(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 107851138 A
(43)申请公布日 2018.03.27

(21)申请号 201680042506.6

(22)申请日 2016.07.25

(30)优先权数据
14/821,174 2015.08.07 US

(85)PCT国际申请进入国家阶段日
2018.01.19

(86)PCT国际申请的申请数据
PCT/US2016/043903 2016.07.25

(87)PCT国际申请的公布数据
WO2017/027196 EN 2017.02.16

(71)申请人 高通股份有限公司
地址 美国加利福尼亚州

(72)发明人 科林·克里斯托弗·夏普
拉梅什·维斯瓦纳坦

(74)专利代理机构 北京律盟知识产权代理有限责任公司 11287
代理人 杨林勳

(51)Int.Cl.
G06F 21/10(2006.01)

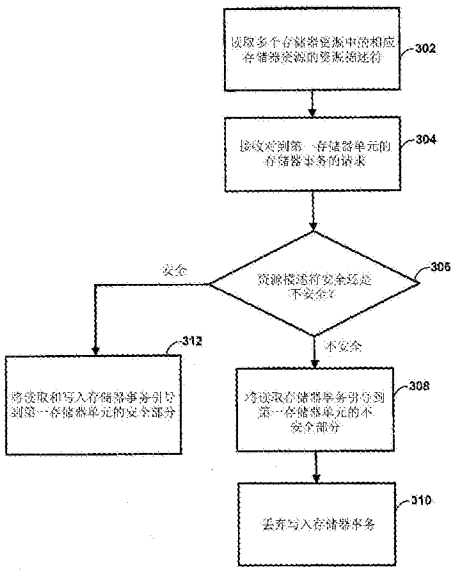
权利要求书5页 说明书18页 附图12页

(54)发明名称

用于图形处理单元的硬件强制内容保护

(57)摘要

本发明提出用于图形处理的技术。在一个实例中，图形处理单元GPU经配置以根据不安全模式和安全模式中的一者来存取存储器。所述GPU可包含存储器存取控制器，其经配置以基于所述不安全模式或安全模式以及与存储器资源相关联的资源描述符，将存储器事务从所述GPU的至少一个硬件单元引导到不安全存储器单元或安全存储器单元。



CN 107851138 A

1.一种用于图形处理的设备，其包括：

图形处理单元GPU，其经配置以根据不安全模式和安全模式中的一者以及与多个存储器资源中的每一者相关联的相应资源描述符，来存取第一存储器单元，所述GPU包括：

存储器存取控制器，其经配置以读取与所述多个存储器资源中的每一者相关联的所述相应资源描述符，

所述存储器存取控制器经配置以接收对到所述第一存储器单元的存储器事务的请求，

所述存储器存取控制器经配置以在所述GPU根据所述安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是安全资源描述符的存储器资源有关的所有读取和写入存储器事务引导到所述第一存储器单元的安全部分，

所述存储器存取控制器经配置以在所述GPU根据所述安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是不安全资源描述符的存储器资源有关的所有读取存储器事务引导到所述第一存储器单元的不安全部分，且

所述存储器存取控制器经配置以在所述GPU根据所述安全模式操作时，响应于所述请求，丢弃与所述多个存储器资源中对于其来说所述相应资源描述符是所述不安全资源描述符的存储器资源有关的所有写入存储器事务。

2.根据权利要求1所述的设备，其中所述存储器存取控制器进一步经配置以在所述GPU根据所述不安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是所述不安全资源描述符的存储器资源有关的读取和写入存储器事务引导到所述第一存储器单元的不安全部分，且

所述存储器存取控制器进一步经配置以在所述GPU根据所述不安全模式操作时，响应于所述请求，丢弃与所述多个存储器资源中对于其来说所述相应资源描述符是所述安全资源描述符的存储器资源有关的读取和写入存储器事务。

3.根据权利要求1所述的装置，其中所述存储器存取控制器经配置以利用安全存储器管理单元将数据写入到所述第一存储器单元的所述安全部分，所述安全存储器管理单元利用含有所述第一存储器单元的所述安全部分的地址范围的安全页表，且

其中所述存储器存取控制器经配置以利用不安全存储器管理单元从所述第一存储器单元的所述不安全部分读取数据，所述不安全存储器管理单元利用含有用于所述第一存储器单元的所述不安全部分的地址范围的不安全页表。

4.根据权利要求3所述的设备，其中所述存储器存取控制器根据来自虚拟存储器地址范围的虚拟存储器地址来读取和写入数据，其中所述虚拟存储器地址范围包含与所述安全存储器管理单元所利用的所述安全页表中的条目有关的第一虚拟存储器地址范围，以及与所述不安全存储器管理单元所利用的所述不安全页表中的条目有关的第二虚拟存储器地址范围。

5.根据权利要求4所述的设备，其进一步包括：

第二存储器单元，其存储图形驱动程序，所述图形驱动程序经配置以将所述GPU置于安全模式或不安全模式。

6.根据权利要求5所述的设备，其进一步包括：

所述安全存储器管理单元；

所述不安全存储器管理单元；以及

中央处理单元CPU，其执行安全操作系统及所述图形驱动程序，所述安全操作系统经配置以将所述安全页表供应到所述安全存储器管理单元且将所述不安全页表供应到所述不安全存储器管理单元。

7.根据权利要求6所述的设备，其中所述GPU进一步包括清除寄存器和一或多个内部存储器，且其中所述安全操作系统经配置以将指令发送到所述清除寄存器，所述指令在所述GPU从所述安全模式转变为所述不安全模式时，致使所述GPU将至少一些内容从所述一或多个内部存储器清除并使其失效。

8.根据权利要求6所述的设备，其中所述GPU进一步包括命令流寄存器和一或多个内部存储器，且其中所述图形驱动程序经配置以将指令发送到所述命令流寄存器，所述指令在所述GPU从所述安全模式转变为所述不安全模式时，致使所述GPU将至少一些内容从所述一或多个内部存储器清除并使其失效。

9.根据权利要求1所述的设备，其中所述GPU进一步包括：

一或多个硬件块，其经配置以将数据写入到所述第一存储器的所述不安全部分，而不管所述GPU是处于所述不安全模式还是所述安全模式中，其中所述一或多个硬件块不具有对所述第一存储器单元的所述安全部分的读取存取权。

10.根据权利要求9所述的设备，其中所述一或多个硬件块包含前端命令处理器。

11.一种方法，其包括：

读取多个存储器资源的相应存储器资源的相应资源描述符；

接收对到第一存储器单元的存储器事务的请求；

当图形处理单元GPU根据安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是安全资源描述符的存储器资源有关的读取和写入存储器事务引导到所述第一存储器单元的安全部分；

当所述GPU根据所述安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是不安全资源描述符的存储器资源有关的读取存储器事务引导到所述第一存储器单元的不安全部分；以及

当所述GPU根据所述安全模式操作时，响应于所述请求，丢弃与所述多个存储器资源中对于其来说所述相应资源描述符是所述不安全资源描述符的存储器资源有关的写入存储器事务。

12.根据权利要求11所述的方法，其进一步包括：

当所述GPU根据所述不安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是不安全资源描述符的存储器资源有关的读取和写入存储器事务引导到所述第一存储器单元的不安全部分；以及

当所述GPU根据所述不安全模式操作时，响应于所述请求，丢弃与所述多个存储器资源中对于其来说所述相应资源描述符是所述安全资源描述符的存储器资源有关的读取和写入存储器事务。

13.根据权利要求11所述的方法，其进一步包括：

利用安全存储器管理单元将数据写入到所述第一存储器单元的所述安全部分，所述安全存储器管理单元利用含有用于所述第一存储器单元的所述安全部分的地址范围的安全页表；以及

利用不安全存储器管理单元从所述第一存储器单元的所述不安全部分读取数据，所述不安全存储器管理单元利用含有用于所述第一存储器单元的所述不安全部分的地址范围的不安全页表。

14.根据权利要求13所述的方法，其进一步包括：

根据来自虚拟存储器地址范围的虚拟存储器地址来读取和写入数据，其中所述虚拟存储器地址范围包含与所述安全存储器管理单元所利用的所述安全页表中的条目有关的第一虚拟存储器地址范围，以及与所述不安全存储器管理单元所利用的所述不安全页表中的条目有关的第二虚拟存储器地址范围。

15.根据权利要求14所述的方法，其进一步包括：

将所述GPU置于安全模式或不安全模式。

16.根据权利要求15所述的方法，其进一步包括：

将所述安全页表供应到所述安全存储器管理单元，且将所述不安全页表供应到所述不安全存储器管理单元。

17.根据权利要求16所述的方法，其进一步包括：

将指令发送到所述GPU的清除寄存器；以及

当所述GPU从所述安全模式转变为所述不安全模式时，响应于所述指令，将至少一些内容从所述一或多个内部存储器清除并使其无效。

18.根据权利要求16所述的方法，其进一步包括：

将指令发送到所述GPU的命令流寄存器；以及

当所述GPU从所述安全模式转变为所述不安全模式时，响应于所述指令，将至少一些内容从所述一或多个内部存储器清除并使其无效。

19.根据权利要求11所述的方法，其进一步包括：

所述GPU的一或多个硬件块将数据写入到所述第一存储器的所述不安全部分，不管所述GPU处于所述不安全模式还是所述安全模式，其中所述一或多个硬件块不具有对所述第一存储器单元的所述安全部分的读取存取权。

20.根据权利要求19所述的方法，其中所述一或多个硬件块包含前端命令处理器。

21.一种用于图形处理的设备，其包括：

用于读取多个存储器资源的相应存储器资源的相应资源描述符的装置；

用于接收对到第一存储器单元的存储器事务的请求的装置；

用于在图形处理单元GPU根据安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是安全资源描述符的存储器资源有关的读取和写入存储器事务引导到所述第一存储器单元的安全部分的装置；

用于在所述GPU根据所述安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是不安全资源描述符的存储器资源有关的读取存储器事务引导到所述第一存储器单元的不安全部分的装置；以及

用于在所述GPU根据所述安全模式操作时，响应于所述请求，丢弃与所述多个存储器资源中对于其来说所述相应资源描述符是所述不安全资源描述符的存储器资源有关的写入存储器事务的装置。

22.根据权利要求21所述的设备，其进一步包括：

用于在所述GPU根据所述不安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是不安全资源描述符的存储器资源有关的读取和写入存储器事务引导到所述第一存储器单元的不安全部分的装置；以及

用于在所述GPU根据所述不安全模式操作时，响应于所述请求，丢弃与所述多个存储器资源中对于其来说所述相应资源描述符是所述安全资源描述符的存储器资源有关的读取和写入存储器事务的装置。

23.根据权利要求21所述的设备，其进一步包括：

用于利用安全存储器管理单元将数据写入到所述第一存储器单元的所述安全部分的装置，所述安全存储器管理单元利用含有用于所述第一存储器单元的所述安全部分的地址范围的安全页表；以及

用于利用不安全存储器管理单元从所述第一存储器单元的所述不安全部分读取数据的装置，所述不安全存储器管理单元利用含有用于所述第一存储器单元的所述不安全部分的地址范围的不安全页表。

24.根据权利要求23所述的设备，其进一步包括：

用于根据来自虚拟存储器地址范围的虚拟存储器地址来读取和写入数据的装置，其中所述虚拟存储器地址范围包含与所述安全存储器管理单元所利用的所述安全页表中的条目有关的第一虚拟存储器地址范围，以及与所述不安全存储器管理单元所利用的所述不安全页表中的条目有关的第二虚拟存储器地址范围。

25.根据权利要求24所述的设备，其进一步包括：

用于将所述GPU置于安全模式或不安全模式的装置。

26.一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：

读取多个存储器资源的相应存储器资源的相应资源描述符；

接收对到第一存储器单元的存储器事务的请求；

当图形处理单元GPU根据安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是安全资源描述符的存储器资源有关的读取和写入存储器事务引导到所述第一存储器单元的安全部分；

当所述GPU根据所述安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是不安全资源描述符的存储器资源有关的读取存储器事务引导到所述第一存储器单元的不安全部分；以及

当所述GPU根据所述安全模式操作时，响应于所述请求，丢弃与所述多个存储器资源中对于其来说所述相应资源描述符是所述不安全资源描述符的存储器资源有关的写入存储器事务。

27.根据权利要求26所述的计算机可读存储媒体，其中所述指令进一步致使所述一或多个处理器：

当所述GPU根据所述不安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说所述相应资源描述符是不安全资源描述符的存储器资源有关的读取和写入存储器事务引导到所述第一存储器单元的不安全部分；以及

当所述GPU根据所述不安全模式操作时，响应于所述请求，丢弃与所述多个存储器资源

中对于其来说所述相应资源描述符是所述安全资源描述符的存储器资源有关的读取和写入存储器事务。

28.根据权利要求26所述的计算机可读存储媒体，其中所述指令进一步致使所述一或多个处理器：

利用安全存储器管理单元将数据写入到所述第一存储器单元的所述安全部分，所述安全存储器管理单元利用含有用于所述第一存储器单元的所述安全部分的地址范围的安全页表；以及

利用不安全存储器管理单元从所述第一存储器单元的所述不安全部分读取数据，所述不安全存储器管理单元利用含有用于所述第一存储器单元的所述不安全部分的地址范围的不安全页表。

29.根据权利要求28所述的计算机可读存储媒体，其中所述指令进一步致使所述一或多个处理器：

根据来自虚拟存储器地址范围的虚拟存储器地址来读取和写入数据，其中所述虚拟存储器地址范围包含与所述安全存储器管理单元所利用的所述安全页表中的条目有关的第一虚拟存储器地址范围，以及与所述不安全存储器管理单元所利用的所述不安全页表中的条目有关的第二虚拟存储器地址范围。

30.根据权利要求29所述的计算机可读存储媒体，其中所述指令进一步致使所述一或多个处理器：

将所述GPU置于安全模式或不安全模式。

用于图形处理单元的硬件强制内容保护

技术领域

[0001] 本发明涉及用于图形处理的技术，且更具体来说，涉及用于内容保护的技术。

背景技术

[0002] 现代操作系统(包含开放平台(例如，安卓或其它开放源平台)以及封闭平台(例如，微软Windows®)通常在保护安全内容方面不受信任，所述安全内容流式传输到所述开放平台或由所述开放平台处理。虽然现代操作系统经由用户内核模式分离而提供某一等级的安全性，但最终内核模式的组件在封闭平台中并且尤其在开放平台中不提供强信任等级。可容易地安装内核模式驱动程序，且恶意的内核模式驱动程序自然地绕过安全边界。此类开放平台中的内核模式硬件驱动程序用于控制可处理安全内容的硬件(例如，图形处理单元(GPU))的操作。然而，因为此类驱动程序常常是开放源，且/或未被认为关于受保护内容是“安全的”，所以它们更易受第三方更改。此类更改可导致通过由此类驱动程序控制的硬件流式传输或由所述硬件处理的受保护内容(例如，数字权利管理(DRM)内容)存储在不安全的存储器中且被复制。由此，常常难以控制开放平台上的安全内容。

发明内容

[0003] 一般来说，本发明描述用于图形处理单元(GPU)的硬件强制内容保护的技术。为了控制硬件平台上的安全内容，可通过例如GPU等硬件来控制对安全存储器的存取。

[0004] 在本发明的一个实例中，一种用于图形处理的设备包括GPU，其经配置以根据不安全模式和安全模式中的一者来存取存储器，所述GPU包括存储器存取控制器，其经配置以在所述GPU在安全模式下操作时，引导从所述GPU的至少一个硬件单元到存储器控制器中的安全上下文组的存储器事务；且经配置以在所述GPU在不安全模式下操作时，引导从所述GPU的至少一个硬件单元到所述存储器控制器中的不安全上下文组的存储器事务。

[0005] 在本发明的另一实例中，一种GPU包括：一或多个硬件单元，其经配置以根据所述GPU的不安全模式和安全模式中的一者来存取存储器；以及存储器存取控制器，其经配置以在所述GPU在安全模式下操作时，引导从所述GPU的所述一或多个硬件单元中的至少一者到存储器控制器中的安全上下文组的存储器事务；且经配置以在所述GPU在不安全模式下操作时，引导从所述GPU的一或多个硬件单元中的所述至少一者到所述存储器控制器中的不安全上下文组的存储器事务。

[0006] 在本发明的另一实例中，一种用于图形处理的方法包括：使用GPU，根据不安全模式，通过引导从所述GPU的至少一个硬件单元到存储器控制器中的不安全上下文组的存储器事务，来存取存储器的不安全部分；以及使用所述GPU，根据安全模式，通过引导从所述GPU的至少一个硬件单元到所述存储器控制器中的安全上下文组的存储器事务，来存取所述存储器的安全部分。

[0007] 在本发明的另一实例中，一种用于图形处理的设备包括：用于根据不安全模式，通过引导从GPU的至少一个硬件单元到存储器控制器中的不安全上下文组，来存取存储器的

不安全部分的装置;以及用于根据安全模式,通过引导从所述GPU的至少一个硬件单元到所述存储器控制器中的安全上下文组的存储器事务,来存取所述存储器的安全部分的装置。

[0008] 在本发明的另一实例中,一种用于图形处理的设备包括GPU,其经配置以根据不安全模式和安全模式中的一者以及与多个存储器资源中的每一者相关联的相应资源描述符来存取第一存储器单元,所述GPU包括存储器存取控制器,其经配置以读取与所述多个存储器资源中的每一者相关联的相应资源描述符,所述存储器无线控制器配置以接收对到第一存储器单元的存储器事务的请求,所述存储器无线控制器经配置以响应于所述请求,在所述GPU根据安全模式操作时,引导与所述多个存储器资源中对于其来说所述相应资源描述符是到第一存储器单元的安全部分的安全资源描述符的存储器资源有关的所有读取和写入存储器事务,所述存储器无线控制器经配置以响应于所述请求,在所述GPU根据安全模式操作时,引导与所述多个存储器资源中对于其来说所述相应资源描述符是到第一存储器单元的不安全部分的不安全资源描述符的存储器资源有关的所有读取存储器事务,且所述存储器无线控制器经配置以响应于所述请求,在所述GPU根据安全模式操作时,丢弃与所述多个存储器资源中对于其来说所述相应资源描述符是不安全资源描述符的存储器资源有关的所有写入存储器事务。

[0009] 在本发明的另一实例中,一种方法包括:读取多个存储器资源中的相应存储器资源的相应资源描述符;接收对到第一存储器单元的存储器事务的请求;响应于所述请求,在GPU根据安全模式操作时,引导与所述多个存储器资源中对于其来说所述相应资源描述符是到第一存储器单元的安全部分的安全资源描述符的存储器资源有关的读取和写入存储器事务;响应于所述请求,在所述GPU根据安全模式操作时,引导与所述多个存储器资源中对于其来说所述相应资源描述符是到所述第一存储器单元的不安全部分的不安全资源描述符的存储器资源有关的读取存储器事务;以及响应于所述请求,当所述GPU根据安全模式操作时,丢弃与所述多个存储器资源中对于其来说所述相应资源描述符是不安全资源描述符的存储器资源有关的写入存储器事务。

[0010] 在本发明的另一实例中,一种用于图形处理的设备包括:用于读取多个存储器资源中的相应存储器资源的相应资源描述符的装置;用于接收对到第一存储器单元的存储器事务的请求的装置;用于响应于所述请求,当GPU根据安全模式操作时,引导与所述多个存储器资源中对于其来说相应资源描述符是到第一存储器单元的安全部分的安全资源描述符的存储器资源有关的读取和写入存储器事务的装置;用于响应于所述请求,当所述GPU根据安全模式操作时,引导与所述多个存储器资源中对于其来说相应资源描述符是到第一存储器单元的不安全部分的不安全资源描述符的存储器资源有关的读取存储器事务的装置;以及用于响应于所述请求,当所述GPU根据安全模式操作时,丢弃与所述多个存储器资源中对于其来说相应资源描述符是不安全资源描述符的存储器资源有关的写入存储器事务的装置。

[0011] 在另一实例中,本发明描述一种计算机可读存储媒体,其存储指令,所述指令在被执行时,致使一或多个处理器:读取对多个存储器资源中的相应存储器资源的相应资源描述符;接收对到第一存储器单元的存储器事务的请求;响应于所述请求,当GPU根据安全模式操作时,引导与所述多个存储器资源中对于其来说相应资源描述符是到第一存储器单元的安全部分的安全资源描述符的存储器资源有关的读取和写入存储器事务;响应于所述请

求，当所述GPU根据安全模式操作时，引导与所述多个存储器资源中对于其来说相应资源描述符是到第一存储器单元的不安全部分的不安全资源描述符的存储器资源有关的读取存储器事务；以及响应于所述请求，当所述GPU根据安全模式操作时，丢弃与所述多个存储器资源中对于其来说相应资源描述符是不安全资源描述符的存储器资源有关的写入存储器事务。

[0012] 随附和以下描述中陈述一或多个实例的细节。其它特征、目标和优点将从所述描述和图式以及从所附权利要求书而显而易见。

附图说明

[0013] 图1是示出经配置以使用本发明的技术的实例计算装置的框图。

[0014] 图2是说明图1的系统存储器的实例物理页的概念图。

[0015] 图3是示出经配置以使用本发明的技术的实例处理单元的框图。

[0016] 图4是示出经配置以执行本发明的硬件强制内容保护技术的实例结构的框图。

[0017] 图5是展示经配置以实施本发明的硬件强制内容保护技术的另一实例结构的框图。

[0018] 图6是示出经配置以实施本发明的硬件强制内容保护技术的另一实例结构的框图。

[0019] 图7A是示出经配置以执行本发明的硬件强制内容保护技术的另一实例结构的框图。

[0020] 图7B是示出经配置以执行本发明的硬件强制内容保护技术的另一实例结构的框图。

[0021] 图8是示出根据本发明的一个实例的高速缓冲存储器清除技术的框图。

[0022] 图9是示出根据本发明的另一实例的高速缓冲存储器清除技术的框图。

[0023] 图10是说明根据本发明的一个实例的方法的流程图。

[0024] 图11是说明根据本发明的另一实例的另一实例方法的流程图。

具体实施方式

[0025] 本发明涉及用于图形处理的技术，且更具体来说，涉及用于图形处理单元(GPU)的硬件强制内容保护的技术。

[0026] 现代操作系统(包含开放平台(例如，安卓或其它开放源平台)以及封闭平台(例如，微软Windows®)通常在保护安全内容方面不受信任，所述安全内容流式传输到所述开放平台或由所述开放平台处理。虽然现代操作系统经由用户内核模式分离而提供某一等级的安全性，但最终内核模式的组件在封闭平台中并且尤其在开放平台中不提供强信任等级。可容易地安装内核模式驱动程序，且恶意的内核模式驱动程序自然地绕过安全边界。此类开放平台中的内核模式硬件驱动程序用于控制可处理安全内容的硬件(例如，图形处理单元(GPU))的操作。然而，因为此类驱动程序常常是开放源，且/或未被认为关于受保护内容是“安全的”，所以它们更易受第三方更改。此类更改可导致通过由此类驱动程序控制的硬件流式传输或由所述硬件处理的受保护内容(例如，数字权利管理(DRM)内容)存储在不安全的存储器中且被复制。由此，常常难以控制开放平台上的安全内容。为解决此问题，本

发明提出一种方法和设备，借此通过硬件自身(例如，通过GPU)来控制对安全存储器的存取。

[0027] 不是直接通过驱动程序代码控制硬件对安全或不安全存储器的存取，在一个实例中，本发明提出使用图形驱动程序(例如，开放源不安全驱动程序)以仅将GPU放置在安全模式或不安全模式中。一次置于安全模式下，所述GPU组件就可经配置以使得可基于所述GPU的模式(即，安全或不安全模式)来限制所述GPU对安全和不安全存储器的读取和/或写入存取。举例来说，在安全模式下，可配置某些GPU组件，使得它们限于仅进行到安全存储器区中的写入。这防止不受信任的驱动程序使用GPU来将存储器内容从安全存储器区复制到不安全存储器区。下文将更详细地论述用于在安全模式下限制GPU对安全存储器的存取，将所述GPU置于不安全模式或安全模式中的一者下，且使某些数据资源与安全存储器或不安全存储器关联的其它技术。

[0028] 在本发明的一个实例中，在此安全模式下，所述GPU可经配置以读取安全(例如复制受保护(CP))内容以及不安全内容(例如存储在不安全存储器中的内容)两者。在不安全模式下，所述GPU可经配置以使得拒绝GPU组件对安全存储器的所有存取。以此方式，即使更改不安全驱动程序以将GPU置于不安全模式下，也将防止GPU自身从安全存储器读取任何数据。由此，防止对安全存储器中的安全内容的存取。

[0029] 图1是说明可用于实施本发明的用于GPU的硬件强制内容保护的技术的实例计算装置2的框图。计算装置2可包括例如个人计算机、桌上型计算机、膝上型计算机、平板计算机、计算机工作站、视频游戏平台或控制台、移动电话(例如蜂窝式或卫星电话)、陆线电话、因特网电话、所谓的智能电话、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、电视机、电视机顶盒、服务器、中间网络装置、主机计算机，或处理和/或显示图形数据的任何其它类型的装置。

[0030] 如图1的实例中所说明，计算装置2可包含用户输入接口4、中央处理单元(CPU)6、一或多个存储器控制器8、系统存储器10、图形处理单元(GPU)12、图形存储器14、显示接口16、显示器18以及总线20和22。注意，在一些实例中，图形存储器14可与GPU 12“在芯片上”。在一些情况下，例如在芯片上系统(SoC)设计中，图1中所示的所有硬件元件可在芯片上。用户输入接口4、CPU 6、存储器控制器8、GPU 12和显示接口16可使用总线20彼此通信。存储器控制器8和系统存储器10也可使用总线22彼此通信。总线20、22可为多种总线结构中的任一者，例如第三代总线(例如，超传输(HyperTransport)总线或无限带宽(InfiniBand)总线)、第二代总线(例如，高级图形端口总线、周边组件互连(PCI)高速总线，或高级eXentisible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于实施本发明的技术。

[0031] 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、文字处理器应用程序、电子邮件应用程序、电子数据表应用程序、媒体播放器应用程序、视频游戏应用程序、图形用户接口应用程序或另一程序。另外，CPU6可执行用于控制GPU 12的操作的GPU驱动程序7。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、触摸屏，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。

[0032] 在CPU 6上执行的软件应用程序可包含指令CPU 6以致使向显示器18渲染图形数据的一或多个图形再现指令。在一些实例中，软件指令可符合应用程序编程接口(API)，例如开放图形库(OpenGL®)API、开放图形库嵌入系统(OpenGL ES)API、开放式计算语言(OpenCL®)API、Direct3D API、X3D API、RenderMan API、WebGL API或任何其它公用或专有标准图形API。为了处理图形渲染指令，CPU 6可将一或多个图形渲染命令发布到GPU 12(例如，通过GPU驱动程序7)，以致使GPU 12执行图形数据的渲染中的一些或全部。在一些实例中，待渲染的图形数据可包含例如点、线、三角形、四边形、三角形条带等图形基元的列表。

[0033] 存储器控制器8促进数据进出系统存储器10的传送。举例来说，存储器控制器8可接收存储器读取和写入命令，且相对于系统存储器10服务此类命令，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经由存储器总线22以通信方式耦合到系统存储器10。尽管在图1中将存储器控制器8说明为与CPU 6和系统存储器10两者分开的处理模块，但在其它实例中，存储器控制器8的一些或全部功能性可在CPU 6、GPU 12和系统存储器10中的一者或任一者上实施。系统存储器10可包括一个或若干存储器单元。可物理上划分存储器单元(例如，单独的物理磁盘或固态存储器单元)或可通过存储器地址范围划分存储器单元。明确地说，可将系统存储器10划分成由“安全”的存储器单元及“不安全”的存储器单元组成的两个或更多个存储器单元。在一些实例中，安全存储器单元可利用加密和/或其它数字权限管理(DRM)技术来防止对存储在其上的数据的存取、复制或解密。

[0034] 存储器控制器8还可包含一或多个存储器管理单元(MMU)，其包含用于控制IO装置对系统存储器10的存取(例如，GPU)的IOMMU(即，输入/输出MMU)。存储器管理单元可实施虚拟存储器系统。可将虚拟存储器空间划分成多个虚拟页。这些虚拟页可为邻接的，但这些虚拟页所对应的系统存储器10中的物理页可在系统存储器10中不是邻接的。可将页视为MMU可能够管理的最小单元。

[0035] 在中央处理单元(CPU)上运行的现代操作系统(OS)通常使用用于将存储器分配给在CPU上操作的多个程序的虚拟存储器方案。虚拟存储器是一种存储器管理技术，其将计算机系统的物理存储器(例如，RAM、磁盘存储装置等)虚拟化，使得应用仅需要涉及一组存储器(即，虚拟存储器)。虚拟存储器由映射到物理存储器中的多个位置的邻接地址空间组成。以此方式，物理存储器的分段“隐藏”起来而不让应用程序看见，所述应用程序可改为与虚拟存储器的邻接块事务。虚拟存储器中的邻接块通常布置成“页”。每一页是虚拟存储器地址的某一固定长度的邻接块。从虚拟存储器到物理存储器的映射常常由存储器管理单元(MMU)处置。当前映射到物理存储器中的位置的虚拟存储器空间被视为“背对”物理存储器。

[0036] 虚拟存储器空间中的位置到物理存储器的映射与翻译旁视缓冲器(TLB)存储在一起。TLB由MMU使用以将虚拟地址快速翻译为物理地址。可将TLB实施为将虚拟存储器地址用作输入且输出物理存储器地址的内容可寻址存储器(CAM)。MMU可随后使用输出的物理存储器地址来快速检索所请求的数据。

[0037] 图2是说明系统存储器10的实例物理页的概念图。举例来说，图2说明包含虚拟页42的IOMMU 40，所述虚拟页包含四个区段(区段0到3)。应理解，虚拟页42是图2中为了易于理解而说明的虚拟构造。在图2中，系统存储器10可包含对应于虚拟页42的物理页44。

[0038] 物理页44可跨系统存储器10的多个存储器单元存储。举例来说，物理页44可包含存储器单元11A和存储器单元11N两者。在一个实例中，存储器单元11A是“安全”存储器单元，且存储器单元11N是“不安全”存储器单元。存储器单元11A可存储物理页44的一部分(指示为部分44A)，且存储器单元11N可存储物理页44的一部分(指示为部分44B)。如所说明，存储器单元11A存储物理页44的区段0和区段2，且存储器单元11N存储物理页44的区段1和区段3。

[0039] 出于说明的目的，图2的实例仅包含两个存储器单元，但可使用任何数目的存储器单元。举例来说，返回参看图1，GPU驱动程序7可发射致使GPU 12存储像素值或任何其它所计算的值的指令，且可发射将在其处存储像素值的虚拟地址。GPU 12继而可请求IOMMU 40根据虚拟地址来存储像素值。IOMMU 40继而可将虚拟地址映射到物理地址，且基于所述物理地址以交错方式将像素值存储在系统存储器10的页中。

[0040] 返回到图1，系统存储器10可存储程序模块和/或指令，其可存取以供CPU 6和/或数据执行，从而供在CPU 6上执行的程序使用。举例来说，系统存储器10可存储窗口管理器应用程序，其由CPU 6使用以在显示器18上呈现图形用户接口(GUI)。另外，系统存储器10可存储用户应用程序以及与所述应用程序相关联的应用程序表面数据。系统存储器10可另外存储由计算装置2的其它组件使用和/或产生的信息。举例来说，系统存储器10可充当GPU 12的装置存储器，且可存储将由GPU 12对其进行操作的数据，以及因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DRM受保护的游戏内容或由GPU 12产生的经解码视频。在此情况下，此类DRM受保护内容优选存储在系统存储器10的安全存储器单元中。作为其它实例，系统存储器10可存储其它图形数据，例如纹理缓冲器、深度缓冲器、模板缓冲器、顶点缓冲器、帧缓冲器或类似者的任何组合。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。

[0041] GPU 12可经配置以执行图形操作以向显示器18呈现一或多个图形图元。因此，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可向GPU 12提供图形命令和图形数据以用于向显示器18渲染。图形数据可包含(例如)绘图命令、状态信息、图元信息、纹理信息等。在一些实例中，GPU 12可内置有高度并行结构，所述高度并行结构提供比CPU 6更高效的复杂图形相关操作的处理。举例来说，GPU 12可包含经配置以便以并行方式对多个顶点或像素操作的多个处理元件。在一些实例中，GPU 12的高度并行本质允许GPU 12比使用CPU 6直接将场景绘图到显示器18更快速地将图形图像(例如，GUI和二维(2D)和/或三维(3D)图形场景)绘图到显示器18上。

[0042] 在一些实例中，可将GPU 12集成到计算装置2的主板中。在其它实例中，GPU 12可存在于图形卡上，所述图形卡安装在计算装置2的主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事务操作的外围装置内。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。

[0043] GPU 12可直接耦合到图形存储器14。因此，GPU 12可在无需使用总线20的情况下从图形存储器14读取数据且将数据写入到图形存储器14。换句话说，GPU 12可使用本地存

储装置，而不是使用其它较慢的系统存储器来本地处理数据。这允许GPU 12通过消除GPU 12经由系统总线20读取和写入数据的需要而以更高效的方式操作，其中经由系统总线操作可经历繁重的总线业务。然而，在一些实例中，GPU 12可不包含单独存储器，而是经由总线20利用系统存储器10。图形存储器14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。

[0044] CPU 6和/或GPU 12可将经渲染的图像数据存储在帧缓冲器15中。通常，帧缓冲器15将被分配在系统存储器10内，但在一些情况下可为独立的存储器。显示接口16可从帧缓冲器15检索数据，且配置显示器18以显示经渲染的图像数据所表示的图像。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以供处理。显示器18可包含监视器、电视机、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(例如有机LED(OLED)显示器)、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耦合到计算装置2的独立装置。举例来说，显示器18可为经由缆线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。

[0045] 图3是进一步详细地说明图1的CPU 6、GPU 12和系统存储器10的实例实施方案的框图。CPU 6可包含至少一个软件应用程序24、图形API 26和GPU驱动程序7，其中的每一者可为在CPU 6上执行的一或多个软体应用程式或服务。GPU 12可包含3D图形处理管线30，所述3D图形处理管线包含一起操作以执行图形处理命令的多个图形处理级。GPU 12可经配置而以多种渲染模式执行图形处理管线30，包含分格渲染模式(也被称为基于瓦片的或延迟渲染模式)以及直接渲染模式。GPU 12也可为可操作以执行通用着色器39的，以用于执行适用于由GPU硬件的高度并行性质执行的更一般的计算。此类通用应用可为所谓的通用图形处理单元(GPGPU)，且可符合通用API，例如OpenCL。

[0046] 如图3中所示，图形处理管线30可包含命令引擎32、几何处理级34、光栅化级36和像素处理管线38。图形处理管线30中的组件中的每一者可实施为固定功能组件、可编程组件(例如作为在可编程着色器单元上执行的着色器程序的一部分)，或作为固定功能和可编程组件的组合。可用于CPU 6和GPU 12的存储器可包含系统存储器10，所述系统存储器本身可包含帧缓冲器15。帧缓冲器15可存储经渲染的图像数据。

[0047] 软件应用程序24可为利用GPU 12的功能性的任何应用程序。举例来说，软件应用24可为GUI应用程序、操作系统、便携式制图应用程序、用于工程或艺术应用的计算机辅助设计程序、视频游戏应用程序，或使用2D或3D图形的另一种类型的软件应用程序。软件应用程序24也可为使用GPU来执行更一般的计算(例如GPGPU应用程序中)的应用程序。

[0048] 软件应用程序24可包含指令GPU 12渲染图形用户接口(GUI)和/或图形场景的一或多个绘图指令。举例来说，绘图指令可包含界定将由GPU 12渲染的一或多个图形基元的集合的指令。在一些实例中，绘图指令可共同地界定用于GUI中的多个开窗表面的全部或部分。在额外实例中，绘图指令可共同地界定图形场景的全部或部分，所述图形场景包含在应用程序所界定的模型空间或世界空间内的一或多个图形对象。

[0049] 软件应用程序24可经由图形API 26调用GPU驱动程序7，以将一或多个命令发布到GPU 12，以用于将一或多个图形基元渲染到可显示的图形图像中。举例来说，软件应用程序24可经由图形API 26调用GPU驱动程序7以将图元定义提供给GPU 12。在一些实例中，可将图元定义以绘图图元(例如，三角形、矩形、三角扇、三角带等)列表的形式提供给GPU 12。图元定义可包含指定与待渲染的图元相关的一或多个顶点的顶点规格。所述顶点规格可包含每一顶点的位置坐标，且在一些实例中，包含与顶点相关联的其它属性，例如色彩坐标、法向量和纹理坐标。所述图元定义还可包含图元类型信息(例如，三角形、矩形、三角扇、三角带等)、按比例缩放信息、旋转信息及类似者。基于由软件应用程序24发布到GPU驱动程序7的指令，GPU驱动程序7可制定指定供GPU 12执行的一或多个操作以便渲染所述图元的一或多个命令。当GPU 12接收到来自CPU 6的命令时，图形处理管线30解码所述命令，且配置图形处理管线30内的一或多个处理元件，以执行所述命令中所指定的操作。在执行指定操作之后，图形处理管线30将经渲染的数据输出到与显示装置相关联的帧缓冲器15。图形管线30可经配置以在多种不同的渲染模式中的一者下执行，包含分格渲染模式和直接渲染模式。

[0050] GPU驱动程序7可进一步经配置以编译一或多个着色器程序，且将经编译的着色器程序下载到GPU 12内所包含的一或多个可编程着色器单元上。可以高级着色语言编写着色器程序，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。经编译的着色器程序可包含控制GPU 12内的可编程着色器单元的操作的一或多个指令。举例来说，着色器程序可包含顶点着色器程序和/或像素着色器程序。顶点着色器程序可控制可编程顶点着色器单元或统一着色器单元的执行，且包含指定一或多个逐顶点操作的指令。像素着色器程序可包含控制可编程像素着色器单元或统一着色器单元的执行的像素着色器程序，且包含指定一或多个每像素操作的指令。根据本发明的一些实例，像素着色器程序还可包含选择性地致使基于源像素的对应目的地阿尔法值而检索源像素的纹理值的指令。

[0051] 图形处理管线30可经配置以经由图形驱动程序7从CPU 6接收一或多个图形处理命令，且执行图形处理命令以产生可显示的图形图像。如上文所论述，图形处理管线30包含一起操作以执行图形处理命令的多个级。然而，应注意，此类级不需要一定在单独硬件块中实施。举例来说，几何处理级34和像素处理管线38的若干部分可实施为统一着色器单元的部分。并且，图形处理管线30可经配置以在多种不同的渲染模式中的一者下执行，包含分格渲染模式和直接渲染模式。

[0052] 命令引擎32可接收图形处理命令，且配置图形处理管线30内的其余处理级以执行用于进行图形处理命令的各种操作。图形处理命令可包含(例如)绘图命令和图形状态命令。所述绘图命令可包含顶点规格命令，所述顶点规格命令指定一或多个顶点的位置坐标，并且在一些情况下，指定与所述顶点中的每一者相关联的其它属性值，例如色彩坐标、法向量、纹理坐标和雾坐标。图形状态命令可包含基元类型命令、变换命令、照明命令等。基元类型命令可指定待渲染基元的类型和/或顶点如何组合以形成基元。所述变换命令可指定对顶点执行的变换的类型。所述照明命令可指定图形场景内的不同光的类型、方向和/或布局。命令引擎32可致使几何处理级34相对于与一或多个所接收命令相关联的顶点和/或基元执行几何处理。

[0053] 几何处理级34可对一或多个顶点执行每顶点操作和/或图元设置操作，以便产生用于光栅化级36的图元数据。每一顶点可与一组属性(例如，位置坐标、色彩值、法向量和纹理坐标)相关联。几何处理级34根据各种每顶点操作来修改这些属性中的一或多者。举例来说，几何处理级34可对顶点位置坐标执行一或多个变换以产生经修改的顶点位置坐标。几何处理级34可(例如)向顶点位置坐标应用模型化变换、检视变换、投影变换、模型视图(ModelView)变换、模型视图投影(ModelViewProjection)变换、视口变换和深度范围按比例缩放变换中的一或多者，以产生经修改的顶点位置坐标。在一些情况下，顶点位置坐标可为模型空间坐标，且经修改的顶点位置坐标可为屏幕空间坐标。在应用模型化、检视、投影和视口变换之后，可获得屏幕空间坐标。在一些情况下，几何处理级34还可对顶点执行每顶点照明操作，以产生顶点的经修改的色彩坐标。几何处理级34还可执行其它操作，包含例如正常变换、正常归一化操作、视图体积剪辑、同类划分和/或隐面消除操作。

[0054] 几何处理级34可产生包含限定将被光栅化的基元的一或多个经修改的顶点的集合的基元数据，以及指定顶点如何组合以形成基元的数据。所述经修改的顶点中的每一者可包含(例如)与所述顶点相关联的经修改的顶点位置坐标和经处理的顶点属性值。基元数据可共同地对应于将由图形处理管线30的其它级光栅化的基元。在概念上，每一顶点可对应于基元的其中所述基元的两个边缘会合的拐角。几何处理级34可将基元数据提供到光栅化级36以用于进一步处理。

[0055] 在一些实例中，几何处理级34的全部或部分可由在一或多个着色器单元上执行的一或多个着色器程序实施。举例来说，在此些实例中，几何处理级34可由顶点着色器、几何着色器或其任何组合实施。在其它实例中，可将几何处理级34实施为固定功能硬件处理管线，或实施为固定功能硬件与在一或多个着色器单元上执行的一或多个着色器程序的组合。

[0056] 光栅化级36经配置以从几何处理级34接收表示待光栅化基元的基元数据，且光栅化所述基元以产生对应于经光栅化基元的多个源像素。在一些实例中，光栅化级36可确定哪些屏幕像素位置由将被光栅化的基元覆盖，且产生用于被确定为由基元覆盖的每一屏幕像素位置的源像素。光栅化级36可通过使用所属领域的技术人员已知的技术，例如边缘行走(edge-walking)技术、评估边缘等式，等等，来确定哪些屏幕像素位置被基元覆盖。光栅化级36可将所得源像素提供到像素处理管线38以供进一步处理。

[0057] 由光栅化级36产生的源像素可对应于屏幕像素位置，例如目的地像素，且与一或多个色彩属性相关联。为特定经光栅化的基元产生的所有源像素可被称为与所述经光栅化的基元相关联。由光栅化级36确定的将由基元覆盖的像素可在概念上包含表示基元的顶点的像素、表示基元的边缘的像素以及表示基元的内部的像素。

[0058] 像素处理管线38经配置以接收与经光栅化基元相关联的源像素，并对源像素执行一或多个每像素操作。可由像素处理管线38执行的每像素操作包含(例如)阿尔法测试、纹理映射、色彩计算、像素着色、逐像素光照、雾处理、混合、像素所有权文本、源阿尔法测试、模板测试、深度测试、剪刀测试和/或点刻操作。另外，像素处理管线38可执行一或多个像素着色器程序以实施一或多个每像素操作。由像素处理管线38产生的所得数据可在本文中被称作目的地像素数据，且存储在帧缓冲器15中。目的地像素数据可与帧缓冲器15中的具有与经处理源像素相同的显示位置的目的地像素相关联。所述目的地像素数据可包含例如色

彩值、目的地阿尔法值、深度值等数据。

[0059] 帧缓冲器15存储用于GPU 12的目的地像素。每一目的地像素可与唯一的屏幕像素位置相关联。在一些实例中，帧缓冲器15可存储每个目的地像素的色彩分量和目的地阿尔法值。举例来说，帧缓冲器15可存储用于每一像素的红色、绿色、蓝色、阿尔法(RGBA)分量，其中“RGB”分量对应于色彩值，且“A”分量对应于目的地阿尔法值。像素值还可由亮度分量(Y)和一或多个色度分量(例如U和V)表示。尽管将帧缓冲器15和系统存储器10说明为单独的存储器单元，但在其它实例中，帧缓冲器15可为系统存储器10的部分。

[0060] 通用着色器39可为可在GPU 12上执行以实施计算的任何应用程序。通常，此类计算是利用GPU处理核心(包含算术逻辑单元(ALU)的高度并行结构的类型。实例通用着色器39可符合OpenCL API。OpenCL是允许应用程序跨越异构系统(例如，包含CPU、GPU、DSP等的系统)中的多个处理器进行存取的API。通常，在符合应用的OpenCL中，GPU 12将用于执行非图形计算。非图形计算应用的实例可包含基于物理学的仿真、快速傅立叶变换、音频信号处理、数字图像处理、视频处理、图像后滤波、计算相机、气候研究、天气预报、神经网络、密码学以及大规模并行数据处理，以及许多其它应用。

[0061] 图4是示出经配置以实施本发明的硬件强制内容保护技术的实例装置的框图。在图4的实例中，GPU 12可经配置以根据安全模式或不安全模式来操作。在本发明的一个实例中，在安全模式下，GPU 12受限制而无法将输出数据(例如比赛数据、视频等)写入到不安全存储器56。实际上，在安全模式下，GPU 12可仅将输出数据写入到安全存储器57。当处于安全模式时，GPU 12可从安全存储器57或不安全存储器56读取数据。在不安全模式下，对于此实例，GPU 12受限而无法从安全存储器57读取任何数据。实际上，在不安全模式下，GPU 12可仅将从不安全存储器56读取数据。同样地，当处于不安全模式时，GPU 12可仅将数据写入到不安全存储器56。

[0062] 不安全存储器56和安全存储器57可为任何类型的存储器，包含一或多个易失性或非易失性存储器或存储装置。实例存储器和存储装置包含RAM、SRAM、DRAM、ROM、EPROM、EEPROM、快闪存储器、磁性数据媒体或光学存储媒体。安全存储器57包含不安全存储器56中找不到的额外特征。举例来说，安全存储器57可利用加密、认证和/或其它数字权限管理技术，以防止对存储在其上的数据的存取、复制或解密。一般来说，安全存储器57可被认为是系统存储器10的一部分，且不安全存储器56可被认为是系统存储器10的另一部分。

[0063] 根据下文描述的本发明的一或多个实例，GPU 12可经配置以控制或以其它方式影响使用存储器存取控制器53从何处读取数据和将数据写入到何处。存储器存取控制器53响应GPU 12在其下操作的模式(即，安全模式或不安全模式)，且基于所述模式做出读取/写入决策。一般来说，存储器存取控制器53可经配置以对GPU 12与存储器控制器50之间的事务的性质强加限制。存储器存取控制器53可经配置以响应GPU 12当前正在其下操作的模式(即，安全模式或不安全模式)，且可根据下文的本发明的实例来对存储器事务强加限制。

[0064] 在本发明的一个实例中，GPU存储器模式(例如安全模式或不安全模式)由在CPU 6上操作的GPU驱动程序7设定。GPU驱动程序7可以若干不同方式来改变GPU 12中的存储器模式。在一个实例中，GPU驱动程序7可将值直接写入到GPU 12中的寄存器，所述值向GPU 12指示将使用哪一存储器模式(例如安全模式或不安全模式)。在另一实例中，GPU 12可在命令流中包含可由GPU 12执行的一或多个指令，其指令GPU 12自身将指示要使用哪一存储器模

式的某一值写入到寄存器。以此方式，GPU驱动程序7可仅选择GPU在其下操作的存储器模式，且不做出指定哪些数据将被写入到哪些存储器的任何直接指令。由此，即使更改GPU驱动程序7以将GPU 12置于不安全模式中，通过存储器存取控制器53的功能，GPU 12也将防止从安全存储器57进行任何读取存取，因为在不安全模式下，存储器存取控制器53仅能够从不安全存储器56读取。同样，即使更改GPU 7以将GPU 12置于安全模式中，通过存储器存取控制器53的功能，GPU 12也将防止对不安全存储器56的任何写入存取，因为在安全模式下，存储器存取控制器53仅能够写入到安全存储器57。由此，本发明的技术仍可防止将数据复制到不安全存储器56，即使更改GPU驱动程序7以将GPU 12置于安全模式也是如此。

[0065] 在本发明的一个实例中，存储器存取控制器53经配置以分别经由安全和不安全存储器管理单元(MMU)页表来存取安全存储器57和不安全存储器56。在此实例中，由GPU驱动程序7将虚拟地址范围提供到GPU 12。所述虚拟地址范围包含用于安全存储器的虚拟地址范围以及用于不安全存储器的虚拟地址范围。在由GPU驱动程序7置于安全模式中时，GPU 12利用用于安全存储器的虚拟地址范围来执行读取和写入。GPU 12还将能够将虚拟地址范围用于不安全存储器，以在安全模式下执行读取，但不执行写入，从而防止受保护数据从安全存储器的未经授权的复制。在由GPU驱动程序7置于不安全模式中时，GPU 12将利用用于不安全存储器的虚拟地址范围来执行读取和写入。

[0066] 在一个实例中，存储器存取控制器53通过确定用于读取或写入请求中的虚拟地址是在不安全虚拟存储器地址范围内还是在安全虚拟地址范围内，来将读取和写入投送到适当的存储器单元(即，安全存储器57或不安全存储器56)。基于范围确定，存储器存取控制器利用存储器控制器50中的不安全IOMMU 51或安全IOMMU 52中的一者。存储器控制器50经配置以促进进入系统存储器10和从系统存储器10出来的数据的传送。为了有效地处置任何此类事务，存储器控制器50可包含用于控制例如GPU 12对系统存储器10的装置存取的一或多个MMU。不安全IOMMU 51和安全IOMMU52含有用于虚拟化存储地址的映射，其为其客户端提供页的连续视图。在此实例中，客户端可为结合一或多个资源或将其提供给GPU 12(例如GPU 12执行的应用程序或在CPU 6上执行的应用程序)的任何实体。资源是供GPU 12以某一方式使用的信息的容器(例如存储器或缓冲器)。在一些实例中，资源可具有描述符，其提供关于将如何使用存储器的信息。

[0067] 在本发明的一个实例中，不安全IOMMU 51是经配置以将虚拟存储器地址映射到不安全存储器56中的物理存储器地址的IOMMU。安全IOMMU 52是经配置以将虚拟存储器地址映射到安全存储器57中的物理存储器地址的IOMMU。不安全IOMMU 51使用不安全页表来执行到不安全存储器56的映射。不安全页表是将虚拟存储器地址范围(例如GPU驱动程序7所提供的范围)映射到不安全存储器56中的位置的页表。同样地，安全IOMMU 52使用安全页表来执行到安全存储器57的映射。安全页表是将虚拟存储器地址范围(例如GPU驱动程序7所提供的范围)映射到安全存储器57中的位置的页表。如图4中所描绘，不安全IOMMU 51和安全IOMMU 52是单个存储器控制器50的一部分。存储器控制器50可为图1中所描绘的存储器控制器8中的一者。实际上，当结合安全页表操作时，存储器控制器50变为安全IOMMU，且当结合不安全页表操作时，变为不安全IOMMU。在其它实例中，不安全IOMMU 51和安全IOMMU 52可为物理上分开的MMU。

[0068] 在本发明的一个实例中，通过在CPU 6上执行的安全操作系统(OS)54，将安全和不

安全页表两者提供到安全IOMMU 52和不安全IOMMU 51。安全OS是在正常“丰富”OS(例如苹果iOS、谷歌安卓、微软视窗等)旁边操作的OS。安全OS提供安全应用程序以保护安全内核以及任何安全外围设备(例如,安全IOMMU 52),且将其与在丰富OS上运行的任何代码(例如,GPU驱动程序7)分离。安全OS的实例是由ARM Holdings制作的TrustZone软件。一般来说,安全OS被视为比在丰富OS上运行的软件(包含例如图形驱动程序等软件)不易受到更改和攻击得多。根据本发明的技术,仅允许安全OS更新用于将虚拟存储器地址范围映射到物理存储器地址的页表。由此,更改图形驱动程序(包含由驱动程序提供的虚拟地址范围)的任何尝试将不导致安全内容存储在不安全存储器中,因为仅安全OS提供到安全和不安全存储器的最终映射。

[0069] 在安全和不安全页表两者在存储器控制器50(例如存储器控制器50包含不安全IOMMU 51和安全IOMMU 52两者)处可用的实例中,GPU 12能够在安全模式下,从不安全存储器56和安全存储器57两者读取数据。其它读取/写入限制仍适用。就是说,在安全模式下,GPU 12仅作出对安全存储器57的写入,且在不安全模式中,GPU 12的读取和写入两者限于不安全存储器56。

[0070] 在本发明的另一实例中,不是使安全及不安全IOMMU两者都可用于GPU,其中数据业务经由存储器存取控制器53被引导到安全或不安全IOMMU,将使仅一个IOMMU(即,不安全IOMMU 51或安全IOMMU 52)可用于GPU 12,其取决于选定的存储器模式。就是说,如果存储器模式为不安全模式,那么安全OS 54仅提供用于不安全IOMMU 51的页表映射。在此情况下,安全IOMMU 52将不可用。如果存储器模式是安全模式,那么安全OS 54仅提供用于安全IOMMU 52的页表映射。在此情况下,不安全IOMMU 51将不可用。每一存储器模式仅使一个IOMMU可用的此实例将提供更简单的实施方案,其中每一存储器模式限制读取和写入两者。就是说,仅在安全模式下,GPU12将允许对安全存储器57的读取和写入,而仅在不安全模式下,GPU 12将允许对不安全存储器56的读取和写入。这略微不同于上文描述的其中两个IOMMU可为都可用的方法,不同之处在于安全模式将不再允许对不安全存储器56的读取。

[0071] 甚至在安全模式中时,存在除GPU 12的最终输出产品之外的某一写入,其将更好地使GPU写入到不安全存储器。这些写入包含GPU 12与图形驱动程序7之间的通信信标。此类数据包含时戳以及其它辅助数据和控制数据,例如计数器数据和查询数据。GPU12使用存储器(例如,不安全存储器56)来将此类时戳和数据传送回到驱动程序。由于图形驱动程序7不受信任,所以参与通信路径的存储器需要是不安全的(例如,不安全存储器56)。作为一个实例,在GPU 12到达处理中的某一点时,GPU 12将时戳/循序标记写入到存储器。图形驱动程序7使用此信息来确定GPU已在特定命令流中进行多远。此确定(例如)允许图形驱动程序7在GPU 12结束时释放GPU 12对其操作的存储器对象。存在许多其它类型的信令和通信路径,GPU 12可用来将信息提供到图形驱动程序7。作为另一实例,在绘图调用之后,图形驱动程序7可请求GPU 12报告性能计数器。GPU 12接着将这些性能计数器写入到由图形驱动程序7指定的存储器位置(例如,不安全存储器56中)。

[0072] 为了解决以上一般规则的此例外,GPU 12在安全模式下不写入到不安全存储器,GPU 12的硬件可经修改以使得某些硬件块经配置以具有不安全存储器存取,而在GPU在安全模式下运行时,也不具有对连接到或含有安全内容的数据路径和高速缓冲存储器的存取。

[0073] 图5描绘实例实施方案，其中GPU 12的某些硬件块仅具有通过GPU 12的存储器接口块(VBIF 60)，且接着通过不安全IOMMU 51对不安全存储器的直接存取，甚至在GPU 12处于安全模式中时也如此。此硬件块的一个实例是GPU的前端处的命令处理器(CP)62块。CP 62可执行命令引擎，例如图3中所示的命令引擎32。CP 62负责将消息(经由不安全存储器)发送回到GPU驱动器7。如图5中所示，CP 62经配置以仅具有通过不安全IOMMU 51到存储器(在此情况下，不安全存储器)的一个物理路径。由此，不管GPU 12的任何其它硬件块是否对安全内容操作，CP 62可从不获得对此类安全内容的存取权。为了进一步确保CP 62不具有对安全内容的存取权，CP 62还可在物理上与可用于存储安全内容的任何寄存器(包含调试总线)隔离(例如，不具有连接)。如图5中所示，CP 62不具有对L2高速缓冲存储器61和图形存储器(GMEM)70的直接存取权。GMEM 70是当渲染内容以供在GPU 12的某一操作模式下显示时，GPU 12用作渲染目标或帧缓冲器的快速存储器(通常SRAM)。L2高速缓冲存储器61是用以存储最近寻址的数据或频繁使用的数据，使得对主存储器(例如安全存储器)的存取的数目可减少的次要高速缓冲存储器。L2高速缓冲存储器61也可用于缓冲程序指令。通常，L2高速缓冲存储器61大于GMEM 70。

[0074] GPU 12的其它硬件块还可经配置以仅对不安全存储器具有存取权。举例来说，基元控制(PC)单元和可见性流压缩器(VSC)可经配置以仅对不安全存储器具有存取权。PC单元控制图元(例如三角形)如何进行或“遍历”通过图形管线(例如图3的图形3D处理管线30)。VSC用于基于瓦片或延迟渲染方案中以压缩和管理可见性流。一般来说，在一些情况下，避免要求某些硬件块写入到安全存储器可为有益的。此类情形包含其中硬件块不写入安全内容的情形，以及在硬件块写入图形驱动程序所需的控制数据时。

[0075] 图5中的其它硬件块基于上文描述的技术而将内容存储到不安全存储器或安全存储器。就是说，在不安全模式下，可仅从不安全存储器读取数据或将数据写入到不安全存储器。在不安全模式下，不可从安全存储器读取数据。在安全模式下，仅可将数据写入到安全存储器。在安全模式下，不可将数据写入到不安全存储器。然而，在一些实例中，在安全模式下，可从安全存储器和不安全存储器两者读取数据。GPU 12的可根据存储器模式存取存储器的这些额外硬件块包含顶点获取解码(VFD)单元65、高级定序器(HLSQ)66、顶点着色器(VS)67、像素着色器(PS)68和渲染后端(RB)69。VFD 65负责应CP 62的请求而获取顶点数据。HLSQ 66控制着色器处理器(即，GPU上的执行着色器代码的可编程处理器)，从而将正执行的工作和发起的工作的正确状态填充到着色器处理器中。VS 67是在着色器处理器上执行的顶点着色器。举例来说，VS 67可包含顶点着色器代码，其执行图3的图形3D处理管线30的几何处理级34。PS 68是在着色器处理器上执行的像素着色器。举例来说，PS 68可包含像素着色器代码，其执行图3的图形3D处理管线30的像素处理管线38。渲染后端(RB)69负责为深度缓冲器和模板缓冲器写入和读取像素。

[0076] 图6是示出经配置以实施本发明的硬件强制内容保护技术的另一实例结构的框图。在图6的实例中，GPU 12和存储器控制器50与上文在图5中所述的相同，不同之处在于存储器存取控制器53的操作。另外，为了简化，存在于GPU 12中的各种硬件单元通常已标记为例如GPU硬件块71。GPU硬件块71可包含VFD单元65、HLSQ 66、VS 67、PS 68和RB 69中的一或多者。

[0077] 在图6的实例中，存储器存取控制器53可经配置以基于GPU 12的存储器模式(即，

不安全模式或安全模式)以及与存储器资源(例如存储数据的缓冲器或高速缓存线,如图6的客户端73中示出)相关联的资源描述符,将数据引导到存储器单元(例如不安全存储器56或安全存储器57)。资源描述符,例如被称作“安全标记”,可与每一资源相关联,以指示用于所述资源的数据是否应根据安全模式(例如通过安全IOMMU 52)来投送,或根据不安全模式(例如通过不安全IOMMU 51)来投送。如图6中所示出,资源描述符可指示将使用安全IOMMU 52的可信“T”资源,以及将使用不安全IOMMU 51的不可信的“U”资源。

[0078] 使用资源描述符和GPU 12的存储器模式,存储器存取控制器53可经配置以基于资源描述符,通过L2高速缓冲存储器61从GPU硬件块71引导存储器读取和写入。L2高速缓冲存储器61中的每一高速缓冲存储器行可包含资源描述符信息。在一个实例中,存储器存取控制器53可经配置以检查存在于用于特定存储器事务(例如读取或写入)的资源中的安全标记信息,且确定使用不安全IOMMU 51或安全IOMMU 52中的哪一者来投送所述事务。

[0079] 举例来说,当将GPU 12设定成在安全模式下操作时,存储器存取控制器53可检查用于存储器事务中的资源的资源描述符中的安全标记信息。如果安全标记指示可信资源“T”,那么存储器存取控制器53将引导此类安全资源到安全IOMMU 52的读取和写入两者。在一些实例中,存储器存取控制器将具有T资源描述符的资源的所有读取和写入引导到安全IOMMU 52。如果安全标记信息指示不可信的资源“U”,那么存储器存取控制器53将此类不安全资源的(例如一些或所有)读取引导到不安全IOMMU 51,但将丢弃或不允许请求不安全资源的写入(例如一些或所有)。

[0080] 根据以上实例,GPU 12可经配置以根据不安全模式和安全模式中的一者,以及与多个存储器资源中的每一者相关联的相应资源描述符,来存取第一存储器单元(例如系统存储器10)。存储器存取控制器53可经配置以读取所述多个存储器资源的资源描述符,且接收对到第一存储器单元的存储器事务的请求。

[0081] 存储器存取控制器53可进一步经配置以在GPU 12根据安全模式操作时,响应于所述请求,将与所述多个存储器资源中具有安全资源描述符的存储器资源有关的读取和写入存储器事务引导到第一存储器单元的安全部分。存储器存取控制器53可进一步经配置以在GPU 12根据安全模式操作时,响应于所述请求,将与所述多个存储器资源中具有不安全资源描述符的存储器资源有关的读取存储器事务引导到第一存储器单元的不安全部分。存储器存取控制器53可进一步经配置以在所述GPU根据安全模式操作时,响应于所述请求,丢弃与所述多个存储器资源中具有不安全资源描述符的存储器资源有关的写入存储器事务。

[0082] 在本发明的另一实例中,存储器存取控制器53可进一步经配置以在所述GPU根据不安全模式操作时,响应于所述请求,将与所述多个存储器资源中具有不安全资源描述符的存储器资源有关的读取和写入存储器事务引导到第一存储器单元的不安全部分。存储器存取控制器53可进一步经配置以在所述GPU根据不安全模式操作时,响应于所述请求,丢弃与所述多个存储器资源中具有安全资源描述符的存储器资源有关的读取和写入存储器事务。

[0083] 图7A和7B是示出经配置以执行本发明的硬件强制内容保护技术的其它实例结构的框图。在图7A和7B的实例中,存储器控制器100可包含一或多个MMU。如上文所描述,MMU实施虚拟化存储器方案,其为其客户端提供页的连续视图。可将虚拟存储器空间分成若干虚拟页。MMU可实施一或多个上下文组来维持这些虚拟页表。所述上下文组可包含将虚拟存储

器地址映射到物理存储器地址的页表(PT)条目以及指示对于每一上下文组中的特定PT条目是否允许读取、写入或读取和写入两者的规则。

[0084] 在图7A和7B的实例中，存储器控制器100的MMU可包含不安全上下文组102和安全上下文组105。不安全上下文组102可包含为只读存取映射的不安全PT条目104。不安全PT条目104可包含从虚拟存储器地址到不安全存储器56中的物理存储器地址的映射。由于为只读存取映射不安全PT条目104，因此使用不安全上下文组102的存储器控制器100仅能够从不安全存储器56读取。安全上下文组105可包含为只读存取映射的不安全PT条目106，以及为读取和写入存取(R/W)两者映射的安全PT条目108。不安全PT条目106可包含从虚拟存储器地址到不安全存储器56中的物理存储器地址的映射。安全PT条目106可包含从虚拟存储器地址到安全存储器57中的物理存储器地址的映射。

[0085] 当使用上文所论述技术中的一者将GPU 12置于安全模式中时，存储器存取控制器53可经配置以将用于GPU硬件块71的存储器事务引导到存储器控制器100的安全上下文组105。如果GPU 12，或来自使用GPU 12的客户端的指令，尝试执行到不安全资源(例如不安全存储器56)中的写入，那么存储器控制器100经配置以发布页错误，因为不安全上下文组102中的PT条目映射为仅在安全上下文组105中读取。页错误向客户端指示此类存储器事务是不允许的。

[0086] 在本发明的一个实例中，CP 62可经配置以总是在不安全模式下操作，无关于GPU12的存储器模式。也就是说，CP 62可经配置以总是使用不安全上下文组102。图7A示出当GPU 12处于安全模式时，存储器事务从GPU 12到不安全上下文组102和安全上下文组105的流动。图7B示出当GPU 12处于不安全模式时，存储器事务从GPU 12到不安全上下文组102和安全上下文组105的流动。

[0087] 为了重申，GPU 12可经配置以根据不安全模式和安全模式中的一者来存取存储器(例如不安全存储器56或安全存储器57)。GPU 12可包含存储器存取控制器53，其经配置以在GPU 12在安全模式下操作时，将存储器事务从GPU 12的至少一个硬件单元(例如GPU硬件块71中的一或多者)引导到存储器控制器100中的安全上下文组105。存储器存取控制器53还可经配置以在GPU 12在不安全模式下操作时，将存储器事务从GPU12的至少一个硬件单元引导到存储器控制器100中的不安全上下文组102。

[0088] 如上文所描述，安全上下文组105可包含到存储器(例如不安全存储器56)的不安全部分的只读页表条目，以及到所述存储器(例如安全存储器57)的安全部分的读取/写入页表条目。不安全上下文组102可包含到所述存储器(例如不安全存储器56)的不安全部分的只读页表条目。在一个实例中，存储器控制器100可经配置以在接收到将数据写入到含于安全上下文组105的只读页表条目内的地址中的请求时，发出页错误。

[0089] 在上文所述的实例中的任一者中，当GPU 12从安全模式转变到不安全模式时，GPU12的各种高速缓冲存储器、存储器和寄存器内能还有安全内容。在本发明的一个实例中，提供一种机制来清除可能保持安全内容的GPU 12的各种存储单元和/或使其失效，之后允许使用不安全存储器模式的不安全工作在GPU 12上启动。

[0090] 在此上下文中，清除存储器意味着存储在所述存储器中的数据被擦除和/或允许被覆写。实际上，清除可涉及解除分配存储器单元的所有存储器地址，使得存储器单元中的所有数据可被覆写。在其它实例中，清除可涉及覆写存储器单元中的所有数据(例如，使用

全1或全0)，使得任何先前存储的数据不再可用。如果未清除存储器单元，那么不安全工作可将安全数据的后续保留复制到不安全存储器。可经由安全软件技术、硬件技术或两种技术的组合来解决此问题。无论如何，清除及到不安全的转变可为原子操作，这是因为此操作是由不安全驱动程序触发的。在此上下文中，不可中断的操作包含与转变回到不安全模式一起清除内部GPU 12存储器(即，不可中断地)。例如，必须存在做两件事(改变模式及清除内部存储器)的单一“命令”，否则，恶意软件可仅执行回到不安全模式的转变，且不执行清除操作。

[0091] 在一些实例中，在从安全模式转变到不安全模式时可能没有必要清除GPU 12的所有存储单元。替代地，仅需要清除存储单元的一部分以有效地防止对安全内容的未授权的存取。作为一个实例，可仅清除所存储的内容的一半。作为另一实例，可清除数据的每隔一个数据块(例如，每隔32个字节的数据)。

[0092] 图8是示出根据本发明的一个实例的高速缓冲存储器清除技术的框图。在图8的实例中，使用安全软件解决方案来使GPU在安全模式与不安全模式之间转变。在一个实例中，GPU寄存器(例如清除寄存器74)处于在主机CPU 6上运行的安全软件(例如安全OS 54)的控制下。如果GPU驱动器7将GPU 12的存储器模式从不安全模式切换到安全模式，那么GPU驱动程序7还将调用安全OS 54中的安全软件来清除GPU 12的高速缓冲存储器、存储器或寄存器上剩余的任何安全内容，包含L2高速缓冲存储器61、GMEM70和其它寄存器72。此时，安全OS 54可首先通过将存储器清除和/或失效指令写入到清除寄存器74中，来启动GPU 12上的工作。此类指令将导致GPU 12中的所有其余安全数据被清除。所述指令可为着色器程序、存储器写入和/或寄存器编程(例如，GPU L2高速缓冲存储器失效)的组合。

[0093] 图9是示出根据本发明的另一实例的高速缓冲存储器清除技术的框图。在图9的实例中，使用硬件解决方案来使GPU 12在安全模式与不安全模式之间转变。在此实例中，外部可见(例如存储器映射的输入/输出(MMIO))或内部(例如命令流)寄存器76可经配置，使得其由图形驱动程序7直接写入。GPU 12的硬件可经配置以使得当写入寄存器76时(例如当从安全模式到不安全模式时)，GPU 12的硬件完成当前安全工作，排放管线(即，去除正处理的任何其余安全内容)，且清除可含有安全内容的所有寄存器、存储器和高速缓冲存储器(包含L2高速缓冲存储器61、GMEM 70和其它寄存器72)和/或使其无效。此清除过程可包含使用驻存在GPU 12上的硬接线或安全地加载且受保护的着色器代码。

[0094] 图10是说明根据本发明的一个实例的方法的流程图。GPU 12，包含存储器存取控制器53，以及存储器控制器100可经配置以执行图10的技术。

[0095] 在本发明的一个实例中，存储器存取控制器53可经配置以通过将存储器事务从GPU12的至少一个硬件单元引导到存储器控制器100中的不安全上下文组，根据不安全模式来存取存储器(例如系统存储器10)的不安全部分(202)。存储器存取控制器53可进一步经配置以通过将存储器事务从GPU 12的至少一个硬件单元引导到存储器控制器100中的安全上下文组，根据安全模式来存取所述存储器安全部分安全部分(204)。在本发明的一个实例中，安全上下文组包含到所述存储器的不安全部分的只读页表条目以及到所述存储器的安全部分的读取/写入页表条目，且不安全上下文组包含到所述存储器的不安全部分的只读页表条目。在本发明的另一实例中，存储器控制器100可经配置以在接收到将数据写入到含于安全上下文组的只读页表条目内的地址中的请求(206)时，发布页错误(208)。

[0096] 在本发明的另一实例中，GPU的至少一个硬件单元包含顶点获取解码单元、高级定序器、顶点着色器、像素着色器和渲染后端单元中的一或多者。

[0097] 在本发明的另一实例中，GPU 12可经配置以使用前端命令处理器，通过不安全上下文组来存取所述存储器的不安全部分，不管GPU 12在不安全模式还是安全模式下操作。

[0098] 在本发明的另一实例中，GPU驱动程序7可经配置以将GPU 12置于安全模式或不安全模式。在本发明的另一实例中，GPU 12可经配置以从GPU驱动程序7接收让GPU12的命令流寄存器清除GPU 12的一或多个内部存储器并使其失效的指令。GPU 12可进一步经配置以基于命令流寄存器中的指令，在GPU 12从安全模式转变为不安全模式时，将至少一些内容从GPU 12的一或多个内部存储器清除并使其失效。

[0099] 在本发明的另一实例中，GPU 12可经配置以在清除寄存器处接收清除GPU的一或多个内部存储器并使其失效的指示，且基于清除寄存器中的所述指示，在GPU 12从安全模式转变为不安全模式时，将至少一些内容从GPU 12的一或多个内部存储器清除并使其失效。

[0100] 图11是说明根据本发明的一个实例的方法的流程图。GPU 12，包含存储器存取控制器53，可经配置以执行图11的技术。

[0101] 在本发明的一个实例中，GPU 12可经配置以根据不安全模式和安全模式中的一者，以及与多个存储器资源中的每一者相关联的相应资源描述符，来存取第一存储器单元(例如系统存储器10)。存储器存取控制器53可经配置以读取与所述多个存储器资源中的每一者相关联的相应资源描述符(302)，且接收对到第一存储器单元的存储器事务的请求(304)。

[0102] 存储器存取控制器53可进一步经配置以确定与和对存储器事务的请求有关的存储器资源相关联的资源描述符是安全还是不安全资源描述符(306)。存储器存取控制器53可进一步经配置以在GPU 12根据安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说相应资源描述符是安全资源描述符的存储器资源有关的读取和写入存储器事务引导到第一存储器单元的安全部分(312)。存储器存取控制器53可进一步经配置以在GPU 12根据安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说相应资源描述符是不安全资源描述符的存储器资源有关的读取和写入存储器事务引导到第一存储器单元的不安全部分(308)。存储器存取控制器53还可经配置以在GPU 12根据安全模式操作时，响应于所述请求，丢弃与所述多个存储器资源中对于其来说相应资源描述符是不安全资源描述符的存储器资源有关的写入存储器事务(310)。

[0103] 在本发明的另一实例中，存储器存取控制器53进一步经配置以在GPU 12根据不安全模式操作时，响应于所述请求，将与所述多个存储器资源中对于其来说相应资源描述符是不安全资源描述符的存储器资源有关的读取和写入存储器事务引导到第一存储器单元的不安全部分，且在GPU 12根据不安全模式操作时，响应于所述请求，丢弃与所述多个存储器资源中对于其来说相应资源描述符是安全资源描述符的存储器资源有关的读取和写入存储器事务。

[0104] 在本发明的另一实例中，存储器存取控制器53经配置以通过利用安全存储器管理单元，将数据写入到第一存储器单元的安全部分，所述安全存储器管理单元利用含有第一存储器单元的安全部分的地址范围的安全页表。在本发明的另一实例中，存储器存取控制

器53经配置以通过利用不安全存储器管理单元从第一存储器单元的不安全部分读取数据，所述不安全存储器管理单元利用含有第一存储器单元的不安全部分的地址范围的不安全页表。

[0105] 在本发明的另一实例中，存储器存取控制器53经配置以根据来自虚拟存储器地址范围的虚拟存储器地址读取和写入数据，其中所述虚拟存储器地址范围包含与安全存储器管理单元所利用的安全页表中的条目有关的第一虚拟存储器地址范围，以及与不安全存储器管理单元所利用的不安全页表中的条目有关的第二虚拟存储器地址范围。

[0106] 在一或多个实例中，上文所描述的功能可以硬件、软件、固件或其任何组合来实施。如果以软件实施，那么功能可作为一或多个指令或代码存储在包括非暂时性计算机可读媒体的制品上。计算机可读媒体可包含计算机数据存储媒体。数据存储媒体可为可由一或多个计算机或者一或多个处理器存取以检索用于实施本发明中所描述的技术的指令、代码和/或数据结构的任何可用媒体。作为实例而非限制，此类计算机可读媒体可包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储装置、磁盘存储装置或其它磁性存储装置、快闪存储器或可用来运载或存储呈指令或数据结构形式的所要程序代码且可由计算机存取的任何其它媒体。如本文中所使用，磁盘和光盘包含压缩光盘(CD)、激光光盘、光学光盘、数字多功能光盘(DVD)、软性磁盘和蓝光光盘，其中磁盘通常以磁性方式再现数据，而光盘用激光以光学方式再现数据。以上各项的组合也应包含在计算机可读媒体的范围内。

[0107] 代码可由一或多个处理器执行，所述一或多个处理器例如一或多个DSP、通用微处理器、ASIC、FPGA，或其它等效集成或离散逻辑电路。另外，在一些方面中，可在专用硬件和/或软件模块内提供本文中所描述的功能性。并且，所述技术可完全实施于一或多个电路或逻辑元件中。

[0108] 本发明的技术可在多种多样的装置或设备中实施，包含无线手持机、集成电路(IC)或一组IC(例如芯片组)。本发明中描述各种组件、模块或单元是为了强调经配置以执行所揭示的技术的装置的功能方面，但未必需要通过不同硬件单元来实现。确切地，如上文所描述，各种单元可结合合适的软件和/或固件组合在编解码器硬件单元中，或由互操作硬件单元的集合来提供，所述硬件单元包含如上文所描述的一或多个处理器。

[0109] 已描述了各种实例。这些和其它实例在所附权利要求书的范围内。

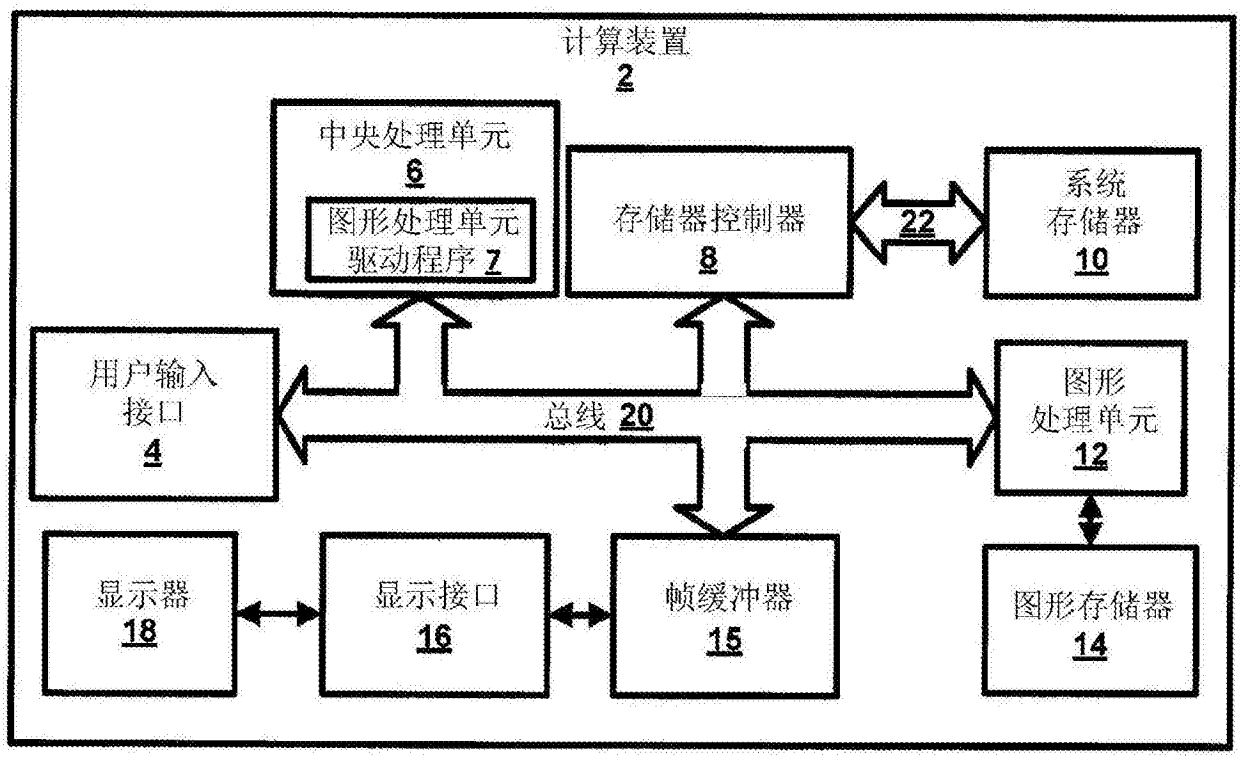

计算装置
2
中央处理单元
6
图形处理单元
驱动程序 7
存储器控制器
8
22
系统
存储器
10
用户输入
接口
4
总线 20
图形
处理单元
12
显示器
18
显示接口
16
帧缓冲器
15
图形存储器
14

图1

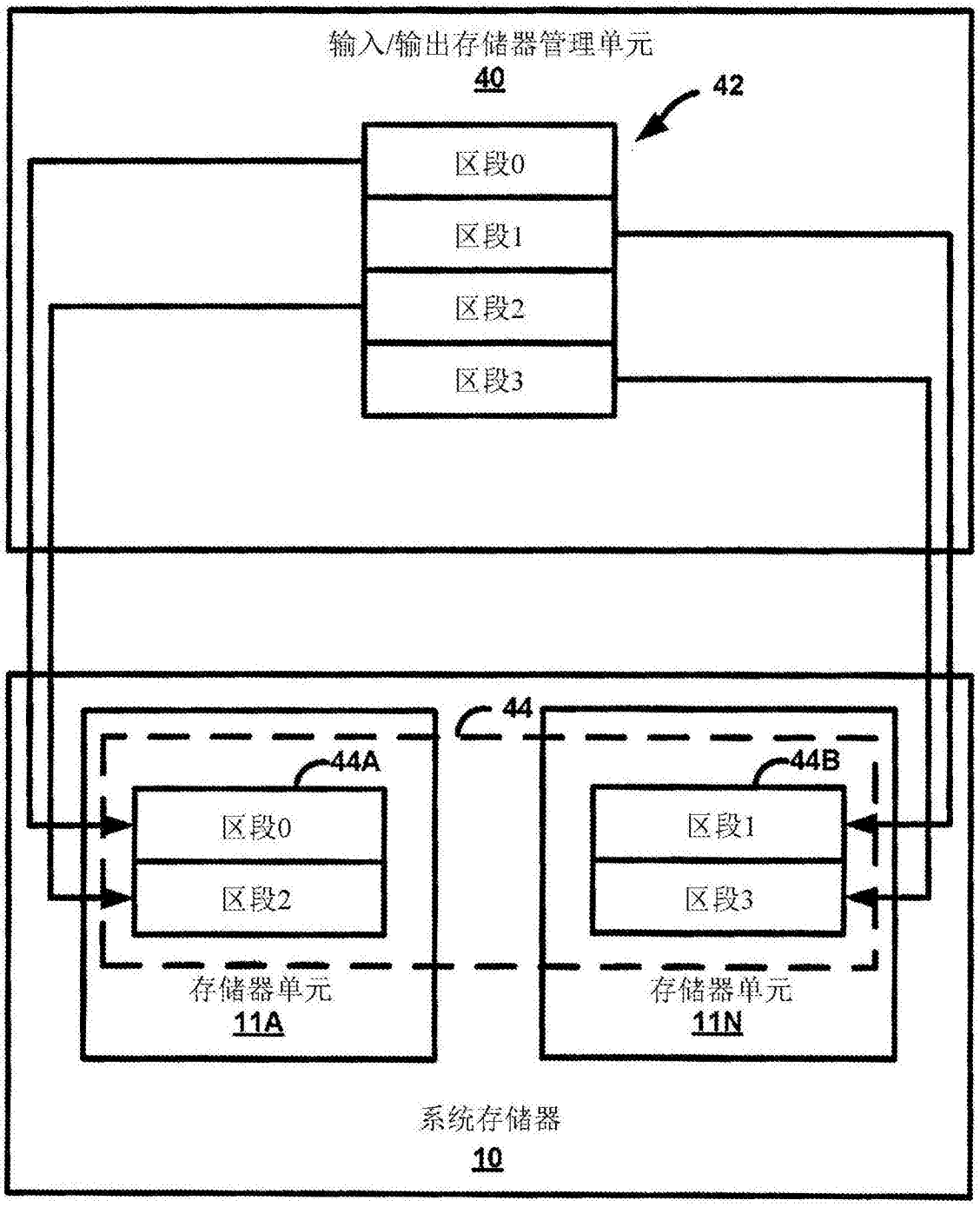

输入/输出存储器管理单元
40
42
区段0
区段1
区段2
区段3
44
44A
44B
区段0
区段2
区段1
区段3
存储器单元
11A
存储器单元
11N
系统存储器
10

图2

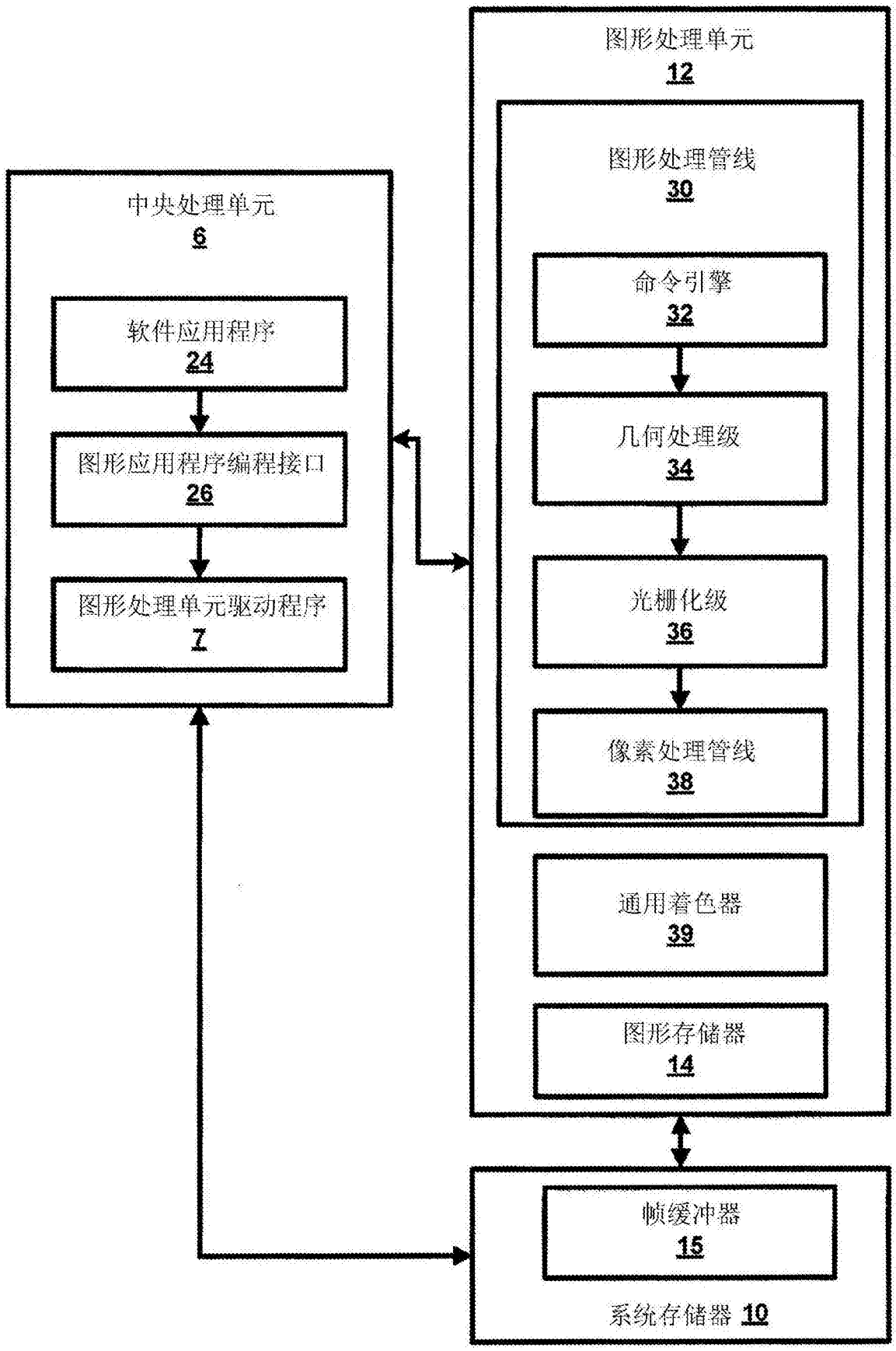

图形处理单元
12
图形处理管线
30
命令引擎
32
几何处理级
34
光栅化级
36
像素处理管线
38
通用着色器
39
图形存储器
14
中央处理单元
6
软件应用程序
24
图形应用程序编程接口
26
图形处理单元驱动程序
7
帧缓冲器
15
系统存储器 10

图3

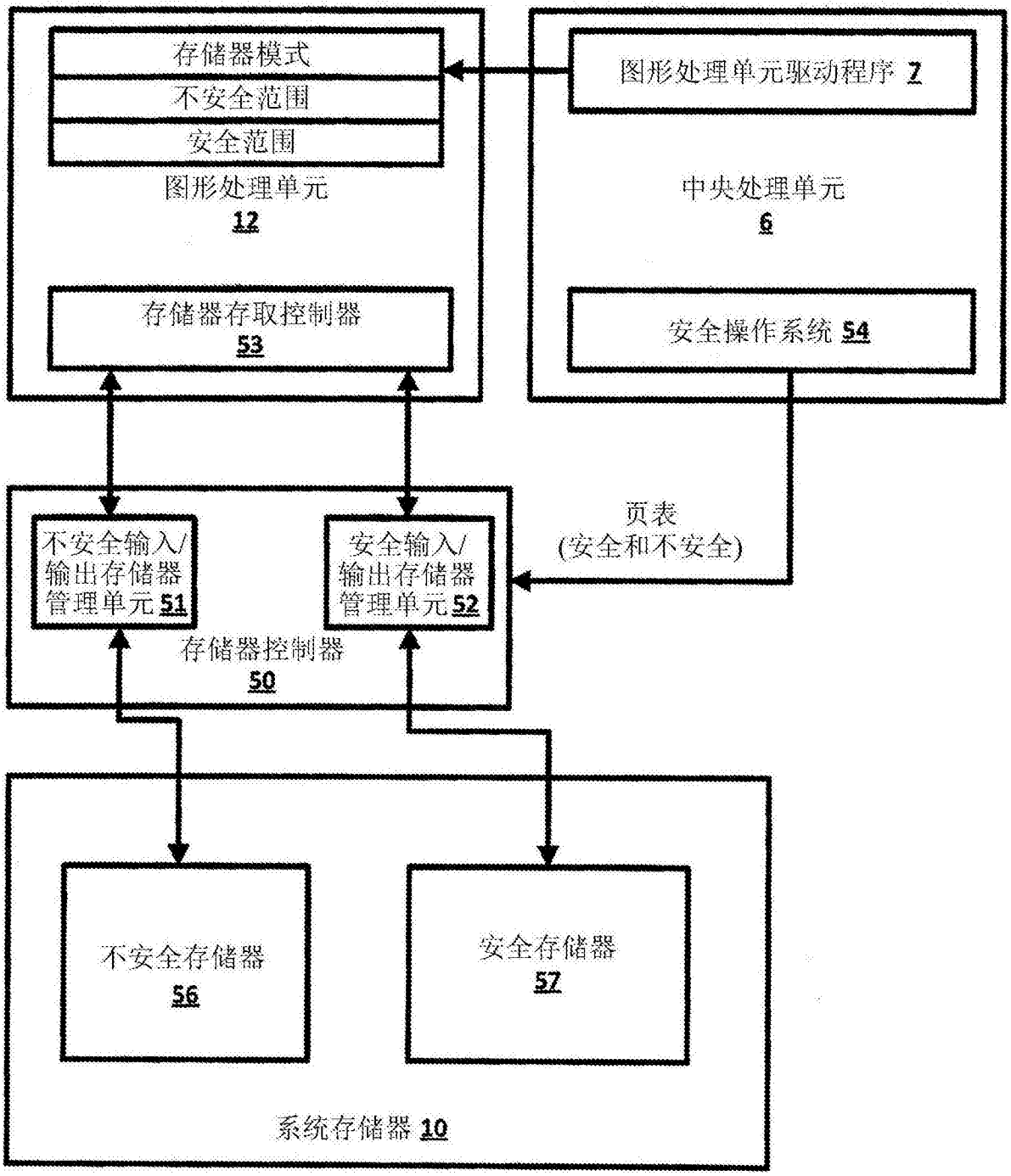

存储器模式
不安全范围
安全范围
图形处理单元
12
图形处理单元驱动程序 7
中央处理单元
6
存储器存取控制器
53
安全操作系统 54
页表
(安全和不安全)
不安全输入/
输出存储器
管理单元 51
安全输入/
输出存储器
管理单元 52
存储器控制器
50
不安全存储器
56
安全存储器
57
系统存储器 10

图4

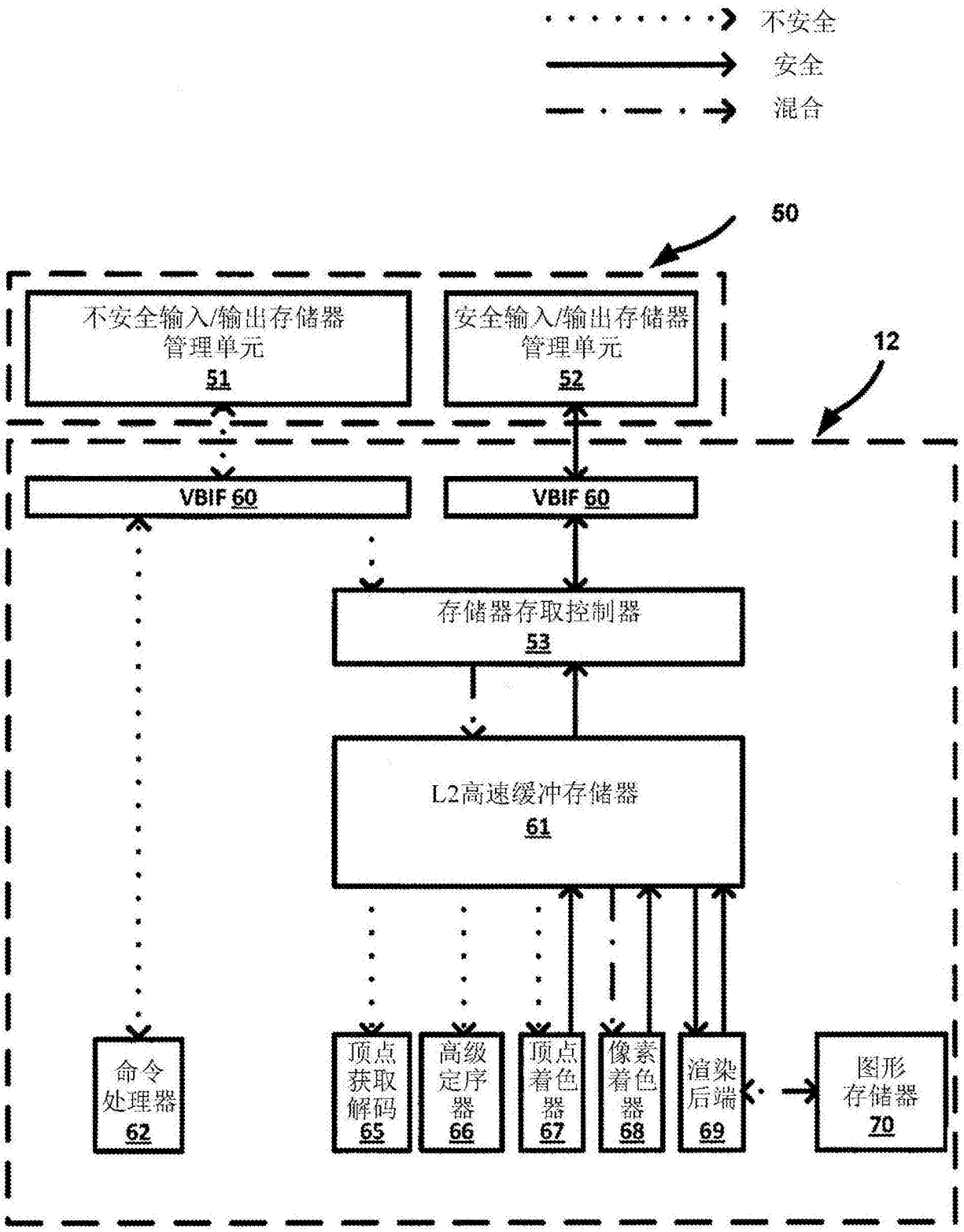

不安全
安全
混合
50
不安全输入/输出存储器管理单元 51
安全输入/输出存储器管理单元 52
12
VBIF 60
VBIF 60
存储器存取控制器 53
L2高速缓冲存储器 61
命令处理器 62
顶点获取解码 65
高级定序器 66
顶点着色器 67
像素着色器 68
渲染后端 69
图形存储器 70

图5

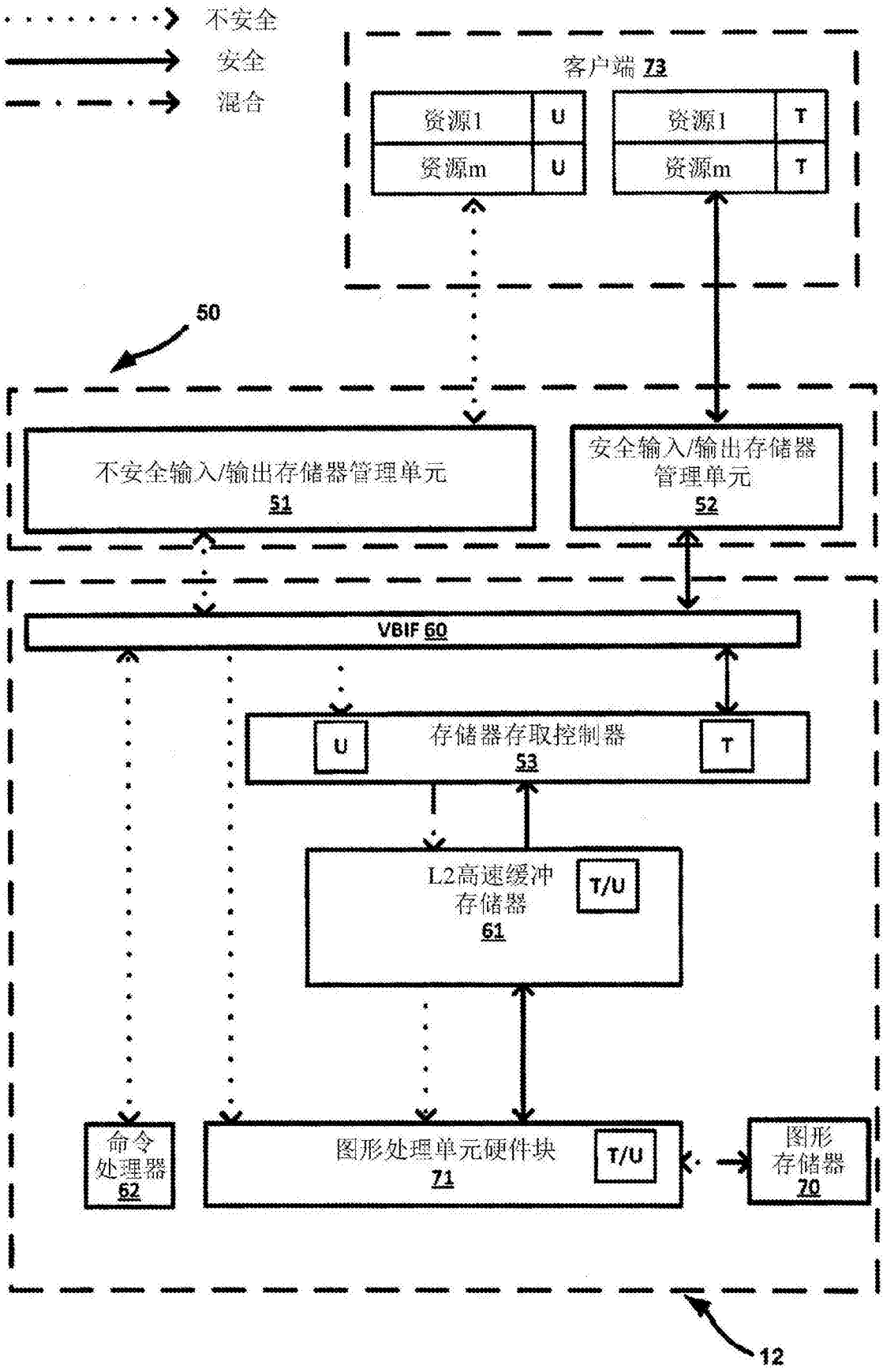

不安全
安全
混合
客户端 73
资源1
U
资源m
U
资源1
T
资源m
T
50
不安全输入/输出存储器管理单元
51
安全输入/输出存储器
管理单元
52
VBIF 60
U
存储器存取控制器
53
T
L2高速缓冲
存储器
61
T/U
命令
处理器
62
图形处理单元硬件块
71
T/U
图形
存储器
70
12

图6

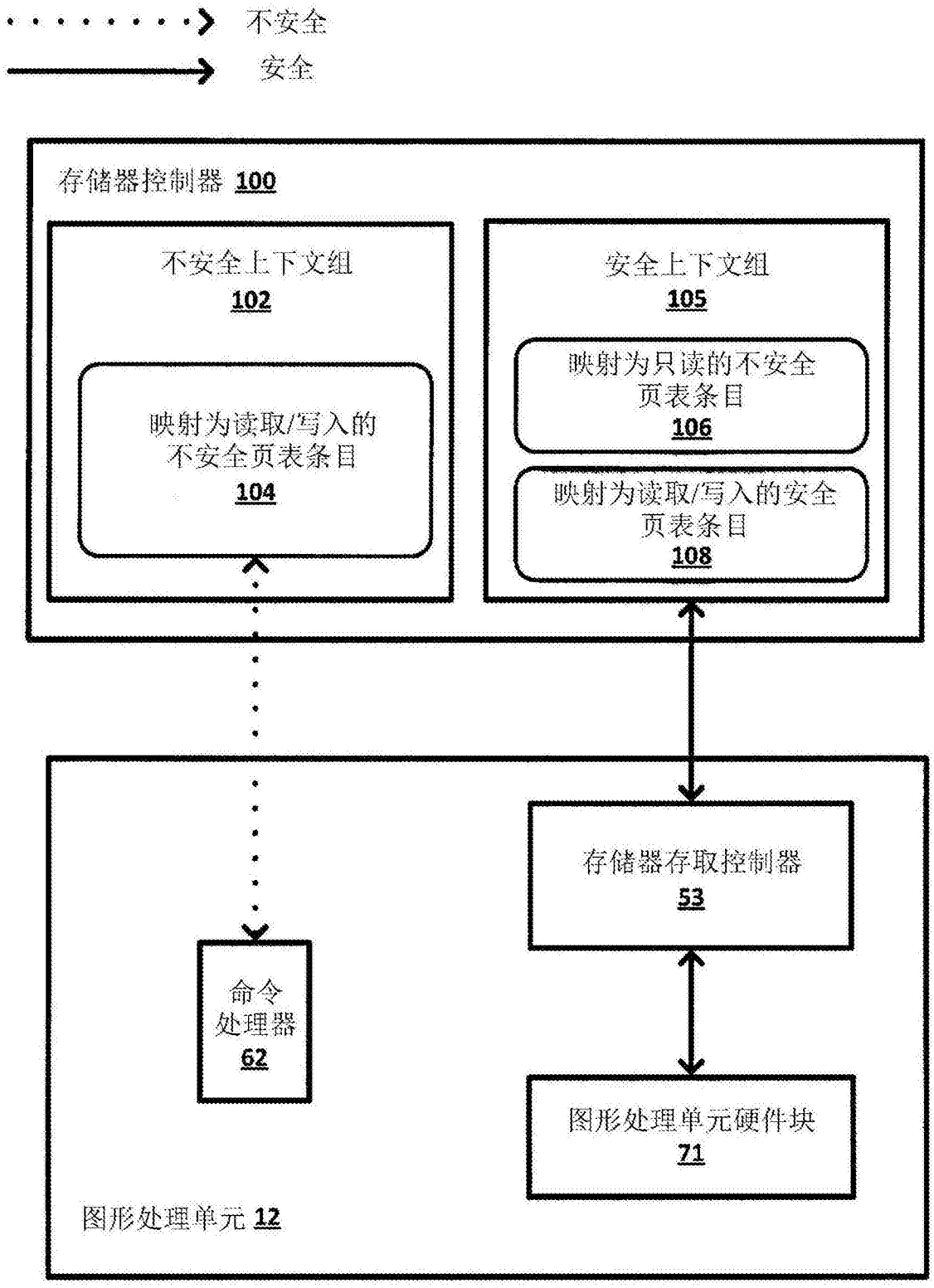
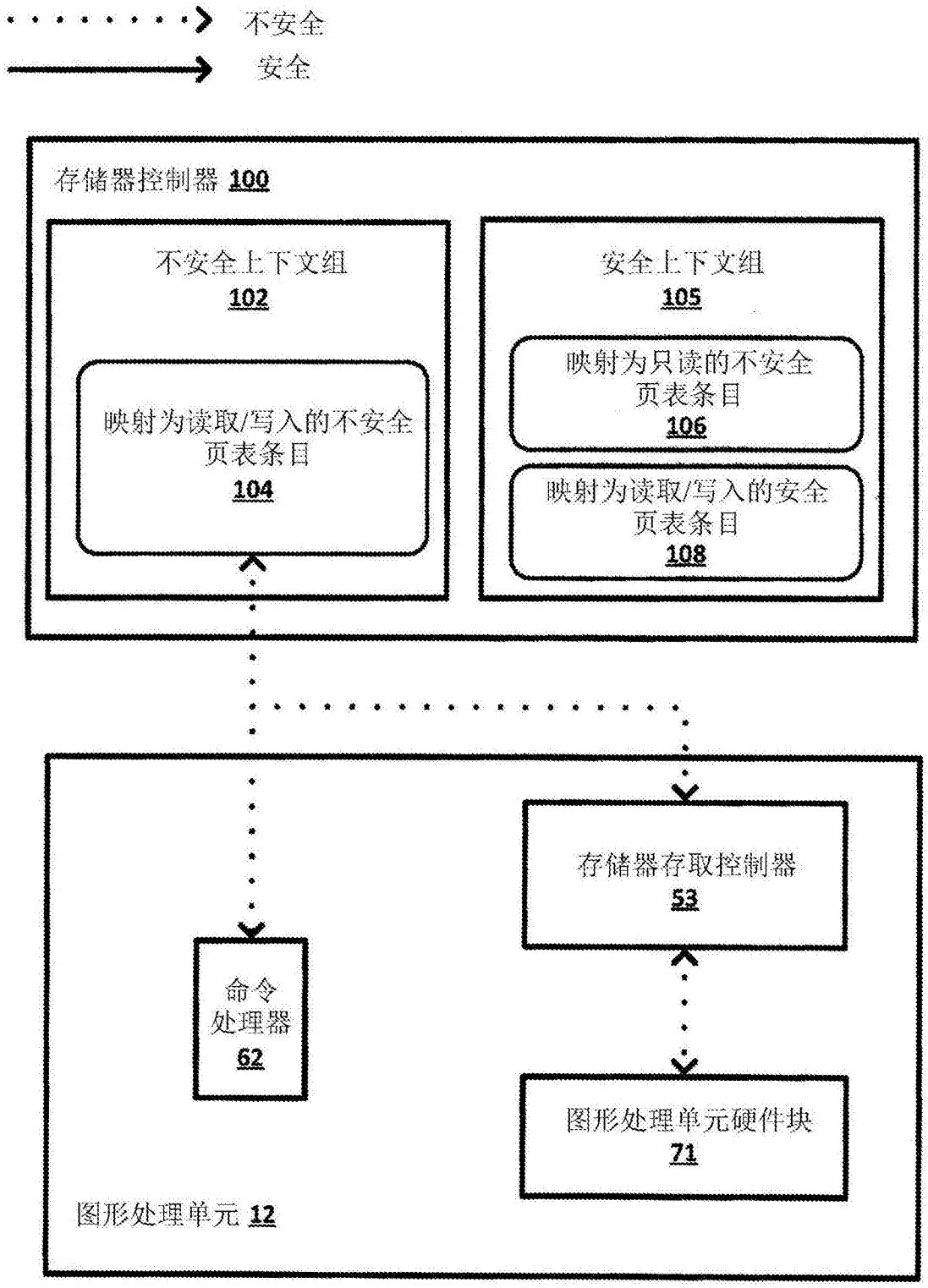

不安全
安全
存储器控制器 100
不安全上下文组
102
映射为读取/写入的
不安全页表条目
104
安全上下文组
105
映射为只读的不安全
页表条目
106
映射为读取/写入的安全
页表条目
108
存储器存取控制器
53
命令
处理器
62
图形处理单元硬件块
71
图形处理单元 12
安全模式

图7A


不安全
安全
存储器控制器 100
不安全上下文组
102
映射为读取/写入的不安全
页表条目
104
安全上下文组
105
映射为只读的不安全
页表条目
106
映射为读取/写入的安全
页表条目
108
存储器存取控制器
53
命令
处理器
62
图形处理单元硬件块
71
图形处理单元 12

不安全模式

图7B

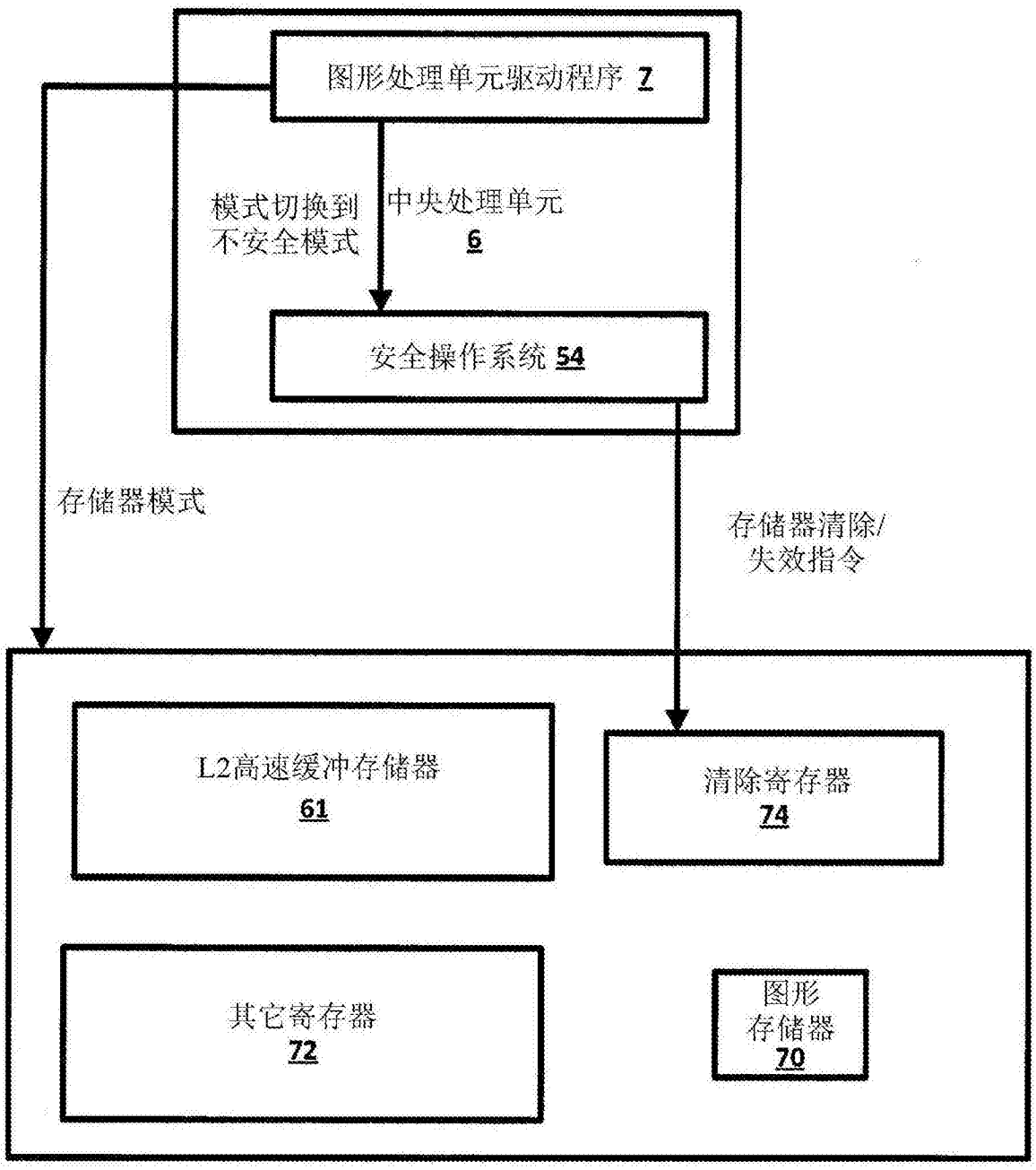

图形处理单元驱动程序 7
模式切换到
不安全模式
中央处理单元
6
安全操作系统 54
存储器模式
存储器清除/
失效指令
L2高速缓冲存储器
61
清除寄存器
74
其它寄存器
72
图形
存储器
70

图8

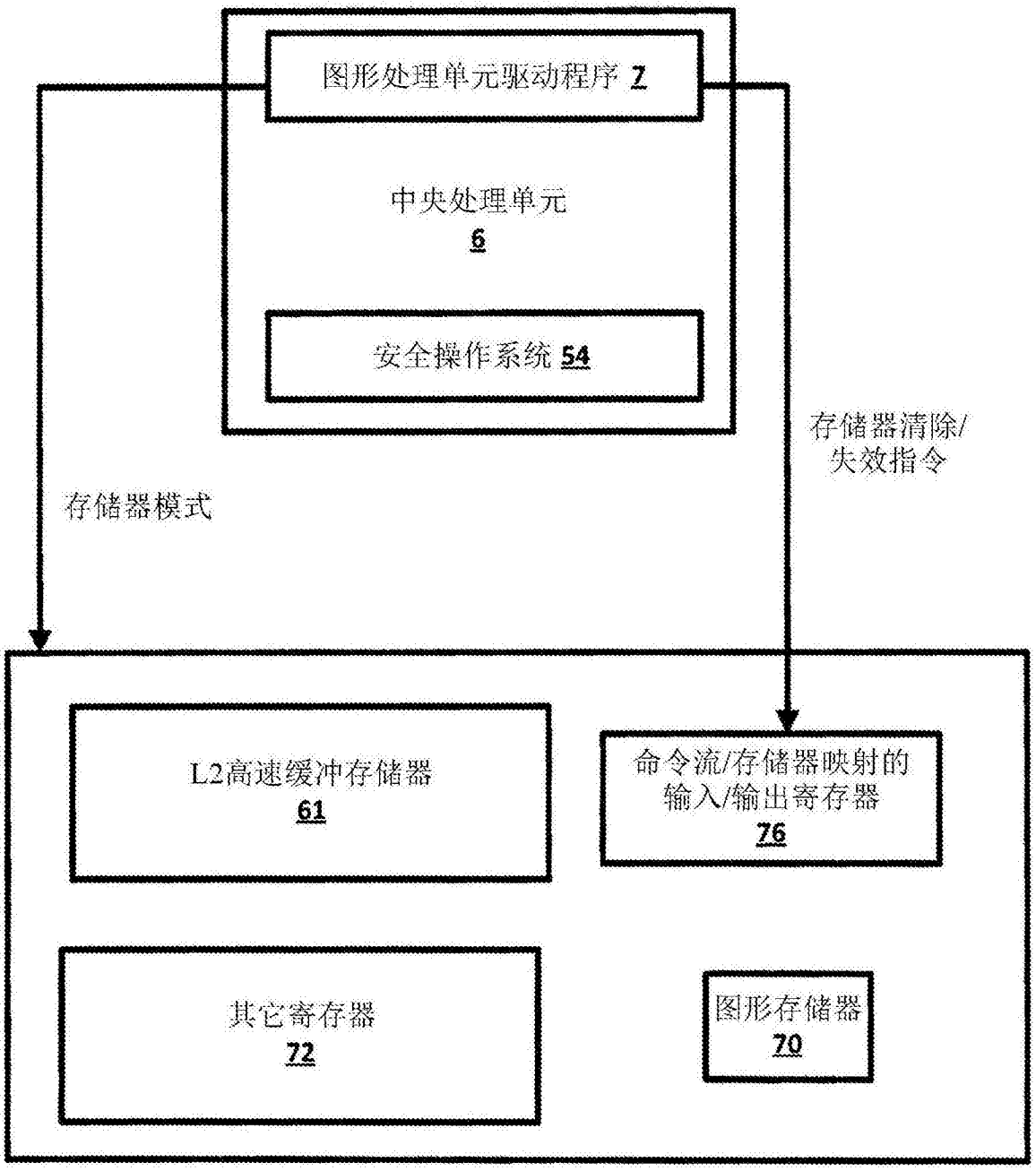

图形处理单元驱动程序 7
中央处理单元
6
安全操作系统 54
存储器清除/
失效指令
存储器模式
L2高速缓冲存储器
61
命令流/存储器映射的
输入/输出寄存器
76
其它寄存器
72
图形存储器
70

图9

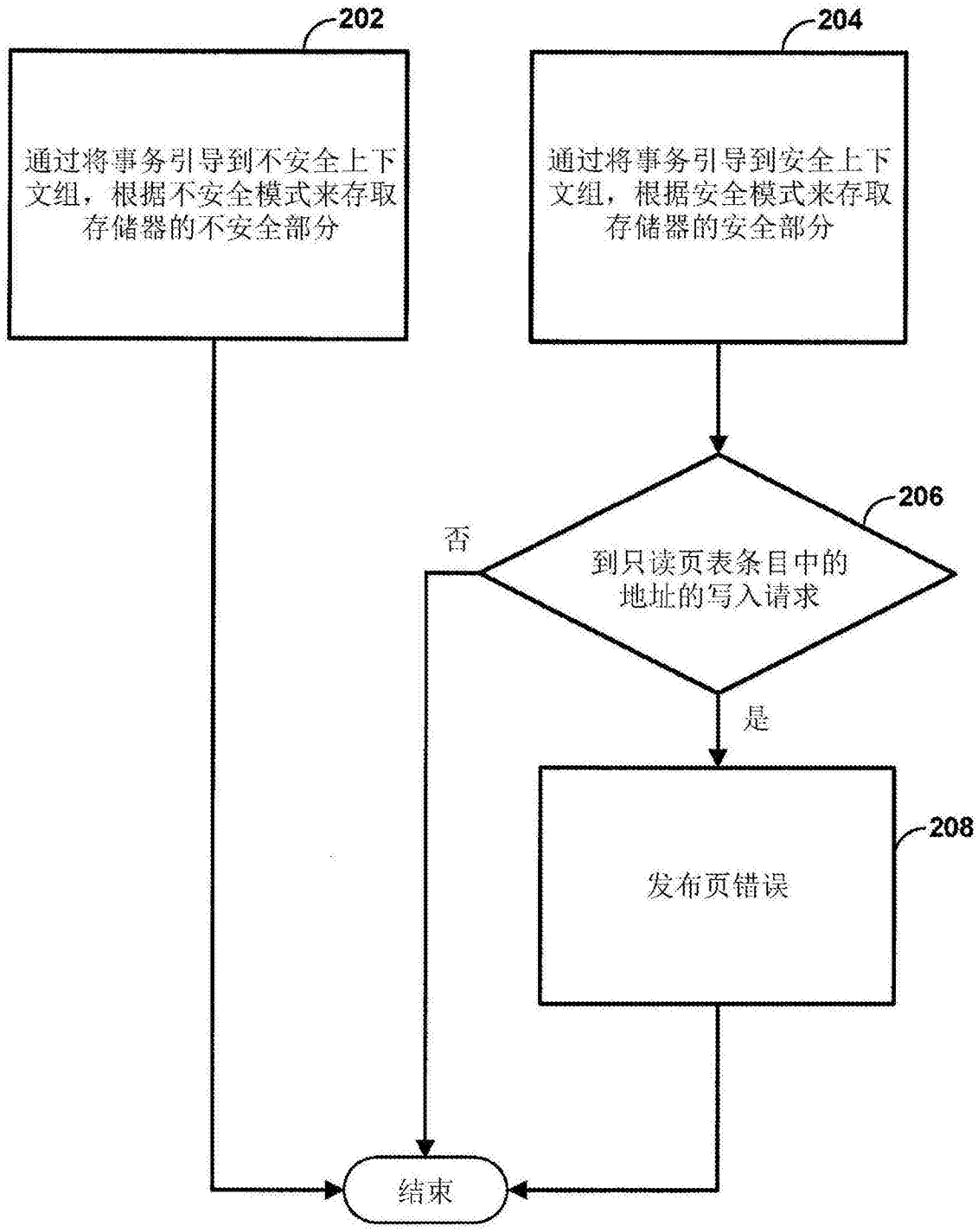

202
通过将事务引导到不安全上下文组，根据不安全模式来存取存储器的不安全部分
204
通过将事务引导到安全上下文组，根据安全模式来存取存储器的安全部分
206
到只读页表条目中的地址的写入请求
否
是
208
发布页错误
结束

图10

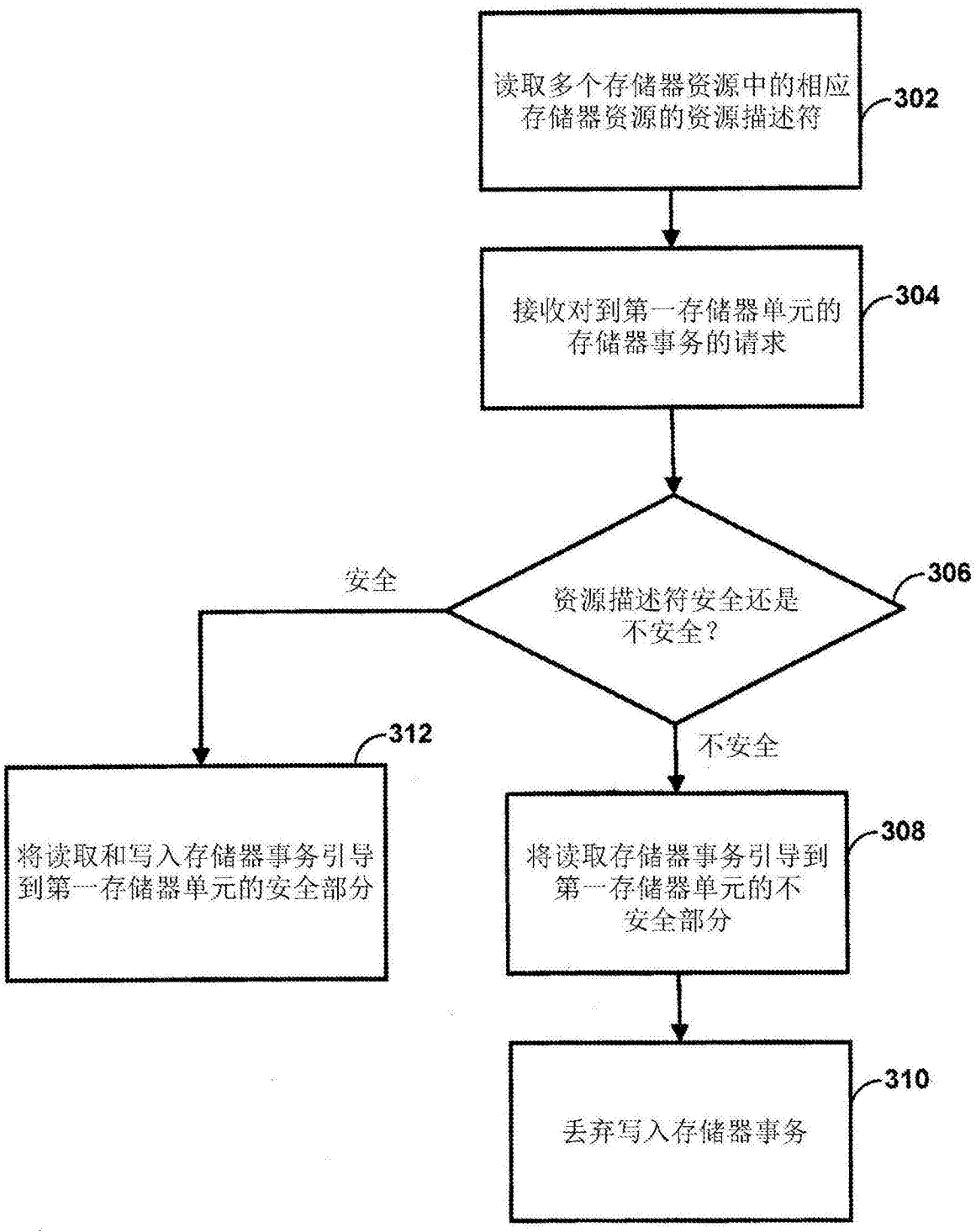

读取多个存储器资源中的相应存储器资源的资源描述符
302
接收对到第一存储器单元的存储器事务的请求
304
资源描述符安全还是不安全?
306
安全
不安全
将读取和写入存储器事务引导到第一存储器单元的安全部分
312
将读取存储器事务引导到第一存储器单元的不安全部分
308
丢弃写入存储器事务
310

图11