(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 118779100 A
(43)申请公布日 2024.10.15

(21)申请号 202410819705.X

(22)申请日 2024.06.21

(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 张锋巍 王晨旭 卢琨 闫守孟

(74)专利代理机构 北京亿腾知识产权代理事务所(普通合伙) 11309
专利代理师 陈霁 周良玉

(51)Int.Cl.
G06F 9/50(2006.01)
G06F 13/28(2006.01)
G06F 21/57(2013.01)

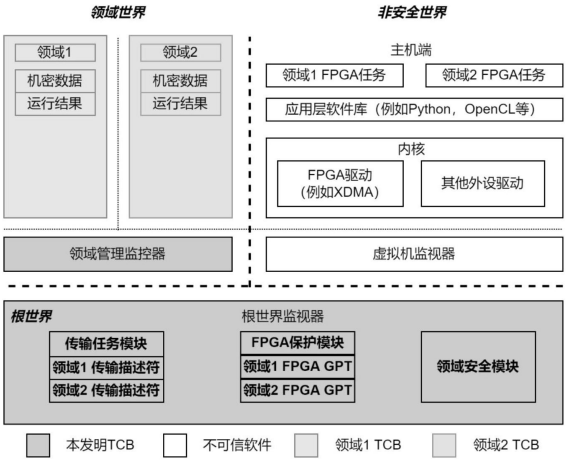
权利要求书2页 说明书11页 附图3页

(54)发明名称

一种在机密计算架构中执行FPGA任务的方法和设备

(57)摘要

本说明书实施例提供一种在机密计算架构中执行FPGA任务的方法和设备。方法包括:首先,非安全世界的FPGA软件在内存的非安全世界区段中,配置FPGA任务的桩数据结构。然后,根世界中的根监视器在内存中对应于第一领域的第一区段中,配置与桩数据结构对应的真实数据结构,其中包括,领域缓存区以及传输描述符;所述领域缓存区用于存储待处理的机密数据;所述传输描述符用于描述有待进行DMA传输的数据。接着,根监视器设置用于CPU的第一GPT表和用于FPGA的第二GPT表,使得在第一GPT表中,FPGA内存映射MMIO属于根世界,在第二GPT表中,仅第一区段为可访问区段。从而,根监视器通过FPGA MMIO与FPGA进行交互,使得FPGA基于传输描述符通过DMA传输读取所述机密数据,执行FPGA任务。



CN 118779100 A

1.一种在机密计算架构中执行FPGA任务的方法,所述机密计算架构包括,安全世界,领域世界,非安全世界,根世界;所述方法包括:

非安全世界的FPGA软件根据FPGA的数据需求信息,在内存的非安全世界区段中,配置FPGA任务的桩数据结构,其中包括数据缓存区;

根世界中的根监视器在所述内存中对应于第一领域的第一区段中,配置与所述桩数据结构对应的真实数据结构,其中包括,与所述数据缓存区对应的领域缓存区,以及传输描述符;所述领域缓存区用于存储待处理的机密数据;所述传输描述符用于描述有待进行直接内存访问DMA传输的数据;

所述根监视器设置颗粒度保护表GPT,其中包括用于CPU的第一GPT表和用于FPGA的第二GPT表,使得在第一GPT表中,FPGA内存映射MMIO属于根世界,在第二GPT表中,仅第一区段为可访问区段;

所述根监视器通过FPGA MMIO与FPGA进行交互,使得FPGA基于所述传输描述符通过DMA传输读取所述机密数据,执行FPGA任务。

2.根据权利要求1所述的方法,其中,在配置FPGA任务的桩数据结构之前,还包括:

所述第一领域向所述FPGA软件提供所述数据需求信息,及其签名;

配置与所述桩数据结构对应的真实数据结构,包括:

所述根监视器在验证所述签名正确后,配置所述真实数据结构。

3.根据权利要求1所述的方法,还包括,在配置FPGA任务的桩数据结构之前:

所述第一领域通过安全信道,接收用户提供的所述机密数据。

4.根据权利要求3所述的方法,还包括:

所述第一领域与所述用户通过密钥协商协议,协商出密钥;

基于所述密钥,构建所述安全信道。

5.根据权利要求1所述的方法,其中,所述传输描述符包括以下信息:待传输数据的起始地址,目的地地址,数据长度,以及所在领域缓存区对应的数据通道信息,所述数据通道选自从FPGA到主机的C2H通道,以及从主机到FPGA的H2C通道。

6.根据权利要求1所述的方法,其中,所述传输描述符包括,通过链表形式连接的多个描述符;单个描述符对应FPGA的单次数据导入;各描述符具有指示下一描述符地址的目标字段;在所述链表中,当前描述符的目标字段中填充有下一个描述符在所述第一区段中存储的起始地址。

7.根据权利要求1所述的方法,其中,所述数据需求信息包括,针对同一FPGA的多份数据需求;在配置与所述桩数据结构对应的真实数据结构之前,所述方法还包括:

所述FPGA软件在针对所述多份数据需求分别创建桩数据结构之后,统一调用smc指令,从而切换到根世界。

8.根据权利要求1所述的方法,其中,所述领域缓存区包括,输入数据缓存区和结果数据缓存区,所述输入数据缓存区存储所述机密数据,所述结果数据缓存区用于存储所述FPGA任务的执行结果。

9.根据权利要求1所述的方法,其中,

配置FPGA任务的桩数据结构,包括:根据所述数据缓存区,生成桩页表;

配置与所述桩数据结构对应的真实数据结构,包括:根据所述桩页表,和所述领域缓存

区,生成真实FPGA页表。

10.根据权利要求9所述的方法,其中,在所述根监视器通过FPGA MMIO与FPGA进行交互之前,还包括:

所述根监视器修改FPGA MMIO,使其指向所述真实FPGA页表。

11.根据权利要求1所述的方法,其中,在所述第二GPT表中,第一区段被设置属于非安全世界,其他所有内存区段被设置为属于根世界。

12.根据权利要求1所述的方法,还包括:

在执行所述FPGA任务之后,令FPGA将其片上内存中最近使用的内存完全清除。

13.一种机密计算架构中的根监视器,所述机密计算架构包括,安全世界,领域世界,非安全世界和根世界;所述根监视器位于所述根世界中,并包括传输任务模块和FPGA保护模块,其中:

传输任务模块配置为,响应于非安全世界的FPGA软件根据FPGA的数据需求信息,在内存的非安全世界区段中配置FPGA任务的桩数据结构,而所述内存中对应于第一领域的第一区段中,配置与所述桩数据结构对应的真实数据结构,其中桩数据结构包括数据缓存区;真实数据结构包括,与所述数据缓存区对应的领域缓存区,以及传输描述符;所述领域缓存区用于存储待处理的机密数据;所述传输描述符用于描述有待进行直接内存访问DMA传输的数据;

FPGA保护模块配置为,设置颗粒度保护表GPT,其中包括用于CPU的第一GPT表和用于FPGA的第二GPT表,使得在第一GPT表中,FPGA内存映射MMIO属于根世界,在第二GPT表中,仅第一区段为可访问区段;通过FPGA MMIO与FPGA进行交互,使得FPGA基于所述传输描述符通过DMA传输读取所述机密数据,执行FPGA任务。

14.一种计算设备,包括存储器和若干处理器,所述计算设备形成机密计算架构,所述机密计算架构包括,安全世界,领域世界,非安全世界和根世界;所述根世界包括权利要求13所述的根监视器。

一种在机密计算架构中执行FPGA任务的方法和设备

技术领域

[0001] 本说明书一个或多个实施例涉及机密计算框架,尤其涉及一种在机密计算框架中执行FPGA任务的方法及装置。

背景技术

[0002] 随着各行业计算技术的发展,以及云端和终端用户的增加,人们将大量数据存储在各种计算机设备中。在行业发展的同时,人们对于设备和数据安全的关注也在日益增加。为了确保设备和数据的安全性,各个架构厂商也分别提出了各自的解决方案,如ARM提出了可信区技术(TrustZone),AMD提出了安全虚拟机加密技术(SEV),英特尔提出了软件防护扩展(SGX)技术,等等。这些解决方案为用户提供一个安全的可信执行环境,用于机密地保存和处理数据,使其免受不可信的内核与传统应用程序的损害。以Arm可信区技术为例,它将传统内核和应用程序的运行环境视作为非安全世界,并创建了一个隔离的安全世界,以及定义了具有最高权限的安全层用于世界切换。非安全世界将无法直接访问安全世界,需要经过安全层的固件验证才能访问特定的资源。

[0003] 虽然ARM机密计算架构有效地确保了用户的数据安全,然而,其仍然存在一些不足,其中之一是无法提供对诸如FPGA的专用加速器上机密计算的支持。这使得在该技术框架下,采用FPGA进行任务加速具有很大的安全方面的挑战,存在对此进行改进的需求。

发明内容

[0004] 本说明书一个或多个实施例描述了一种在机密计算架构中执行FPGA任务的方法及装置,能够基于已有的机密计算架构的硬件特性,为FPGA任务的执行提供机密计算环境,支持FPGA机密计算。

[0005] 根据第一方面,提供一种在机密计算架构中执行FPGA任务的方法,所述机密计算架构包括,安全世界,领域世界,非安全世界,根世界;所述方法包括:

[0006] 非安全世界的FPGA软件根据FPGA的数据需求信息,在内存的非安全世界区段中,配置FPGA任务的桩数据结构,其中包括数据缓存区;

[0007] 根世界中的根监视器在所述内存中对应于第一领域的第一区段中,配置与所述桩数据结构对应的真实数据结构,其中包括,与所述数据缓存区对应的领域缓存区,以及传输描述符;所述领域缓存区用于存储待处理的机密数据;所述传输描述符用于描述有待进行直接内存访问DMA传输的数据;

[0008] 所述根监视器设置颗粒度保护表GPT,其中包括用于CPU的第一GPT表和用于FPGA的第二GPT表,使得在第一GPT表中,FPGA内存映射MMIO属于根世界,在第二GPT表中,仅第一区段为可访问区段;

[0009] 所述根监视器通过FPGA MMIO与FPGA进行交互,使得FPGA基于所述传输描述符通过DMA传输读取所述机密数据,执行FPGA任务。

[0010] 根据第二方面,提供了一种机密计算架构中的根监视器,所述机密计算架构包括,

安全世界,领域世界,非安全世界和根世界;所述根监视器位于所述根世界中,并包括传输任务模块和FPGA保护模块,其中:

[0011] 传输任务模块配置为,响应于非安全世界的FPGA软件根据FPGA的数据需求信息,在内存的非安全世界区段中配置FPGA任务的桩数据结构,而所述内存中对应于第一领域的第一区段中,配置与所述桩数据结构对应的真实数据结构,其中桩数据结构包括数据缓存区;真实数据结构包括,与所述数据缓存区对应的领域缓存区,以及传输描述符;所述领域缓存区用于存储待处理的机密数据;所述传输描述符用于描述有待进行直接内存访问DMA传输的数据;

[0012] FPGA保护模块配置为,设置颗粒度保护表GPT,其中包括用于CPU的第一GPT表和用于FPGA的第二GPT表,使得在第一GPT表中,FPGA内存映射MMIO属于根世界,在第二GPT表中,仅第一区段为可访问区段;通过FPGA MMIO与FPGA进行交互,使得FPGA基于所述传输描述符通过DMA传输读取所述机密数据,执行FPGA任务。

[0013] 根据第三方面,提供了一种计算设备,包括存储器和若干处理器,所述计算设备形成机密计算架构,所述机密计算架构包括,安全世界,领域世界,非安全世界和根世界;所述根世界包括如第二方面所述的根监视器。

[0014] 在本说明书实施例提供的方案中,通过传输任务实现兼容Arm机密计算架构CCA的FPGA机密计算。根据传输任务机制,由非安全世界的FPGA软件创建不包含真实数据的桩任务,并如常规流程调度和管理桩任务。在桩任务提交后,根监视器在领域世界区段中创建包含真实数据的真实FPGA任务的数据结构,其中包括,用于存储机密数据的领域缓存区,以及用于描述有待进行DMA传输的数据的传输描述符。根监视器还通过GPT表的设置为真实FPGA任务提供隔离的执行环境。从而使得,FPGA基于传输描述符通过DMA传输读取机密数据,执行FPGA任务。如此,在Arm机密计算架构CCA中实现FPGA机密计算。

附图说明

[0015] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其它的附图。

[0016] 图1示出Arm机密计算架构的示意图;

[0017] 图2示出机密计算架构中各个世界对物理地址空间的访问权限控制;

[0018] 图3示出根据一个实施例在机密计算架构中运行FPGA任务的示意图;

[0019] 图4示出根据一个实施例在机密计算架构中执行FPGA任务的方法;

[0020] 图5示出根据一个实施例的多个传输描述符的示意图;

[0021] 图6示出在一个示例场景中,根监视器维护的GPT表。

具体实施方式

[0022] 下面结合附图,对本说明书提供的方案进行描述。

[0023] 为了保证数据的安全性,ARM提供了TrustZone可信区技术。在该技术中,将传统内核和应用程序的运行环境视为非安全世界(Norma l Wor ld),在此之外创建了一个隔离的

安全世界(Secure Wor ld),并定义了具有最高权限的安全层用于世界切换。

[0024] 具体的,在Armv8-A架构中,CPU核基于特权划分将异常划分为4个等级,EL0至EL3,其中,EL0表示应用级,EL1用于系统内核(kerna l),EL2表示虚拟机管理器(hypervi sor),EL3表示安全层监视器。这四个等级也可用于表示运行环境的权限等级。在TrustZone可信区技术中,CPU安全状态被划分为非安全(Norma l)状态和安全状态。EL0和EL1可以运行于任意状态,例如,可以在非安全世界的EL1中执行非可信的操作系统OS(untrusted OS),在安全世界的EL1中执行可信OS。EL2可用于安全状态。EL3即安全层监视器,永远存在于安全世界,用于执行安全状态的切换。

[0025] 在该架构下,非安全世界无法直接访问安全世界,需要经过安全层监视器的验证才能访问特定的资源。敏感或机密的数据,以及高权限的软件应用运行于安全世界,从而为这些机密数据提供一种可信执行环境TEE。

[0026] 在以上的TrustZone基础架构基础上,ARM近来又发布了改进的Arm机密计算架构CCA(Confident ia l Compute Arch itecture)。Arm机密计算架构是Armv9-A架构的一部分,其在原本的TrustZone架构基础上引入了领域管理扩展,该扩展在可信区技术中已经存在的非安全世界与安全世界之外,额外引入领域(Rea lm)世界和根(Root)世界。为了支持不同世界的隔离,CCA架构在硬件层提供领域管理扩展RME(Rea lm Management Extens ion)组件,来扩展隔离模式。

[0027] 图1示出Arm机密计算架构的示意图。如图1所示,在Arm机密计算机构CCA中,运行环境被划分为四个世界:安全世界,非安全世界(Norma l),领域世界和根世界。根世界中运行着拥有最高权限的根世界监视器,负责世界之间的隔离和通信。领域世界用于为虚拟机提供名为机密领域的受保护的虚拟机机密计算环境。领域世界中运行着领域管理监视器RMM,负责管理领域虚拟机的执行以及与非安全世界的交互。用户可以将虚拟机作为领域虚拟机放入机密领域中,隔离来自外界软件的非法访问。具体的,用户可以通过非安全世界中的虚拟机管理器创建虚拟机,通过领域管理监视器RMM将其转入到领域世界,成为领域虚拟机。领域管理监视器RMM会负责与机密领域安全相关的检查和保护。领域虚拟机之间使用虚拟化技术相互隔离,领域管理监视器会负责管理不同领域虚拟机的可访问地址空间。领域虚拟机不需要相信非安全世界和安全世界,只需要相信领域管理监视器和根世界监视器。

[0028] 相应的,Arm机密计算架构CCA将内存的物理地址空间PAS(Phys ica l address spaces)也划分为四个世界。图2示出机密计算架构中各个世界的安全状态对物理地址空间的访问权限控制。如图2所示,根世界具有最高的访问权限,可以访问所有四个世界的地址空间。非安全世界具有最低的访问权限,仅可以访问非安全世界的地址空间。安全世界和领域世界,则可以访问非安全世界的地址空间,以及属于自己世界的地址空间。

[0029] 在Arm机密计算架构中,不同世界的地址空间访问控制,通过构建颗粒度保护表GPT(Granu le Protect ion Tab le),并基于该GPT表进行颗粒度保护检查GPC(Granu le Protect ion Check)而实现。具体的,机密计算架构CCA在内存中维护颗粒度保护表GPT,其中记录细粒度的每段物理内存的安全状态。典型的,记录的粒度是以内存页(4KB大小的区段)为单位。如此,GPT表中记录每个内存页的安全状态和访问权限。当内存页的分配在不同世界发生迁移和变更,则可以动态更新GPT中的条目。

[0030] 当处理器访问内存时,硬件层中的前述RME组件执行颗粒度保护检查GPC。在检查

中,获取当前CPU的安全状态,并通过读取GPT表获取请求访问的内存页的安全状态,检查二者是否匹配。如果没有通过GPC检查(例如,如果非安全世界的主机OS请求访问领域世界的内存),则会发出颗粒度保护异常信号,从而拒绝此次内存访问,由此保证世界之间的隔离。通过以上的隔离机制,Arm机密计算架构进一步为领域世界的领域虚拟机提供了隔离的机密计算环境。

[0031] 另一方面,许多任务希望能够通过专用加速器来进行加速执行,其中FPGA是一种用于加速任务执行的硬件。FPGA(Fie ld Programmab le Gate Array)即现场可编程门阵列,是一种可编程的硬件电路结构,可通过Ver i log等语言构建电路逻辑并执行特定的计算指令。FPGA在神经网络加速运算,工业控制等领域,体现了其加速效能,逐渐受到云提供商的重视,扩大其应用。

[0032] 然而,现有的Arm计算框架难以为FPGA计算任务提供有效的机密保护。这一方面是因为,Arm计算框架将FPGA视为非可信的普通外设。尽管FPGA具有其专有的片上内存,但是仍需要与主机内存进行交互,在交互时FPGA需要与CPU和诸多不可信外设共同访问主机内存,因此容易受到攻击。另一方面,根据目前大多数ARM设备的工作流程,FPGA的任务执行和调度受到FPGA软件(如FPGA驱动和相关的编程软件库)的管理。而该FPGA软件处于非安全世界中,容易受到攻击。

[0033] 具体的,FPGA软件用于管理FPGA的计算环境,并与FPGA硬件进行交互。在常规流程中,为了准备执行环境,FPGA软件根据任务要求,分配物理内存,将待处理的数据加载到主机内存中,并通过FPGA内存映射MMIO,使得FPGA硬件经由直接内存访问DMA访问主机内存中的数据,执行FPGA任务。

[0034] 假设一个很强的敌手,他控制了非安全世界和安全世界的整个软件栈,包括,FPGA软件,非可信OS,虚拟机管理器hypervi sor,以及安全世界中同样层次的软件。该敌手想要窥探甚至篡改FPGA任务的机密数据,包括任务的输入数据,中间数据或执行结果。那么,敌手有可能访问主机内存,读取其中存储的机密数据,或控制能够进行DMA的外设来读取上述内存数据,从而进行攻击。

[0035] 目前已针对GPU加速器设计了一些机密保护方案。但是,由于硬件设计差异,GPU和FPGA加速器的工作流程存在不同,因此无法直接将针对GPU的机密计算方案直接迁移到FPGA加速的系统上。最近在针对Arm CCA的一项研究中,提出了一种兼容GPU和FPGA的机密加速计算系统。虽然该系统确保了FPGA计算安全,但其没有很好地符合Arm CCA的系统设计:它依赖较为庞大的加速器软件栈来管理加速计算,因此极大增加了机密虚拟机的可信计算基,也需要对FPGA软件栈进行大量改动。此外,一些研究者也使用非Arm的安全硬件,或传统的Arm安全硬件(如StrongBox系统,其依赖Arm TrustZone;Cronus系统,其依赖Arm安全虚拟化等)构建机密加速计算系统,然而此类系统无法直接兼容基于Arm CCA的系统设计,且传统的Arm安全硬件在Arm CCA中已不再完全受信,导致系统的安全性降低。另外,还有研究者提出构建在FPGA加速器内部的防御机制,例如在FPGA内部构建加密引擎、远程认证模块等,但此类设计需要修改加速器硬件,导致硬件兼容性降低。

[0036] 因此,总体来说,针对Arm CCA架构,如何对FPGA加速器的任务执行提供机密计算保护,仍然是一项挑战。

[0037] 有鉴于此,在本说明书的实施例中提出一种方案,基于Arm机密计算架构CCA的硬

件特性,在不影响Arm CCA架构原有功能设计的基础上,提供对FPGA计算任务的安全保护,从而支持FPGA上的机密计算。

[0038] 图3示出根据一个实施例在机密计算架构中运行FPGA任务的示意图。图3所示的系统架构符合Arm机密计算架构CCA,其中,非安全世界中运行有主机端,主机端包含FPGA驱动(例如XDMA)和其他外设驱动。虚拟机管理器hypervi sor创建若干机密计算“领域”,并对其进行管理和调度。而在机密计算架构CCA新引入的领域世界中,部署领域管理监控器RMM来实现不同“领域”的内存隔离。根世界部署具有最高权限的根监视器,用于管理世界之间的隔离和切换,以及提供秘钥管理、远程验证等安全认证机制。根监视器可以实现为安全固件的形式。

[0039] 在以上CCA架构中,认为领域世界的领域管理监控器RMM和根世界为完全可信,这是因为这些组件所需的内存和代码量非常小,因此暴露的攻击面较小,不易受到攻击。此外,其他组件则被视为不可信,包括安全世界的软件。

[0040] 在本说明书实施例的方案中,为了实现FPGA机密计算,在根世界的根监视器中引入几个附加的组件:传输任务模块,FPGA保护模块和领域安全模块。其中,传输任务模块用于针对在非安全内存中为FPGA创建的不含真实数据的传输任务,在领域内存中创建真实任务;FPGA保护模块用于保护FPGA运行环境不受攻击,领域安全模块用于进行领域的构建,机密数据传输与FPGA硬件认证。此外,本实施例的方案中,FPGA软件(包括FPGA驱动和相关函数库)仍然运行在非安全世界的主机端,但是需要进行少量修改,以协助完成传输任务。

[0041] 传输任务是为了使得FPGA工作流程兼容于Arm机密计算架构所引入的机制,其核心思想是,允许主机端中FPGA软件在非安全世界创建和管理与领域对应的FPGA任务(又可称为桩任务),具体包括例如分配内存、创建缓冲区、调度和提交任务等。这些桩任务,例如图3中对应于领域1的FPGA任务1,对应于领域2的FPGA任务2,其数据结构与普通FPGA任务相似,例如包含所需的数据缓冲区,但这些桩任务的数据缓冲区内并不包含真实的待处理数据,至多提供对这些数据缓冲区的描述。与此对应的,在对应领域中构建对应的真实FPGA任务的数据结构。FPGA软件可以照常提交这些桩任务,与常规不同的是,在其提交时,位于根世界的根监视器将桩任务替换为对应领域中的真实FPGA任务。真实FPGA任务具有与桩任务类似的数据结构,但是其中填入有真实待处理的机密数据。并且,根据FPGA的执行特点,在对应领域中,创建和维护FPGA进行直接内存访问DMA所需的数据描述符(图3中的传输描述符)。从而,在根监视器的控制下,FPGA根据对应领域中的传输描述符,读取领域中存储的机密数据,执行FPGA中的运算逻辑。在此过程中,通过FPGA保护模块提供对应的GPT表,确保执行环境的隔离。由此,本实施例的方案允许非安全世界在不访问真实机密数据的前提下,调度和管理来自不同领域的FPGA任务,符合Arm机密计算架构的设想。

[0042] 下面结合单个FPGA任务,描述通过传输任务机制,调度和执行FPGA计算任务并为其提供机密隔离环境的过程。

[0043] 图4示出根据一个实施例在机密计算架构中执行FPGA任务的方法;可以理解,图4中的方法,基于图3所示的机密计算架构执行。

[0044] 首先在初始化阶段或预备阶段,用户可以针对一个FPGA申请一个领域,并通过加密信道将需要FPGA任务处理的真实数据传入其中。这里,领域的创建可以通过图3所示的领域安全模块进行。具体的,领域安全模块可以根据用户的请求,通过hypervi sor创建一个

虚拟机,通过与领域管理监控器RMM交互,将其部署到领域世界中作为一个机密领域。为了描述的方便(以及必要时区分于其他具体领域),下文中,将该用户申请的领域称为第一领域。在创建第一领域之后,用户可以与该第一领域进行密钥协商,以建立安全信道。具体的,用户可以通过DH(Diffie-He l lman)协议,基于椭圆曲线的DH协议,或其他各种协议,与第一领域交换密钥,从而协商出加密密钥。于是,基于协商出的密钥,二者可以建立一个安全的加密信道。通过该安全加密信道,第一领域可以接收用户传输的机密数据,并将其存储于领域世界。

[0045] 不同于GPU通过任务代码指示计算逻辑,FPGA的执行逻辑已经由设计者通过编程的方式固化于硬件电路中,也就是说,FPGA在设计完成后,具有固定的执行逻辑。通常,设计完成的FPGA具有对应的描述文件,其中记录该FPGA的功能逻辑,暴露的接口,接口对应的输入/输出数据等。通过该描述文件,可以确定FPGA任务的数据需求信息。

[0046] 在一个实施例中,在创建上述第一领域后,将对应FPGA的描述文件记录于第一领域中。当需要执行FPGA任务时,第一领域可以根据该描述文件,向非安全世界的主机端中的FPGA软件提供数据需求信息,其中包括,目标传输数据,数据长度,数据类型等。为了防止FPGA软件被攻击后篡改数据需求,在一个实施例中,第一领域还提供签名信息,即对数据需求信息进行签名,将签名附加在数据需求信息之后。在其他实施例中,FPGA软件也可以通过其他方式,获取FPGA任务的数据需求信息,例如由用户提供给FPGA软件,或根据用户指示在指定存储位置读取上述描述文件,得到数据需求信息。

[0047] 根据上述初始化阶段,第一领域获得用户提供的机密数据/真实数据,将其存储于受保护的领域世界。主机端获得FPGA任务的数据需求信息,将其存储于非安全世界对应的未保护区域。

[0048] 在此基础上,非安全世界的主机端中的FPGA软件,就可以基于数据需求信息创建传输任务。如前所述,FPGA软件主要包括FPGA驱动软件,例如使用Xi l inx的XDMA驱动,也包括与之相关的一些函数库,例如用户层运行时函数库。该FPGA软件经过修改,基于传输任务机制,在非安全世界创建桩任务。

[0049] 具体的,如图4中步骤S41所示,非安全世界的FPGA软件根据FPGA的数据需求信息,在内存的非安全世界区段中,配置FPGA任务的桩数据结构,其中包括数据缓存区。

[0050] 具体的,FPGA软件可以创建一个传输任务,并相应配置桩数据结构。如前所述,数据需求信息指示了FPGA任务处理数据所要求的数据信息,包括,数据缓存区的数量、大小、填充数据等特点。根据这样的数据需求信息,FPGA软件可以在内存的非安全世界区段中,分配相应的内存空间,在其中创建符合数据需求信息要求的若干桩数据缓存区。

[0051] 在一些实施例中,数据需求信息会要求创建多个数据缓存区,例如其中之一用于存储输入数据,另一个用于存储执行结果。可选的,有时数据需求信息还会指示创建存储中间结果的数据缓存区。FPGA软件根据数据需求信息的要求,对应分配这些数据缓存区作为桩数据缓存区。不过与常规处理不同的,FPGA只是根据数据需求信息如实创建桩数据结构,但并不在桩数据缓存区填充真实数据。

[0052] 由于并不需要在这些桩数据缓存区中存储真实数据,在一个实施例中,在创建桩数据缓存区时,可以只按照数据需求信息中缓存区数量、数据类型的要求进行创建,而不必按照需求数据的大小。并且,创建的桩数据缓存区中可以仅存储数据描述信息。

[0053] 需要说明的是,FPGA硬件在执行任务时,通过DMA与主机内存进行交互。在此过程中,FPGA通常使用描述符(descr iptor)进行DMA操作。描述符是一种数据结构,它包含了一次DMA传输所需的信息,比如源地址、目的地址、传输大小等。为此,FPGA软件在创建桩任务时,通常会根据驱动文件,在上述非安全区段中创建或加载描述符。然而,如上所述,桩数据结构中并不存储真实数据,因此,这里加载的描述符只是用于对齐和符合FPGA软件的常规处理流程,其中并不反映真实数据的存储信息,因此此时加载的描述符又可称为桩描述符。

[0054] 可选的,一些FPGA可能要求创建执行任务所需的页表,即FPGA页表。在这样的情况下,FPGA软件还根据分配的内存,生成用于执行该传输任务的FPGA页表,可以称为桩页表。该页表中记录有FPGA任务执行过程中虚拟地址与内存物理地址的映射。初始地,FPGA内存映射MMIO(Memory-Mapped I nput/Output)被配置为指向该桩页表。具体地,通过FPGA内存映射MMIO,将FPGA硬件中存储页表基地址的寄存器映射为存储所述桩页表的内存地址,从而指向该桩页表。

[0055] 如此,FPGA软件创建了传输任务,为其配置了桩数据结构。在一个具体示例中,在非安全世界的内存区段中,FPGA软件分配了数据缓冲区1,数据缓冲区2,其中数据缓冲区1和2中可以仅存储对应的数据描述,而不存储真实数据。此外,FPGA软件还加载有桩描述符,以及生成了桩页表。

[0056] 可以看到,创建传输任务的过程与创建常规任务的过程是相似的,只是并不向其中的数据缓存中填充真实数据。因此,创建的传输任务是一个不具有真实数据的“空壳”任务,但是具有与真实任务相同的数据结构,可以进行任务的管理和调度。

[0057] 因此,FPGA软件在创建上述传输任务后,如常地将其插入到FPGA任务队列中,安排任务的执行顺序,并经由根监视器向FPGA硬件提交该传输任务。具体的,FPGA软件在提交任务时可以调用smc(安全监视器调用)指令,该指令能够被根世界固件捕获,调用后CPU将进入根世界执行代码。

[0058] 根监视器捕获到上述指令,就会在领域世界创建真实任务,即执行图4的步骤S42。在该步骤中,根监视器在内存中对应于第一领域的第一区段中,配置与桩数据结构对应的真实数据结构,即领域数据结构;所述领域数据结构包括,与数据缓存区对应的领域缓存区,以及新创建的传输描述符。领域缓存区用于存储真实的机密数据,传输描述符用于描述有待进行DMA传输的真实数据的存储信息。根监视器将待处理的机密数据存储于上述领域缓存区,或将其暂时留空(例如用于存储结果的领域缓存区)。

[0059] 具体的,根监视器可以复制数据需求信息,并通过签名验证其正确性。在验证其正确性后,可以根据数据需求信息,在第一领域对应的第一区段中,创建真实FPGA任务的数据结构,其中包括与桩数据结构中的数据缓存区对应的领域缓存区。根监视器根据数据需求信息,将真实的机密数据存储于对应的领域缓存区中。此外,根监视器根据真实数据在领域缓存区中的存储信息,重新构建描述符,称为传输描述符。在一个实施例中,在第一领域中构建的传输描述符至少包括以下信息:待传输数据的起始地址(Src_adr),目的地地址(Dst_adr),长度(Len)等。

[0060] 进一步的,如本领域技术人员所知,FPGA硬件可以与主机之间建立数据通道,以便进行直接内存访问DMA。不同的FPGA硬件,可以通过不同的具体实现方式,建立这样的数据通道。从数据传输方向来说,上述数据通道可以分为C2H通道(Card to Host,从FPGA存储数

据到主机的通道),以及H2C通道(Host to Card,从主机加载数据到FPGA的通道)。两种通道对应不同的数据传输方向。相应的,针对这样的情况,在一个实施例中,在第一领域中创建的传输描述符还可以包括,数据所在领域缓存区对应的数据通道信息(通道ID)。

[0061] 在一个实施例中,存储数据描述信息所需的内存长度较小(即桩数据结构中的缓存区可以很小),而一次DMA请求实际要求传输数据的内存长度可能较大,无法通过一次传输完全向FPGA导入所有数据。为此,在一个实施例中,根监视器根据数据需求信息中的总的数据需求长度,以及单次传输可以向FPGA导入的数据长度,计算出总传输所需次数,进而在第一领域区段中构建多个传输描述符,并通过链表形式将其连接。具体地,使得传输描述符包含Nxt_adr属性字段,用于存储下一传输描述符的起始地址。可选的,还可以包含Nxt_adj属性字段,用于存储剩余传输次数。在构建多个传输描述符时,根监视器将下个传输描述符的起始地址写入到当前传输描述符的Nxt_adr属性中,并将剩余的传输次数填入Nxt_adj属性中,如此形成链表结构。

[0062] 图5示出根据一个实施例的多个传输描述符的示意图。如图5所示,假定在第一领域区段中创建了两个buffer(即领域缓存区),其中buf1用于存储输入数据,大小为0x3000。buf2用于存储输出数据,大小为0x2000。同时,FPGA具有片上硬件内存M。

[0063] 在一个示例中,为了执行FPGA任务,需要发起两次DMA请求,第一次请求使用H2C数据通道,将buf1中的数据加载到FPGA。第二次请求使用C2H数据通道,将FPGA片上内存中的数据存储于buf2。

[0064] 假定buf1中的数据较长,无法通过一次传输导入到FPGA。为此,在H2C传输中,如图5所示,准备了3个描述符desc0-1,desc0-2和desc0-3。

[0065] 具体的,desc0-1中的起始地址src_adr指向buf1的地址A,目的地地址dst_adr指向FPGA硬件内存M的地址B,数据长度s ize可以是0x1000,nxt_adr字段指向desc0-2的地址。

[0066] Desc0-2中起始地址src_adr指向buf1的地址A+0x1000,目的地地址dst_adr指向FPGA硬件内存M的地址B+0x1000,数据长度s ize为0x1000,nxt_adr字段指向desc0-3的地址。

[0067] Desc0-3中起始地址src_adr指向buf1的地址A+0x2000,目的地地址dst_adr指向FPGA硬件内存M的地址B+0x2000,数据长度s ize为0x1000,nxt_adr字段设置为0,表示没有后续传输。

[0068] 如此,通过串接的3个描述符desc0-1,desc0-2和desc0-3,可以针对一次H2C的DMA传输请求,将buf1中的数据加载到FPGA。

[0069] 类似的,针对C2H传输,可以准备2个描述符desc1-1和desc1-2。

[0070] desc1-1中起始地址src_adr指向FPGA硬件内存M的地址C,目的地地址dst_adr指向buf2的地址D,数据长度s ize为0x1000,nxt_adr字段设置为desc1-2的地址。

[0071] desc1-2中起始地址src_adr指向FPGA硬件内存M的地址C+0x1000,目的地地址dst_adr指向buf2的地址D+0x1000,数据长度s ize为0x1000,nxt_adr字段设置为0,表示没有后续传输。

[0072] 如此,通过串接的2个描述符desc1-1和desc1-2,可以针对一次C2H的DMA传输请求,将FPGA中的数据存储到buf2。

[0073] 除此之外,注意到某些FPGA加速计算任务需要传输多种不同类型的数据,需要频繁调用smc指令,而FPGA通常等待所有数据均传输完成后才开始计算。为此,在一个实施例中,可以批量记录针对同一FPGA的多份数据需求,最后合并发送至根世界的传输任务模块中。换而言之,在针对同一FPGA的多个数据需求信息分别创建多个桩任务/桩数据结构后,再统一调用smc指令,使得根监视器针对该多个数据需求信息,统一构建真实数据结构和传输描述符,从而减少smc调用与世界切换次数,降低性能负担。

[0074] 另外,如前所述,在一些实施例中,FPGA要求创建页表以执行计算任务,为此,FPGA软件在非安全内存区段生成有桩页表。与此对应的,根监视器需要根据桩页表创建真实FPGA页表,存储于第一领域区段。为此,根监视器可以首先对桩页表中记录的页表条目进行校验,例如检查其中是否存在重复映射或非法映射。如果校验通过,根监视器通过复制或重放其中的页表条目,构建真实FPGA页表。需要注意的是,由于桩数据结构中的数据缓存区并不存储真实数据,也不参与真实FPGA计算,因此,在真实FPGA页表中,将与数据缓存区相关的条目修改为指向真实数据结构中的领域缓存区。

[0075] 延续之前的示例,在真实任务创建阶段,在受保护的领域区段中,创建与桩任务中的2个数据缓冲区对应的真实的数据缓冲区1和数据缓冲区2。将真实的机密数据存储于数据缓冲区1,而将数据缓冲区2暂时留空,用于存储结果数据。此外,根监视器还生成真实FPGA页表,存储于该第一领域区段。

[0076] 在需要执行真实FPGA任务时,根监视器中的FPGA保护模块为真实FPGA任务的执行提供受保护的执行环境,实现该执行环境与其他软件(包括不受信的软件栈,以及其他“领域”)隔离。具体来说,FPGA与主机的内存交互主要包括两个方面:(1)DMA数据交互,(2)FPGA MMIO的指令交互(即通过写入FPGA寄存器,传输任务执行操作)。针对以上两个方面,根监视器通过Arm机密计算架构中提供的基于颗粒度保护检查(GPC)的内存保护机制,实现FPGA运行环境的隔离。

[0077] 为此,根据GPC机制,根监视器执行步骤S43,设置用于CPU的第一GPT表和用于FPGA的第二GPT表,使得在第一GPT表中,FPGA内存映射MMIO属于根世界,在第二GPT表中,仅第一区段为可访问区段。

[0078] 如前所述,机密计算架构CCA在内存中维护颗粒度保护表GPT,其中记录细粒度的每段物理内存的安全状态,用于进行GPC检查,从而实施内存隔离。根据本实施例的方案,根监视器可以维护多个版本的GPT表,使得不同对象具有不同访问权限。

[0079] 具体地,根监视器至少维护第一GPT表和第二GPT表。第一GPT表用于CPU和其他外设对内存的访问。在第一GPT表中,第一区段属于第一领域,具有领域世界的权限。非安全世界的应用、其他领域的应用,均无权访问该第一区段,由此实现领域间的隔离。此外,为了保护FPGA内存映射MMIO区段,禁止不受信的软件栈访问该区域,可以在上述第一版本GPT表中将其设置为属于根世界。根据图2所示的不同世界的权限,通过CPU或其他外设请求内存访问的各种应用,包括安全世界中的软件,均不能访问上述第一区段和FPGA内存映射MMIO区段。

[0080] 第二GPT表是用于前述FPGA的针对第一领域的GPT表,该GPT表可以在创建第一领域时生成并初始化,其中,仅将第一领域对应的第一区段设置为可访问区段。当第一领域执行FPGA机密计算时,根监视器向控制FPGA的SMMU(一种内存访问控制硬件,支持GPC)写入第

一GPT表的地址,使其按照第一GPT表进行GPC检查,从而实现对来自FPGA的内存访问的控制。

[0081] 进一步的,在一个实施例中,为了简化权限状态划分,对于每个“领域”的FPGA GPT表,可以将各自的领域内存标记为“非安全”状态,将其他内存空间标记为“根”状态,以此仅允许FPGA硬件访问其对应领域的内存。也就是说,对于第一领域的FPGA,将对应的第一GPT表设置为,第一领域的第一区段为“非安全”状态,可以访问;其他所有区段均为“根”状态,均不能访问。如此,用于FPGA的GPT表的配置仅需要两种状态(即“非安全”和“根”状态,分别表示FPGA可访问和不可访问的区域),而不需要特定地识别其他状态(如安全和领域状态)的内存,极大简化FPGA GPT配置的过程,进一步优化性能开销。

[0082] 图6示出在一个示例场景中,根监视器维护的GPT表。如图6所示,在该示例场景中,领域世界中至少包括领域R1和领域R2。假定两个领域各自的用户分别要求,基于领域R1执行FPGA任务1,基于领域2执行FPGA任务2,与图3对应。其中,结合图4描述的在第一领域中执行的FPGA任务,假定对应于在领域R1中执行的FPGA任务1。

[0083] 为了给FPGA任务1和FPGA任务2分别提供隔离的执行环境,根监视器至少要维护图6所示的4个GPT表。

[0084] 在针对CPU的GPT表中,领域R1和领域R2对应的内存区段照常属于领域世界区段。此外,FPGA内存映射MMIO区段被设置为属于根世界。

[0085] 针对不可信外设的GPT表与针对CPU的GPT表总体相似,其中关于领域R1,领域R2的设置完全相同。不同的是,外设具有其对应的内存访问限制,CPU可以访问的部分内存区段(例如图中最前面的一小段),在外设的GPT表被设置为属于根世界,外设无权访问。

[0086] 针对领域1的FPGA GPT表,是FPGA执行领域1对应的FPGA任务1时适用的GPT表,即前述的第二GPT表。在该表中,领域R1被设置为属于非安全世界,可以访问;所有其他区段均被设置为属于根世界,不能访问。这意味着,在FPGA执行领域1对应的FPGA任务1时,仅仅可以访问领域R1的内存数据,无权访问任何其他区段数据。

[0087] 针对领域2的FPGA GPT表,是FPGA执行领域2对应的FPGA任务2时适用的GPT表。在该表中,领域R2被设置为属于非安全世界,可以访问;所有其他区段均被设置为属于根世界,不能访问。这意味着,在FPGA执行领域2对应的FPGA任务2时,仅仅可以访问领域R2的内存数据,无权访问任何其他区段数据。

[0088] 当硬件(CPU,FPGA或外设)请求访问内存时,硬件层中的RME根据对应适用的GPT表进行GPC检查,从而进行内存访问控制。

[0089] 通过针对领域1和领域2的两个FPGA GPT表可见,不同领域的FPGA任务之间,也进行了内存隔离,确保执行环境的安全。可以理解,如果需要基于更多领域执行更多FPGA任务,则需要维护更多的GPT表。

[0090] 通过以上设置,可以执行步骤S44,根监视器通过FPGA内存映射MMIO与FPGA进行交互,使得FPGA基于所述描述符读取所述机密数据,执行FPGA任务。

[0091] 具体地,在基于前述GPT表进行GPC的内存访问控制下,根世界将代替原有的软件栈与FPGA MMIO进行交互,执行FPGA传输任务,即向特定的寄存器提供传输任务的来源地址(包括描述符的地址),写入控制指令,实现数据发送与运行结果接收。具体的,在需要FPGA页表的情况下,首先将FPGA页表基地址指向存储于第一领域中的真实FPGA页表,如此,用真

实FPGA页表替换桩页表,基于真实PFGA页表执行内存访问。一些FPGA任务也可不需要FPGA页表而执行。此时,根监视器通过FPGA MMIO与FPGA进行指令交互,使得FPGA基于描述符读取机密数据,执行FPGA任务。

[0092] 为确保FPGA执行过程中数据的机密性,在如上所述执行FPGA任务之后,令FPGA将其片上内存中最近使用的内存完全清除,之后才会取消上述执行环境的隔离保护,例如,将FPGA MMIO的权限恢复为常规状态。

[0093] 回顾以上过程,通过传输任务实现兼容Arm机密计算架构CCA的FPGA机密计算。根据传输任务机制,由非安全世界的FPGA软件创建不包含真实数据的桩任务,并如常规流程调度和管理桩任务。在桩任务提交后,根监视器创建包含真实数据的真实FPGA任务,生成真实的描述符,并通过设置GPT表提供受保护的执行环境。然后,根监视器通过受保护的FPGA MMIO与FPGA交互,使得FPGA硬件在受保护的执行环境中根据描述符读取数据,执行真实FPGA任务。如此,在Arm机密计算架构CCA中实现FPGA机密计算。

[0094] 另一方面,与上述方法过程相对应的,本说明书实施例还披露一种机密计算架构中的根监视器,所述机密计算架构包括,安全世界,领域世界,非安全世界和根世界;所述根监视器位于所述根世界中。根监视器可以包括传输任务模块和FPGA保护模块。

[0095] 传输任务模块配置为,响应于非安全世界的FPGA软件根据FPGA的数据需求信息,在内存的非安全世界区段中配置FPGA任务的桩数据结构,而在所述内存中对应于第一领域的第一区段中,配置与所述桩数据结构对应的真实数据结构,其中所述桩数据结构包括包括数据缓存区,所述真实数据结构包括,与所述数据缓存区对应的领域缓存区,以及传输描述符;所述领域缓存区用于存储待处理的机密数据;所述传输描述符用于描述有待进行直接内存访问DMA传输的数据的存储信息。

[0096] FPGA保护模块配置为,设置用于CPU的第一GPT表和用于FPGA的第二GPT表,使得在第一GPT表中,FPGA内存映射MMIO属于根世界,在第二GPT表中,仅第一区段为可访问区段;通过FPGA内存映射MMIO与FPGA进行交互,使得FPGA基于所述描述符通过DMA传输读取所述机密数据,执行FPGA任务。

[0097] 传输任务模块和FPGA保护模块的具体执行过程示例,可以参照之前结合图4和图5的描述,不复赘述。

[0098] 在典型实施例中,所述根监视器实现为安全固件。

[0099] 根据再一方面的实施例,还提供一种计算设备,包括存储器和若干处理器,所述计算设备形成机密计算架构,所述机密计算架构包括,安全世界,领域世界,非安全世界和根世界;所述根世界包括前述的根监视器。

[0100] 本领域技术人员应该可以意识到,在上述一个或多个示例中,本发明所描述的功能可以用硬件、软件、固件或它们的任意组合来实现。当使用软件实现时,可以将这些功能存储在计算机可读介质中或者作为计算机可读介质上的一个或多个指令或代码进行传输。

[0101] 以上所述的具体实施方式,对本发明的目的、技术方案和有益效果进行了进一步详细说明,所应理解的是,以上所述仅为本发明的具体实施方式而已,并不用于限定本发明的保护范围,凡在本发明的技术方案的基础之上,所做的任何修改、等同替换、改进等,均应包括在本发明的保护范围之内。

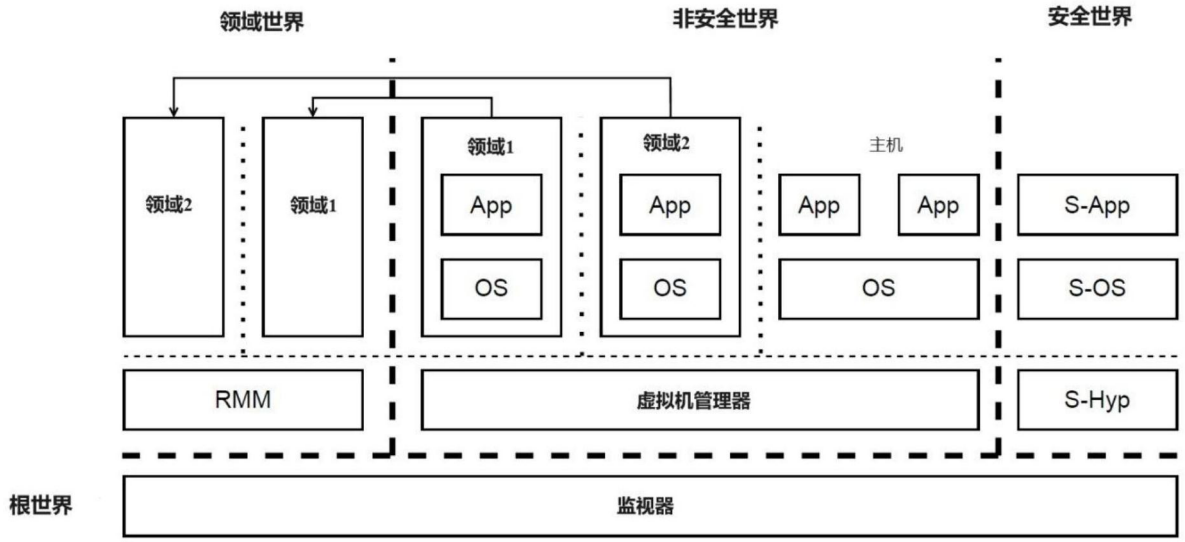


图1

安全状态	非安全 PAS	安全 PAS	领域 PAS	根 PAS
非安全	✓	×	×	×
安全	✓	✓	×	×
领域	✓	×	✓	×
根	✓	✓	✓	✓

图2

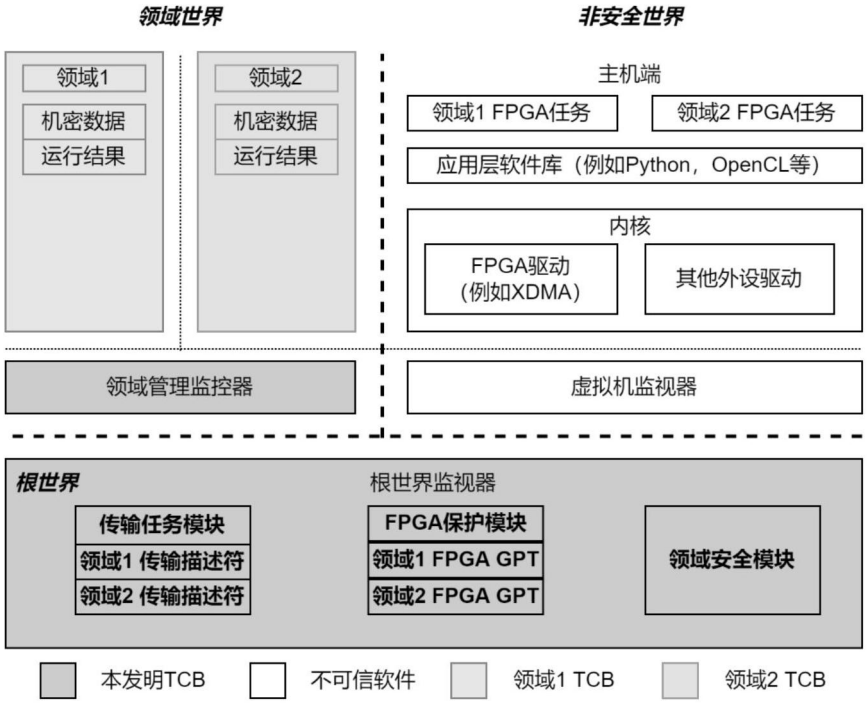

领域世界
非安全世界
领域1
机密数据
运行结果
领域2
机密数据
运行结果
主机端
领域1 FPGA任务
领域2 FPGA任务
应用层软件库（例如Python，OpenCL等）
内核
FPGA驱动
（例如XDMA）
其他外设驱动
领域管理监控器
虚拟机监视器
根世界
根世界监视器
传输任务模块
领域1 传输描述符
领域2 传输描述符
FPGA保护模块
领域1 FPGA GPT
领域2 FPGA GPT
领域安全模块
本发明TCB
不可信软件
领域1 TCB
领域2 TCB

图3

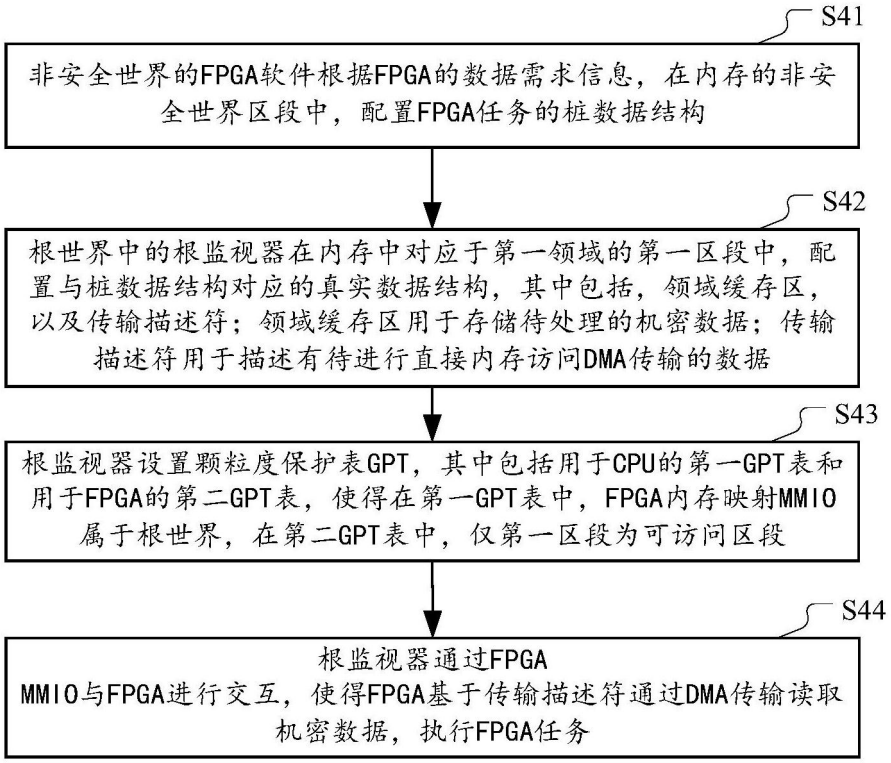

S41
非安全世界的FPGA软件根据FPGA的数据需求信息，在内存的非安全世界区段中，配置FPGA任务的桩数据结构
S42
根世界中的根监视器在内存中对应于第一领域的第一区段中，配置与桩数据结构对应的真实数据结构，其中包括，领域缓存区，以及传输描述符；领域缓存区用于存储待处理的机密数据；传输描述符用于描述有待进行直接内存访问DMA传输的数据
S43
根监视器设置颗粒度保护表GPT，其中包括用于CPU的第一GPT表和用于FPGA的第二GPT表，使得在第一GPT表中，FPGA内存映射MMIO属于根世界，在第二GPT表中，仅第一区段为可访问区段
S44
根监视器通过FPGA MMIO与FPGA进行交互，使得FPGA基于传输描述符通过DMA传输读取机密数据，执行FPGA任务

图4

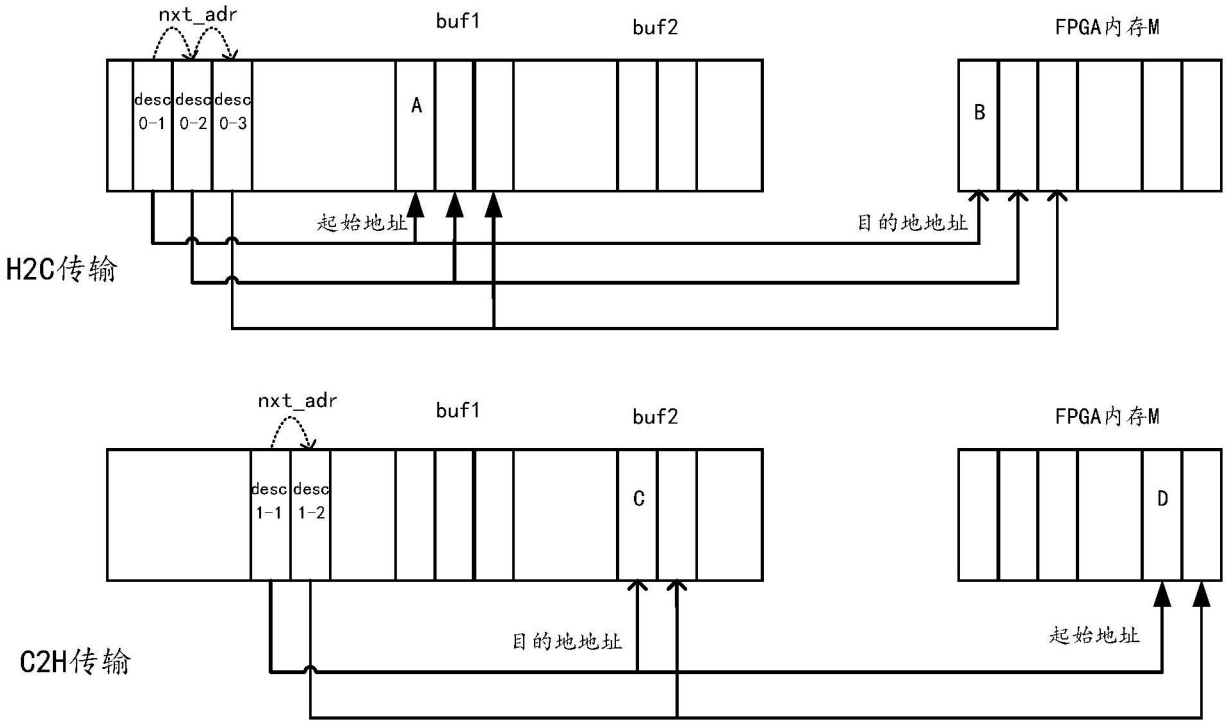

nxt_adr
buf1
buf2
FPGA内存M
desc 0-1
desc 0-2
desc 0-3
A
B
起始地址
目的地地址
H2C传输
nxt_adr
buf1
buf2
FPGA内存M
desc 1-1
desc 1-2
C
D
目的地地址
起始地址
C2H传输

图5

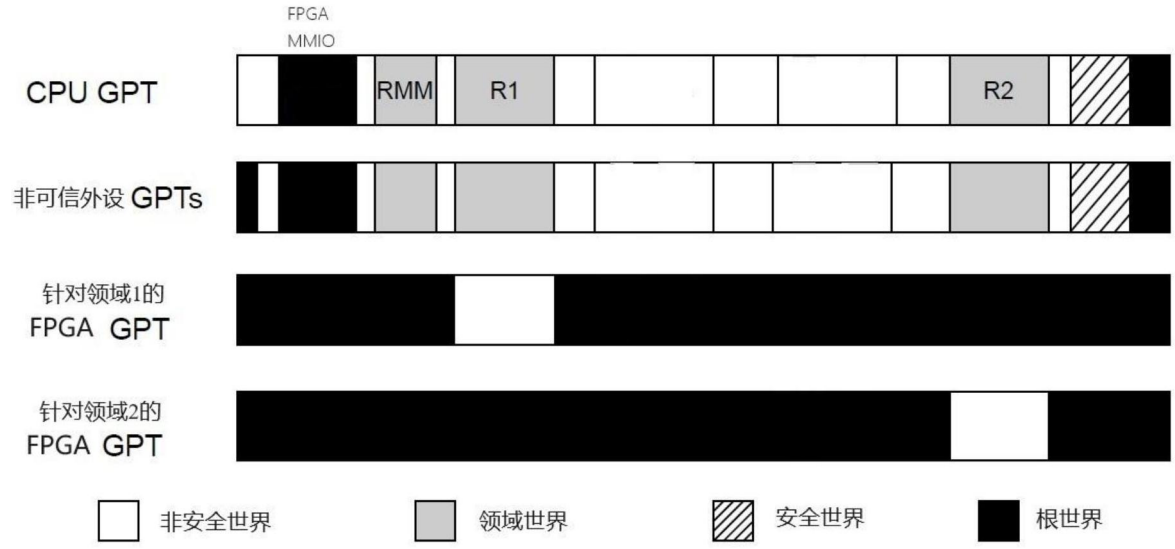

FPGA MMIO
CPU GPT
RMM
R1
R2
非可信外设 GPTs
针对领域1的 FPGA GPT
针对领域2的 FPGA GPT
非安全世界
领域世界
安全世界
根世界

图6