(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 116823585 A
(43)申请公布日 2023.09.29

(21)申请号 202310694362.4

(22)申请日 2023.06.12

(71)申请人 南方科技大学
地址 518055 广东省深圳市南山区学苑大道1088号
申请人 支付宝(杭州)信息技术有限公司

(72)发明人 张峰巍 闫守孟 宁振宇 何征宇 邓韵杰 王晨旭

(74)专利代理机构 北京永新同创知识产权代理有限公司 11376
专利代理师 林锦辉 刘景峰

(51)Int.Cl.
G06T 1/20(2006.01)

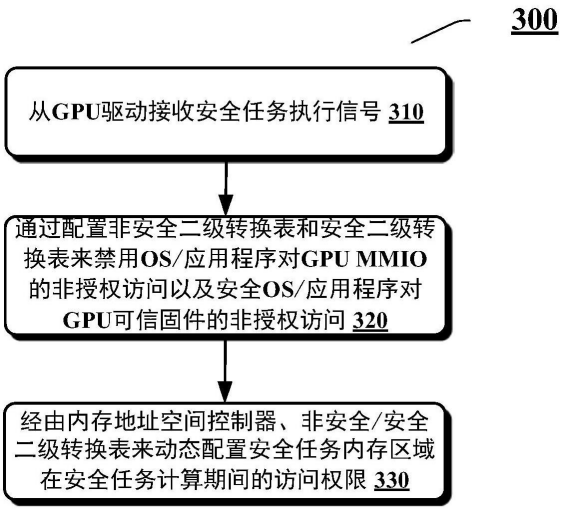
权利要求书5页 说明书17页 附图5页

(54)发明名称

GPU可信执行环境的构建方法、GPU可信计算执行方法及装置

(57)摘要

本说明书实施例提供GPU可信执行环境的构建方法、GPU可信计算执行方法及装置。Arm终端设备的内存空间被划分为可信内存区域、非安全世界内存区域、安全任务内存区域和非安全任务内存区域。响应于从GPU驱动接收到安全任务执行信号，分别配置可信内存区域中存储的非安全二级转换表和安全二级转换表，以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对GPU MMIO接口和GPU可信固件的非授权访问。经由内存地址空间控制器、非安全二级转换表和安全二级转换表动态配置安全任务内存区域在安全任务计算期间的访问权限，以在GPU执行安全任务计算期间实现针对安全任务内存区域所存储的所述安全任务的数据和代码的隔离保护。



CN 116823585 A

1.一种用于构建面向Arm终端设备的GPU可信执行环境的方法,所述Arm终端设备的内存空间被划分为可信内存区域、非安全世界内存区域、安全任务内存区域和非安全任务内存区域,所述方法包括:

响应于从GPU驱动接收到安全任务执行信号,分别配置所述可信内存区域中存储的非安全二级转换表和安全二级转换表,以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对所述GPU MMIO接口和GPU可信固件的非授权访问;以及

经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限,以在GPU执行安全任务计算期间实现针对所述安全任务内存区域所存储的所述安全任务的数据和代码的隔离保护。

2.如权利要求1所述的方法,其中,所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域,通过配置所述非安全二级转换表和所述安全二级转换表来对所述任务区域中包含不同安全任务处理阶段的数据和代码的内存区域执行页级保护以及监视对所述GPU页表区域的修改请求,并且通过配置所述内存地址空间控制器来管理DMA、GPU和/或外围设备对所述任务区域的访问以及禁用外围设备对所述GPU页表区域的写访问。

3.如权利要求2所述的方法,其中,经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限包括:

在安全任务提交期间,将所述任务区域和所述GPU页表区域的访问权限配置为完全可访问;

在安全任务执行期间,将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域的访问权限配置为禁止操作系统和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止外围设备通过DMA进行读/写操作;

在安全任务切换期间,将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域中用于后续安全任务的内存区域的访问权限配置为禁止操作系统、GPU和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止GPU和外围设备通过DMA进行读/写操作;

在安全任务完成期间,将所述任务区域和所述GPU页表区域的访问权限被置为完全可访问。

4.如权利要求3所述的方法,其中,针对所述任务区域中的数据,在所述安全任务执行期间,所述安全任务的对应内存区域所存储的数据为明文数据,其它内存区域所存储的数据为密文数据,以及在所述安全任务切换期间,所述安全任务的对应内存区域中用于后续安全任务的内存区域所存储的数据为明文数据,其它内存区域所存储的数据为密文数据。

5.如权利要求3所述的方法,其中,经由DMA传输的数据经过加密并且执行完整性检查。

6.一种用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的方法,所述Arm终端设备的内存空间被划分为可信内存区域、非安全世界内存区域、安全任务内存区域和非安全任务内存区域,所述方法包括:

响应于从GPU驱动接收到安全任务执行信号,分别配置所述可信内存区域中存储的非

安全二级转换表和安全二级转换表,以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对所述GPU MMIO接口和GPU可信固件的非授权访问;

经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限;以及

利用所配置的所述安全任务内存区域在安全任务计算期间的访问权限,在GPU执行安全任务计算期间访问所述安全任务内存区域来获取所述安全任务的数据和代码,以供GPU进行GPU可信计算。

7.如权利要求6所述的方法,其中,所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域,通过配置所述非安全二级转换表和所述安全二级转换表来对所述任务区域中包含不同安全任务处理阶段的数据和代码的内存区域执行页级保护以及监视对所述GPU页表区域的修改请求,并且通过配置所述内存地址空间控制器来管理DMA、GPU和/或外围设备对所述任务区域的访问以及禁用外围设备对所述GPU页表区域的写访问。

8.如权利要求7所述的方法,其中,经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限包括:

在安全任务提交期间,将所述任务区域和所述GPU页表区域的访问权限配置为完全可访问;

在安全任务执行期间,将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域的访问权限配置为禁止操作系统和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止外围设备通过DMA进行读/写操作;

在安全任务切换期间,将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域中用于后续安全任务的内存区域的访问权限配置为禁止操作系统、GPU和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止GPU和外围设备通过DMA进行读/写操作;

在安全任务完成期间,将所述任务区域和所述GPU页表区域的访问权限被置为完全可访问。

9.如权利要求6所述的方法,其中,在安全任务提交之前,所述方法还包括:

检查GPU任务状态寄存器,以确认所述GPU当前不存在其它正在处理的任务。

10.如权利要求6所述的方法,其中,所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域,在安全任务提交之前,所述方法还包括:

对所述GPU页表区域进行地址映射检查以及对所述任务区域中存储的数据和代码执行完整性检查。

11.如权利要求6所述的方法,还包括:

在安全任务提交之前,经由通用终端控制器将GPU中断配置为安全状态;以及

在安全任务完成后,经由通用终端控制器将GPU中断配置为非安全状态。

12.一种用于构建面向Arm终端设备的GPU可信执行环境的装置,所述Arm终端设备的内存空间被划分为可信内存区域、非安全世界内存区域、安全任务内存区域和非安全任务内

存区域,所述装置包括:

GPU保护单元,响应于从GPU驱动接收到安全任务执行信号,分别配置所述可信内存区域中存储的非安全二级转换表和安全二级转换表,以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对所述GPU MMIO接口和GPU可信固件的非授权访问;以及

安全任务保护单元,经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限,以在GPU执行安全任务计算期间实现针对所述安全任务内存区域所存储的所述安全任务的数据和代码的隔离保护。

13.如权利要求12所述的装置,其中,所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域,

所述安全任务保护单元通过配置所述非安全二级转换表和所述安全二级转换表来对所述任务区域中包含不同安全任务处理阶段的数据和代码的内存区域执行页级保护以及监视对所述GPU页表区域的修改请求,并且通过配置所述内存地址空间控制器来管理DMA、GPU和/或外围设备对所述任务区域的访问以及禁用外围设备对所述GPU页表区域的写访问。

14.如权利要求13所述的装置,其中,

在安全任务提交和安全任务完成期间,所述安全任务保护单元将所述任务区域和所述GPU页表区域的访问权限配置为完全可访问;

在安全任务执行期间,所述安全任务保护单元将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域的访问权限配置为禁止操作系统和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止外围设备通过DMA进行读/写操作;

在安全任务切换期间,所述安全任务保护单元将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域中用于后续安全任务的内存区域的访问权限配置为禁止操作系统、GPU和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止GPU和外围设备通过DMA进行读/写操作。

15.如权利要求14所述的装置,其中,所述安全任务保护单元对经由DMA传输的数据进行加密并且执行完整性检查。

16.一种用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的装置,所述Arm终端设备的内存空间被划分为可信内存区域、非安全世界内存区域、安全任务内存区域和非安全任务内存区域,所述装置包括:

GPU保护单元,响应于从GPU驱动接收到安全任务执行信号,分别配置所述可信内存区域中存储的非安全二级转换表和安全二级转换表,以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对所述GPU MMIO接口和GPU可信固件的非授权访问;

安全任务保护单元,经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限;以及

安全访问单元,利用所配置的所述安全任务内存区域在安全任务计算期间的访问权

限,在GPU执行安全任务计算期间访问所述安全任务内存区域来获取所述安全任务的数据和代码,以供GPU进行GPU可信计算。

17.如权利要求16所述的装置,其中,所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域,

所述安全任务保护单元通过配置所述非安全二级转换表和所述安全二级转换表来对所述任务区域中包含不同安全任务处理阶段的数据和代码的内存区域执行页级保护以及监视对所述GPU页表区域的修改请求,并且通过配置所述内存地址空间控制器来管理DMA、GPU和/或外围设备对所述任务区域的访问以及禁用外围设备对所述GPU页表区域的写访问。

18.如权利要求17所述的装置,其中,

在安全任务提交和安全任务完成期间,所述安全任务保护单元将所述任务区域和所述GPU页表区域的访问权限配置为完全可访问;

在安全任务执行期间,所述安全任务保护单元将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域的访问权限配置为禁止操作系统和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止外围设备通过DMA进行读/写操作;

在安全任务切换期间,所述安全任务保护单元将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域中用于后续安全任务的内存区域的访问权限配置为禁止操作系统、GPU和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止GPU和外围设备通过DMA进行读/写操作。

19.如权利要求16所述的装置,其中,所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域,在安全任务提交之前,所述安全任务保护单元对所述GPU页表区域进行地址映射检查以及对所述任务区域中存储的数据和代码执行完整性检查。

20.如权利要求16所述的装置,其中,在安全任务提交之前,所述GPU保护单元经由通用终端控制器将GPU中断配置为安全状态,以及在安全任务完成后,所述GPU保护单元经由通用终端控制器将GPU中断配置为非安全状态。

21.一种Arm终端设备,包括:

基于Arm架构的GPU可信固件,包括如权利要求16到20中任一所述的用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的装置;以及

GPU,在经由GPU可信固件所提供的可信执行环境中,使用安全任务的数据和代码来进行GPU可信计算。

22.一种用于构建面向Arm终端设备的GPU可信执行环境的装置,包括:

至少一个处理器;

与所述至少一个处理器耦合的存储器;以及

存储在所述存储器中的计算机程序,所述至少一个处理器执行所述计算机程序来实现如权利要求1到5中任一所述的用于构建面向Arm终端设备的GPU可信执行环境的方法。

23.一种用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的装置,包括:

至少一个处理器;

与所述至少一个处理器耦合的存储器;以及

存储在所述存储器中的计算机程序,所述至少一个处理器执行所述计算机程序来实现如权利要求6到11中任一所述的用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的装置的方法。

GPU可信执行环境的构建方法、GPU可信计算执行方法及装置

技术领域

[0001] 本说明书实施例通常涉及可信计算领域，尤其涉及面向Arm终端设备的GPU可信执行环境的构建方法、GPU可信计算执行方法及装置。

背景技术

[0002] GPU被广泛用于高性能应用，比如，3D游戏、视频处理和视频压缩、移动虚拟现实以及神经网络训练和推理。此外，GPU不仅适用于服务器和云环境，而且也适用于小型嵌入式系统，比如，智能手机和自动驾驶汽车，以满足日益增长的性能需求。

[0003] GPU越来越受欢迎并且被广泛应用，但是并没有出现相应级别的GPU可信计算方案。攻击者可以利用操作系统级别的众多漏洞来控制GPU驱动，随后可以通过内存映射I/O(MMIO)接口来访问GPU内存，从而能够访问GPU应用程序所处理的机密数据。此外，攻击者可以通过篡改GPU页表来打破GPU应用程序之间的隔离，从而泄露GPU上处理的潜在机密数据。随着GPU计算所使用的个人身份信息和机密秘密逐步增加，迫切需要高性能的GPU可信计算方案来确保GPU计算安全。

发明内容

[0004] 本说明书实施例提供面向Arm终端设备的GPU可信执行环境的构建方法、GPU可信计算执行方法及装置。利用该GPU可信执行环境构建方法及装置，可以从GPU可信固件中删除易受攻击的安全操作系统和安全应用程序来实现小型GPU可信固件，该小型GPU可信固件可以通过利用内存地址空间控制器、非安全二级转换和安全二级转换组件来防止来自非安全操作系统/非安全应用程序以及安全操作系统/安全应用程序的攻击，从而确保安全地执行GPU可信计算。

[0005] 根据本说明书实施例的一个方面，提供一种用于构建面向Arm终端设备的GPU可信执行环境的方法，所述Arm终端设备的内存空间被划分为可信内存区域、非安全世界内存区域、安全任务内存区域和非安全任务内存区域，所述方法包括：响应于从GPU驱动接收到安全任务执行信号，分别配置所述可信内存区域中存储的非安全二级转换表和安全二级转换表，以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对所述GPU MMIO接口和GPU可信固件的非授权访问；以及经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限，以在GPU执行安全任务计算期间实现针对所述安全任务内存区域所存储的所述安全任务的数据和代码的隔离保护。

[0006] 可选地，在上述方面的一个示例中，所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域，通过配置所述非安全二级转换表和所述安全二级转换表来对所述任务区域中包含不同安全任务处理阶段的数据和代码的内存区域执行页级保护以及监视对所述GPU页表区域的修改请求，并且通过配置所述内存地址空间控制器来管理DMA、GPU和/或外围设备对所述任务区域的访问以及禁用外围设备对所述GPU页表区域

的写访问。

[0007] 可选地,在上述方面的一个示例中,经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限可以包括:在安全任务提交期间,将所述任务区域和所述GPU页表区域的访问权限配置为完全可访问;在安全任务执行期间,将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域的访问权限配置为禁止操作系统和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止外围设备通过DMA进行读/写操作;在安全任务切换期间,将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域中用于后续安全任务的内存区域的访问权限配置为禁止操作系统、GPU和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止GPU和外围设备通过DMA进行读/写操作;在安全任务完成期间,将所述任务区域和所述GPU页表区域的访问权限被置为完全可访问。

[0008] 可选地,在上述方面的一个示例中,针对所述任务区域中的数据,在所述安全任务执行期间,所述安全任务的对应内存区域所存储的数据为明文数据,其它内存区域所存储的数据为密文数据,以及在所述安全任务切换期间,所述安全任务的对应内存区域中用于后续安全任务的内存区域所存储的数据为明文数据,其它内存区域所存储的数据为密文数据。

[0009] 可选地,在上述方面的一个示例中,经由DMA传输的数据经过加密并且执行完整性检查。

[0010] 根据本说明书的实施例的另一方面,提供一种用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的方法,所述Arm终端设备的内存空间被划分为可信内存区域、非安全世界内存区域、安全任务内存区域和非安全任务内存区域,所述方法包括:响应于从GPU驱动接收到安全任务执行信号,分别配置所述可信内存区域中存储的非安全二级转换表和安全二级转换表,以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对所述GPU MMIO接口和GPU可信固件的非授权访问;经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限;以及利用所配置的所述安全任务内存区域在安全任务计算期间的访问权限,在GPU执行安全任务计算期间访问所述安全任务内存区域来获取所述安全任务的数据和代码,以供GPU进行GPU可信计算。

[0011] 可选地,在上述方面的一个示例中,所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域,通过配置所述非安全二级转换表和所述安全二级转换表来对所述任务区域中包含不同安全任务处理阶段的数据和代码的内存区域执行页级保护以及监视对所述GPU页表区域的修改请求,并且通过配置所述内存地址空间控制器来管理DMA、GPU和/或外围设备对所述任务区域的访问以及禁用外围设备对所述GPU页表区域的写访问。

[0012] 可选地,在上述方面的一个示例中,经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限包括:在安全任务提交期间,将所述任务区域和所述GPU页表区域的访问权限配置为完全可访问;在安全任务执行期间,将所述GPU页表区域的访问权限配置为写保护,针对所述

任务区域,将所述安全任务的对应内存区域的访问权限配置为禁止操作系统和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止外围设备通过DMA进行读/写操作;在安全任务切换期间,将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域中用于后续安全任务的内存区域的访问权限配置为禁止操作系统、GPU和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止GPU和外围设备通过DMA进行读/写操作;在安全任务完成期间,将所述任务区域和所述GPU页表区域的访问权限被置为完全可访问。

[0013] 可选地,在上述方面的一个示例中,在安全任务提交之前,所述方法还包括:检查GPU任务状态寄存器,以确认所述GPU当前不存在其它正在处理的任务。

[0014] 可选地,在上述方面的一个示例中,所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域。在安全任务提交之前,所述方法还包括:对所述GPU页表区域进行地址映射检查以及对所述任务区域中存储的数据和代码执行完整性检查。

[0015] 可选地,在上述方面的一个示例中,所述方法还包括:在安全任务提交之前,经由通用终端控制器将GPU中断配置为安全状态;以及在安全任务完成后,经由通用终端控制器将GPU中断配置为非安全状态。

[0016] 根据本说明书的实施例的另一方面,提供一种用于构建面向Arm终端设备的GPU可信执行环境的装置,所述Arm终端设备的内存空间被划分为可信内存区域、非安全世界内存区域、安全任务内存区域和非安全任务内存区域,所述装置包括:GPU保护单元,响应于从GPU驱动接收到安全任务执行信号,分别配置所述可信内存区域中存储的非安全二级转换表和安全二级转换表,以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对所述GPU MMIO接口和GPU可信固件的非授权访问;以及安全任务保护单元,经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限,以在GPU执行安全任务计算期间实现针对所述安全任务内存区域所存储的所述安全任务的数据和代码的隔离保护。

[0017] 可选地,在上述方面的一个示例中,所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域。所述安全任务保护单元通过配置所述非安全二级转换表和所述安全二级转换表来对所述任务区域中包含不同安全任务处理阶段的数据和代码的内存区域执行页级保护以及监视对所述GPU页表区域的修改请求,并且通过配置所述内存地址空间控制器来管理DMA、GPU和/或外围设备对所述任务区域的访问以及禁用外围设备对所述GPU页表区域的写访问。

[0018] 可选地,在上述方面的一个示例中,在安全任务提交和安全任务完成期间,所述安全任务保护单元将所述任务区域和所述GPU页表区域的访问权限配置为完全可访问。在安全任务执行期间,所述安全任务保护单元将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域的访问权限配置为禁止操作系统和外围设备通过DMA进行读/写操作,将其它内存区域的访问权限配置为禁止外围设备通过DMA进行读/写操作。在安全任务切换期间,所述安全任务保护单元将所述GPU页表区域的访问权限配置为写保护,针对所述任务区域,将所述安全任务的对应内存区域中用于后续安全任务的内存区域的访问权限配置为禁止操作系统、GPU和外围设备通过DMA进行读/写操作,将

其它内存区域的访问权限配置为禁止GPU和外围设备通过DMA进行读/写操作。

[0019] 可选地，在上述方面的一个示例中，所述安全任务保护单元对经由DMA传输的数据进行加密并且执行完整性检查。

[0020] 根据本说明书的实施例的另一方面，提供一种用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的装置，所述Arm终端设备的内存空间被划分为可信内存区域、非安全世界内存区域、安全任务内存区域和非安全任务内存区域，所述装置包括：GPU保护单元，响应于从GPU驱动接收到安全任务执行信号，分别配置所述可信内存区域中存储的非安全二级转换表和安全二级转换表，以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对所述GPU MMIO接口和GPU可信固件的非授权访问；安全任务保护单元，经由内存地址空间控制器、所述非安全二级转换表和所述安全二级转换表动态配置所述安全任务内存区域在安全任务计算期间的访问权限；以及安全访问单元，利用所配置的所述安全任务内存区域在安全任务计算期间的访问权限，在GPU执行安全任务计算期间访问所述安全任务内存区域来获取所述安全任务的数据和代码，以供GPU进行GPU可信计算。

[0021] 可选地，在上述方面的一个示例中，所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域。所述安全任务保护单元通过配置所述非安全二级转换表和所述安全二级转换表来对所述任务区域中包含不同安全任务处理阶段的数据和代码的内存区域执行页级保护以及监视对所述GPU页表区域的修改请求，并且通过配置所述内存地址空间控制器来管理DMA、GPU和/或外围设备对所述任务区域的访问以及禁用外围设备对所述GPU页表区域的写访问。

[0022] 可选地，在上述方面的一个示例中，在安全任务提交和安全任务完成期间，所述安全任务保护单元将所述任务区域和所述GPU页表区域的访问权限配置为完全可访问。在安全任务执行期间，所述安全任务保护单元将所述GPU页表区域的访问权限配置为写保护，针对所述任务区域，将所述安全任务的对应内存区域的访问权限配置为禁止操作系统和外围设备通过DMA进行读/写操作，将其它内存区域的访问权限配置为禁止外围设备通过DMA进行读/写操作。在安全任务切换期间，所述安全任务保护单元将所述GPU页表区域的访问权限配置为写保护，针对所述任务区域，将所述安全任务的对应内存区域中用于后续安全任务的内存区域的访问权限配置为禁止操作系统、GPU和外围设备通过DMA进行读/写操作，将其它内存区域的访问权限配置为禁止GPU和外围设备通过DMA进行读/写操作。

[0023] 可选地，在上述方面的一个示例中，所述安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域。在安全任务提交之前，所述安全任务保护单元对所述GPU页表区域进行地址映射检查以及对所述任务区域中存储的数据和代码执行完整性检查。

[0024] 可选地，在上述方面的一个示例中，在安全任务提交之前，所述GPU保护单元经由通用终端控制器将GPU中断配置为安全状态，以及在安全任务完成后，所述GPU保护单元经由通用终端控制器将GPU中断配置为非安全状态。

[0025] 根据本说明书的实施例的另一方面，提供一种Arm终端设备，包括：基于Arm架构的GPU可信固件，包括如上所述的用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的装置；以及GPU，在经由GPU可信固件所提供的可信执行环境中，使用安全任务的数据

和代码来进行GPU可信计算。

[0026] 根据本说明书的实施例的另一方面，提供一种用于构建面向Arm终端设备的GPU可信执行环境的装置，包括：至少一个处理器；与所述至少一个处理器耦合的存储器；以及存储在所述存储器中的计算机程序，所述至少一个处理器执行所述计算机程序来实现如上所述的用于构建面向Arm终端设备的GPU可信执行环境的方法。

[0027] 根据本说明书的实施例的另一方面，提供一种用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的装置，包括：至少一个处理器；与所述至少一个处理器耦合的存储器；以及存储在所述存储器中的计算机程序，所述至少一个处理器执行所述计算机程序来实现如上所述的用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的装置的方法。

附图说明

[0028] 通过参照下面的附图，可以实现对于本说明书内容的本质和优点的进一步理解。在附图中，类似组件或特征可以具有相同的附图标记。

[0029] 图1示出了Arm终端设备上的GPU任务执行流程的示例流程图。

[0030] 图2示出了根据本说明书的实施例的Arm终端设备的示例架构图。

[0031] 图3示出了根据本说明书的实施例的用于构建面向Arm终端设备的GPU可信执行环境的方法的示例流程图。

[0032] 图4示出了根据本说明书的实施例的GPU应用程序的任务计算期间安全任务内存区域的访问权限动态变化图。

[0033] 图5示出了根据本说明书的实施例的用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的方法的示例流程图。

[0034] 图6示出了根据本说明书的实施例的用于构建面向Arm终端设备的GPU可信执行环境的装置的示例方框图。

[0035] 图7示出了根据本说明书的实施例的用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的装置的示例方框图。

[0036] 图8示出了根据本说明书的实施例的基于计算机系统实现的可信执行环境构建装置的示例示意图。

[0037] 图9示出了根据本说明书的实施例的基于计算机系统实现的可信计算执行装置的示例示意图。

具体实施方式

[0038] 现在将参考示例实施方式讨论本文描述的主题。应该理解，讨论这些实施方式只是为了使得本领域技术人员能够更好地理解从而实现本文描述的主题，并非是对权利要求书中所阐述的保护范围、适用性或者示例的限制。可以在不脱离本说明书内容的保护范围的情况下，对所讨论的元素的功能和排列进行改变。各个示例可以根据需要，省略、替代或者添加各种过程或组件。例如，所描述的方法可以按照与所描述的顺序不同的顺序来执行，以及各个步骤可以被添加、省略或者组合。另外，相对一些示例所描述的特征在其它例子中也可以进行组合。

[0039] 如本文中使用的,术语“包括”及其变型表示开放的术语,含义是“包括但不限于”。术语“基于”表示“至少部分地基于”。术语“一个实施例”和“一实施例”表示“至少一个实施例”。术语“另一个实施例”表示“至少一个其他实施例”。术语“第一”、“第二”等可以指代不同的或相同的对象。下面可以包括其他的定义,无论是明确的还是隐含的。除非上下文中明确地指明,否则一个术语的定义在整个说明书中是一致的。

[0040] 本说明书中使用的流程图示出了根据本说明书中的一些实施例的系统实现的操作。应该清楚地理解,流程图的操作可以不按顺序实现。相反,操作可以以反转顺序或同时实现。此外,可以向流程图添加一个或多个其他操作。可以从流程图中移除一个或多个操作。

[0041] 为了防止在GPU计算时发生机密数据泄漏,提出了基于可信执行环境(TEE)的GPU可信计算方案。通过使用专用硬件和软件,使得GPU可以在TEE所提供的隔离运行时环境中安全地执行GPU应用程序。例如,可以通过使用具有定制化TEE的改进Intel Software Guard eXtensions(SGX)、Graviton和HETEE,将TEE应用来隔离GPU计算。然而,基于Arm的终端设备通常采用片上系统(SoC),其中,GPU和CPU之间共享统一内存(并且因此具有不可信操作系统),而基于Intel的终端设备上的GPU被自然隔离,并且隔离的GPU具有专用内存,Intel GPU平台和Arm GPU平台之间的这种架构差异,使得上述技术不能应用于Arm终端设备上的GPU。

[0042] 此外,GPU计算过程涉及高度耦合的软件栈(例如,内核层中的GPU驱动和用户层中的闭源用户运行时)。软件栈是共同工作来保证软件正常运行的独立组件集合。软件栈中的组件例如可以包含操作系统、架构层、协议、运行时环境、数据库系统、功能调用,并且以一个在另一个上面的架构组织在一起。软件栈的底层组件会直接与硬件进行交换,并且高层组件为终端用户执行任务或者提供服务。为了提供可信执行环境,需要将软件栈移植到Enclave,Enclave以受保护的机密GPU应用程序的名义执行。然而,这可能增加了系统内的漏洞。一方面,所移植入的软件栈增加了Enclave/TEE的可信代码基。另一方面,这种被移植的软件栈的实现可能存在漏洞,其严重威胁了GPU计算期间的数据安全。

[0043] 而且,基于Intel的设备上的GPU TEE机制需要大量的硬件修改,如果将该GPU TEE机制应用于Arm终端设备,将会导致与现有软件系统之间的兼容性差。Arm终端设备上的安全计算需要将整个GPU驱动移植到可信计算基,并且只关注特定的应用程序(例如,深度学习推理)。

[0044] 为此,提出一种面向Arm终端设备的GPU TEE构建方案。所构建的GPU TEE旨在确保Arm终端设备中的GPU执行安全且隔离的任务计算。利用该GPU TEE构建方案,可以从GPU可信固件中删除易受攻击的安全操作系统和安全应用程序来实现小型GPU可信固件,该小型GPU可信固件可以通过利用内存地址空间控制器、非安全二级转换和安全二级转换来防止来自非安全操作系统/非安全应用程序以及安全操作系统/安全应用程序的攻击,从而在执行GPU可信计算时确保所使用的数据和代码的安全。此外,所实现的小型GPU可信固件可以减少潜在的攻击面,从而降低攻击风险。而且,所构建的GPU TEE既不依赖于特定Arm终端设备的特征,也不需要对GPU芯片或CPU芯片进行硬件修改,从而实现针对Arm终端设备的高兼容性。

[0045] 在描述根据本说明书的实施例的GPU TEE构建方案及GPU可信计算执行方案之前,

针对Arm终端设备所具有的能力进行简要说明。

[0046] Arm终端设备支持Arm TrustZone。Arm TrustZone是一种基于硬件的安全机制,其为Arm终端设备上的数据和代码提供多种隔离保障。TrustZone将GPU应用程序的执行隔离为非安全世界(Normal World)和安全世界(Security World)。不可信应用程序和不可信操作系统(例如,主机操作系统)位于非安全世界中,以及可信应用程序和可信操作系统位于安全世界中。安全世界中的机密计算由TrustZone通过内存中的硬件隔离进行严格保护,并且可以通过几种机制来在非安全世界中调度请求,例如,smc指令。

[0047] 作为TrustZone体系架构一部分的硬件组件可以用于确保非安全世界和安全世界之间的隔离。这种组件的一个示例例如是内存地址空间控制器,比如,TrustZone地址空间控制器(TZASC)。TZASC嵌入在内存总线中,并且位于DRAM和CPU/外围设备之间,用于监控对安全地址空间和非安全地址空间的访问。此外,TZASC将非安全访问标识(NSAID)分配给每个不可信的外围设备。当外围设备需要对地址进行读/写访问时,TZASC查找对应存储区域的相应配置(通常存储在寄存器中)来验证访问有效性。然而,TZASC只支持配置8个存储区域,这限制了内存保护机制的灵活性。

[0048] TrustZone还可以配置通用中断控制器(Generic Interrupt Controller),以响应于设备I/O来隔离安全中断和非安全中断。TrustZone可以创建两组中断,中断组0(仅在安全世界中可访问)和中断组1(在安全世界和非安全世界中都可以访问)。在中断发生时,TrustZone识别该中断及其组号,随后将该中断以其相关安全状态分发到CPU。

[0049] Arm终端设备中定义了两级转换机制(一级转换和二级转换)来将操作系统和应用程序的内存空间映射到物理内存空间。一级转换用于将内核或用户空间的虚拟地址(VA)转换为中间物理地址(IPA),以及二级转换用于将中间物理地址映射到实际物理地址(PA)。然而,由于二级转换通常不适合于多租户管理,从而Arm终端设备通常禁用二级转换。在所提出的GPU TEE中,启用二级转换来实现针对GPU MMIO寄存器和GPU任务存储器的页表级访问控制。

[0050] 自Armv8.4起的Arm终端设备具有安全虚拟化扩展功能。在安全虚拟化扩展中,引入了安全管理程序层(即,S-EL2)来虚拟化整个安全世界(例如,安全存储器和安全外围设备),从而引入安全二级转换机制来管理安全操作系统/安全应用程序所使用的内存。与非安全二级转换相比,安全二级转换拥有几个专用系统寄存器来配置转换页表(VSTTBR_EL2)和转换控制(VSTCR_EL2)的基地址,但是安全二级转换和非安全二级转换共享相同的寄存器(HCR_EL2),以确定它们是否启用。

[0051] 为了在软件级别控制Arm终端设备上的GPU,Arm终端设备提供两个GPU软件栈:(1)用户层中的闭源用户运行时(例如,OpenCL),以及(2)内核层的开源GPU驱动。用户级运行时提供了各种高级API、内置函数等以及特定数据结构,以支持GPU应用程序开发。内核级GPU驱动经由内存映射接口(MMIO)控制内存分配、任务调度和任务提交。

[0052] GPU应用程序由一个或多个GPU任务组成,每个GPU任务可以包含若干GPU线程。图1示出了Arm终端设备上的GPU任务执行流程100的示例流程图。

[0053] 如图1所示,在GPU应用程序执行期间,针对待执行的GPU任务,在110,GPU软件栈为GPU任务中的基本组件分配GPU内存,并构建对应的GPU页表。GPU任务的基本组件例如可以包括GPU数据缓存、GPU代码缓存和元数据(meta)等,元数据用于指示GPU数据/代码缓存在

内存中的位置。

[0054] 在120,操作系统通过直接内存访问(DMA)控制器将数据加载到所分配的对应内存,并且在130,GPU软件栈将程序代码的二进制代码加载到所分配的对应内存中。

[0055] 在140,操作系统通过配置GPU MMIO寄存器来向GPU发送任务提交命令,即,向GPU提交待计算的GPU任务。

[0056] 在接收到任务提交命令后,在150,GPU基于所加载的代码和数据来执行任务计算,并将任务计算结果存储到特定内存。一旦GPU任务计算完成,GPU发送硬件中断以通知GPU软件栈中的中断处理程序。对于多任务的GPU应用程序,GPU软件栈重复执行任务所需数据和代码加载,任务提交和GPU任务计算。在处理完GPU应用程序的所有任务后,操作系统通过DMA控制器访问或导出GPU应用程序的程序执行结果。

[0057] 下面将参照附图描述根据本说明书的实施例的可信执行环境构建方法及装置以及可选计算执行方法及装置。

[0058] 图2示出了根据本说明书的实施例的Arm终端设备200的示例架构图。

[0059] 如图2所示,Arm终端设备200包括软件组件和硬件组件。软件组件被分层为用户层,内核层,虚拟化层以及安全监视层EL3。用户层,内核层和虚拟化层被隔离为非安全世界中的用户层EL0、内核层EL1和虚拟化层EL2以及安全世界中的用户层S-EL0、内核层S-EL1和虚拟化层S-EL2。GPU应用程序和用户GPU运行时位于用户层EL0,以及安全应用程序位于用户层S-EL0。非安全操作系统(非安全OS)位于内核层EL1,以及安全操作系统(安全OS)位于内核层S-EL1。非安全OS可以包括GPU驱动、DMA控制器和中断处理器。GPU驱动例如可以包括任务调度器和内存管理器。中断处理器例如可以是通用中断控制器(Generic Interrupt Controller,GIC)。

[0060] 在图2所示的Arm终端设备中,安全监视层EL3中部署基于Arm架构的GPU可信固件。GPU可信固件用于为机密GPU应用程序构建可信执行环境,并且使用所构建的可信执行环境来执行安全任务的安全隔离计算。该GPU可信固件可以再用用户GPU运行时以及非安全操作系统(EL1)中的GPU驱动,并且删除易受攻击的安全操作系统和安全应用程序,从而实现GPU可信固件的小型化,并且防止来自安全操作系统/安全应用程序的攻击以确保GPU可信计算时的数据和代码安全。在该Arm终端设备中,在虚拟化层EL2和S-EL2中未部署非安全世界和安全世界中的虚拟机监控程序,并且无需对安全应用程序和安全操作系统进行修改。

[0061] 如图2所示,硬件组件包括CPU设备和GPU设备,CPU设备和GPU设备分别设置对应的内存管理单元(Memory Management Unit)。硬件组件还可以包括内存地址空间控制器,例如,TZSAC。此外,Arm终端系统的内存空间被划分为四个内存区域:可信内存区域、非安全世界内存区域、安全任务内存区域和非安全任务内存区域。这里,每个内存区域可以被实现为一个存储器,例如,RAM。非安全世界内存区域和非安全任务内存区域属于不可信区域,分别用于内核任务以及非安全任务。可信内存区域和安全任务内存区域属于可信区域。可信内存区域被预留给GPU可信固件中的GPU可信运行时、非安全二级转换表和安全二级转换表。安全任务内存区域被预留给机密GPU应用程序来动态请求内存并创建GPU页表映射。

[0062] GPU可信固件(GPU可信运行时)利用内存管理单元和专门配置的TZASC来保护可信内存区域和安全任务内存区域。在内存管理单元中,GPU可信固件执行非安全二级转换和安全二级转换,以控制从非安全操作系统到GPU MMIO接口和两个可信内存区域的访问,以及

从安全操作系统到GPU MMIO接口、GPU可信固件(GPU可信运行时)和两个可信内存区域的访问。此外,硬件组件可以使用Arm终端设备中的现有可软件配置组件,以提升GPU可信固件的设备兼容性。此外,GPU可信固件利用TZASC控制GPU和外围设备对两个可信区域的访问。要说明的是,基于Arm架构的GPU可信固件和GPU可以构成基于Arm架构的GPU可信计算架构。

[0063] 图3示出了根据本说明书的实施例的用于构建面向Arm终端设备的GPU可信执行环境的方法300的示例流程图。

[0064] 如图3所示,在执行机密GPU应用程序时,在310,GPU可信固件从GPU驱动接收到安全任务执行信号。例如,GPU可信固件可以从GPU驱动接收到SMC调用。在本说明书所提供的方案中,为了提供安全任务的独占性执行,在GPU驱动的任务调度器中为安全任务设置专用的调度规则。一旦安全任务准备好执行,任何其它任务的计算都将被迫重安排,并等待该安全任务完成。对于正在运行的任务,GPU驱动评估GPU寄存器的内容,以确定是否有其它任务正在执行。一旦确定GPU没有执行其它任务,GPU驱动使用专用SMC调用来触发GPU可信执行环境的构建和GPU可信计算。

[0065] 响应于接收到安全任务执行信号,在320,GPU可信固件分别配置可信内存区域中存储的非安全二级转换表和安全二级转换表,以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对GPU MMIO接口和GPU可信固件的非授权访问。

[0066] 在一些实施例中,可以通过配置非安全二级转换表或安全二级转换表中的地址映射,无效非安全世界组件或安全世界组件对GPU MMIO接口、GPU可信固件(例如,GPU可信运行时)以及TZASC的地址应用,从而无效非安全世界组件或安全世界组件对GPU MMIO接口、GPU可信固件(例如,GPU可信运行时)以及TZASC的访问。此外,还可以通过配置非安全二级转换表或安全二级转换表中的地址映射,无效非安全世界组件或安全世界组件对安全任务内存区域的地址映射,由此无效非安全世界组件或安全世界组件对安全任务内存区域的访问。

[0067] 一旦接收到smc调用,GPU可信固件可以配置非安全二级转换表的转换表条目,以防止对GPU MMIO接口的任何未授权访问。具体地,将对应的非安全二级PTE的最后一位设置为0,以无效到GPU MMIO接口的映射,然后无效每个CPU内核的TLB条目。对于安全二级转换,还将相应的PTE配置为0以无效来自安全操作系统和安全应用程序的访问。通过上述操作,试图通过GPU MMIO接口访问GPU寄存器的攻击者将会发生转换失败。此外,为了切换到安全执行,GPU可信固件利用GPU驱动来设置控制寄存器和关键状态寄存器,然后安全地验证关键寄存器。

[0068] 利用上述配置过程,可以锁定安全任务计算期间对GPU MMIO接口和GPU可信固件的授权访问。在接收到来自非安全操作系统/非安全应用程序或安全操作系统/安全应用的未授权访问时,生成页面错误异常,从而可以捕获任何针对GPU MMIO接口和GPU可信固件的恶意操作,例如,篡改GPU寄存器、提交恶意任务、篡改GPU可信运行时等。

[0069] 为了在GPU可信执行环境构建时使用安全二级转换表,需要为安全二级转换表预留内存区域,并为其分配可支持配置的TZASC区域来实现安全二级转换表的隔离保护。具体地,利用TZASC来无效到安全二级转换表的物理地址的映射。如果攻击者打算通过修改安全二级转换表来绕过,则会遍历安全二级转换表来获得无效条目,由此可以通过转换错误来

终止该修改。

[0070] 在对安全二级转换表进行保护后，利用安全二级转换表，无效对GPU MMIO接口、GPU可信固件(GPU可信执行运行时)以及例如TZASC的安全组件的访问。因此，攻击者无法通过篡改GPU可信执行运行时的配置来泄漏敏感数据。此外，通过利用非安全/安全二级转换表来无效非安全世界组件或安全世界组件对整个安全任务内存区域的地址映射，可以实现针对安全任务内存区域的代码、数据和页表条目的保护。

[0071] 在如上完成GPU MMIO接口和GPU可信固件的锁定后，在330，经由内存地址空间控制器、非安全二级转换表和安全二级转换表动态配置安全任务内存区域在安全任务计算期间的访问权限，以在GPU执行安全任务计算期间实现针对安全任务内存区域所存储的所述安全任务的数据和代码的隔离保护。

[0072] 在一些实施例中，安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域。在这种情况下，可以通过组合非安全二级转换表、安全二级转换表和TZASC，提供动态且细粒度的内存保护机制。在该内存保护机制中，通过配置非安全二级转换表和安全二级转换表，对任务区域中包含不同安全任务处理阶段的数据和代码的内存区域执行页级保护并且监视对GPU页表区域的修改请求。通过配置内存地址空间控制器，管理DMA、GPU和/或外围设备对任务区域的访问并且禁用外围设备对GPU页表区域的写访问。

[0073] 在一些实施例中，针对GPU页表，可以通过禁用TZASC NSAIDW寄存器中除了位CPU(AP)之外的大多数位，拒绝所有外围设备和DMA对GPU页表的写操作。对于来自不可信操作系统的写操作，通过异常来监控针对GPU页面的修改，并验证该写操作在异常处理程序中是否非法。除了保护GPU页表之外，还检查每个安全任务的GPU页表基址寄存器AS_TRANSTAB。针对任务内存区域，使用TZASC来配置NSAIDW和NSAIDR寄存器中的对应位，由此管理DMA、GPU和外围设备。此外，通过利用TLB无效来动态更改非安全二级映射，随机限制来自不可信操作系统的数据和代码访问。此外，通过触发非安全二级转换错误，禁止对代码和数据的任何非法读取或写入访问。针对安全二级转换表，无效对整个安全任务内存区域的访问，包括任务内存区域和相应的GPU页表区域。

[0074] 图4示出了根据本说明书的实施例的GPU应用程序的任务计算期间安全任务内存区域的访问权限动态变化图。

[0075] 在图4的示例中，安全任务内存区域被细分为GPU页表区域(PET)和任务区域(例如，缓存1、缓存2等)。如图4所示，在机密GPU应用程序执行期间，GPU页表区域和任务区域的访问权限被分类为六种类型：

[0076] • 完全可访问(Full Accessible)：允许任何读/写操作；

[0077] • 写保护(Write Protected)：允许来自任何组件的读操作，但监视写操作；

[0078] • 禁止DMA(DMA Prohibited)：禁止外围设备通过DMA进行读/写操作；

[0079] • 禁止OS-DMA(OS-DMA Prohibited)：禁止OS和外围设备通过DMA进行读/写操作；

[0080] • 禁止GPU-DMA(GPU-DMA Prohibited)：禁止GPU和外围设备通过DMA进行读/写操作；和

[0081] • 禁止OS-GPU-DMA(OS-GPU-DMA Prohibited)：禁止OS、GPU和外围设备通过DMA进行读/写操作。

[0082] 在安全任务提交期间，任务区域和GPU页表区域的访问权限都被配置为完全可访

问。在安全任务执行期间,GPU页表区域的访问权限被配置为写保护。针对任务区域,当前执行的安全任务的对应内存区域的访问权限被配置为禁止操作系统和外围设备通过DMA进行读/写操作,而其它内存区域的访问权限被配置为禁止外围设备通过DMA进行读/写操作。

[0083] 在安全任务切换期间,GPU页表区域的访问权限被配置为写保护。针对任务区域,当前执行的安全任务的对应内存区域中用于后续安全任务的内存区域的访问权限被配置为禁止操作系统、GPU和外围设备通过DMA进行读/写操作,而其它内存区域的访问权限被配置为禁止GPU和外围设备通过DMA进行读/写操作。

[0084] 在安全任务完成期间,将所述任务区域和所述GPU页表区域的访问权限被置为完全可访问。

[0085] 如图4所示,机密GPU应用程序包括两个GPU任务,即,任务1和任务2。在任务1中,缓存1和缓存2用于输入,以及缓存2用于输出,并且包括代码段C1。在任务2中,缓存2和缓存3用于输入,以及缓存4用于输出,并且包括代码段C2。

[0086] 在安全任务提交阶段,安全任务内存区域中的所有内存区域(GPU页表区域、缓存1到缓存4、代码区域C1和C2)的初始访问权限都被配置为“Full Accessible”,以允许经由GPU软件栈准备用于提交的GPU应用程序。在任务1执行期间,GPU页表区域被配置为“Write Protected”,以避免机密数据的潜在泄露。GPU可信固件中的GPU可信运行时可以捕获对GPU页表的修改以及自检任何恶意内存映射(例如,双映射和映射到不可信区域)。由于GPU页表最初由GPU驱动准备,GPU可信固件可以在运行第一个安全任务之前,验证整个GPU页表。

[0087] 在任务1执行阶段,缓存1、缓存2和代码区域C1(即,当前正在执行任务的数据和代码内存区域)的访问权限被配置为“OS-DMA Prohibited”,以保护后续加密以及代码和数据区域的完整性验证的安全。缓存3、缓存4和代码区域C2(即,整个任务区域中除了当前正在执行任务的数据和代码内存区域之外的内存区域)的访问权限被配置为“DMA Prohibited”。

[0088] 在任务1切换阶段,缓存2(即,用于后续任务2的内存区域)的访问权限被配置为“OS-GPU-DMA Prohibited”。缓存1、缓存3、缓存4、代码区域C1和C2(即,整个任务区域中除了用于后续任务2的内存区域之外的内存区域)的访问权限被配置为“GPU-DMA Prohibited”。

[0089] 在任务2执行阶段,缓存2、缓存3、缓存4和代码区域C2(即,当前正在执行任务的数据和代码内存区域)的访问权限被配置为“OS-DMA Prohibited”,以保护后续加密以及代码和数据区域的完整性验证的安全。缓存1和代码区域C1(即,整个任务区域中除了当前正在执行任务的数据和代码内存区域之外的内存区域)的访问权限被配置为“DMA Prohibited”。

[0090] 在任务2切换阶段,由于不存在用于后续任务2的内存区域,从而缓存1、缓存2、缓存3、缓存4、代码区域C1和C2的访问权限都被配置为“GPU-DMA Prohibited”。

[0091] 在GPU应用程序的所有任务完成阶段,安全任务内存区域中的所有内存区域(GPU页表区域、缓存1到缓存4、代码区域C1和C2)的访问权限都被配置为“Full Accessible”。

[0092] 此外,GPU缓存(例如,任务1中的缓存1和缓存2,以及任务2中的缓存2、缓存3和缓存4)中的数据被默认为是密文数据。在安全任务执行期间,安全任务的对应内存区域所存储的数据为明文数据,其它内存区域所存储的数据为密文数据。在安全任务切换期间,安全

任务的对应内存区域中用于后续安全任务的内存区域所存储的数据为明文数据，其它内存区域所存储的数据为密文数据。

[0093] 例如，对于安全任务1，在任务1执行期间，缓存1和缓存2所存储的数据为明文数据，缓存3和缓存4所存储的数据为密文数据。在任务1切换阶段，用于后续安全任务的缓存2保留明文数据，缓存1、缓存3和缓存4的数据为密文数据。在这种情况下，将明文数据存储器(缓存2)配置为“OS-GPU-DMA Prohibited”，并且将除了明文数据存储器(缓存2)之外的所有存储器都配置为“GPU-DMAProhibited”，直到提交下一安全任务。在所有任务完成后，所有缓存中的数据都被加密，并且将整个安全任务内存区域配置为“Full Accessible”，以允许用户加载结果。

[0094] 此外，出于安全目的，GPU可信固件禁止不同的机密GPU应用程序中的安全任务共享安全任务内存区域。在上一机密GPU应用程序完成所有安全任务并且安全终止之前，不能启动其他机密GPU应用程序中的任何任务。按照这种存储器组织方式，可以为安全任务所使用的任何机密数据提供高效的隔离保证。

[0095] 此外，可以通过提供安全数据传输路径来避免数据泄露。在该安全数据传输路径中，经由DMA传输的任何机密数据都需要进行加密并且执行完整性检查。例如，可以使用基于哈希的消息认证码(HMAC)来进行数据完整性检查。GPU可信固件执行安全自检，以利用共享密钥对机密数据进行解密或加密，并根据明文数据或明文代码计算每个HMAC。由于秘密密钥、中间数据或明文数据以及对应任务页表的存储器受非安全二级转换和TZASC机制保护，针对所计算的哈希值的TOCTTOU攻击不可行。接下来，响应于验证成功而继续完成任务提交，或响应于验证失败而中止提交任务。在完成安全任务后，GPU可信固件恢复执行环境。在恢复到非安全状态之前，需要对执行结果进行加密和HASH处理。在这种情况下，仅在安全任务执行期间存在明文数据。

[0096] 利用上述GPU可信执行环境构建过程，可以从GPU可信固件中删除易受攻击的安全操作系统和安全应用程序来实现小型GPU可信固件，该小型GPU可信固件可以通过利用内存地址空间控制器、非安全二级转换和安全二级转换来防止来自非安全操作系统/非安全应用程序以及安全操作系统/安全应用程序的攻击，从而在执行GPU可信计算时确保所使用的数据和代码的安全。

[0097] 在如上完成GPU可信执行环境构建后，可以在所构建的可信执行环境中，执行针对安全任务的GPU可信计算。

[0098] 图5示出了根据本说明书的实施例的用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的方法500的示例流程图。图5中例示的过程由GPU可信固件执行。

[0099] 如图5所示，在执行机密GPU应用程序时，在510，GPU可信固件从GPU驱动接收到安全任务执行信号。例如，GPU可信固件可以从GPU驱动接收到SMC调用。安全任务执行信号由GPU驱动发出。GPU驱动可以读取当前正在运行的GPU应用程序队列，并等待GPU上正在运行的GPU应用程序运行结束。当获取到GPU上正在运行的GPU应用程序运行结束时，将除当前机密GPU应用程序中所包含的GPU任务之外的GPU应用程序全部置于等待队列，并产生安全任务执行信号。

[0100] 响应于接收到安全任务执行信号，在520，GPU可信固件分别配置可信内存区域中存储的非安全二级转换表和安全二级转换表，以禁用非安全操作系统/非安全应用程序对

GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对GPU MMIO接口和GPU可信固件的非授权访问。

[0101] 在如上完成GPU MMIO接口和GPU可信固件的锁定后,在530,经由内存地址空间控制器、非安全二级转换表和安全二级转换表动态配置安全任务内存区域在安全任务计算期间的访问权限,以在GPU执行安全任务计算期间实现针对安全任务内存区域所存储的所述安全任务的数据和代码的隔离保护。

[0102] 在540,利用所配置的安全任务内存区域在安全任务计算期间的访问权限,在GPU执行安全任务计算期间访问安全任务内存区域来获取安全任务的数据和代码。在获取到安全任务的数据和代码后,GPU使用所获取的数据和代码来执行安全任务的可信计算,并将所计算出的任务计算结果存储到特定内存,以供输出给用户端。在一些实施例中,在完成GPU任务可信计算后,GPU可以与用户端进行通信交互获取加密算法,并根据加密算法对任务计算结果进行加密处理,生成加密后的任务计算结果。通过对任务计算结果进行加密处理,可以进一步提高任务计算结果在传输过程中的安全性。

[0103] 利用上述GPU可信计算执行过程,可以通过利用内存地址空间控制器、非安全二级转换和安全二级转换来构建GPU可信执行环境,并在所构建的GPU可信执行环境下获取安全任务的数据和代码来执行GPU可信计算,由此在执行GPU可信计算时确保所使用的数据和代码的安全。

[0104] 可选地,在一些实施例中,在安全任务提交之前,GPU可信固件还可以检查GPU任务状态寄存器,以确认GPU当前是否存在其它正在处理的任务(例如,隐藏任务)。如果GPU当前存在其它正在处理的任务,则认为当前提交的安全任务是不可信安全任务,拒绝向GPU提交该安全任务。如果GPU当前不存在其它正在处理的任务,则认为当前提交的安全任务是可信安全任务,允许向GPU提交该安全任务。此外,可选地,在一些实施例中,GPU可信固件还可以检查例如页表基址寄存器和GPU任务代码寄存器等的关键GPU寄存器。利用上述检查过程,可以检测攻击者是否在GPU驱动配置GPU执行环境时恶意修改GPU执行环境,或者隐藏GPU任务

[0105] 可选地,在一些实施例中,安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域。在安全任务提交之前,还可以对GPU页表区域进行地址映射检查以及对任务区域中存储的数据和代码执行完整性检查。通过对GPU页表区域进行地址映射检查,可以防止攻击者将敏感的GPU缓存地址映射到未控存储器中。在当前安全任务执行之前,GPU页表区域被GPU可信固件锁定并检查,并且在完成GPU应用的最后一个安全任务后解锁。在一些实施例中,可以采用哈希值检查方法来检查GPU任务所使用的数据和代码的完整性。利用数据和代码完整性检查,可以提高GPU任务执行过程的安全性,由此避免因为传输错误或非法攻击而导致的由于安全任务的内容异常带来的系统风险。

[0106] 可选地,在一些实施例中,为了能够处理GPU可信固件所接收到的GPU中断,还可以在安全任务提交之前,经由通用终端控制器将GPU中断配置为安全状态。随后,GPU可信固件将准备好的安全任务写入任务提交寄存器并提交给GPU。GPU在接收到任务提交命令后,使用对应的数据和代码来执行GPU可信计算。在提交安全任务后,GPU可信固件返回GPU驱动并释放CPU。因此,在GPU可信计算期间,GPU可信固件不会阻塞CPU内核。此外,为了实现安全任务同步,GPU可信固件要求GPU驱动将GPU任务调度为提交,但是它不支持并发提交。此外,在

GPU可信计算期间,GPU可信固件不会阻止不与GPU交互的其他smc调用。

[0107] 在GPU可信计算完成(即,安全任务完成)后,GPU向GPU可信固件发送GPU中断(该GPU中断被预先配置为安全),以通知GPU计算完成。因此,GPU可信固件中的GPU执行运行时拦截GPU中断,并恢复GPU MMIO和GPU内存的访问权限。此外,GPU可信固件还通过配置GIC,将GPU中断配置为非安全状态,以允许GPU驱动处理中断。在恢复MMIO的访问权限和中断状态后,GPU被允许处理新任务。

[0108] 在一些实施例中,在GPU可信固件从GPU获取加密后的GPU应用程序执行结果后,GPU可信固件根据加密后的GPU应用程序执行结果,配置安全空间与非安全空间的地址映射关系,并根据所配置的地址映射关系,调取安全空间内的加密后的GPU应用程序执行结果至非安全空间,以使GPU驱动获取加密后的GPU应用程序执行结果,并将加密处理后的GPU应用程序执行结果发送至用户端。

[0109] 利用上述处理过程,通过使用基于ARM架构的GPU可信固件配置安全空间与非安全空间之间的地址映射关系,使得加密后的GPU应用程序执行结果可以通过重新配置后的地址映射关系从安全空间释放到非安全空间。

[0110] 在一些实施例中,在GPU可信固件从GPU接收到加密后的GPU应用程序执行结果后,可以通过配置内存地址空间控制器和非安全/安全二级转换表,对所配置的安全空间地址再次配置其地址映射,由此将加密的GPU应用程序执行结果从安全空间释放至非安全空间,以实现GPU驱动读取加密后的GPU应用程序执行结果并转发至用户端。

[0111] 在一些实施例中,在安全任务执行期间,安全任务计算所使用的数据可以通过与用户端进行通信交互获取到的解密算法和密钥文件,对密文数据和进行解码处理来获取明文数据。

[0112] 如上参照图1到图5描述了根据本说明书实施例的可信执行环境构建方法和可信计算执行方法。

[0113] 图6示出了根据本说明书的实施例的用于构建面向Arm终端设备的GPU可信执行环境的装置(下文称为可信执行环境构建装置)600的示例方框图。如图6所示,可信执行环境构建装置600包括GPU保护单元610和安全任务保护单元620。

[0114] GPU保护单元610被配置为响应于从GPU驱动接收到安全任务执行信号,分别配置可信内存区域中存储的非安全二级转换表和安全二级转换表,以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对GPU MMIO接口和GPU可信固件的非授权访问。GPU保护单元610的操作可以参考上面参照图3的320描述的操作。

[0115] 安全任务保护单元620被配置为经由内存地址空间控制器、非安全二级转换表和安全二级转换表动态配置安全任务内存区域在安全任务计算期间的访问权限,以在GPU执行安全任务计算期间实现针对安全任务内存区域所存储的安全任务的数据和代码的隔离保护。安全任务保护单元620的操作可以参考上面参照图3的330描述的操作。

[0116] 在一些实施例中,安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域。安全任务保护单元620通过配置非安全二级转换表和安全二级转换表来对所述任务区域中包含不同安全任务处理阶段的数据和代码的内存区域执行页级保护以及监视对GPU页表区域的修改请求,并且通过配置内存地址空间控制器来管理DMA、GPU和/或外

围设备对任务区域的访问以及禁用外围设备对GPU页表区域的写访问。

[0117] 在一些实施例中,在安全任务提交和安全任务完成期间,安全任务保护单元将任务区域和GPU页表区域的访问权限配置为完全可访问。在安全任务执行期间,安全任务保护单元将GPU页表区域的访问权限配置为写保护,并且针对任务区域,将安全任务的对应内存区域的访问权限配置为禁止操作系统和外围设备通过DMA进行读/写操作,以及将其它内存区域的访问权限配置为禁止外围设备通过DMA进行读/写操作。

[0118] 在安全任务切换期间,安全任务保护单元将GPU页表区域的访问权限配置为写保护,并且针对任务区域,将安全任务的对应内存区域中用于后续安全任务的内存区域的访问权限配置为禁止操作系统、GPU和外围设备通过DMA进行读/写操作,以及将其它内存区域的访问权限配置为禁止GPU和外围设备通过DMA进行读/写操作。

[0119] 在一些实施例中,安全任务保护单元还可以对经由DMA传输的数据进行加密并且执行完整性检查。

[0120] 图7示出了根据本说明书的实施例的用于在面向Arm终端设备的GPU可信执行环境中执行GPU可信计算的装置(下文中称为可行计算执行装置)700的示例方框图。如图7所示,可行计算执行装置700包括GPU保护单元710、安全任务保护单元720和安全访问单元730。

[0121] GPU保护单元710被配置为响应于从GPU驱动接收到安全任务执行信号,分别配置可信内存区域中存储的非安全二级转换表和安全二级转换表,以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对所述GPU MMIO接口和GPU可信固件的非授权访问。GPU保护单元710的操作可以参考上面参照图5的520描述的操作。

[0122] 安全任务保护单元720被配置为经由内存地址空间控制器、非安全二级转换表和安全二级转换表动态配置安全任务内存区域在安全任务计算期间的访问权限。安全任务保护单元720的操作可以参考上面参照图5的530描述的操作。

[0123] 安全访问单元730被配置为利用所配置的安全任务内存区域在安全任务计算期间的访问权限,在GPU执行安全任务计算期间访问安全任务内存区域来获取安全任务的数据和代码,以供GPU进行GPU可信计算。安全访问单元730的操作可以参考上面参照图5的540描述的操作。

[0124] 在一些实施例中,安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域。安全任务保护单元通过配置非安全二级转换表和安全二级转换表来对任务区域中包含不同安全任务处理阶段的数据和代码的内存区域执行页级保护以及监视对GPU页表区域的修改请求,并且通过配置内存地址空间控制器来管理DMA、GPU和/或外围设备对任务区域的访问以及禁用外围设备对GPU页表区域的写访问。

[0125] 在一些实施例中,在安全任务提交和安全任务完成期间,安全任务保护单元将任务区域和所述GPU页表区域的访问权限配置为完全可访问。在安全任务执行期间,安全任务保护单元将GPU页表区域的访问权限配置为写保护,并且针对任务区域,将安全任务的对应内存区域的访问权限配置为禁止操作系统和外围设备通过DMA进行读/写操作,以及将其它内存区域的访问权限配置为禁止外围设备通过DMA进行读/写操作。在安全任务切换期间,安全任务保护单元将所述GPU页表区域的访问权限配置为写保护,并且针对任务区域,将安全任务的对应内存区域中用于后续安全任务的内存区域的访问权限配置为禁止操作系统、

GPU和外围设备通过DMA进行读/写操作,以及将其它内存区域的访问权限配置为禁止GPU和外围设备通过DMA进行读/写操作。

[0126] 在一些实施例中,安全任务内存区域被显式划分为两个物理上连续的任务区域和GPU页表区域。在安全任务提交之前,安全任务保护单元对GPU页表区域进行地址映射检查以及对任务区域中存储的数据和代码执行完整性检查。在一些实施例中,在安全任务提交之前,GPU保护单元经由通用终端控制器将GPU中断配置为安全状态,以及在安全任务完成后,GPU保护单元经由通用终端控制器将GPU中断配置为非安全状态。

[0127] 如上参照图1到图7,对根据本说明书实施例的可信执行环境构建方法及可信执行环境构建装置、可信计算执行方法及可信计算执行装置、对象推荐方法及对象推荐装置进行了描述。上面的模型训练装置、交互转化率预测装置和对象推荐装置可以采用硬件实现,也可以采用软件或者硬件和软件的组合来实现。

[0128] 图8示出了根据本说明书的实施例的基于计算机系统实现的可信执行环境构建装置800的示例示意图。如图8所示,可信执行环境构建装置800可以包括至少一个处理器810、存储器(例如,非易失性存储器)820、内存830和通信接口840,并且至少一个处理器810、存储器820、内存830和通信接口840经由总线860连接在一起。至少一个处理器810执行在存储器中存储或编码的至少一个计算机可读指令(即,上述以软件形式实现的元素)。

[0129] 在一个实施例中,在存储器中存储计算机可执行指令,其当执行时使得至少一个处理器810:响应于从GPU驱动接收到安全任务执行信号,分别配置可信内存区域中存储的非安全二级转换表和安全二级转换表,以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对GPU MMIO接口和GPU可信固件的非授权访问;以及经由内存地址空间控制器、非安全二级转换表和安全二级转换表动态配置安全任务内存区域在安全任务计算期间的访问权限,以在GPU执行安全任务计算期间实现针对述安全任务内存区域所存储的安全任务的数据和代码的隔离保护。

[0130] 应该理解,在存储器中存储的计算机可执行指令当执行时使得至少一个处理器810进行本说明书的各个实施例中以上结合图1-图4和图6描述的各种操作和功能。

[0131] 图9示出了根据本说明书的实施例的基于计算机系统实现的可信计算执行装置900的示例示意图。如图9所示,可信计算执行装置900可以包括至少一个处理器910、存储器(例如,非易失性存储器)920、内存930和通信接口940,并且至少一个处理器910、存储器920、内存930和通信接口940经由总线960连接在一起。至少一个处理器910执行在存储器中存储或编码的至少一个计算机可读指令(即,上述以软件形式实现的元素)。

[0132] 在一个实施例中,在存储器中存储计算机可执行指令,其当执行时使得至少一个处理器910:响应于从GPU驱动接收到安全任务执行信号,分别配置可信内存区域中存储的非安全二级转换表和安全二级转换表,以禁用非安全操作系统/非安全应用程序对GPU MMIO接口的非授权访问以及安全操作系统/安全应用程序对GPU MMIO接口和GPU可信固件的非授权访问;经由内存地址空间控制器、非安全二级转换表和安全二级转换表动态配置安全任务内存区域在安全任务计算期间的访问权限;以及利用所配置的安全任务内存区域在安全任务计算期间的访问权限,在GPU执行安全任务计算期间访问安全任务内存区域来获取安全任务的数据和代码,以供GPU进行GPU可信计算。

[0133] 应该理解,在存储器中存储的计算机可执行指令当执行时使得至少一个处理器

910进行本说明书的各个实施例中以上结合图5和图7描述的各种操作和功能。

[0134] 根据一个实施例,提供了一种比如机器可读介质(例如,非暂时性机器可读介质)的程序产品。机器可读介质可以具有指令(即,上述以软件形式实现的元素),该指令当被机器执行时,使得机器执行本说明书的各个实施例中以上结合图1-图6描述的各种操作和功能。具体地,可以提供配有可读存储介质的系统或者装置,在该可读存储介质上存储着实现上述实施例中任一实施例的功能的软件程序代码,且使该系统或者装置的计算机或处理器读出并执行存储在该可读存储介质中的指令。

[0135] 在这种情况下,从可读介质读取的程序代码本身可实现上述实施例中任何一项实施例的功能,因此机器可读代码和存储机器可读代码的可读存储介质构成了本发明的一部分。

[0136] 可读存储介质的实施例包括软盘、硬盘、磁光盘、光盘(如CD-ROM、CD-R、CD-RW、DVD-ROM、DVD-RAM、DVD-RW、DVD-RW)、磁带、非易失性存储卡和ROM。可选择地,可以由通信网络从服务器计算机上或云上下载程序代码。

[0137] 根据一个实施例,提供一种计算机程序产品,该计算机程序产品包括计算机程序,该计算机程序当被处理器执行时,使得处理器执行本说明书的各个实施例中以上结合图1-图6描述的各种操作和功能。

[0138] 本领域技术人员应当理解,上面公开的各个实施例可以在不偏离发明实质的情况下做出各种变形和修改。因此,本发明的保护范围应当由所附的权利要求书来限定。

[0139] 需要说明的是,上述各流程和各系统结构图中不是所有的步骤和单元都是必须的,可以根据实际的需要忽略某些步骤或单元。各步骤的执行顺序不是固定的,可以根据需要进行确定。上述各实施例中描述的装置结构可以是物理结构,也可以是逻辑结构,即,有些单元可能由同一物理实体实现,或者,有些单元可能分由多个物理实体实现,或者,可以由多个独立设备中的某些部件共同实现。

[0140] 以上各实施例中,硬件单元或模块可以通过机械方式或电气方式实现。例如,一个硬件单元、模块或处理器可以包括永久性专用的电路或逻辑(如专门的处理器,FPGA或ASIC)来完成相应操作。硬件单元或处理器还可以包括可编程逻辑或电路(如通用处理器或其它可编程处理器),可以由软件进行临时的设置以完成相应操作。具体的实现方式(机械方式、或专用的永久性电路、或者临时设置的电路)可以基于成本和时间上的考虑来确定。

[0141] 上面结合附图阐述的具体实施方式描述了示例性实施例,但并不表示可以实现的或者落入权利要求书的保护范围的所有实施例。在整个本说明书中使用的术语“示例性”意味着“用作示例、实例或例示”,并不意味着比其它实施例“优选”或“具有优势”。出于提供对所描述技术的理解的目的,具体实施方式包括具体细节。然而,可以在没有这些具体细节的情况下实施这些技术。在一些实例中,为了避免对所描述的实施例的概念造成难以理解,公知的结构和装置以框图形式示出。

[0142] 本公开内容的上述描述被提供来使得本领域任何普通技术人员能够实现或者使用本公开内容。对于本领域普通技术人员来说,对本公开内容进行的各种修改是显而易见的,并且,也可以在不脱离本公开内容的保护范围的情况下,将本文所定义的一般性原理应用于其它变型。因此,本公开内容并不限于本文所描述的示例和设计,而是与符合本文公开的原理和新颖性特征的最广范围相一致。

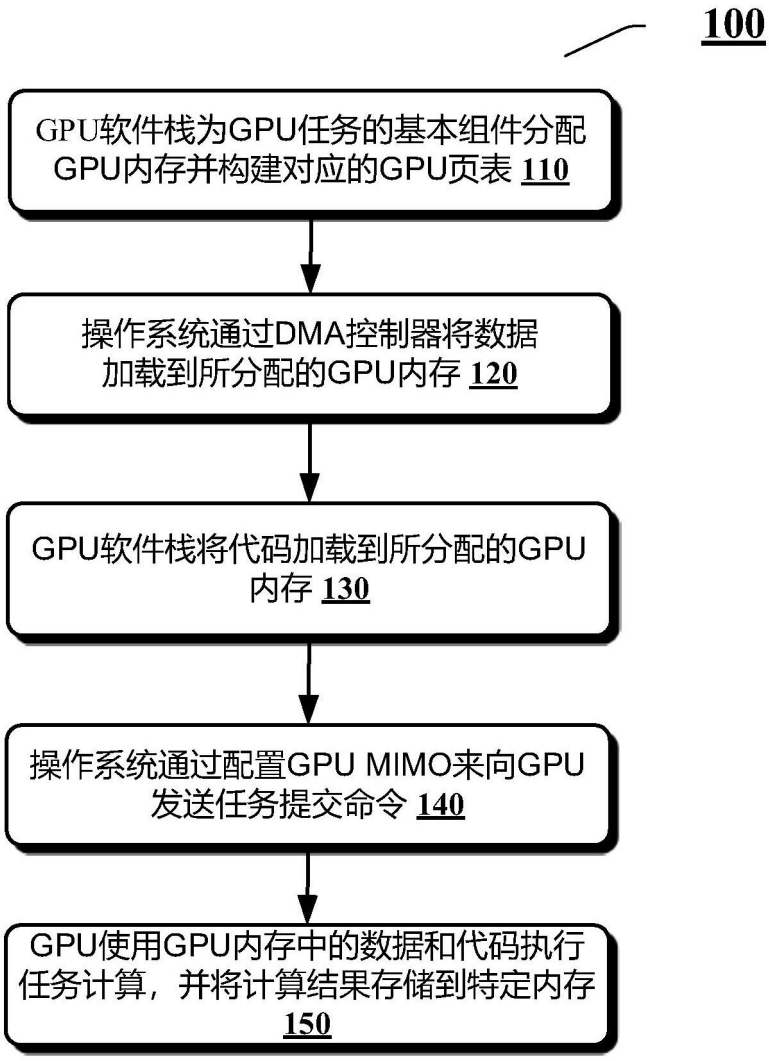

100
GPU软件栈为GPU任务的基本组件分配GPU内存并构建对应的GPU页表 110
操作系统通过DMA控制器将数据加载到所分配的GPU内存 120
GPU软件栈将代码加载到所分配的GPU内存 130
操作系统通过配置GPU MIMO来向GPU发送任务提交命令 140
GPU使用GPU内存中的数据和代码执行任务计算，并将计算结果存储到特定内存 150

图1

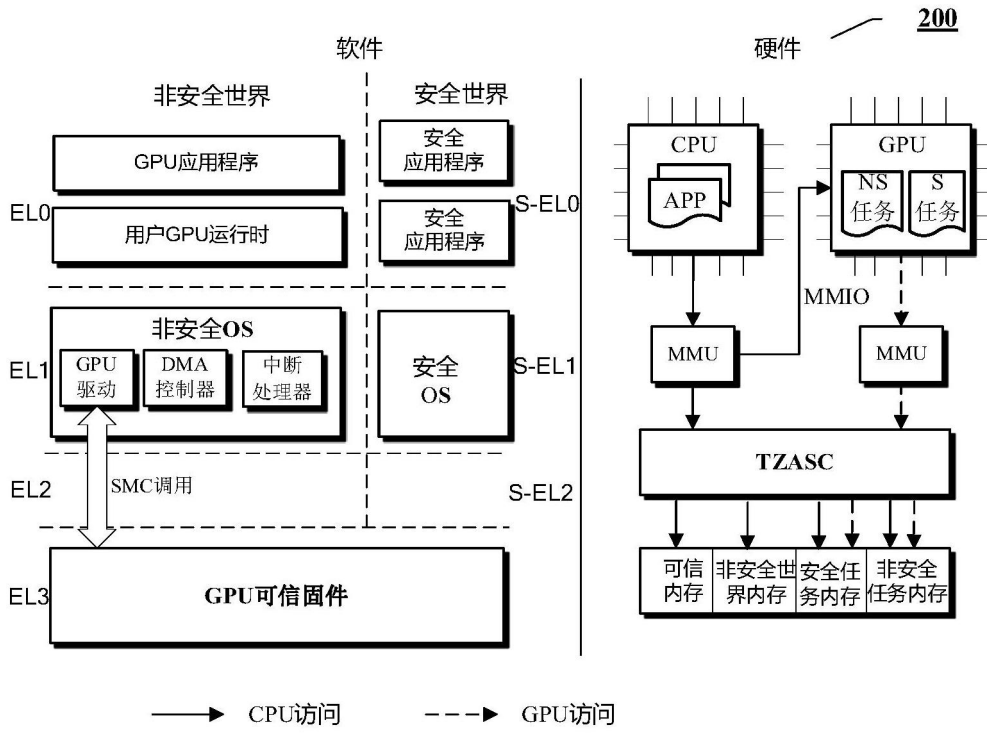

200
软件
硬件
非安全世界
安全世界
GPU应用程序
安全
应用程序
EL0
用户GPU运行时
安全
应用程序
S-EL0
非安全OS
EL1
GPU
驱动
DMA
控制器
中断
处理器
安全
OS
S-EL1
EL2
SMC调用
S-EL2
EL3
GPU可信固件
CPU
APP
GPU
NS
任务
S
任务
MMIO
MMU
MMU
TZASC
可信
内存
非安全世
界内存
安全任
务内存
非安全
任务内存
CPU访问
GPU访问

图2

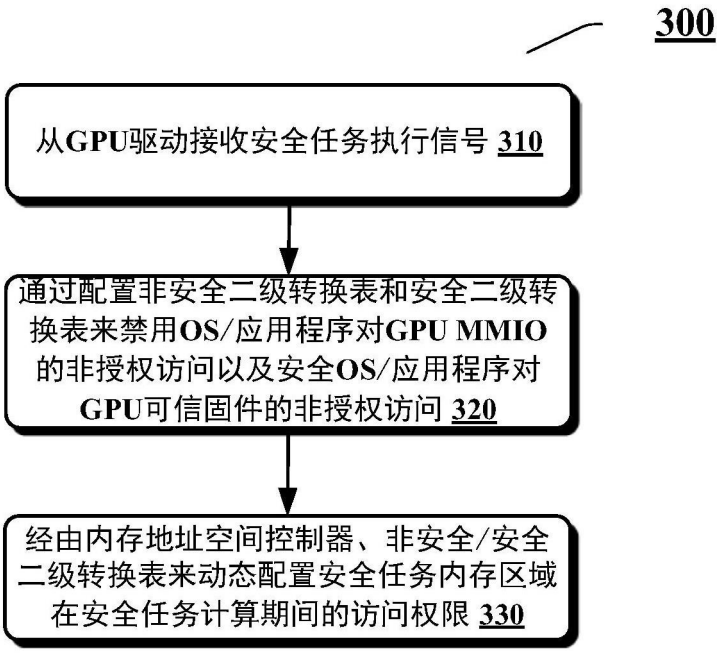

300
从GPU驱动接收安全任务执行信号 310
通过配置非安全二级转换表和安全二级转换表来禁用OS/应用程序对GPU MMIO的非授权访问以及安全OS/应用程序对GPU可信固件的非授权访问 320
经由内存地址空间控制器、非安全/安全二级转换表来动态配置安全任务内存区域在安全任务计算期间的访问权限 330

图3

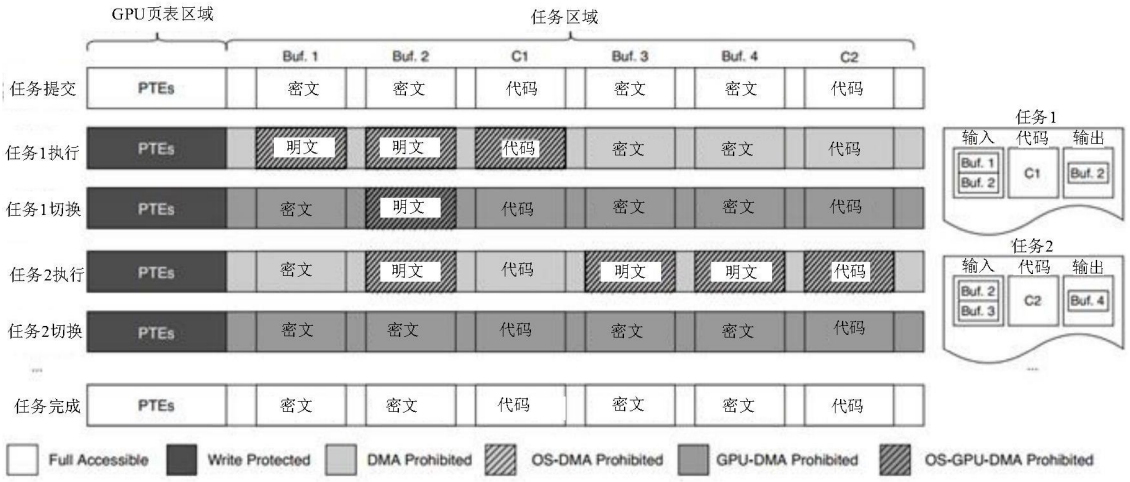

GPU页表区域
任务区域
Buf. 1
Buf. 2
C1
Buf. 3
Buf. 4
C2
任务提交
任务1执行
任务1切换
任务2执行
任务2切换
任务完成
PTEs
密文
明文
代码
任务1
输入
代码
输出
Buf. 1
Buf. 2
C1
Buf. 2
任务2
Buf. 2
Buf. 3
C2
Buf. 4
Full Accessible
Write Protected
DMA Prohibited
OS-DMA Prohibited
GPU-DMA Prohibited
OS-GPU-DMA Prohibited

图4

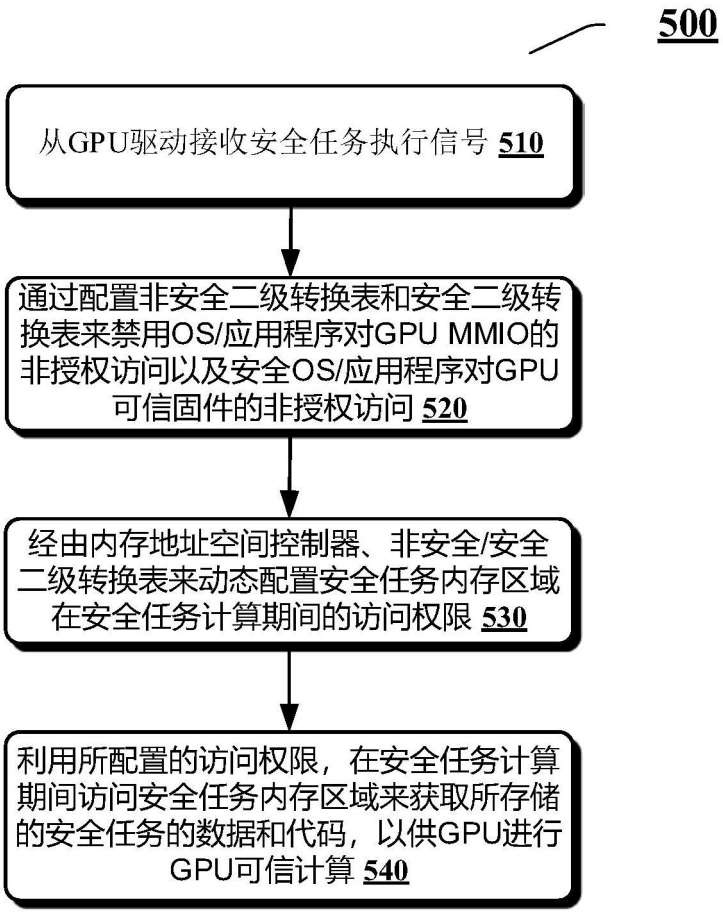

500
从GPU驱动接收安全任务执行信号 510
通过配置非安全二级转换表和安全二级转换表来禁用OS/应用程序对GPU MMIO的非授权访问以及安全OS/应用程序对GPU可信固件的非授权访问 520
经由内存地址空间控制器、非安全/安全二级转换表来动态配置安全任务内存区域在安全任务计算期间的访问权限 530
利用所配置的访问权限，在安全任务计算期间访问安全任务内存区域来获取所存储的安全任务的数据和代码，以供GPU进行GPU可信计算 540

图5


GPU保护
单元 610
安全任务保护
单元 620
可信执行环境
构建装置 600

图6


GPU保护
单元 710
安全任务保护
单元 720
安全访问
单元 730
可信计算执行装置 700

图7

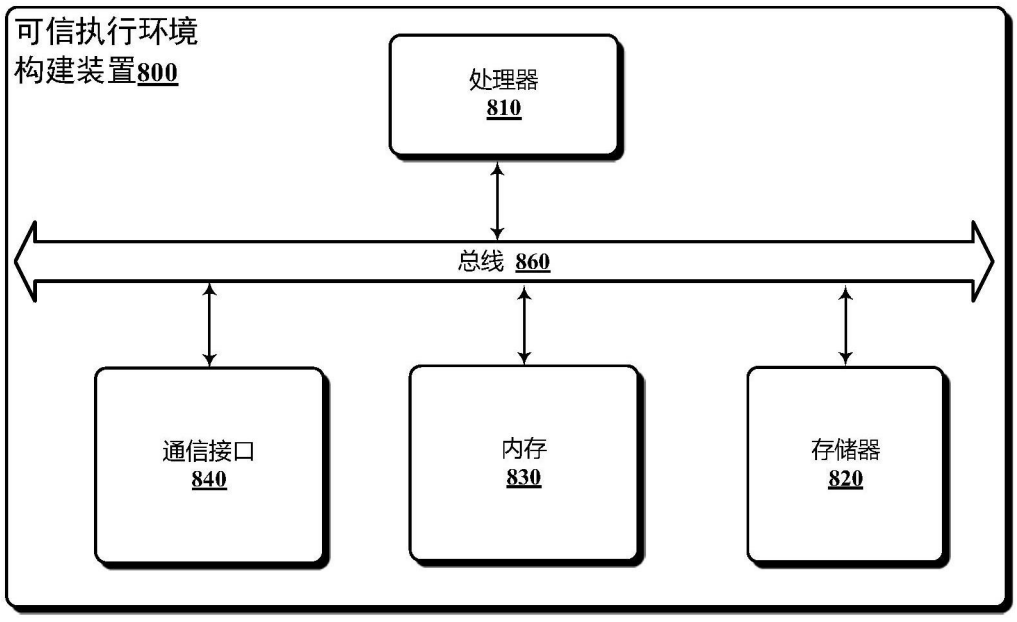

可信执行环境
构建装置800
处理器
810
总线 860
通信接口
840
内存
830
存储器
820

图8

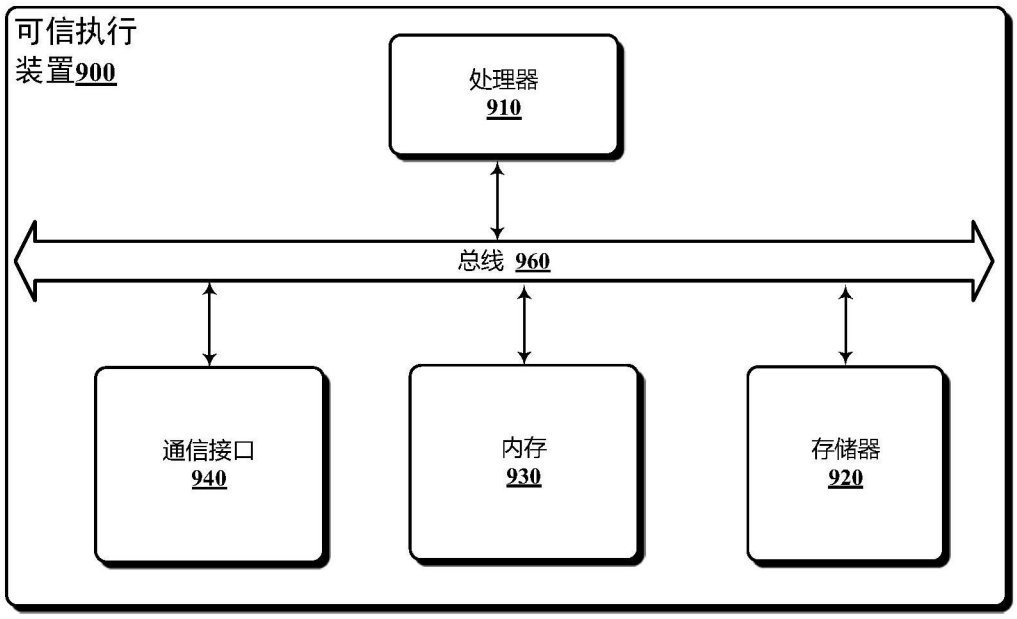


图9