(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103718156 B
(45)授权公告日 2018.05.01

(21)申请号 201280037791.4

(22)申请日 2012.06.29

(65)同一申请的已公布的文献号
申请公布号 CN 103718156 A

(43)申请公布日 2014.04.09

(30)优先权数据
13/193,779 2011.07.29 US

(85)PCT国际申请进入国家阶段日
2014.01.28

(86)PCT国际申请的申请数据
PCT/US2012/044805 2012.06.29

(87)PCT国际申请的公布数据
WO2013/019350 EN 2013.02.07

(73)专利权人 英特尔公司
地址 美国加利福尼亚州

(72)发明人 B·靳兹伯格 E·纳坦森
I·殴萨池依 Y·扎克

(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 张东梅

(51)Int.Cl.
G06F 9/06(2006.01)
G06F 15/163(2006.01)

(56)对比文件
CN 101685391 A,2010.03.31,
CN 102103567 A,2011.06.22,
US 2007222798 A1,2007.09.27,
US 2011067038 A1,2011.03.17,
CN 1311351 C,2007.04.18,
CN 100578454 C,2010.01.06,
CN 101398753 A,2009.04.01,
CN 101685391 A,2010.03.31,

审查员 邵娜娜

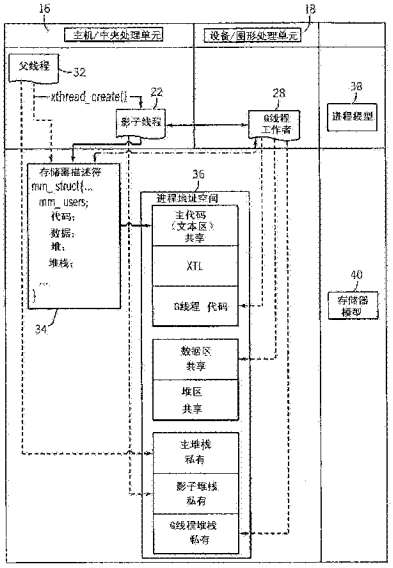
权利要求书2页 说明书3页 附图4页

(54)发明名称

CPU/GPU同步机制

(57)摘要

可以使用一个处理器上的线程来使另一个处理器能对互斥上锁或释放。例如,中央处理单元线程可以由图形处理单元使用以保护共享存储器的互斥。



CN 103718156 B

1.一种用于同步的方法,所述方法包括:

使用中央处理单元上的第一线程和线程库在所述中央处理单元上派生第二线程,专门适用于同步所述第一线程以及图形处理单元上的第三线程,所述同步是通过:在所述第二线程中接收来自所述第三线程的请求、作为响应向所述中央处理单元的操作系统发送系统调用、所述第二线程将响应从所述操作系统发送至所述第三线程,所述线程库用于管理所述图形处理单元上的用户线程;以及

使用所述第二线程来同步图形处理单元上的所述第三线程以及所述中央处理单元上的所述第一线程。

2.如权利要求1所述的方法,其特征在于,包括使用运行在所述中央处理单元上的所述第二线程来对所述图形处理单元上的线程的互斥上锁和解锁。

3.如权利要求1所述的方法,其特征在于,所述线程库具有用于同步的库函数。

4.如权利要求3所述的方法,其特征在于,包括使所述中央处理单元能在互斥可用时通知所述图形处理单元。

5.如权利要求1所述的方法,其特征在于,包括使用与所述中央处理单元和所述图形处理单元共享的虚拟存储器。

6.一种用于同步的装置,包括:

存储器;以及

与所述存储器耦合的处理器,所述处理器适用于:

使用中央处理单元上的第一线程和线程库在所述中央处理单元上派生第二线程,专门适用于同步所述第一线程以及图形处理单元上的第三线程,所述同步是通过:在所述第二线程中接收来自所述第三线程的请求、作为响应向所述中央处理单元的操作系统发送系统调用、所述第二线程将响应从所述操作系统发送至所述第三线程,所述线程库用于管理所述图形处理单元上的用户线程;以及

使用所述第二线程来同步图形处理单元上的所述第三线程以及所述中央处理单元上的所述第一线程。

7.如权利要求6所述的装置,其特征在于,所述处理器还适用于使用运行在中央处理单元上的所述第二线程来对所述图形处理单元上的线程的互斥上锁。

8.如权利要求7所述的装置,其特征在于,所述处理器还适用于使用运行在所述中央处理单元上的所述第二线程来对所述图形处理单元上的线程的互斥解锁。

9.如权利要求7所述的装置,其特征在于,所述线程库具有用于同步的库函数。

10.如权利要求9所述的装置,其特征在于,所述处理器还适用于使所述中央处理单元能在互斥可用时通知所述图形处理单元。

11.如权利要求6所述的装置,其特征在于,所述处理器还适用于使用与所述中央处理单元和所述图形处理单元共享的虚拟存储器。

12.一种用于同步的装置,所述装置包括:

图形处理单元;

中央处理单元,所述中央处理单元耦合到所述图形处理单元;以及

存储器,所述存储器由所述图形处理单元和所述中央处理单元共享,所述中央处理单元提供中央处理单元上的第一线程和线程库在所述中央处理单元上派生第二线程、专门适

用于同步所述第一线程以及图形处理单元上的第三线程，所述同步是通过：在所述第二线程中接收来自所述第三线程的请求、作为响应向所述中央处理单元的操作系统发送系统调用、所述第二线程将响应从所述操作系统发送至所述第三线程，以及使用所述第二线程来同步所述图形处理单元上的所述第三线程以及所述中央处理单元上的所述第一线程，所述线程库用于管理所述图形处理单元上的用户线程。

13.如权利要求12所述的装置，其特征在于，包括所述中央处理单元上的图形处理单元驱动器。

14.如权利要求12所述的装置，其特征在于，所述第二线程对所述图形处理单元上的线程的互斥进行上锁和解锁。

15.如权利要求14所述的装置，其特征在于，所述中央处理单元在互斥可用时通知所述图形处理单元。

16.如权利要求12所述的装置，其特征在于，所述存储器是共享的虚拟存储器。

17.一种用于同步的装置，所述装置包括：

中央处理单元；

存储器，所述存储器耦合到所述中央处理单元；以及

所述中央处理单元启动第一线程来启用图形处理单元上的第三线程和所述中央处理单元上的第二线程之间的同步，所述同步是通过：在所述第二线程中接收来自所述第三线程的请求、作为响应向所述中央处理单元的操作系统发送系统调用、所述第二线程将响应从所述操作系统发送至所述第三线程，所述第一线程由所述第二线程派生。

18.如权利要求17所述的装置，其特征在于，所述中央处理单元使用所述第一线程对运行在图形处理单元上的线程的互斥进行上锁和解锁。

19.如权利要求18所述的装置，其特征在于，所述中央处理单元在互斥可用时通知所述图形处理单元。

20.如权利要求17所述的装置，其特征在于，包括耦合到所述中央处理单元的图形处理器。

CPU/GPU同步机制

背景技术

[0001] 这一般地涉及具有通用处理器和图形处理单元的计算机。

[0002] 运行在通用或中央处理单元上由用户应用所使用的存储器和由图形处理单元所使用的处理器通常是分开的。图形处理单元驱动器将数据从用户空间复制到驱动器存储器以在图形处理单元上进行处理。在共享的虚拟存储器模型中，数据没有被复制到图形处理单元，而是相反地，它在图形处理单元和中央处理单元之间共享。

[0003] 当前，在多线程应用中，共享的数据由称为互斥的锁来保护。每个想要访问共享数据的线程都必须首先对相对应的互斥上锁以阻止其它线程访问该互斥。此上锁可以通过锁上的“回旋(spinning)”来进行，但这个技术从功率和性能的角度效率不高。

[0004] 为了优化中央处理单元，操作系统提供允许线程休眠直到有互斥可用、然后当互斥被解锁时通知其它线程的系统调用。但这个机制只对于运行在中央处理单元核上的线程生效。

附图说明

[0005] 图1是本发明的一个实施例的示意图；

[0006] 图2是根据本发明的一个实施例的扩展线程和存储器模型；

[0007] 图3是根据本发明的一个实施例的页错误处理的流程图；以及

[0008] 图4是一个实施例的系统图。

具体实施方式

[0009] 在一个实施例中，对每个运行在图形处理器上的线程群，可以创建运行在中央处理单元上的辅助影子线程。当运行在图形处理单元上的线程想要对互斥上锁时，如果它想要等待直到该线程由其它任务释放，则图形线程向中央处理单元上的影子线程发送请求。中央处理单元上的影子线程向操作系统发出相对应的系统调用。当操作系统将锁给影子线程时，影子线程向图形处理单元上的线程发送通知。

[0010] 当在本申请中使用术语图形处理单元时，应理解图形处理单元可以是或可以不是单独的集成电路。本发明对图形处理单元和中央处理单元集成到一个集成电路的情形适用。如此处所使用的，处理器或处理单元可以是处理器、控制器或协处理器。

[0011] 参考图1，在基于处理器的系统10中，主/中央处理单元16与图形处理单元18通信。主/中央处理单元16可以是包括硬连线或移动设备的任何基于处理器的系统的一部分。举几个示例，移动设备的示例包括蜂窝电话、膝上型计算机、平板计算机和移动因特网设备。主中央处理单元16包括用户应用20，它向影子线程22提供控制信息。然后影子线程22向图形处理单元驱动器26通信关于同步和控制信息的同步。影子线程也和主操作系统24通信。

[0012] 如图1中所示，用户级别12包括影子线程22和用户应用20，而内核级别14包括主操作系统24和图形处理单元驱动器26。图形处理单元驱动器26是图形处理单元的驱动器，即使该驱动器驻留在中央处理单元16中。

[0013] 在用户级别12中，图形处理单元18包括g线程(gthread)28，它向操作系统(pOS)30发送控制和同步消息并从操作系统30接收消息。g线程是运行在图形处理单元上的用户代码，与运行在中央处理单元上的父线程共享虚拟存储器。操作系统30可以是相对小的操作系统，它运行在图形处理单元上，负责图形处理单元异常。作为一个示例，它相对于主操作系统24是小的。

[0014] 用户应用20是运行在中央处理单元16上的任何用户进程。用户应用20派生图形处理单元18上的线程。

[0015] 扩展线程库或XTL是创建和管理图形处理单元上的用户线程的扩展。这个库为每个g线程创建影子线程并具有用于同步的库函数。

[0016] 用户应用使用传统多线程模型的扩展将计算卸载到图形处理单元，扩展诸如：

[0017] xthread_create(thread,attr,gpu_worker,arg)。

[0018] 在图形处理单元上创建的g线程或工作者线程与父线程共享虚拟存储器。在可以使用所有标准的过程间同步机制诸如互斥和信号机的意义上，它以与常规线程相同的方式行为。同时，在主中央处理单元16上创建新影子线程。这个影子线程充当异常处理单元的代理和中央处理单元与图形处理单元上的线程之间的同步。

[0019] 在一些实施例中，父线程、主影子线程和图形处理单元工作者线程可以共享虚拟存储器，如图2中所示。主/中央处理单元16包括为影子线程22生成xthread_create()的父线程32。影子线程22访问影子堆栈，影子堆栈是进程地址空间36中的私有地址空间。父线程32也访问存储器描述符34和主堆栈，主堆栈是进程地址空间36内的私有地址空间。存储器描述符34也可以与g线程工作者28通信。g线程工作者28可以访问进程空间36内的g线程代码以及共享数据区和私有g线程堆栈。较上的框中的材料对应于进程模型38，并且较下的框相对应于存储器模型40。

[0020] 参考图3，同步算法可以用硬件、软件和/或固件来实现。在软件实施例中，算法可以作为存储在非瞬态计算机可读介质上的计算机可执行指令来实现，介质诸如光学、半导体或磁存储设备。在图3中，中央处理单元16的主操作系统24、影子线程22、驱动器26和图形处理单元18中的操作系统30和环357的流程示为并行的垂直流程路径，且一般由水平箭头指示的它们之间的交互。

[0021] 继续在图3中，在一个实施例中，在环3，g线程28试图通过回旋(spinning)一小段时间来从用户空间获取互斥，如列57中的框60中所指示的。如果成功地获取了互斥，如菱形61中所确定的，则程序继续，如框62中所指示的。如果由于互斥已经由另一线程上锁而没有被获取，则用ACQUIRE操作码进行对操作系统30的调用(SYSCALL)。

[0022] 在操作系统30中接收SYSCALL。操作系统30向驱动器发送称为PassHostCommand(通过主机命令)的消息。PassHostCommand包括SYSCALL的操作码和地址加操作的数据，如框50中所指示的。然后操作系统30将调用线程置为休眠，如框54中所指示的。然后调用线程进入空闲，如框56中所指示的。

[0023] 中央处理单元16中的驱动器26使用传输命令将消息传输给Xthread22。在Xthread22中，接收传输命令。Xthread等待来自驱动器的这个消息，如框44中所指示的。然后Xthread执行对互斥上锁，如框46中所指示的，并可以包括对主操作系统24的SYSCALL。操作系统24等待，直到获取了互斥(即，存在上下文切换)，如42处所指示的。在获取了互斥之

后，Xthread向驱动器26发送恢复执行命令，然后等待来自驱动器的下一消息，如框48中所指示的。

[0024] 驱动器26接收恢复执行命令并向图形处理单元中的操作系统30发送PassGpuCommand(通过Gpu命令)消息。PassGpuCommand可以包括不带数据的resume_execution的操作码，如框52中所指示的。

[0025] 操作系统30唤醒线程，如框58中所指示的，并向g线程28发送来自SYSCAL的恢复。结果是，g线程28继续工作，如框62中所指示的。影子线程也可以用来释放互斥。

[0026] 图4中所示的计算机系统130可以包括通过总线104耦合到芯片组核逻辑110的硬驱动器134和可移动介质136。键盘和鼠标120或其它传统组件可以经由总线108耦合到芯片组核逻辑。在一个实施例中，核逻辑可以经由总线105耦合到图形处理器112，和耦合到中央处理器100。图形处理器112还可以通过总线106耦合到帧缓冲器114。帧缓冲器114可以通过总线107耦合到显示屏幕118。在一个实施例中，图形处理器112可以是使用单指令多数据(SIMD)架构的多线程、多核并行处理器。

[0027] 在软件实现的情况中，相关代码可以存储在任何适合的半导体、磁或光学存储器中，包括主存储器132(如139处所指示的)或图形处理器内任何可用的存储器。因此，在一个实施例中，执行图3的序列的代码可以存储在非瞬态机器或计算机可读介质(诸如存储器132和/或图形处理器112和/或中央处理器100)中，并在一个实施例中可以由处理器100和/或图形处理器112执行。

[0028] 图3是流程图。在一些实施例中，在此流程图中描绘的序列可以用硬件、软件或固件实现。在软件实施例中，非瞬态计算机可读介质，诸如半导体存储器、磁存储器或光学存储器，可以用来存储指令并可以由处理器执行以实现图3中所示的序列。

[0029] 此处描述的图形处理技术可以用各种硬件架构实现。例如，图形功能可以集成在芯片组中。替换性地，可以使用离散图形处理器。如又一实施例，图形功能可以通过通用处理器(包括多核处理器)来实现。

[0030] 贯穿本说明书的对“一个实施例”和“实施例”的引用意味着结合实施例所描述的特定特征、结构或特性包括在涵盖于本发明的至少一个实现中。因此，短语“一个实施例”或“在实施例中”的出现不必全都指的是同一实施例。而且，除了所示的特定实施例之外，特定特征、结构和特性还可以用其它适合的形式建立，且所有这样的形式都可以涵盖于本申请的权利要求书中。

[0031] 尽管参考有限数量的实施例描述了本发明，本领域的技术人员从中可以理解众多修改和变化。所附权利要求书旨在覆盖所有这样落入本发明的真正的精神和范围内的修改和变化。

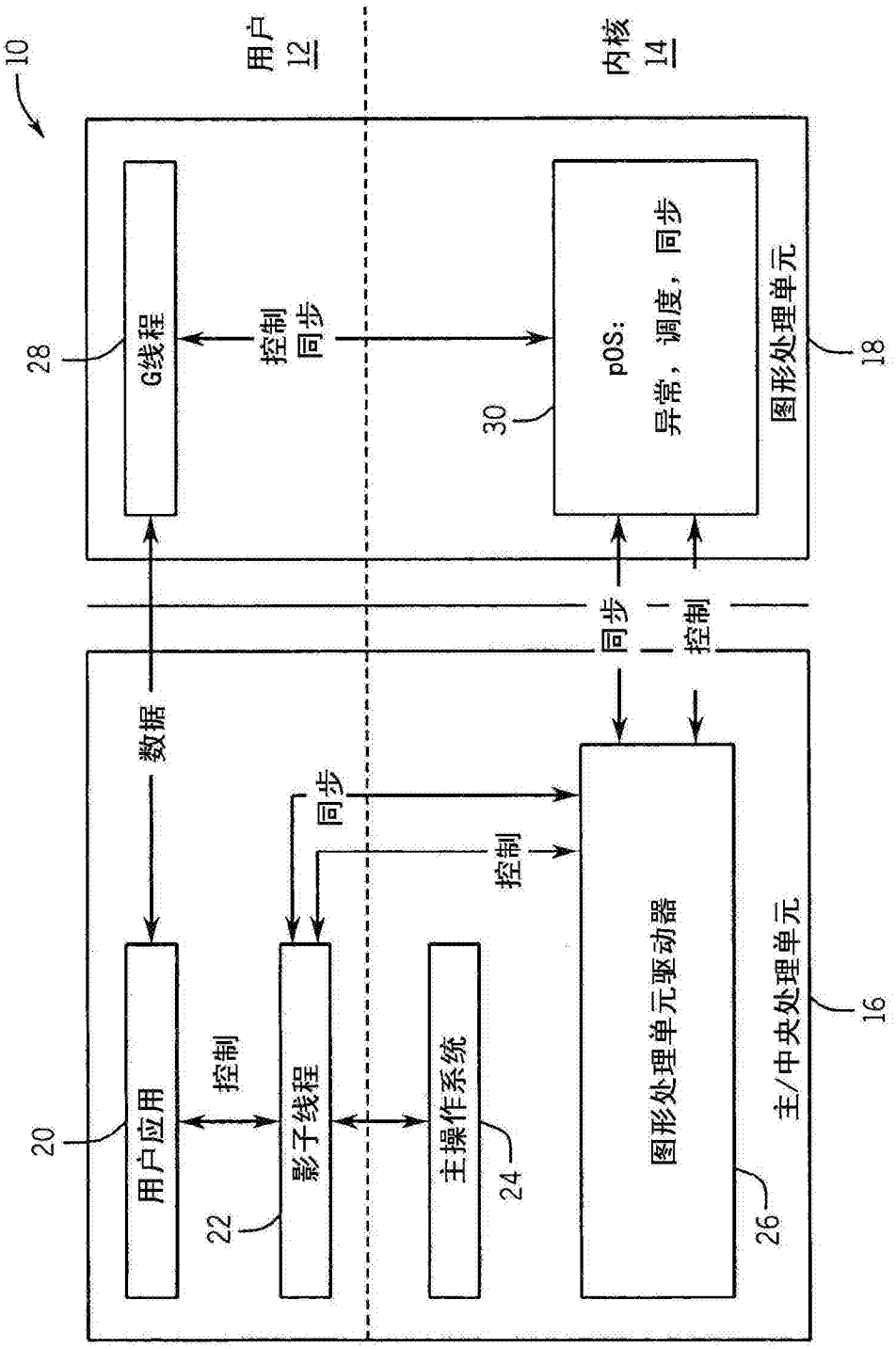

10
用户
12
内核
14
28
G线程
控制
同步
30
pOS:
异常，调度，同步
图形处理单元
18
同步
控制
数据
同步
控制
20
用户应用
控制
影子线程
主操作系统
图形处理单元驱动器
主/中央处理单元
16
22
24
26

图1

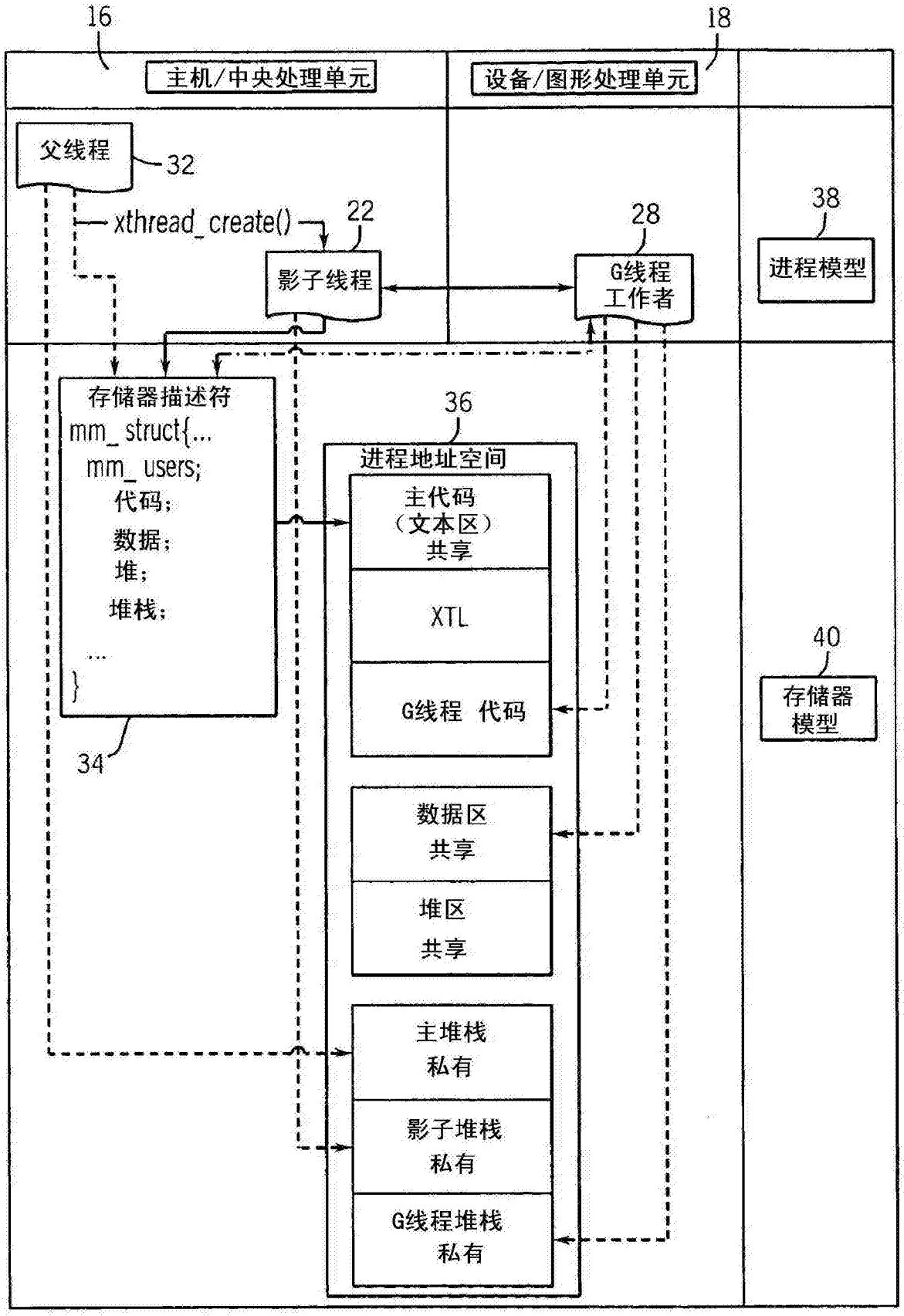

16
18
主机/中央处理单元
设备/图形处理单元
父线程
32
xthread_create()
22
影子线程
28
G线程
工作者
38
进程模型
存储器描述符
mm_ struct{...
mm_ users;
代码;
数据;
堆;
堆栈;
...
}
34
36
进程地址空间
主代码
（文本区）
共享
XTL
G线程 代码
数据区
共享
堆区
共享
主堆栈
私有
影子堆栈
私有
G线程堆栈
私有
40
存储器
模型

图2

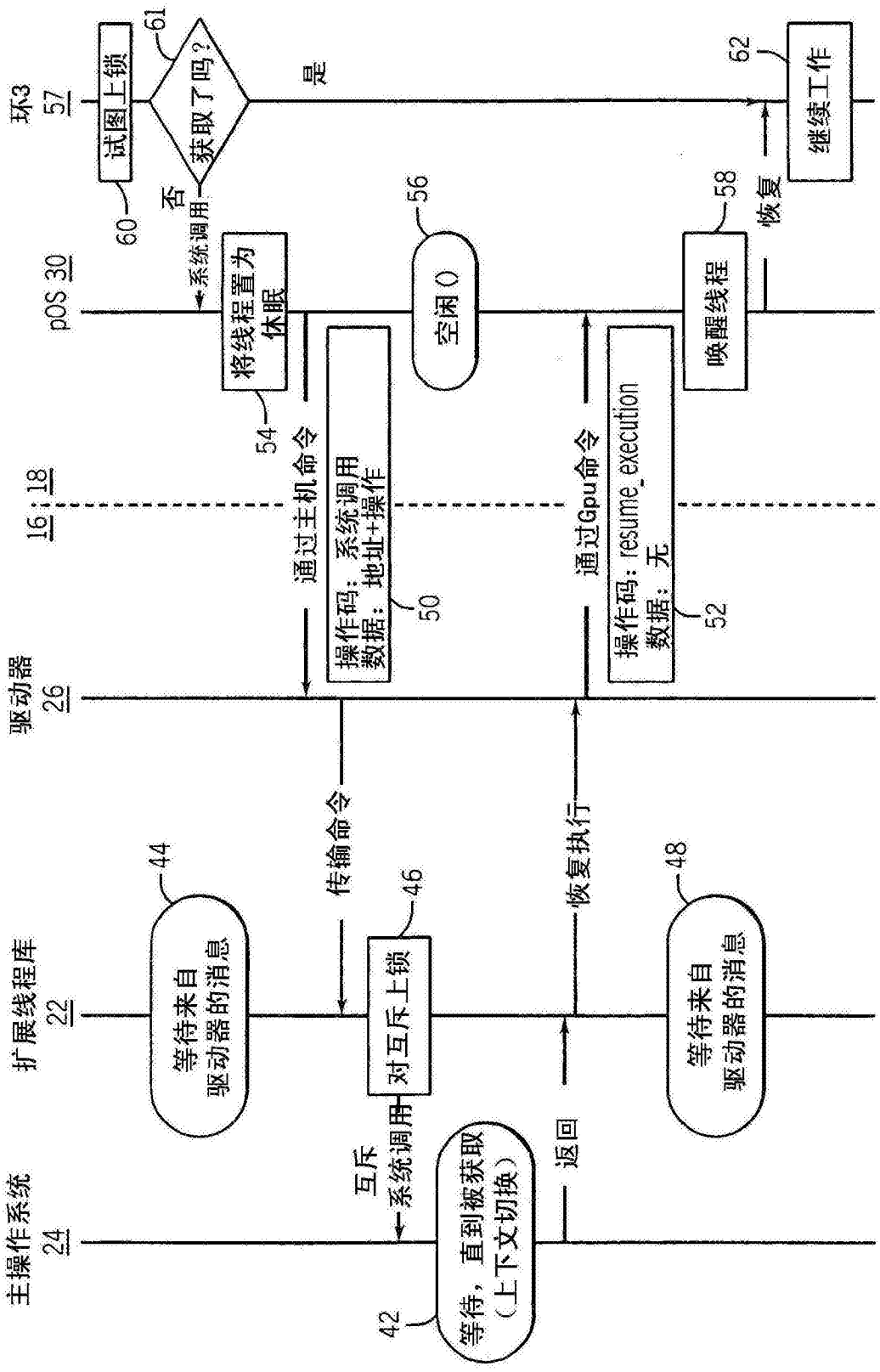

环3
57
试图上锁
61
获取了吗？
是
否
62
继续工作
60
系统调用
恢复
pOS 30
将线程置为休眠
54
56
空闲()
58
唤醒线程
16 ; 18
通过主机命令
操作码：系统调用
数据：地址+操作
50
通过Gpu命令
操作码：resume_execution
数据：无
52
驱动器
26
传输命令
恢复执行
扩展线程库
22
44
等待来自驱动器的消息
46
对互斥上锁
48
等待来自驱动器的消息
互斥
系统调用
返回
主操作系统
24
42
等待，直到被获取（上下文切换）

图3

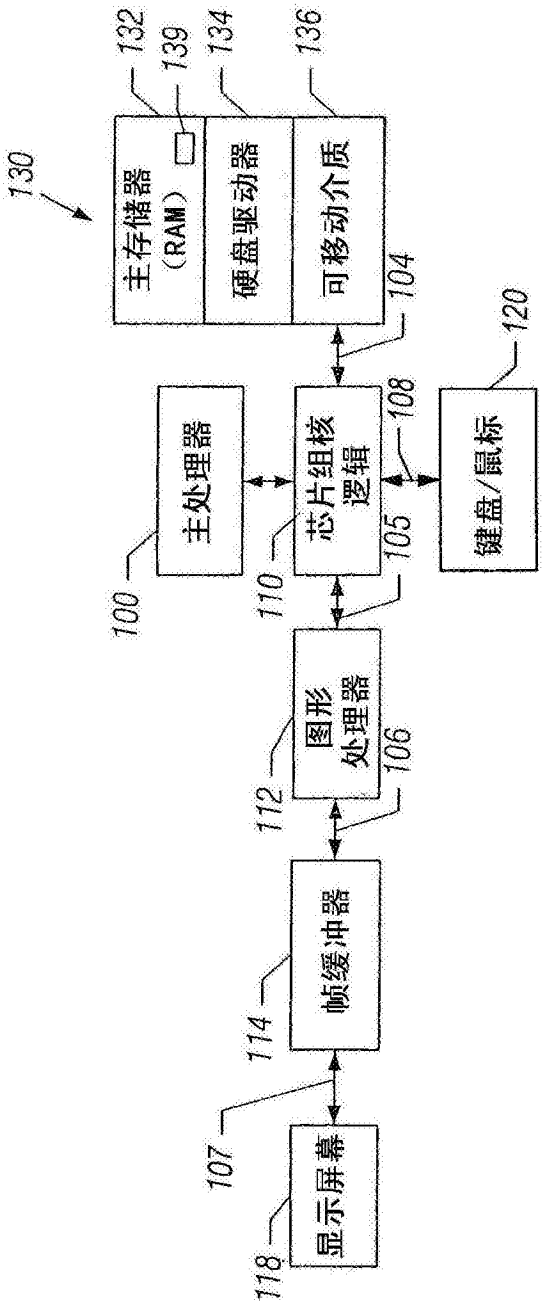

130
主存储器
(RAM)
硬盘驱动器
可移动介质
132
139
134
136
104
120
主处理器
芯片组核
逻辑
键盘/鼠标
108
100
110
105
图形
处理器
112
106
帧缓冲器
114
107
显示屏幕
118

图4