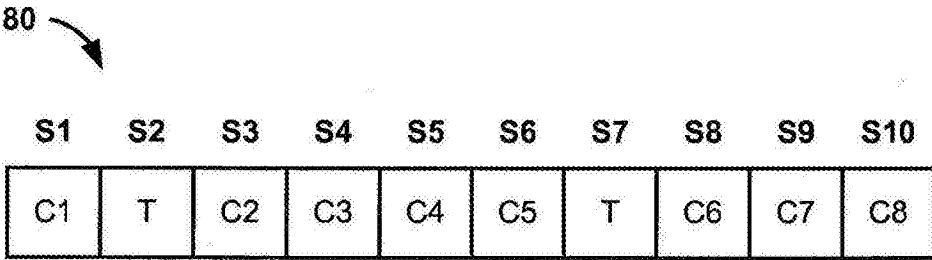


图5

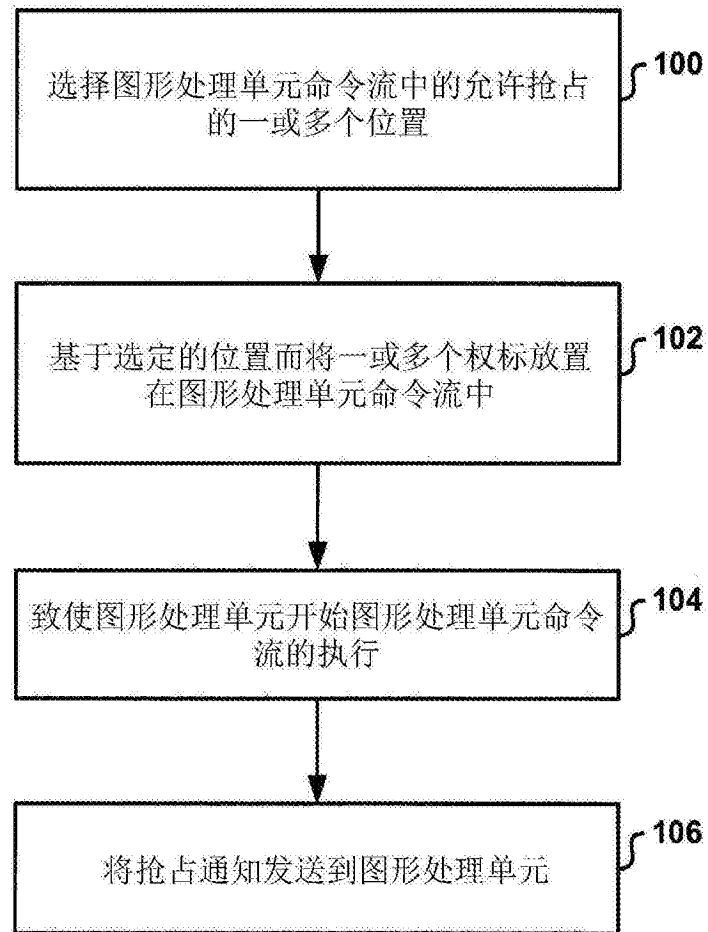


图6

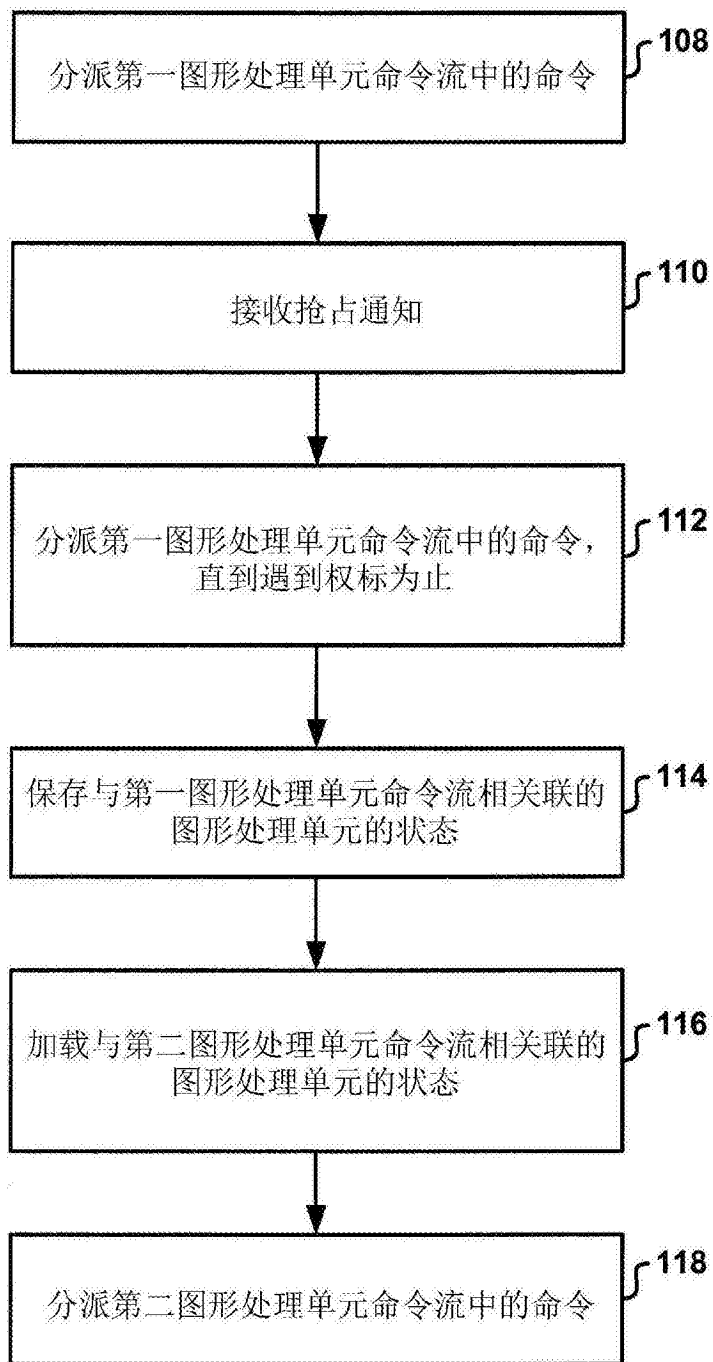


图7

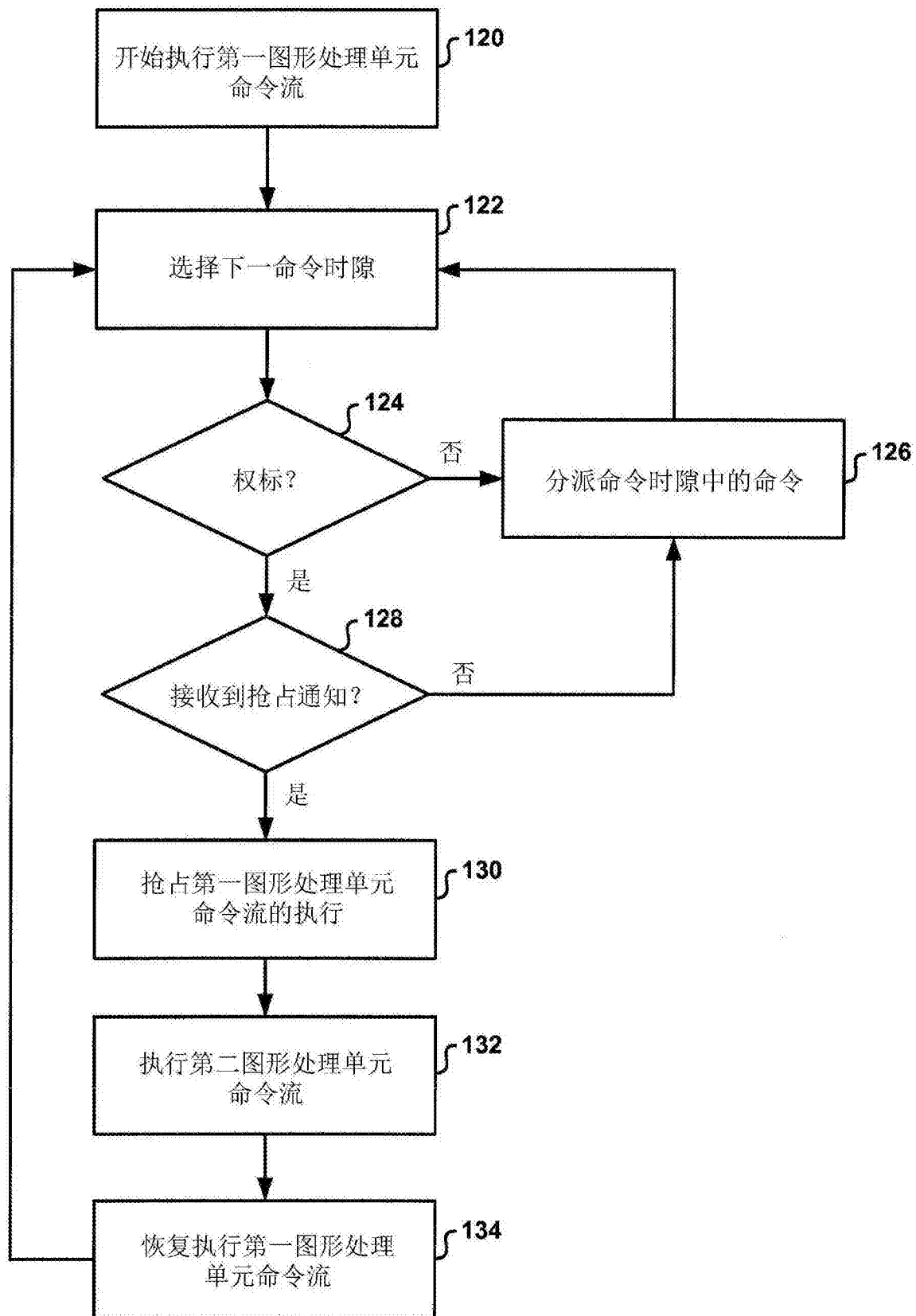


图8

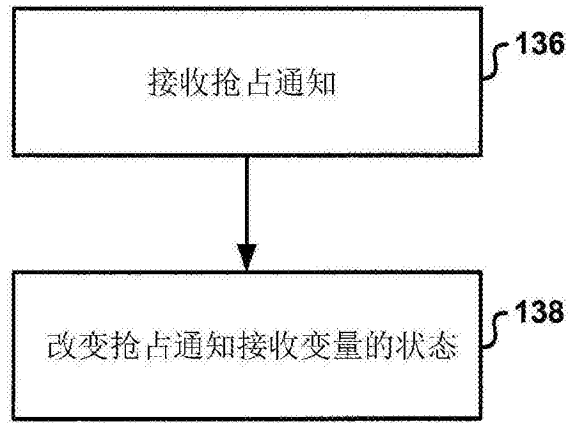


图9

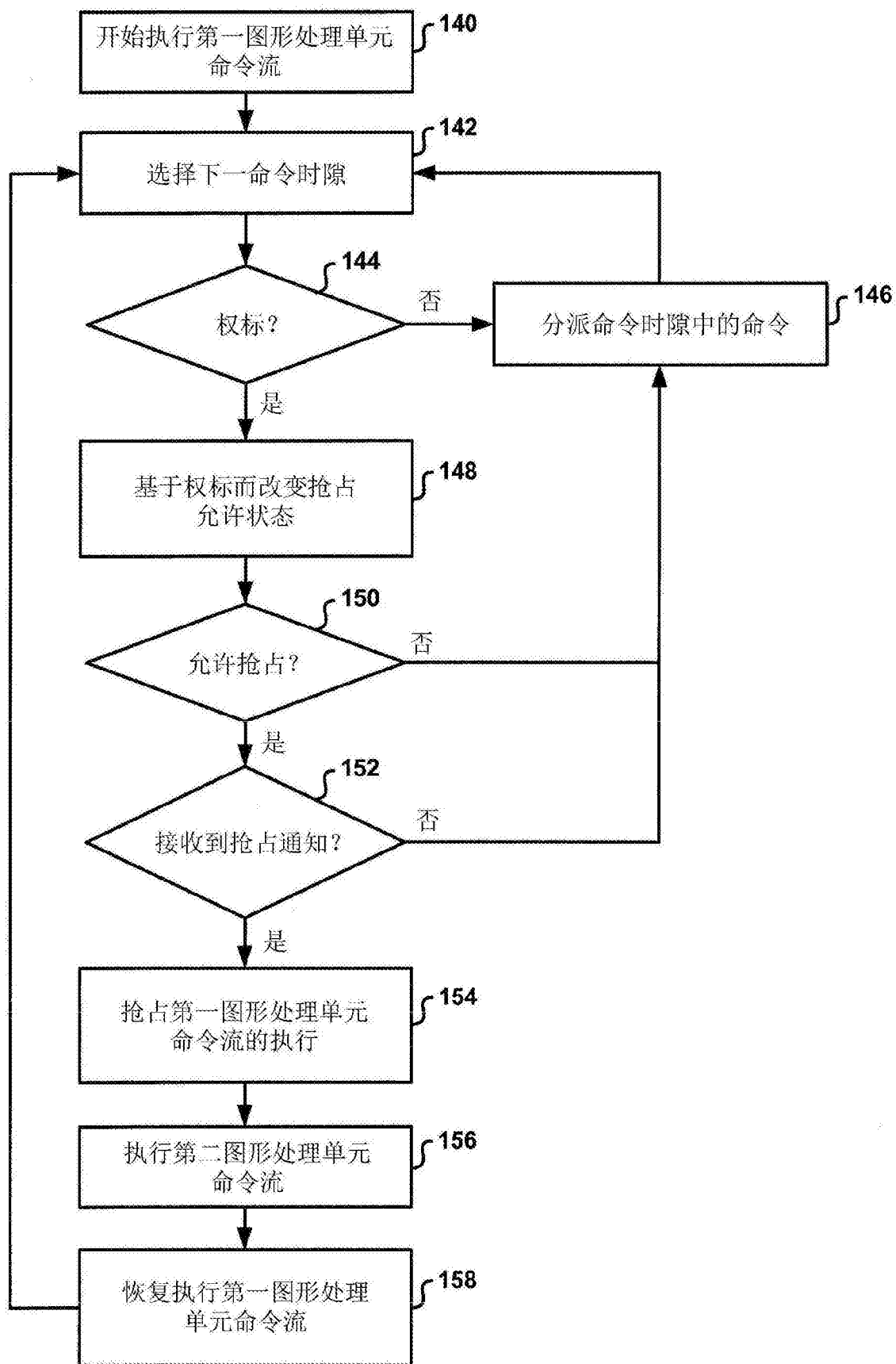


图10

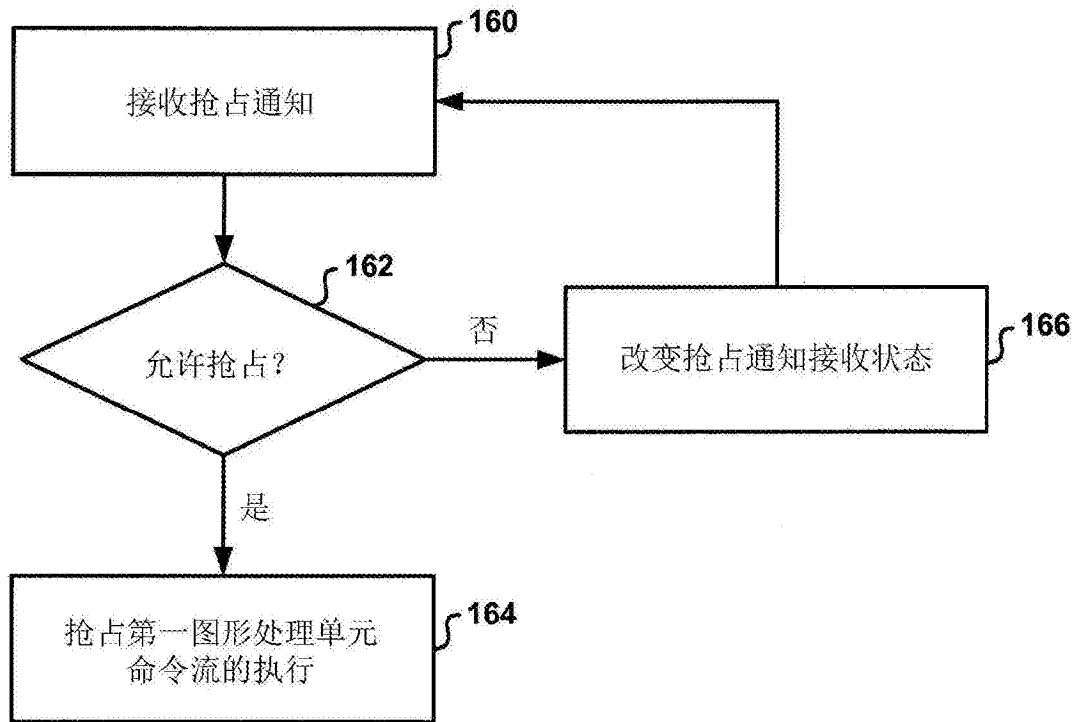


图11