



(12) 发明专利申请

(10) 申请公布号 CN 103544417 A

(43) 申请公布日 2014. 01. 29

(21) 申请号 201310247403. 1

(22) 申请日 2013. 06. 20

(30) 优先权数据

13/528, 438 2012. 06. 20 US

(71) 申请人 微软公司

地址 美国华盛顿州

(72) 发明人 B·A·拉马恰 E·B·南丁格尔

(74) 专利代理机构 上海专利商标事务所有限公司 31100

代理人 胡利鸣

(51) Int. Cl.

G06F 21/31 (2013. 01)

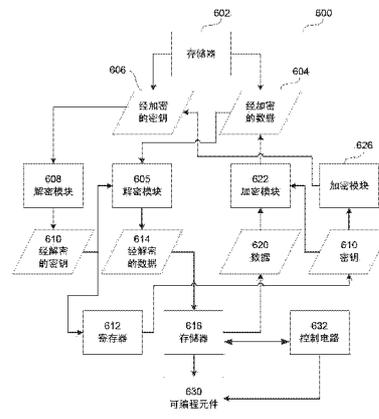
权利要求书3页 说明书7页 附图6页

(54) 发明名称

用可重新编程的密码操作来管理对现场可编程门阵列的使用

(57) 摘要

描述了用可重新编程的密码操作来管理对现场可编程门阵列的使用。描述了对操作系统中多个进程对现场可编程门阵列的使用的管理。现场可编程门阵列能在通用计算系统中用作共享可编程协作处理器资源。FPGA 的组件被隔离来保护 FPGA 以及 FPGA 和计算机系统的其它组件之间传输的数据。被传输的数据可由 FPGA 或其它组件来数字地签名以提供认证。用于编程 FPGA 的代码可由作者来加密并签名、以被加密的状态加载到 FPGA 中并接着在用该代码编程 FPGA 之前由 FPGA 自己来解密和认证。该代码可被用于改变在 FPGA 中执行的密码操作,包括密钥、或解密和加密算法或两者。



1. 一种计算机系统,包括:

具有作为协作处理器的现场可编程门阵列的处理器,所述处理器具有用于编程所述现场可编程门阵列的安全通道;

其中所述现场可编程门阵列包括可编程元件;

存储用于所述现场可编程门阵列的程序库的存储器,所述程序库包括用于执行密码操作的一个或多个程序库;

其中,在选择程序库后,所述现场可编程门阵列被安全地重新编程以根据所选的程序库来执行密码操作。

2. 如权利要求 1 所述的计算机系统,其特征在于,安全地重新编程包括提供经加密的程序逻辑。

3. 如权利要求 2 所述的计算机系统,其特征在于,所述现场可编程门阵列在所述现场可编程门阵列内对经加密的程序逻辑进行解密,并使用经解密的程序逻辑来对所述可编程元件进行编程。

4. 如权利要求 3 所述的计算机系统,其特征在于,经加密的程序逻辑包括嵌入在其中的经加密的密钥,使得当被解密时,该密钥是经解密的程序逻辑的一部分。

5. 如权利要求 3 所述的计算机系统,其特征在于,还包括提供经加密的密钥,其中所述现场可编程门阵列在所述现场可编程门阵列内解密所述经加密的密钥,并使用经解密的密钥来解密经加密的程序逻辑。

6. 如权利要求 3 所述的计算机系统,其特征在于,经加密的程序逻辑被数字地签名并且所述现场可编程门阵列在对所述可编程元件进行重新编程之前认证经加密的程序逻辑。

7. 如权利要求 1 所述的计算机系统,其特征在于,所述现场可编程门阵列包括:

第一存储器,所述第一存储器用于接收包括经加密的程序逻辑和至少一个经加密的密钥的输入数据;

第一解密模块,所述第一解密模块具有输入以接收所述至少一个经加密的密钥、对应于所述经加密的密钥的发送者的公钥以及与所述现场可编程门阵列相关联的私钥,并具有提供经解密的密钥的输出;

第二解密模块,所述第二解密模块具有输入以接收来自所述第一解密模块的经解密的密钥以及来自所述存储器的经加密的程序逻辑,并具有提供经解密的程序逻辑的输出;

第二存储器,所述第二存储器用于接收经解密的程序逻辑;以及

控制电路,所述控制电路用于使用来自所述第二存储器的经解密的程序逻辑来对所述现场可编程门阵列的可编程元件进行编程。

8. 如权利要求 1 所述的计算机系统,其特征在于,所述现场可编程门阵列包括:

存储器访问通道,包括:

加密模块,所述加密模块具有输入以接收来自所述现场可编程门阵列的处理元件的数据以及将经加密的数据提供到存储器的输出;以及

解密模块,所述解密模块具有输入以接收来自所述存储器的数据以及将经解密的数据提供到所述现场可编程门阵列内的处理元件的输出。

9. 一种现场可编程门阵列,包括:

多个可编程处理元件,包括被编程以执行密码操作的元件;

多个用于存储密码秘密的隔离的寄存器,所述密码秘密被用在由所述处理元件执行的密码操作中,使得由所述现场可编程门阵列接收的经加密的数据只能由所述现场可编程门阵列使用所述密码秘密来解密;以及

加载机制,所述加载机制用于安全地更新由经编程的处理元件执行的密码操作。

10. 如权利要求 9 所述的现场可编程门阵列,其特征在于,所述加载机制安全地更新所述可编程处理元件来执行不同的密码操作。

11. 如权利要求 9 所述的现场可编程门阵列,其特征在于,所述加载机制安全地更新存储在隔离的寄存器中的密码秘密。

12. 如权利要求 9 所述的现场可编程门阵列,其特征在于,所述加载机制更新在程序逻辑中编译的密码秘密以用于对可编程处理元件进行编程。

13. 如权利要求 9 所述的现场可编程门阵列,其特征在于,所述加载机制包括:

第一存储器,所述第一存储器用于接收包括经加密的程序逻辑和至少一个经加密的密钥的输入数据;

第一解密模块,所述第一解密模块具有输入以接收所述至少一个经加密的密钥、对应于所述经加密的密钥的发送者的公钥以及与所述现场可编程门阵列相关联的私钥,并具有提供经解密的密钥的输出;

第二解密模块,所述第二解密模块具有输入以接收来自所述第一解密模块的经解密的密钥以及来自所述存储器的经加密的程序逻辑,并具有提供经解密的程序逻辑的输出;

第二存储器,所述第二存储器用于接收经解密的程序逻辑;以及

控制电路,所述控制电路用于使用来自所述第二存储器的经解密的程序逻辑来对所述现场可编程门阵列的可编程元件进行编程。

14. 如权利要求 9 所述的现场可编程门阵列,其特征在于,所述加载机制包括:

存储器访问通道,包括:

加密模块,所述加密模块具有输入以接收来自所述现场可编程门阵列的处理元件的数据以及将经加密的数据提供到存储器的输出;以及

解密模块,所述解密模块具有输入以接收来自所述存储器的数据以及将经解密的数据提供到所述现场可编程门阵列内的处理元件的输出。

15. 一种用于编程现场可编程门阵列的方法,包括:

将用于密码操作的经加密的程序逻辑接收到存储器中;

将经加密的程序逻辑解密到所述现场可编程门阵列内的存储器中;以及

使用经解密的程序逻辑来对所述现场可编程门阵列进行编程以在所述现场可编程门阵列中实现所述密码操作。

16. 如权利要求 15 所述的方法,其特征在于,还包括:

将用于经加密的程序逻辑的经加密的密钥接收到存储器中;

在所述现场可编程门阵列中解密所述经加密的密钥;以及

在对加密程序逻辑解密时使用经解密的密钥。

17. 如权利要求 16 所述的方法,其特征在于,使用所述现场可编程门阵列的公钥以及所述经加密的程序逻辑的提供者的私钥来对所述经加密的密钥进行加密。

18. 如权利要求 17 所述的方法,其特征在于,所述经加密的程序逻辑是使用对称密钥

来加密的。

19. 如权利要求 16 所述的方法,其特征在于,所述经加密的程序逻辑是使用对称密钥来加密的。

20. 如权利要求 15 所述的方法,其特征在于,所述经加密的程序逻辑包括用于所述密码操作的密码秘密。

## 用可重新编程的密码操作来管理对现场可编程门阵列的使用

### 技术领域

[0001] 本申请涉及用可重新编程的密码操作来管理对现场可编程门阵列的使用。

### 背景技术

[0002] 在大多数通用计算机内,操作系统是管理对计算机内资源的访问的主要软件。主要资源是执行被设计成在计算机上运行的应用程序的中央处理单元(CPU)、主存储器和存储。在一些计算机体系结构中,可出现附加的处理单元(诸如处理器中的多个核)和/或附加的处理器(称为协作处理器)。这样的协作处理器的示例包括图形处理单元(GPU)和数字信号处理器(DSP)。操作系统也管理多个进程对这些资源的访问。

[0003] 现场可编程门阵列(FPGA)是一种通常被用在专用计算设备中的逻辑器件。FPGA 通常被用于执行此门阵列尤其适用于的特定的、专用的功能。FPGA 通常位于外围设备或其它专用硬件(诸如连接到诸如 PCI 总线的系统总线并通过该系统总线被访问的印刷电路板)中。一般而言,这样的器件被编程一次并被使用多次。因为这些器件是可编程的,相比于其它专用逻辑器件,它们具有能被在现场更新的优势。

### 发明内容

[0004] 提供本发明内容以便以简化形式介绍将在以下具体实施方式中进一步描述的一些概念。本发明内容并不旨在标识所要求保护主题的关键特征或必要特征,也不旨在用于限制所要求保护主题的范围。

[0005] 一个或多个现场可编程门阵列(FPGA)能在通用计算系统中用作共享可编程协作处理器资源。FPGA 能被编程来执行功能,这些功能进而能与一个或多个进程相关联。在多个进程的情况下,FPGA 能被共享,并且进程能在访问 FPGA 的时间间隙期间被分配到 FPGA 的至少一个部分。用硬件描述语言所写的用于编程 FPGA 的程序被用作硬件库。操作系统对以下进行管理:将 FPGA 资源分配到进程、根据要由进程使用 FPGA 来执行的功能来编程该 FPGA、以及对这些进程对 FPGA 的使用进行调度。

[0006] 如果 FPGA 被用作通用计算平台的组件,那么其可能容易受到不安全代码的攻击和执行。例如,对数据传输和存储器的检查可能暴露与安全操作相关的密钥、算法和其它信息。为了提升安全性,FPGA 的各个组件被隔离来保护 FPGA 以及 FPGA 和计算机系统的其它组件之间传输的数据。

[0007] 例如,由 FPGA 写入到存储器的数据被加密,并在从存储器中读回时在 FPGA 中被解密。在 FPGA 和诸如 CPU 或 GPU 等其它组件之间传输的数据(无论是直接地或通过存储器的)可使用为通信中的组件所知的密钥(无论是使用共享秘密密钥还是公钥/私钥对)来被类似地加密。被传输的数据还可由 FPGA 或其它组件来数字地签名以提供认证。用于编程 FPGA 的代码可由作者来加密并签名、以被加密的状态加载到 FPGA 中并接着在用该代码编程 FPGA 之前由 FPGA 自己来解密和认证。密钥还可被嵌入在用于编程 FPGA 的已编译的程

序逻辑中。

[0008] 在 FPGA 内执行的密码操作能使用 FPGA 的可编程元件来实现并可被重新编程来提供灵活的密码环境。密钥、或解密和加密算法之一或它们两者能被重新编程。

[0009] 在以下描述中,对附图进行了参考,附图构成了实施方式的一部分且在其中作为示例示出了本发明技术的具体示例实现。可以理解,可以使用其它实施例并且可以做出结构上的改变而不背离本发明的范围。

[0010] 附图描述

[0011] 图 1 是对其操作系统能被实现的具有 FPGA 资源的示例计算系统的框图。

[0012] 图 2 是 FPGA 功能单元的说明性示例的示意图。

[0013] 图 3 是使用具有 FPGA 资源的计算机系统上的硬件和软件库的应用的示例体系结构的示意图。

[0014] 图 4 是示出随着时间的对 FPGA 使用的图。

[0015] 图 5 是具有支持隔离组件的现场可编程门阵列的计算系统的框图。

[0016] 图 6 是现场可编程门阵列的更详细的框图。

[0017] 图 7 是描述用新的密码操作来安全地编程 FPGA 的流程图。

## 具体实施方式

[0018] 以下部分提供了对示例计算环境的简要的、一般的描述,在该示例计算环境中能实现用于管理对 FPGA 资源的使用的操作系统。该系统可以用众多通用或专用计算设备来实现。适合的公知计算设备的示例包括但不限于:个人计算机、服务器计算机、手持式或膝上型设备(例如,媒体播放器、笔记本计算机、蜂窝电话、个人数据助理、语音记录器)、多处理器系统、基于微处理器的系统、机顶盒、可编程消费电子产品、网络 PC、小型机、大型计算机、包括以上系统或设备的任一个的分布式计算环境等等。

[0019] 图 1 仅仅示出示例计算环境,并不旨在对适合的计算环境的使用范围或功能提出任何限制。

[0020] 参考图 1,示例计算环境包括计算设备 100。在一个基本配置中,计算设备 100 包括至少一个处理单元 102(诸如通用计算机的典型中央处理单元(CPU))和存储器 104。

[0021] 计算设备可包括多个处理单元和/或附加的协作处理单元,诸如图形处理单元(GPU)。计算设备还包括一个或多个现场可编程门阵列(FPGA),其被表示为可用作共享(在运行在计算机上的进程间共享)的协作处理资源的 FPGA 单元 120。FPGA 可位于其自己的 CPU 插孔中或位于分开的被插入到扩展槽(诸如快速外围部件互连(PCI-E)槽)中的卡上。通过提供这样的 FPGA 单元,能在得到硬件加速的益处的前提下实现各种非常适合于门阵列来实现的功能。

[0022] 取决于处理单元和 FPGA 单元的配置,该单元或单元内的每个功能单元具有相关联的输入/输出通道来用于与主操作系统进程进行通信。例如,能提供专用于该功能单元并在其与使用该功能单元的进程之间共享的存储器区域。一种请求队列和响应队列还能被用于使得能够实现在 FPGA 单元内实现的操作的异步调用。此外,FPGA 单元中的功能单元针对进程的状态能被保存到用于该功能单元和该进程的存储器区域并从该存储器区域中还原。替换地,其它技术能被用于确保功能单元在被其进程使用前处于已知状态。

[0023] 取决于计算设备的配置和类型,存储器 104 可以是易失性的(诸如 RAM)、非易失性的(诸如 ROM、闪存等)或是两者的某种组合。处理单元、协作处理器和存储器的该配置在图 1 中用虚线 106 示出。

[0024] 计算设备 100 还可具有附加的资源和设备。例如,计算设备 100 还可包含附加存储(可移动和 / 或不可移动),包括但不限于磁盘、光盘或磁带。在图 1 中通过可移动存储 108 和不可移动存储 110 示出这样的附加存储。计算机存储介质包括以用于存储诸如计算机程序指令、数据文件、数据结构、程序模块或其他数据等信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。存储器 104、可移动存储 108 和不可移动存储 110 全部都是计算机存储介质的示例。计算机存储介质包括但不限于, RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘(DVD)或其它光存储、磁带盒、磁带、磁盘存储或其它磁存储设备,或者可用于存储所需信息并且可由计算设备 100 访问的任何其它介质。任何这样的计算机存储介质都可以是计算设备 100 的一部分。

[0025] 计算设备 100 还可包括通信连接 112,其允许设备通过通信介质与其它设备进行通信。通信连接 112 的实现是取决于正由计算设备访问的通信介质的种类的,这是因为其提供了对这样的介质的接口以允许通过该通信介质的数据的传输和 / 接收。通信介质通常承载诸如载波或其他传输机制等已调制数据信号中的计算机程序指令、数据文件、数据结构、程序模块或其他数据,并包括任何信息传递介质。术语已调制数据信号意指其一个或多个特征以这样的方式设置或改变以便在信号中对信息进行编码的信号。作为示例而非限制,通信介质包括诸如有线网络或直接线连接之类的有线介质,以及诸如声学、RF、红外及其他无线介质之类的无线介质。

[0026] 计算设备 100 可具有各种输入设备 114,如键盘、鼠标、笔、相机、触摸输入设备等。还可包括诸如显示器、扬声器、打印机等输出设备 116。所有这些设备在本领域中是公知的并且不必在此详细讨论。

[0027] 使用由计算设备处理的诸如程序模块等计算机可执行指令和 / 或计算机解释的指令来实现在计算设备上执行的应用。一般而言,程序模块包括在由处理单元处理时指示处理单元执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。在分布式计算环境中,这样的任务能由通过通信网络链接的远程处理设备来执行。在分布式计算环境中,程序模块可以位于包括存储器存储设备在内的本地和远程计算机存储介质中。

[0028] 在计算设备上执行的操作系统管理进程对计算设备的各种资源的访问。通常,在计算机系统上运行应用导致一个或多个进程被创建,其中每个进程随着时间被分配到不同的资源。如果资源在进程间共享,并且如果进程不能并发地共享资源,那么操作系统随着时间调度对资源的访问。这样的资源之一是图 1 的 FPGA 单元 120,其可包括一个或多个分立的 FPGA。

[0029] 参考图 2,FPGA 单元内的资源之一是一组或多组可编程门,在此称为功能单元。每个功能单元通过一组门和 / 或门阵列中的其它资源来定义。一般而言,功能单元是不重叠的,即,不共享门阵列中的可编程元件。例如,如图 2 中示意地示出的,功能单元 200、202、204 和 206 是不重叠的。大多数 FPGA 只有一个功能单元。然而,图 1 中的 FPGA 单元 120 可具有一个或多个 FPGA。在多个 FPGA 的情况下,每个 FPGA 可被视为功能单元。参考图 3,每

个功能单元是以下资源：其能被分配给一个或多个进程、被操作系统使用实现一操作的硬件库来编程并接着被分配给其的进程用于执行该操作。参考图 3，作为一个示例，应用 300 可使用传统的软件库 302 以及 FPGA 硬件库 304 来执行各种操作。如果应用依赖硬件库 304，则操作系统 306 使用该硬件库来编程 FPGA 资源 310 以允许应用 300 使用库。FPGA 可在应用开始执行之前被编程。如果 FPGA 可被足够快地重新编程，那么库可在操作系统的调度量子 (quantum) 内被加载到 FPGA 中。操作系统 306 还执行来自应用 300 和 CPU308 上的软件库 302 的软件命令。当应用作出对由软件库执行的功能的调用时，操作系统执行来自 CPU308 上的软件库的功能。当应用作出对由 FPGA 执行的功能的调用时，操作系统确保 FPGA 是使用硬件库来编程的并使用 FPGA 来执行功能。

[0030] 为了示出不同的功能单元能随着时间如何被使用，现在参考图 4。在图 4 中，在时间 T1，使用功能单元 400 和 402。在时间 T2，使用功能单元 400 和 404。在时间 T2，再次使用功能单元 400 和 402。在时间 T1，功能单元 400 能被分配给进程 P1，而功能单元 402 能被分配给进程 P2。在时间 T2，进程 P2 可能是不活动的，而进程 P1 能使用功能单元 400 并且进程 P3 能使用功能单元 404。在时间 T3，另一进程 (诸如进程 P4) 能开始使用功能单元 400；并且进程 P2 能再次活动来使用功能单元 402。通过当前的 FPGA 实现，在同一时间由不同的进程对多个功能单元的使用暗示多个 FPGA 的使用。就 FPGA 能支持由不同的进程在同一时间使用的多个功能单元而言，这些功能单元能在同一 FPGA 上。实际上，操作系统在时间和空间方面在统计学上复用 FPGA。

[0031] 为了允许这种随着时间由不同的进程对 FPGA 资源的使用，操作系统具有调度器，该调度器确定在每个调度量子 (即，时间段) 哪个进程能访问 FPGA 资源以及何时 FPGA 功能单元将用硬件库来编程使得功能单元可用于由该进程使用。由此，用于 FPGA 单元的调度器的实现部分地取决于 FPGA 单元的性质以及其包括的一个或多个 FPGA。要考虑的与 FPGA 有关的因素包括但不限于以下。例如，在一些情况下，如果一个功能单元不能独立于其它功能单元而被编程，那么整个 FPGA 要被刷新来编程功能单元。另一考虑是功能单元能被编程的速度以及功能单元的编程是否阻止其它功能模块在编程阶段期间被使用。要考虑的另一因素是进程是否能通过共享功能单元来共享硬件库。调度器还考虑诸如以下的因素：并发进程的数量、应用性能保证、应用的优先级、进程上下文切换花费、对存储器和总线的访问以及在无功能单元在 FPGA 单元中可用的情况下软件库的可用性。

[0032] 可以存在其它情况，其中 FPGA 单元向应用或操作系统提供通用设施，其因此被调度为应用实例化的长度。例如，自定义网络协议或卸载可作为 FPGA 单元上的加速服务来提供。相反，一般在通用 CPU 中执行的系统调用或标准库调用能使用 FPGA 来被加速。此外，操作系统能基于进程优先级的偏好来复用 CPU。在另一情况中，操作系统能使用应用的简档 (统计地或动态地生成) 来预测最适合于在 FPGA 单元上运行的功能并接着预先加载该功能，使得其可用于调度。通过将简档用作向导，操作系统能确保空间和时间均在 FPGA 单元上可用来加速应用。最终，操作系统能使用来自应用的简单提示来知道何时在 FPGA 单元上调度时间。例如，某些到操作系统内的调用 (系统调用) 可指示长的延迟 (对盘或网络的调用)，其提供了 FPGA 单元能空闲某一时间量来供其它线程或进程使用的提示。因此，操作系统使用各种提示和偏好来创建对复用对 FPGA 单元的访问的调度。由于操作系统控制调度器，因此其具有关于正在执行和即将到来的工作、可用的硬件库以及在编程 FPGA 所花费的

时间的详细知识。因此,它能使用该知识来确定在执行期间哪些进程利用 FPGA。

[0033] 现在已经描述了这样的计算机体系结构的一般概览,现在将描述示例实现。

[0034] 参考图 5,显示了使用具有隔离组件的现场可编程门阵列 502 的计算机系统 500 的一般体系结构。在该示例中,FPGA 连接到存储器 504、中央处理单元 506 和图形处理单元 508。这种连接是通过传统的高速计算机总线 510 来提供的,诸如具有超传输总线的 CPU 插槽、PCI、PCI-E 或 PCI-X 总线。

[0035] 现场可编程门阵列可包括一个或多个寄存器,该一个或多个寄存器包括密钥(诸如对称密钥或公钥 / 私钥对)。它还包括使用那些密钥来执行对应的密码操作的能力。密码组件可以是对 FPGA 的可编程元件进行编程的一部分。这些组件可用对策(countermeasure)来实现以增加对芯片进行直接分析(这诸如能使用受信平台模块(TPM)组件来实现)的难度。

[0036] 在一个实现中,密钥能被存储在 TPM 组件中,其中 FPGA 仅能在密钥被使用时从该 TPM 组件中加载这样的密钥。如果 TPM 能访问由 FPGA 持有的公共 / 私有对中的公钥,那么 TPM 能使用 FPGA 的公钥来加密它对 FPGA 所持有的密钥。由此,密钥本身只在从 TPM 传送到 FPGA 之后被解密。这种配置允许被加密的密钥通过不安全的总线(诸如标准 PC 高速互连)来传送。

[0037] 通过以下方式在 FPGA502 和主存储器 505 之间创建逻辑通道:在所有数据离开 FPGA 之前用对称密钥对所有数据进行加密,将经加密的数据存储在主存储器中。随后,当经加密的数据从主存储器读回到 FPGA 中时,用 FPGA 内的对称密钥对经加密的数据进行解密。在一个实现中,对数据的加密还可包括完整性保护。例如,可使用用于对称密码的经认证的加密操作模式。作为另一示例,数据可被散列并且散列值可被附加到该数据,并接着具有附加的散列值的数据能在被写入到主存储器之前被加密。

[0038] 通过相互认证和密钥传输协议,在 FPGA502 和图形处理单元(GPU)508 或其它组件(诸如 CPU 或外围设备)之间创建逻辑通道。在这种情况下,FPGA 使用公钥 / 私钥对来向组件(例如,GPU)认证自己,并且该组件使用第二公钥 / 私钥对(其中私钥仅仅为 GPU 所知)来这样做。作为相互认证过程的一部分,FPGA 和 GPU 建立一个或多个共享秘密(例如,两个共享秘密,一个用于完整性保护而一个用于机密性)。这些共享秘密接着被用于作为 FPGA 和 GPU 之间安全会话的一部分,对这两个组件之间随后的通信进行加密和认证。

[0039] 现在参考图 6,现在显示了提供隔离组件的现场可编程门阵列的更多细节。

[0040] FPGA600 包括输入 / 输出存储器 602,经加密的数据 604 和经加密的密钥 606 通过该存储器来传输。

[0041] 当被从其它设备接收到时,经加密的数据 604 通过解密模块 605(其可实现例如对称密钥密码操作)被解密。在一些情况下,经加密的密钥 606(其可由解密模块 605 使用)被接收并通过解密模块 608(其可实现例如公钥 / 私钥密码操作)来解密。经解密的密钥 610 可被存储在寄存器 612 中。在其他情况下,经解密的数据 614 可被存储在存储器 616 中。

[0042] 当传输到其它设备时,数据 620(诸如从存储器 616)由加密模块 622(其可实现例如对称密钥密码操作)加密以提供经加密的数据 604。加密模块 622 可使用存储在寄存器 612 中的密钥 610。在一些情况下,加密模块 626(其可实现例如公钥 / 私钥密码操作)可对加密模块 622 使用的密钥 610 进行加密以作为经加密的密钥 606 来传输。经加密的数据

604 和密钥 606 可在传输到计算机系统内的另一组件(诸如存储器、GPU、CPU、外围卡或其它设备)之前被存储在存储器 602 中。

[0043] 存储器 616 一般可由 FPGA 的可编程元件 630 来访问以用于对数据的读取和写入两者。有可能具有一些仅仅能被可编程元件读取但不能被修改的寄存器。

[0044] 在存储器 616 中接收到的数据还可以是用于编程 FPGA 单元的功能单元的编程代码。控制电路 632 从存储器 616 中读取编程代码并对可编程元件 630 进行编程。如将在以下更加详细描述,这样的结构允许经加密和签名的代码被安全地下载到 FPGA,在 FPGA 处其被认证和解密,接着被用于编程 FPGA。

[0045] 在一些实现中,可使用 FPGA 的可编程元件来实现解密模块和加密模块中的各种密码操作。可使用可编程元件来实现附加的解密和加密模块以在 FPGA 内提供进一步的加密和解密能力。

[0046] 在给定这样的结构的情况下,FPGA 可在它自己和计算机系统内的其它组件之间安全地传输数据,这是因为数据在所有可访问的总线上被加密。

[0047] 例如,为了将数据传输到其它组件,FPGA 在 FPGA 内加密数据。经加密的数据接着被传输到主存储器或被直接传输到组件。

[0048] 如果该组件是 FPGA 自己,那么经加密的数据从主存储器中读回到 FPGA,并且用 FPGA 内部的密钥和密码操作来对经加密的数据进行解密。在这个示例中,FPGA 将主存储器用作附加的存储器。

[0049] FPGA 可使用主存储器来将数据传输到其它组件。在这个示例中,该其它组件从存储器中读取经加密的数据并解密该数据。由此,CPU、GPU 或其它组件还包括类似于 FPGA 中使用的加密/解密模块。

[0050] 类似地,其它组件可将数据直接地传输到 FPGA 或通过存储器来传输到 FPGA。其它组件对数据进行加密并将其传输到存储器或 FPGA。FPGA 接着从存储器中读取数据或接收数据,并接着解密它。

[0051] 如果解密使用共享秘密,那么该秘密还可直接地从 FPGA 传输到该组件或通过存储器传输到该组件(或可能已经通过该组件传输到 FPGA)。共享秘密的传输可使用公钥/私钥加密来执行以保护该秘密。具体而言,为了提供相互认证,FPGA 使用公钥/私钥对来向组件(例如,GPU)认证自己,并且该组件使用具有仅仅为 GPU 所知的私钥的第二公钥/私钥对来这样做。

[0052] 作为相互认证过程的一部分,FPGA 和 GPU 建立一个或多个共享秘密(例如,两个共享秘密,一个用于完整性保护而一个用于机密性)。这些共享秘密接着被用于作为 FPGA 和 GPU 之间安全会话的一部分,对这两个组件之间随后的通信进行加密和认证。

[0053] 作为另一示例,在图 7 中显示的,现在描述用于对 FPGA 的密码操作进行安全地编程的过程。

[0054] 一般而言,安全地编程 FPGA 涉及将经加密的程序逻辑接收到存储器中。经加密的程序逻辑在现场可编程门阵列中被解密并被解密到 FPGA 中的存储器中。接着使用经解密的程序逻辑来对现场可编程门阵列的可编程元件进行编程。

[0055] 如图 7 中显示的,由于程序逻辑旨在实现密码操作,因此所期望的是,确保经加密的程序逻辑是经认证的。例如,经加密的程序逻辑可使用认证加密协议来加密,或者经加密

的程序逻辑可包括对未经加密的程序逻辑的数字签名。在一个实现中,经加密的程序逻辑可使用对称密钥来加密,该对称密钥用 FPGA 的公钥来加密并还由受信源以 FPGA 能够密码地验证的方式来数字地签名。

[0056] 具体而言,在其中 FPGA 的密钥将被生成并且 FPGA 的代码能被创作和编译的安全设施中,生成 700 针对 FPGA 的密码秘密 K。通过使用密码秘密 K 和实现用于与密码秘密 K 一起使用的所期望的密码操作的一般性 FPGA 程序 P,程序 P 的专用于秘密 K 的版本  $P_k$  被编译 702。该版本可用属于安全设施的密钥来数字地签名 704。

[0057] 经编译的程序的版本  $P_k$  接着被加密 706,从而产生经加密的程序逻辑  $E(K_{\text{PUB}}(\text{公共}), \text{FPGA}, P_k)$ 。这个经加密的程序逻辑被发送 708 到包括具有密码秘密 K 的 FPGA 的计算机系统。这个计算机系统接收 710 经加密的程序逻辑。在该系统中的 CPU 将经加密的程序发送 712 到 FPGA。在 FPGA 上,FPGA 用 FPGA 的私钥来解密 714 该程序。FPGA 可验证 716 经解密的程序上的数字签名是来自安全设施的。FPGA 接着使用现在经验证和经解密的程序来进行自编程 718,由此安装新的密码功能。存储在 FPGA 内的经验证和经解密的程序接着可被用于以和未经加密的程序相同的方式来对 FPGA 进行编程。

[0058] 将用于新的密码操作的密钥的秘密组件编译到现场编程指令中以供编程 FPGA 也是可能的。由此,当 FPGA 用接收到的代码来对自己进行编程时,它具有被构建到算法定义中的密钥。在这个实现中,不存在密钥的单独的存储,这是因为密钥被构建到实际的 FPGA 实现中。

[0059] 在所附权利要求的主题中的术语“制品”、“过程”、“机器”和“物质组成”旨在将权利要求限制到被认为落入 35U. S. C. § 101 中的这些术语的使用所定义的可被专利保护的主体范围内。

[0060] 上文中提到的此处描述的替换实施方式中的任一个或全部可以按形成附加混合实施方式所需的任何组合使用。应该理解,在所附权利要求中定义的主题没有必要限于上述的特定实现。上述特定实现仅作为例子被揭示。

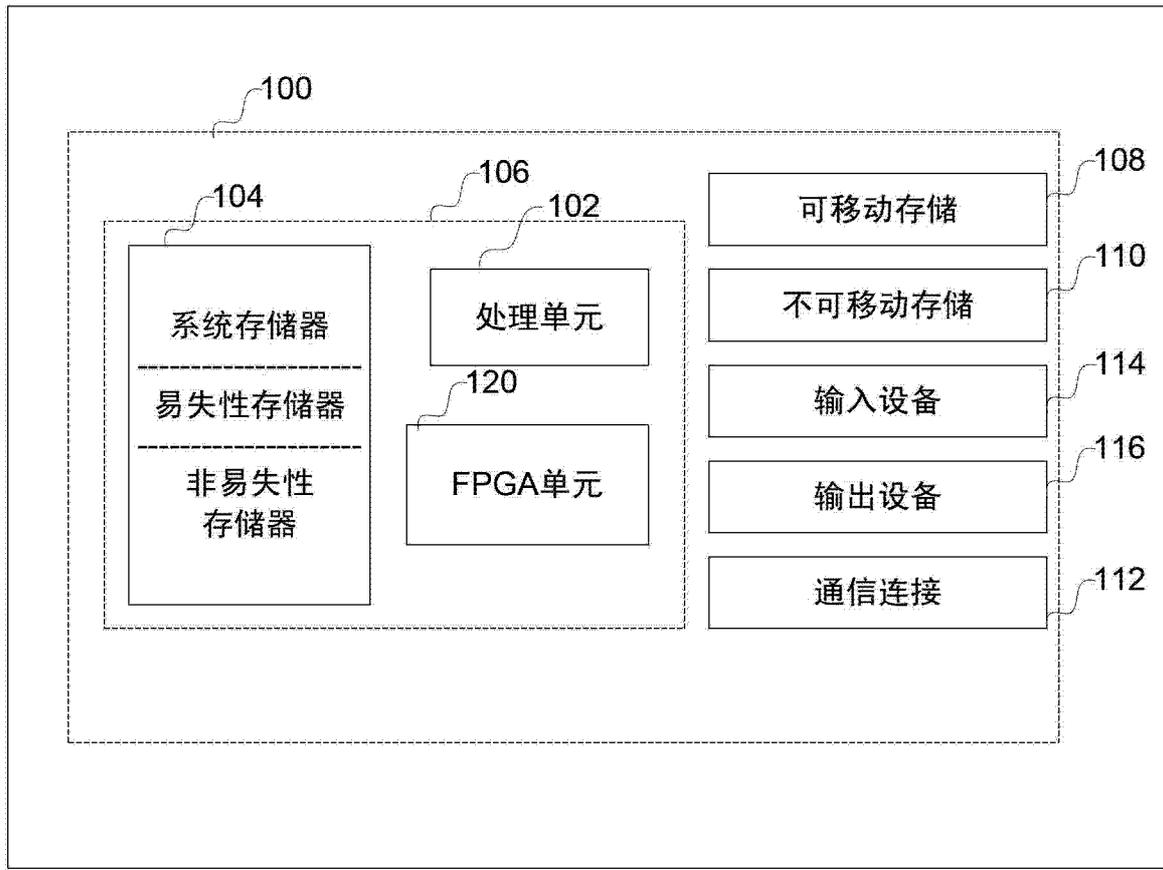


图 1

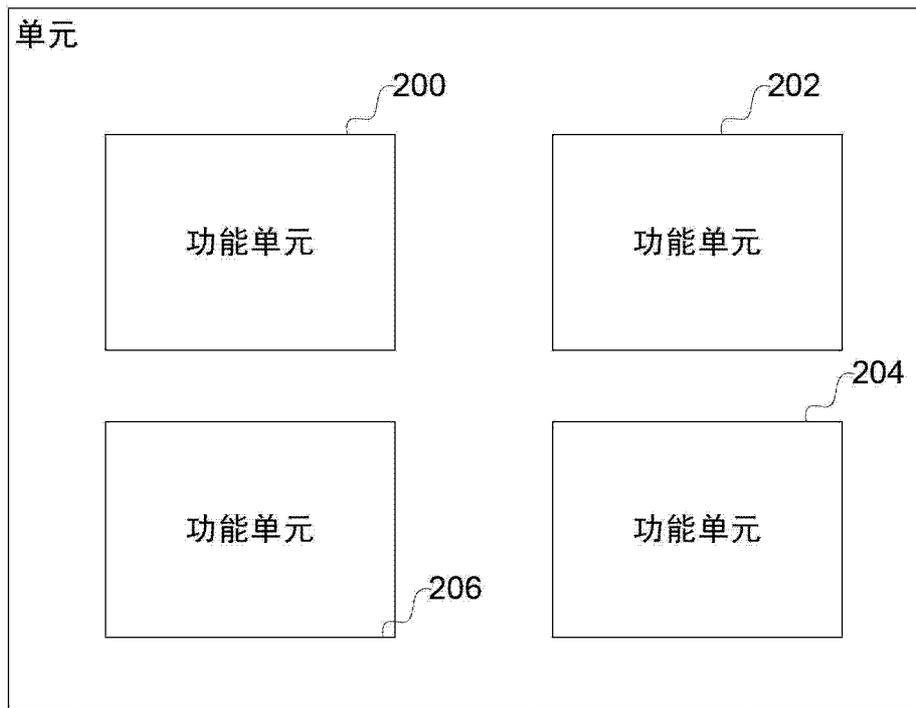


图 2

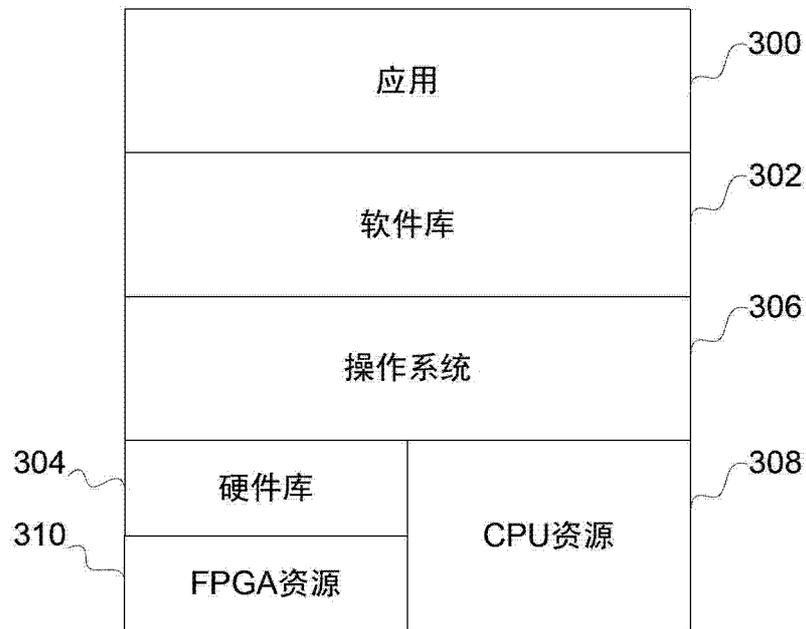


图 3

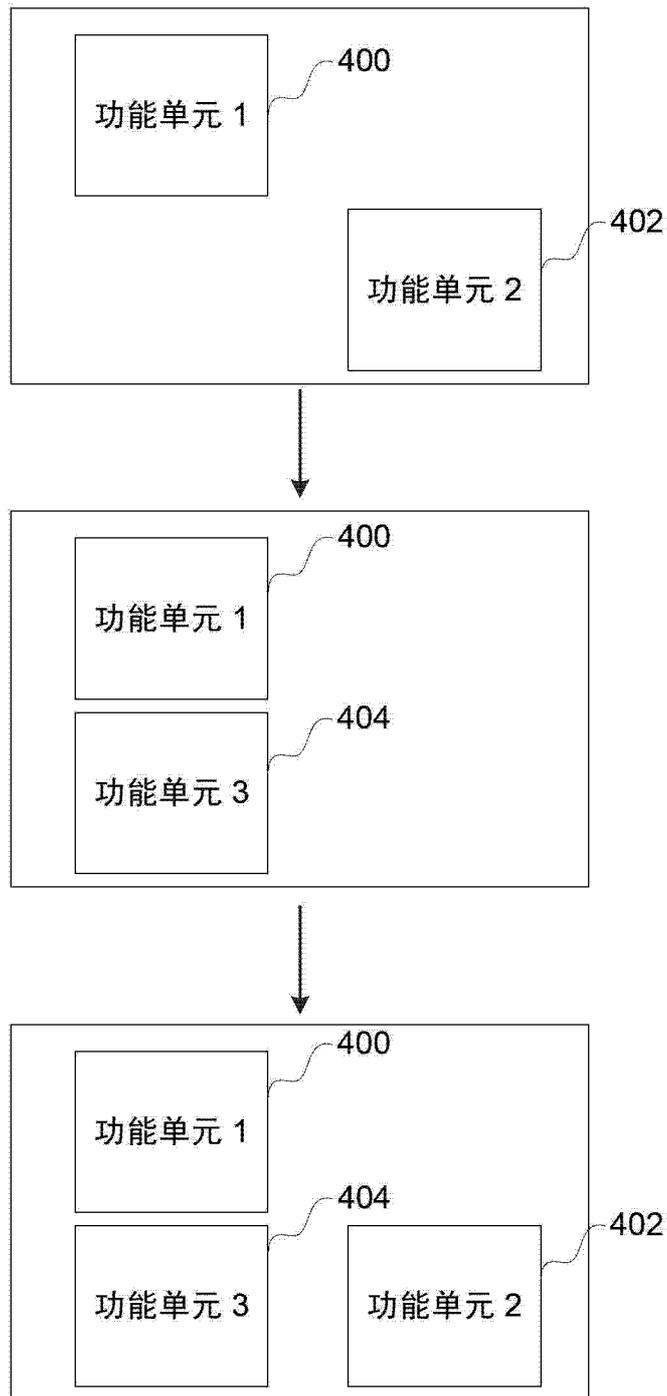


图 4

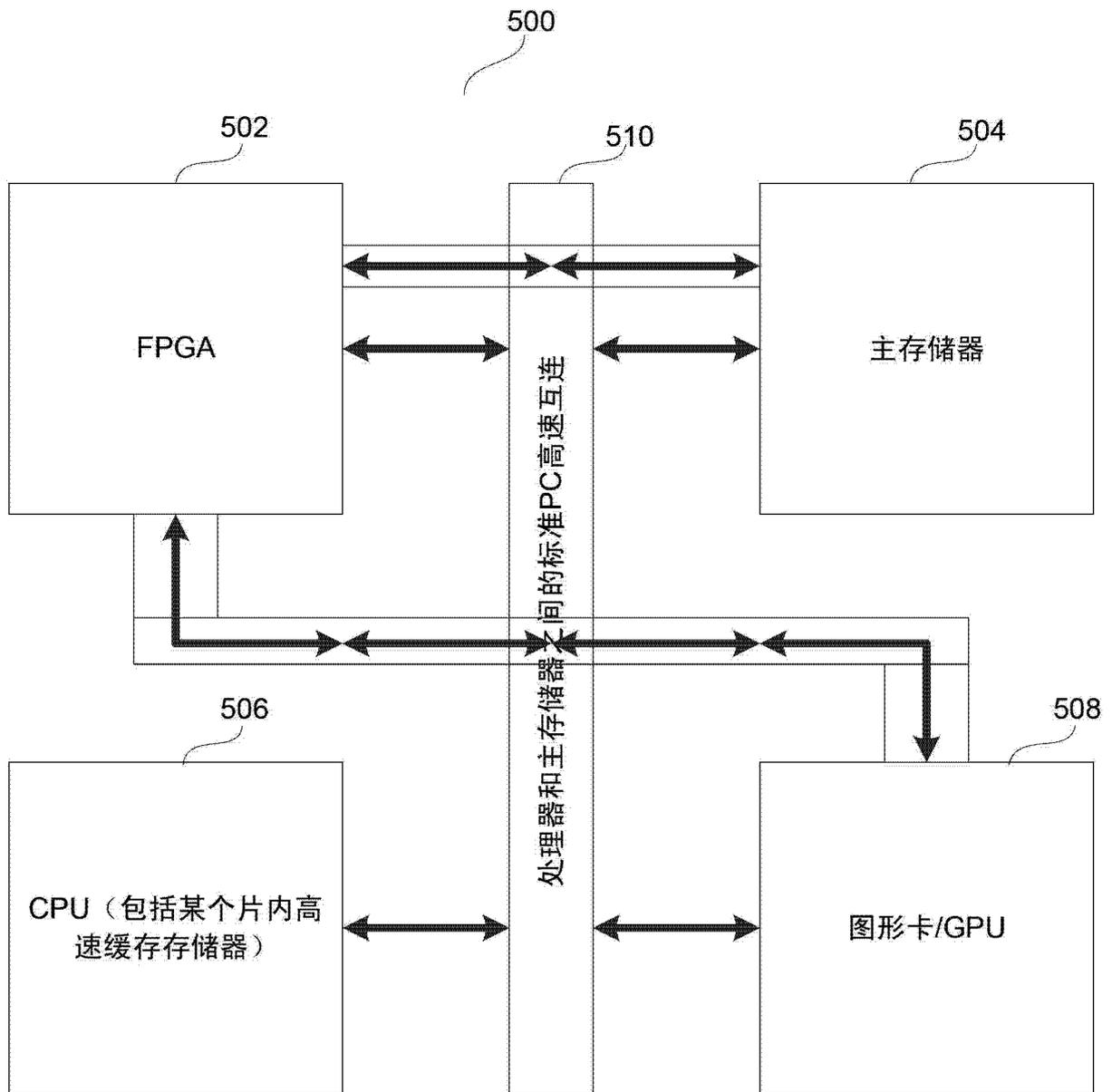


图 5

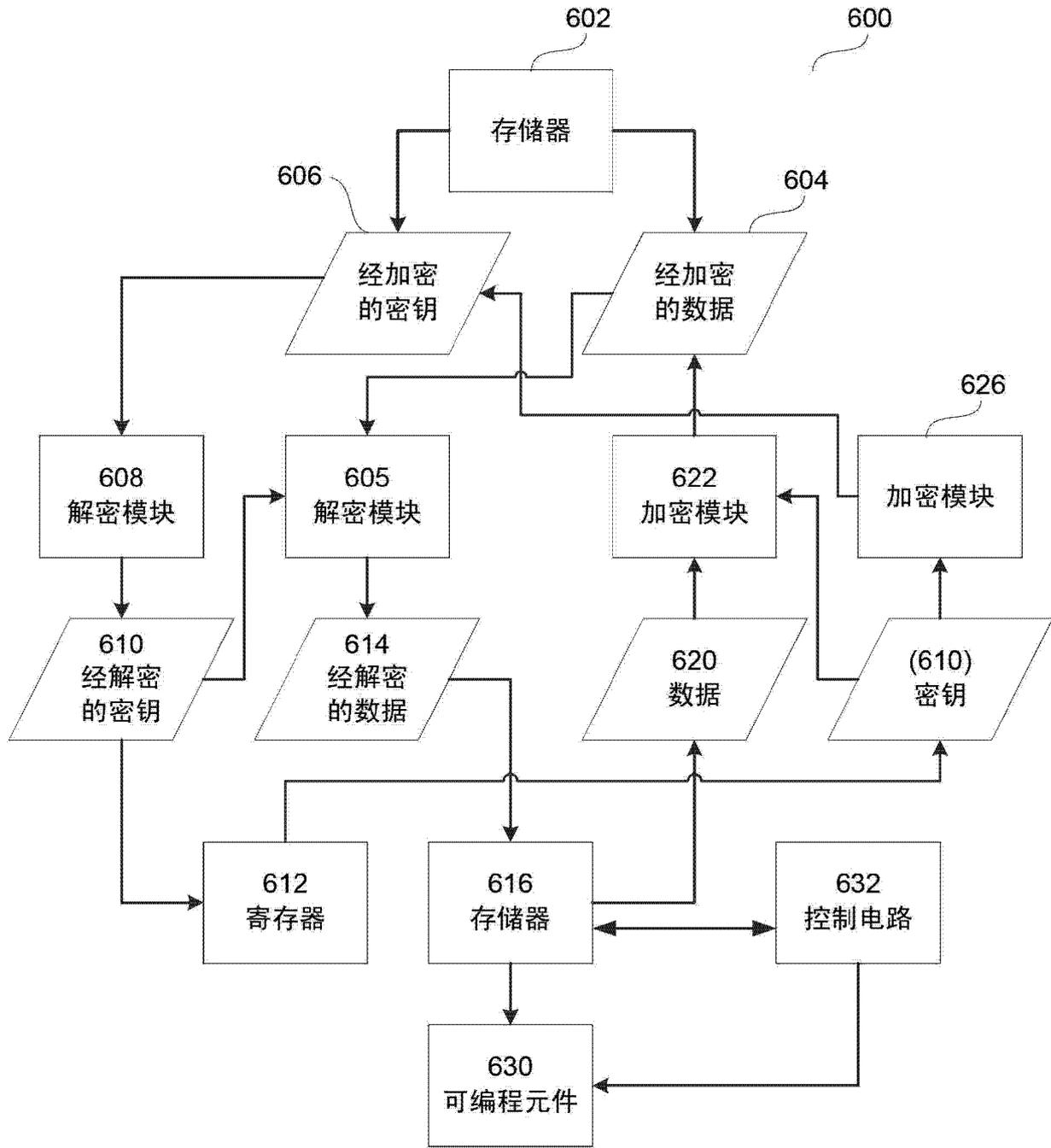


图 6

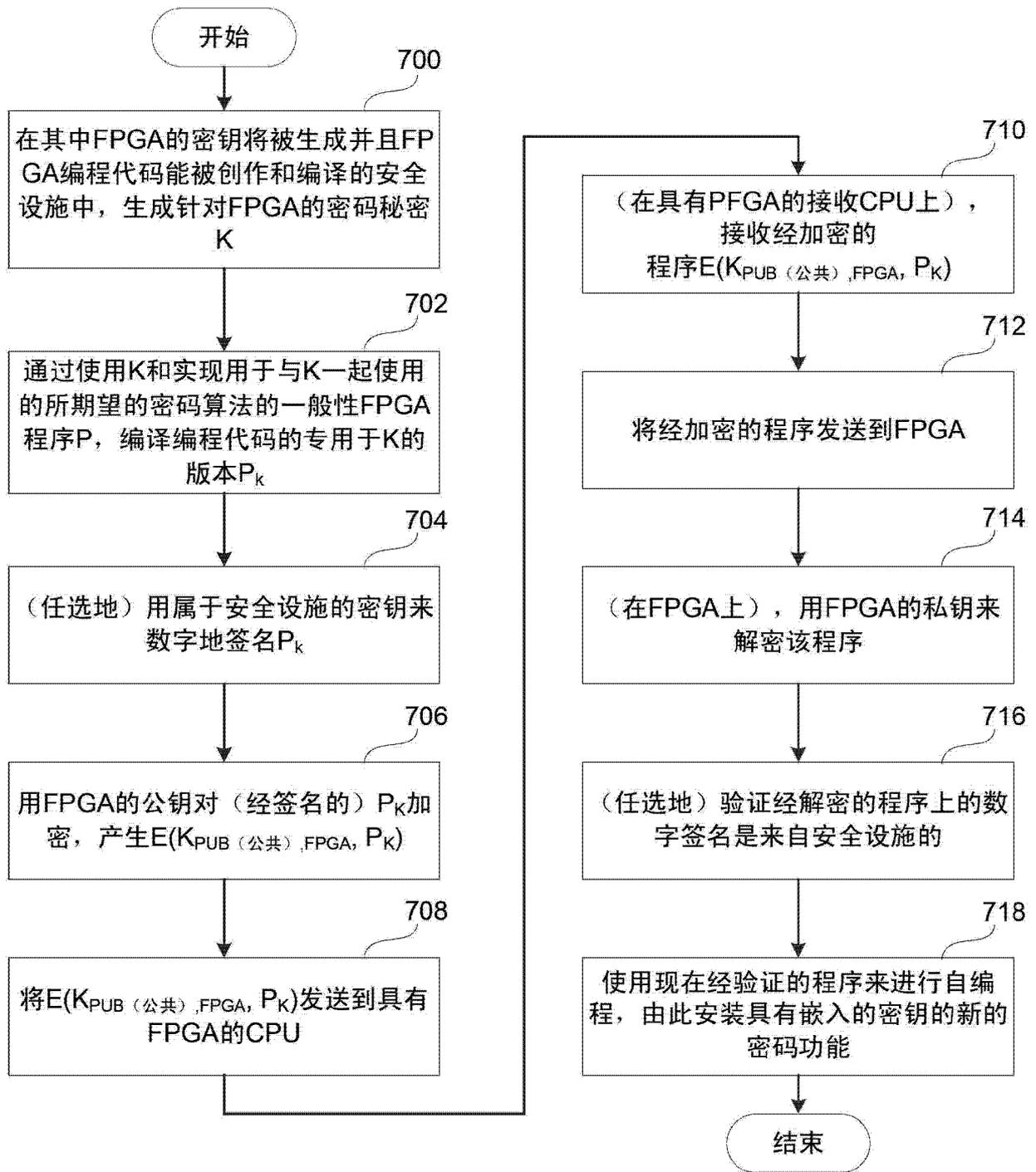


图 7