



(12)发明专利

(10)授权公告号 CN 105027136 B

(45)授权公告日 2018.02.02

(21)申请号 201380062635.8

李江滔 A·拉詹

(22)申请日 2013.06.19

(74)专利代理机构 上海专利商标事务所有限公司 31100

(65)同一申请的已公布的文献号

代理人 张欣

申请公布号 CN 105027136 A

(51)Int.Cl.

G06F 21/73(2006.01)

(43)申请公布日 2015.11.04

(56)对比文件

(30)优先权数据

CN 101755269 A, 2010.06.23, 说明书第[0044]-[0085]段及附图1-3.

13/730,829 2012.12.29 US

US 7685436 B2, 2010.03.23, 说明书第4栏第20-60行, 附图1,2.

(85)PCT国际申请进入国家阶段日

US 2012/0072737 A1, 2012.03.22, 说明书第[0144]段.

2015.05.29

US 7373506 B2, 2008.05.13, 全文.

(86)PCT国际申请的申请数据

CN 1779689 A, 2006.05.31, 全文.

PCT/US2013/046636 2013.06.19

审查员 叶珊

(87)PCT国际申请的公布数据

W02014/105146 EN 2014.07.03

(73)专利权人 英特尔公司

权利要求书3页 说明书20页 附图16页

地址 美国加利福尼亚州

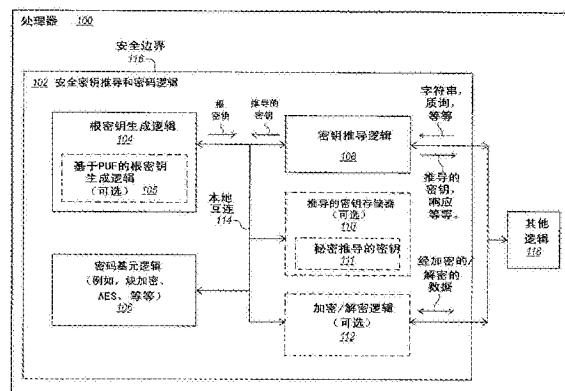
(72)发明人 G·W·考克斯 D·约翰斯顿

(54)发明名称

用于集成电路的安全密钥推导和密码逻辑

(57)摘要

一个方面的处理器包括用于生成根密钥的根密钥生成逻辑。根密钥生成逻辑包括静态且熵性的比特的源。处理器还包括与根密钥生成逻辑耦合的密钥推导逻辑。密钥推导逻辑用于从根密钥推导出一个或多个密钥。处理器还包括与根密钥生成逻辑耦合的密码基元逻辑。密码基元逻辑用于执行加密操作。处理器还包括包含根密钥生成逻辑、密钥推导逻辑,以及密码基元逻辑的安全边界。还公开了其他处理器、方法,以及系统。



1. 一种处理器,包括:

管芯;

根密钥生成逻辑,被包括在所述管芯上,用于生成根密钥,所述根密钥生成逻辑包括静态且熵性的比特的源;

密钥推导逻辑,被包括在所述管芯上并与所述根密钥生成逻辑耦合,所述密钥推导逻辑用于从所述根密钥推导一个或多个密钥;

密码基元逻辑,被包括在所述管芯上并与所述根密钥生成逻辑耦合,用于执行加密操作;以及

包含所述根密钥生成逻辑、所述密钥推导逻辑,以及所述密码基元逻辑的安全边界。

2. 如权利要求1所述的处理器,其特征在于,还包括至少一个通用核,并且所述静态且熵性的比特的源包括在物理上不可克隆的函数(PUF),其用于生成PUF比特。

3. 如权利要求2所述的处理器,进一步包括错误校正逻辑和熵提取逻辑,分别用于对所述PUF比特执行错误校正和熵提取,所述错误校正逻辑和所述熵提取逻辑被包括在所述安全边界内,并且其中所述密钥推导逻辑从中推导出所述一个或多个密钥的所述根密钥已经受由所述熵提取逻辑进行的熵提取。

4. 如权利要求1所述的处理器,其特征在于,所述静态且熵性的比特的源包括熔丝。

5. 如权利要求1所述的处理器,进一步包括所述安全边界内的测试逻辑,用于从所述安全边界内测试至少所述密钥推导逻辑。

6. 如权利要求1所述的处理器,其特征在于,还包括至少一个无序核,并且所述安全边界包括被联邦政府安全认证的边界。

7. 如权利要求6所述的处理器,其特征在于,所述安全认证的边界包括联邦信息处理标准FIPS认证的边界。

8. 如权利要求1所述的处理器,进一步包括多个不同的域标识符,每一个域标识符都对应于不同的域,并且其中所述密钥推导逻辑用于基于所述不同的对应的域标识符推导所述不同的域的不同的密钥。

9. 如权利要求1所述的处理器,进一步包括加密与解密逻辑,用于使用所述一个或多个推导的密钥来执行加密与解密。

10. 如权利要求9所述的处理器,其特征在于,所述根密钥生成逻辑包括基于PUF的根密钥生成逻辑,且进一步包括熔丝控制逻辑,用于向所述加密与解密逻辑提供熔丝值,并且在熔丝阵列中编程经加密的熔丝值,所述经加密的熔丝值已由所述加密与解密逻辑基于以PUF为基础的根密钥进行加密。

11. 如权利要求1所述的处理器,其特征在于,还包括寄存器重命名单元,并且所述根密钥生成逻辑、所述密钥推导逻辑,以及所述密码基元逻辑是垂直、可缩放的,并且可重复使用的知识产权块的一部分。

12. 如权利要求1—11中任一项所述的处理器,其特征在于,在所述安全边界内,由外部实体对逻辑状态的扫描被禁用。

13. 如权利要求1—11中任一项所述的处理器,进一步包括所述安全边界内的数字随机数生成器逻辑,其特征在于,所述数字随机数生成器逻辑和所述密钥推导逻辑共享所述密码基元逻辑。

14. 如权利要求1—11中任一项所述的处理器，其特征在于，还包括至少一个无序核，并且所述根密钥生成逻辑、所述密钥推导逻辑，以及所述密码基元逻辑由硬件构成。

15. 一种在处理器中的方法，包括：

在所述处理器的安全边界内，从静态且熵性的比特的源生成根密钥，其中所述静态且熵性的比特的源被包括在管芯上；

在所述管芯上、在所述安全边界内，从所述根密钥推导一个或多个密钥；以及
在所述管芯上、在所述安全边界内，基于所述根密钥，执行密码基元操作。

16. 如权利要求15所述的方法，其特征在于，生成包括从物理上不可克隆的函数PUF比特生成所述根密钥。

17. 如权利要求16所述的方法，进一步包括在所述安全边界内对所述PUF比特执行错误校正和熵提取，并且其中从所述根密钥提取所述一个或多个密钥包括：从已经受熵提取的根密钥提取所述一个或多个密钥。

18. 如权利要求15所述的方法，其特征在于，生成包括从熔丝生成所述根密钥。

19. 如权利要求15所述的方法，进一步包括用于从所述安全边界内执行所述密码基元操作的测试逻辑，其中，从所述安全边界外面对用于执行所述密码基元操作的所述逻辑的测试被禁用。

20. 如权利要求15所述的方法，其特征在于，推导包括在被联邦政府安全认证的安全边界内从所述根密钥推导所述一个或多个密钥。

21. 如权利要求20所述的方法，其特征在于，所述安全认证的边界包括联邦信息处理标准FIPS边界。

22. 如权利要求15所述的方法，其特征在于，推导包括，对于不同的域，基于不同的对应的域标识符在所述安全边界内从所述根密钥推导密钥。

23. 如权利要求15所述的方法，进一步包括在所述安全边界内，使用所述一个或多个推导的密钥来执行加密与解密。

24. 一种机器可读介质，包括存储在所述机器可读介质上的多条指令，所述多条指令当被执行时使计算设备执行如权利要求15—23中任一项所述的方法。

25. 一种设备，包括用于执行权利要求15—23中任一项所述的方法的装置。

26. 一种电子设备，包括：互连；如权利要求1—11中任一项所述的处理器，与所述互连耦合；以及动态随机访问存储器，与所述互连耦合。

27. 一种处理器，包括：

至少一个乱序执行核；

根密钥生成逻辑，用于生成根密钥，所述根密钥生成逻辑包括静态且熵性的比特的源；
和所述根密钥生成逻辑耦合的密钥推导逻辑，所述密钥推导逻辑用于从所述根密钥推导一个或多个密钥，其中所述密钥推导逻辑用于从根密钥推导所述一个或多个密钥，所述根密钥是在其上已经执行熵提取的熔丝值和PUF比特中的一者；

和所述根密钥生成逻辑耦合的加密与解密逻辑，用于执行密码操作；以及

包含所述根密钥生成逻辑、所述密钥推导逻辑、和所述加密与解密逻辑的安全边界，
其中，所述至少一个乱序执行核、所述根密钥生成逻辑、密钥推导逻辑和所述加密与解密逻辑全部被包括在同一管芯上。

28. 如权利要求27所述的处理器,其特征在于,所述处理器是中央处理单元CPU。

用于集成电路的安全密钥推导和密码逻辑

技术领域

[0001] 各实施例涉及集成电路领域。具体而言，各实施例涉及集成电路中的安全领域。

背景技术

[0002] 台式机、膝上型计算机、上网本、平板电脑、智能电话、蜂窝电话、多媒体内容播放器、智能电视机、机顶盒、服务器，以及各种其他类型的消费电子产品或电子设备，通常用于处理敏感的或安全信息。作为示例，敏感的或安全信息可包括财务信息、保密文档、个人电子邮件、数字权限保护的内容等等。

[0003] 这样的电子设备中所使用的处理器、芯片组组件、片上系统 (SoC)、安全相关的集成电路，及其他类型的集成电路通常设置有用于保护敏感的或安全信息的秘密，诸如秘密密钥。例如，可以使用秘密密钥，通过加密/解密来保护信息。

附图说明

[0004] 可以通过参考用来说明本发明的各实施例的下列描述和附图来理解本发明。在附图中：

[0005] 图1是具有安全密钥推导和密码逻辑的处理器的实施例的框图。

[0006] 图2是基于PUF的根密钥生成系统的实施例的框图。

[0007] 图3是具有安全密钥推导和密码逻辑的处理器的实施例的框图，该安全密钥推导和密码逻辑可任选地包括测试和调试逻辑的实施例的。

[0008] 图4是具有安全密钥推导和密码逻辑的处理器的实施例的框图，该安全密钥推导和密码逻辑可任选地包括物理和/或逻辑安全认证的边界的。

[0009] 图5是具有安全密钥推导和密码逻辑的处理器的实施例的框图，该安全密钥推导和密码逻辑可任选地包括数字随机数生成器逻辑和相关联的缓冲区。

[0010] 图6是具有能够用于提供密码学地实施的域分离的安全密钥推导和密码逻辑的处理器的计算系统的框图。

[0011] 图7是能够用于利用从PUF推导的密钥来加密和解密熔丝值的熔丝逻辑的实施例的框图。

[0012] 图8A是示出根据本发明的各实施例的示例性有序流水线和示例性的寄存器重命名的无序发布/执行流水线的框图。

[0013] 图8B是示出根据本发明的各实施例的要包括在处理器中的有序架构核的示例性实施例和示例性的寄存器重命名的无序发布/执行架构核的框图。

[0014] 图9A是根据本发明的各实施例的单个处理器核的框图，以及其与管芯上的互连网络的连接以及其第2级 (L2) 高速缓存的本地子集。

[0015] 图9B是根据本发明的实施例的图9A中的处理器核的一部分的展开图。

[0016] 图10是根据本发明的各实施例的可以具有一个以上的核，可以具有集成的存储器控制器，并可以具有集成的图形器件的处理器的框图。

- [0017] 图11,所示是根据本发明一实施例的系统的框图。
- [0018] 图12,所示是根据本发明的一个实施例的第一更具体的示例性系统的框图。
- [0019] 图13,所示是根据本发明的一个实施例的第二更具体的示例性系统的框图。
- [0020] 图14,所示是根据本发明一实施例的SoC的框图。
- [0021] 图15是根据本发明的各实施例的对照使用软件指令转换器将源指令集中的二进制指令转换成目标指令集中的二进制指令的框图。

具体实施方式

[0022] 此处公开了用于处理器及其他集成电路的安全密钥推导和密码(cryptography)逻辑。在以下描述中,阐述了大量具体细节(例如,特定密码算法、逻辑分区/集成细节、逻辑实现、微架构细节、操作序列,系统组件的类型和相互关系,等等)。然而,应该理解,本发明的各实施例可以在没有这些具体细节的情况下实施。在其他情况下,没有详细示出已知的电路、结构,以及技术,以便不至于使对本描述的理解变得模糊。

[0023] 图1是具有安全密钥推导和密码逻辑102的处理器100的实施例的框图。在某些实施例中,处理器可以是通用处理器(例如,用于台式机、膝上型计算设备、上网本、平板电脑、智能电话、手机、服务器、智能电视机、机顶盒等计算设备的类型的通用处理器)。替换地,处理器可以是专用处理器。合适的专用处理器的示例包括,但不仅限于,密码处理器、安全处理器、网络处理器、通信处理器、协处理器、嵌入式处理器、数字信号处理器(DSP),仅举几个例子而已。

[0024] 处理器包括安全密钥推导和密码逻辑102。安全密钥推导和密码逻辑包括根密钥生成逻辑104、密码基元逻辑106、密钥推导逻辑108、可选的推导的密钥存储器110,以及可选的加密/解密逻辑112。这些组件全部都与本地互连114耦合,并通过本地互连114耦合在一起。安全密钥推导和密码逻辑可以完全包含在管芯上和/或处理器上(例如,不需要由从存储器加载到处理器中的软件执行任何处理)。在某些实施例中,安全密钥推导和密码逻辑可以主要、几乎完全,或完全地实现在管芯上的和/或处理器上的硬件中。在某些实施例中,安全密钥推导和密码逻辑可以被实现为基本上垂直知识产权(IP)块,该块基本上可重复使用,且基本上可缩放。在某些实施例中,垂直IP块可以被设计为从一种处理器设计可移植到另一种,可缩放的,可重复使用的,基本上自含式的(self-contained)并通过意义明确的接口连接到其他组件,这可以帮助垂直IP块从一个设计到另一种设计被重复使用。在某些实施例中,安全密钥推导和密码逻辑的所有组件都可以基本上在物理和/或逻辑边界116内是自含式的(self-contained),安全或秘密信息不会离开边界。

[0025] 安全密钥推导和密码逻辑102包括根密钥生成逻辑104。根密钥生成逻辑能够用于(operable to)生成一个或多个秘密或安全根密钥。在某些实施例中,根密钥可以不离开安全逻辑的边界。可以使用根密钥来推导一个或多个其他秘密(secret)或安全推导的密钥(secure derived key)。秘密或安全推导的密钥可以用于各种目的(例如,执行加密/解密、提供对质询的响应,等等),如下文进一步描述的。如图所示,在某些实施例中,根密钥生成逻辑可以可任选地包括基于在物理上不可克隆的函数(PUF)的根密钥生成逻辑105。下面将更详细地描述PUF。基于PUF的根密钥生成逻辑可以能够用于生成一个或多个秘密或基于PUF的安全根密钥。作为另一个选项,在某些实施例中,根密钥生成逻辑可以可任选地包括

基于熔丝 (fuse-based) 的根密钥生成逻辑 (未示出)。基于熔丝的根密钥生成逻辑可以能够用于生成一个或多个秘密或基于熔丝的安全根密钥。在其他实施例中,可以使用基于PUF的以及基于熔丝的根密钥生成逻辑的组合。可另选地,代可以使用替熔丝和PUF的替代项 (substitute),或基本上静态的且基本上熵性的 (entropic) /随机的比特的其他源。

[0026] 安全密钥推导和密码逻辑102还包括密码基元逻辑106。在某些实施例中,密码基元逻辑可包括能够用于执行块加密的块加密逻辑(例如,块加密引擎或模块)和/或能够用于计算散列操作的密码散列函数逻辑(例如,安全散列算法逻辑)。块加密 (block cipher) 一般表示确定性的 (deterministic) 密码算法,这些确定性的密码算法利用带有用于加密和解密数据的对称密钥的不变的变换,对叫做块的固定长度的比特组进行操作。作为示例,块加密逻辑可包括高级加密标准 (AES) 逻辑(例如,AES引擎或模块)。AES是美国国家标准与技术协会 (NIST) 批准的密码算法,在2001年11月26日发布的美国联邦信息处理标准 (FIPS) PUB 197中进一步描述了该密码算法。可另选地,代替AES,可以可任选地使用其他块加密管芯上的逻辑实现的算法(例如,NIST或FIPS批准的其它)。在某些实施例中,密码基元逻辑可以主要、几乎完全,或完全地实现在安全密钥推导和密码逻辑的边界116内的硬件和管芯上的和/或处理器上的硬件中。

[0027] 密码基元逻辑可以用于各种目的。例如,在某些实施例中,密码基元逻辑可以被用来加密和解密数据。作为另一个示例,密码基元逻辑可以被用来支持安全密钥推导和密码逻辑内的其他密码逻辑或协议。例如,密码基元逻辑可以被用来支持密钥推导逻辑108和/或密钥推导,如下文进一步描述的。作为另一个示例,密码基元逻辑可以被用来支持结合基于PUF的根密钥生成而执行的熵提取和/或PUF比特调节,如下文进一步描述的。作为再一个示例,密码基元逻辑可以被用来支持加密/解密逻辑112和/或如下文进一步描述的加密/解密。作为更进一步的示例,密码基元逻辑可以被用来支持数字随机数生成 (DRNG),如下文进一步描述的。

[0028] 再次参考图1,安全密钥推导和密码逻辑102还包括密钥推导逻辑108(例如,密钥推导引擎或模块)。密钥推导逻辑可以能够用于从一个或多个根密钥推导一个或多个推导的密钥。作为示例,可以通过利用一个或多个根密钥,对管芯上的逻辑实现的密钥推导算法求值 (evaluate),生成或推导一个或多个推导的密钥。不同的密钥推导算法适合于不同的实施例。合适的密钥推导算法的示例包括,但不仅限于,NIST SP800-108、SP800-56C、基于散列函数的密钥推导算法、基于块加密的密钥推导算法、其他NIST或FIPS批准的密钥推导算法,等等。在某些实施例中,密钥推导逻辑可以主要、几乎完全,或完全地实现在安全密钥推导和密码逻辑的边界116内的硬件和管芯上的和/或处理器上的硬件中。在某些实施例中,根密钥生成逻辑和密钥推导逻辑可以实现在安全认证的边界内的硬件中,作为垂直知识产权块。有利地,一个或多个推导的密钥可以密码学地从一个或多个根密钥(例如,一个或多个基于PUF的根密钥)推导出。一方面,这可以被用来提供多个不同的推导的密钥,无需具有多个不同组的PUF或熔丝,否则,将往往导致增大逻辑的大小、制造成本,以及功率消耗。

[0029] 在某些实施例中,密钥推导逻辑108可以被用来为安全密钥推导和密码逻辑102外部的处理器100的各种组件(例如,其他逻辑118)和/或处理器部署于其中的系统的各种组件生成或推导密钥。例如,在某些实施例中,其他组件(例如,其他逻辑118)可以提供用于对

管芯上的逻辑实现的密钥推导算法求值的额外的比特或数据。作为一个示例,可以通过利用由正在请求推导的密钥的其他组件(例如,其他逻辑118)所提供的一个或多个根密钥和一组比特(例如,推导字符串、个性化字符串等等)来对密钥推导算法求值,以生成推导的密钥。作为示例,其他逻辑118可以表示在处理器或其他集成电路中存在的各种不同类型的密钥利用率和/或安全逻辑。这样的逻辑的示例包括,但不仅限于,加密逻辑、解密逻辑、密码逻辑或模块、可信平台模块、安全引擎、安全控制器、密码处理器、密码协处理器,等等。

[0030] 还可构想密钥推导逻辑108的其他用途。例如,在某些实施例中,推导的密钥可以被用作响应于作为输入接收到的质询,作为输出提供的响应。例如,可以通过另一个组件(例如,其他逻辑118),向密钥推导逻辑提供质询(例如,一组比特)。密钥推导逻辑可以利用一个或多个根密钥和质询,对密钥推导算法求值。推导的密钥可以表示对质询的响应。有利地,基于根密钥(例如,基于PUF的根密钥),密码学地推导响应。提供质询-响应功能的另一种可能的方式将是直接向PUF单元提供质询,并直接从PUF单元提供响应(或许带有错误校正),无需经过密码学密钥推导或处理。然而,后一方法一般具有某些缺点。一方面,PUF单元所能提供的质询-响应对的数量一般而言是有限的(例如,在静态随机存取存储器(SRAM)类型的PUF中,质询可以表示物理地址,从而可能有的是数量有限的响应)。此外,在仲裁器类型的PUF中,大量的质询-响应对可能会易于导致允许建模攻击。其次,来自PUF单元的响应一般易于导致有噪声。相比之下,使用推导的密钥作为对质询的响应可以允许大量的,或者甚至几乎无限的无错误的质询-响应对。此质询-响应能力可以用于各种目的,诸如,例如,用于认证、仿造检测、及已知的其他目的。

[0031] 在某些实施例中,密钥推导逻辑108可以被用来从一个或多个根密钥推导出一个或多个秘密或安全密钥,它们将安全地保留在安全密钥推导和加密逻辑102的边界116内。例如,可以通过利用可能与额外的比特(例如,来自熔丝、存储在RAM中、以别的方式保存在加密逻辑内,等等)相结合的一个或多个根密钥,对密钥推导算法求值,来生成秘密或安全的推导的密钥。在某些实施例中,这些一个或更多秘密推导的密钥111可以存储在可选的推导的密钥存储器110中。在某些实施例中,这些一个或更多秘密推导的密钥111可包括可以被加密和/或解密逻辑112使用的一个或多个加密和/或解密密钥。在某些实施例中,不仅是一个或多个根密钥(例如,基于PUF的根密钥)不暴露在边界116的外面,而且基于一个或多个根密钥推导出的一个或多个秘密或安全推导的密钥也不暴露在边界的外面。

[0032] 再次参考图1,安全密钥推导和密码逻辑102还包括可选的加密和/或解密逻辑112(例如,加密和/或解密引擎或模块)。加密/解密逻辑可以能够用于加密和/或解密数据。合适的加密/解密算法的示例包括,但不仅限于,AES-ECB、AES-CBC、AES-CTR、其他块加密算法、其他NIST或FIPS批准的加密/解密算法,等等。与密码基元逻辑106相比,加密/解密逻辑112一般对较大的并且大小灵活的数据操作,而密码基元逻辑一般对较小并且大小固定的(例如,64比特块、128比特块等等)块操作,虽然这不是必需的。在某些实施例中,加密/解密逻辑可以主要、几乎完全,或完全地实现在安全密钥推导和加密逻辑的边界116内的硬件和管芯上的和/或处理器上的硬件中。在其他实施例中,如果不希望安全密钥推导和密码逻辑来执行加密和/或解密,则加密/解密逻辑可以可任选地被省略。例如,这种情况可以是,逻辑102用于质询响应用途但不用于加密/解密。

[0033] 在各实施例中,加密/解密逻辑可以使用来自根密钥生成逻辑104的根密钥,由密

钥推导逻辑108从根密钥推导出的密钥,来自推导的密钥存储器110的秘密推导的密钥111,或其他密钥。作为一个示例,在某些实施例中,组件(例如,其他逻辑118)可以提供一组比特(例如,推导字符串、个性化字符串等等)以及要被加密的明文数据。密钥推导逻辑可以基于根密钥和所提供的一组比特(例如,推导字符串或个性化字符串),推导出密钥。然后,加密/解密逻辑可以利用推导的密钥,加密明文数据,并将经加密的数据(例如,密文)提供回给作出请求的组件。作为另一个示例,在某些实施例中,组件(例如,其他逻辑118)可以提供一组比特(例如,推导字符串、个性化字符串等等)以及要被解密的经加密的数据(例如,密文(ciphertext))。密钥推导逻辑可以基于根密钥和所提供的一组比特(例如,推导字符串或个性化字符串),推导出密钥。然后,解密/解密逻辑可以利用推导的密钥,解密经加密的数据,并将未加密的或明文数据(例如,密文)提供回给作出请求的组件。

[0034] 如上文所提及的,图1的根密钥生成逻辑104可以可任选地包括基于熔丝的根密钥生成逻辑。取决于特定实现和对安全的需要,使用熔丝的一个潜在缺点在于,存储在熔丝中的秘密在某些情况下可能部分地由于对反向工程师而言太过简单,因此不足够安全。例如,可以在实验室中在物理上严密地(rigorously)检测集成电路,以便确定存储在熔丝中的密钥的值。允许存储在熔丝中的密钥被确定可能会危害集成电路的安全性和它处理的安全或敏感数据,或至少对这种危害有贡献。在某些实施例中,可以使用PUF代替熔丝和/或作为其补充,以便帮助提高安全级别。

[0035] 图2是基于PUF的根密钥生成系统205的实施例的框图。在某些实施例中,图2的基于PUF的根密钥生成系统可以用于图1的处理器和/或安全密钥推导和密码逻辑中。可另选地,图2的基于PUF的根密钥生成系统可以用于类似的或完全不同的处理器和/或安全密钥推导和密码逻辑中。此外,图1的处理器和/或安全密钥推导和密码逻辑可以使用与图2的相同、类似的,或者完全不同的基于PUF的根密钥生成系统。

[0036] 基于PUF的根密钥生成系统205包括一组PUF单元220。PUF单元此处也可以被简单地称为PUF。PUF有时也被称为物理单向函数(POWF),或其他名称。PUF单元中的每一个都可以能够用于生成对应的PUF比特。PUF单元或PUF比特的数量可以是任何常规的或适当的数量,但不仅限于本发明的范围。通常,在相对高度安全的通用处理器的情况下,可以有从大约几十个、几百个、到成千上万个PUF单元和/或PUF比特间的任何数量级,虽然本发明的范围不仅限于任何数量。天然的制造过程变化可以导致每个设备基本上唯一的PUF比特(例如,基本上平台唯一)。PUF比特也往往会在设备的寿命内对于每一个设备基本上是静态的,基本上是熵性的(entropic)或随机的。PUF比特可以是从其获取PUF根密钥的源。

[0037] 可以使用本领域内已知的各种不同类型的PUF 220。这往往导致难以在已知是PUF的所有不同类型的设备、电路,以及物理系统的周围放置准确的周边。此讨论并不旨在,并且不应该用于排除被视为PUF的设备、电路,以及物理系统。大多数PUF表示函数(例如,它们从输入/质询产生输出/响应),它们是物理的(例如,嵌入在介质中,包括集成电路,包括结构或微结构(例如,微电子结构),包括材料,在物理介质中实现,等等),基本上是不可克隆的。术语“基本上不可克隆的(substantially unclonable)”意味着,即便是对于一组PUF的制造商而言,随后即便使用相同制造过程,制造该组PUF的将具有相同显著特征(例如,将提供相同输出/响应(例如,PUF比特))的另一个副本,就算不是不可行的话,也将是非常困难的。由PUF单元所生成的PUF比特的特定二进制值一般取决于对应的PUF单元的物理特征,物

理特征进而又取决于用于制造对应的PUF单元的特定制造过程,包括取决于在制造过程期间所遇到的通常不可控的过程变化,为实用目的准确地再现这些过程变化是不切实际的或不可行的。

[0038] 在某些实施例中,PUF 220可以表示硅本征的(silicon intrinsic) PUF,或更一般性地,半导体本征的PUF,或互补金属氧化物半导体(CMOS) PUF。在某些实施例中,PUF单元可能是使用也用于制造晶体管和/或集成电路的其他逻辑的CMOS制造过程来制造的。在某些实施例中,PUF单元中的每一个都可以被嵌入在集成电路衬底内,例如,包括由通过CMOS过程形成的半导体和/或结构或器件构成的集成电路和/或结构或器件的某些部分。PUF的合适的类型的示例包括,但不仅限于,延迟PUF(例如,基于数字延迟测量值的本征的PUF)、延返回路PUF、存储器PUF(例如,基于数字存储器元件的稳定(settling)状态的本征的PUF)、SRAM PUF、交叉耦合PUF、仲裁器PUF(例如,基于多路复用器和仲裁器的PUF)、环形振荡器PUF、双稳定环形PUF、蝴蝶PUF、锁存PUF、触发器PUF、D-型触发器PUF、涂层PUF、本领域已知的另外的半导体或CMOS PUF,以及其组合。在硅、半导体、或CMOS PUF的情况下,取决于PUF的特定类型,所生成的PUF比特的二进制值可能往往会取决于诸如半导体材料中的掺杂剂浓度、集成电路的线宽度、层厚度、一个区域与下一区域的变化、等等之类的因素,这些因素以无法预测的方式取决于制造过程变化。

[0039] 再次参考图2,基于PUF的根密钥生成逻辑205还包括用于与PUF单元220连接的PUF接口逻辑221。PUF接口逻辑221可以能够用于从PUF单元读取PUF比特(例如,一组二进制或比特值)。在某些实施例中,PUF接口逻辑可以可任选地能够用于提供PUF驱回(driveback),虽然这不是必需的。PUF驱回可以获取从每一个PUF单元中读取的值,颠倒它,并将它驱回PUF单元。有利地,这可能帮助导致PUF单元的成熟(aging),以将其进一步驱向定义的或静止的状态,而并非驱向未定义的或变化的状态。

[0040] 一般而言,当从PUF单元读取PUF比特时,PUF比特一般往往会有合理地静态。例如,当多次从PUF单元读取PUF比特时,通常,大多数PUF比特的一个读取同下一个读取往往会有相同二进制值。被称为“较弱的”PUF比特的某些PUF比特,相比其它的比特,可能倾向于更频繁地在从一个读取到下一个读取时翻转或改变二进制值。例如,对五个PUF单元的第一读取可能会导致PUF比特“01101”,而对相同的五个PUF单元的第二读取可能会导致PUF比特“01111”。注意,在从第一读取到第二读取时,一个PUF比特已经从二进制0翻转到二进制1。这表示PUF比特错误。当用于安全性时,这样的PUF比特错误一般是不希望的(例如,因为它们可能会导致不同的,不可预测的,或不可重复的密钥被生成)。

[0041] 如图所示,PUF接口逻辑221可以将从PUF单元中读取的PUF比特提供给错误校正逻辑222。错误校正逻辑可以能够用于对PUF比特执行错误校正,并校正可能存在的任何错误(至少是高达给定级别的错误)。如图所示,在某些实施例中,错误校正逻辑可以接收错误校正数据。此错误校正数据有时也被称为纠错码或助手(helper)数据。在某些实施例中,此错误校正数据可以存储在熔丝、其他非易失性存储器等等中。在某些实施例中,错误校正逻辑可以能够用于生成并存储错误校正数据,例如,在制造时,以供随后使用。随后,当向错误校正逻辑提供具有错误的PUF比特时,错误校正逻辑可以能够用于使用错误校正数据来校正PUF比特中的错误(至少高达某一级别的错误),以便获得经过错误校正的PUF比特。可以使用本领域中已知的各种错误校正技术。在某些实施例中,错误校正逻辑完全在管芯上和/或

在处理器上，并主要，几乎完全，或完全以硬件逻辑来实现。

[0042] PUF比特一般需要具有足够级别的熵(entropy)或随机性(randomness)。当存在高级别的熵或随机性时，来自两个不同的集成电路的相同的PUF比特组的似然性一般来说往往相对地低。例如，来自第一组五个PUF单元的PUF比特可以是“01101”，来自第二组五个PUF单元的PUF比特可以是“10100”，来自第三组五个PUF单元的PUF比特可以是“10111”，仅作为一个示例。注意，这些PUF比特的组不同。当存在相对较高级别的熵时，每一个比特具有二进制0或者二进制的似然性应该是大致相等的，从而在给定足够的PUF比特的组的情况下，一串PUF比特应该大致横跨所有可能的二进制值，以提供值的相对较高级别的随机性。当用于安全性时，通常希望PUF比特能至少合理地熵性(entropic)的或随机，因为这帮助增强安全性。

[0043] 再次参考图2，错误校正逻辑可以将经过错误校正的PUF比特提供到熵提取逻辑224。熵提取逻辑可以能够用于对经过错误校正的PUF比特执行熵提取。熵提取一般表示帮助鼓励或促进熵或随机性的调节。合适的熵提取方法的示例包括，但不仅限于，基于AES-CMAC算法的那些方法、基于块加密算法的那些方法、基于消息认证代码的那些方法、基于散列函数的那些方法，或已知的其他熵提取方法。在某些实施例中，熵提取逻辑完全在管芯上和/或在处理器上，并主要，几乎完全，或完全以硬件逻辑来实现。熵提取逻辑可以提供表示PUF根密钥的经熵提取的PUF比特，作为输出。

[0044] 应该理解，这只是合适的基于PUF的根密钥生成逻辑的一个说明性示例。其他实施例可包括对PUF比特执行更多或更少处理的更多或更少组件。例如，其他实施例可以可任选地省略PUF驱动。作为另一个示例，其他实施例可以可任选地省略熵提取(例如，如果对于特定实现，已经确保PUF比特是足够熵性的)。上文所描述的PUF比特可以用于此处对于根密钥生成逻辑所公开的各实施例中的任何一个中。另外，尽管此处常常描述PUF，但是，其他实施例可以替代地使用熔丝，或适用于根密钥的其它静态、熵性的比特的源。

[0045] 将PUF用于安全性的一个优点是，PUF比特和/或PUF根密钥往往比存储在熔丝(以及ROM、RTL、tie-up电路/tie-down电路等等)中的密钥更安全和/或不易发现(例如，通过反向工程、物理攻击等等)。PUF生成PUF比特，在运行期间从该PUF比特获取根密钥。为了确定PUF比特和/或PUF根密钥，集成电路或其他设备可能需要运行或被通电，和/或不以将会改变PUF比特的值的方式而改变。这些因素往往使得通过反向工程确定PUF比特和/或PUF根密钥的值困难得多。有利地，这可以帮助提高集成电路或器件以及它们用于处理的敏感信息的安全性。另外，PUF比特和/或PUF根密钥可以在集成电路上的安全密钥推导和密码逻辑内是知道的，但是，可能不能在外部被知道，在某些情况下，甚至不能被集成电路制造商的大多数受信任的实体知道，这可以帮助限制制造商的职责、风险，以及责任。

[0046] 在制造过程中，常见的是在制造的各阶段测试和/或调试集成电路以及集成电路封装。可以为各种目的进行此举，诸如，例如，测试集成电路衬底看其是否正确操作，检测漏洞(bug)或缺陷(defect)，试图修复漏洞或缺陷，将正常地运转的集成电路从将被丢弃或返工的不正确地运转的集成电路中分选出来，将基于测试的数据编程到集成电路中等等。此测试或调试可以通过集成电路测试和/或调试设备(例如，测试器和探测器)和/或其他集成电路制造设备来进行。作为示例，设备可以具有一组电探针，这些电探针用于与集成电路的暴露的电触点(例如，针脚)耦合。根据测试模式，设备可以通过探针和针脚，与集成电路

衬底交换电信号。例如，设备可以向集成电路传输电信号，并接收作为响应的对应电信号，可以作为测试或调试的一部分，分析电信号。一项挑战是，设备可能没有所希望的那样高度安全。有可能，多个雇员（例如，操作员）或其他人可以访问设备。存在一项风险，即操作员、雇员，或具有对设备（例如，测试器和探测器）的访问权限的其他个人可能秘密地安装恶意软件（例如，损坏的（corrupted）测试程序）以企图获取秘密信息（例如，密钥）。此外，甚至在制造设施的外面，攻击者也可能通过外部触点企图使用它们自己的设备来企图访问秘密（例如，密钥）。如果秘密被发现，这可能危害，集成电路的安全性和/或在其中处理的秘密信息，或或至少对危害有贡献。

[0047] 图3是具有安全密钥推导和密码逻辑302的处理器300的实施例的框图，安全密钥推导和密码逻辑302可任选地包括测试和调试逻辑330的实施例。安全密钥推导和密码逻辑302包括根密钥生成逻辑304、密码基元逻辑306、可选的密钥推导逻辑308、推导的密钥存储器310，可选的加密/解密逻辑312，以及本地互连314。处理器还包括其他逻辑318。这些组件中的每一个都可以可任选地与图1的处理器100的相应地命名的组件具有相同或类似的特征。此外，在某些实施例中，根密钥生成逻辑304可以与图2的逻辑相同或类似。为避免模糊描述，下面将主要详细地描述图3的处理器300的不同的和/或附加结构和特征。

[0048] 在某些实施例中，安全密钥推导和密码逻辑302可以被包括在物理的和/或逻辑的、外部测试和/或调试被阻止的边界316内。在某些实施例中，由边界316外部的实体（例如，边界外面的测试器和探测器、制造设备、外部扫描链、处理器上的逻辑等等）对边界316内的逻辑所进行的扫描、测试，和/或调试被予以阻止或被禁用（即，不允许执行）。边界可以表示外部扫描阻止或禁用的边界、外部测试阻止或禁用的边界、外部调试阻止或禁用的边界等等。外部实体可不被允许检查或知道边界内的逻辑的内部状态。例如，在某些实施例中，可以没有线、线路，或其他互连和/或逻辑以允许边界内的逻辑的内部状态被通过处理器的暴露的或外部的触点（例如，针脚、接触垫等等）而访问或检查。类似地，在某些实施例中，可以没有互连和/或逻辑以允许边界内的逻辑的内部状态被边界外面的处理器的管芯上的逻辑而访问或检查。作为示例，一般的芯片范围内的调试能力，诸如扫描链和芯片范围内的内装自测试（BIST），可能不被在边界内使用。相比之下，外部实体能够扫描、测试，或调试边界外面的处理器的其他逻辑（例如，逻辑318）。可任选地，如果安全密钥推导和密码逻辑包括没有对任何秘密或安全信息的访问权限并且可以被外部实体扫描、测试，和/或调试而不会有任何明显的危害安全的风险的某种逻辑，那么，这样的逻辑可以可能地并可任选地被从阻止的边界中排除。有利地，这可以帮助防止被损坏（corrupted）的测试设备、或其他攻击实体扫描边界内的安全密钥推导和密码逻辑的内部状态并可能地获取秘密或安全信息（例如，根密钥、推导的密钥，等等）或试图配置逻辑以不安全的方式操作，等等。

[0049] 在某些实施例中，扫描、测试，和/或调试逻辑330可以被包括在边界316内。在某些实施例中，扫描、测试，和/或调试逻辑可以能够用于扫描、测试，和/或调试边界的范围内的至少某些或所有逻辑302。就是说，逻辑302，或至少其一部分，可以使用扫描、测试，和/或调试逻辑来执行自含式的（self-contained）扫描、测试，和/或调试。扫描或测试可以允许判定逻辑是否正确地和/或根据需要运转（例如，是否有一个或多个漏洞）。在某些情况下，如果逻辑不是正确地和/或根据需要运转，则可以有某些修复漏洞的能力，例如，通过重新配置逻辑，禁用不正确地运转的逻辑，等等，所有这些都从边界的范围之内采取。在某些实施

例中,扫描、测试,和/或调试逻辑可包括基于扫描的内装自测试(BIST)和可选的调试逻辑。可另选地,在其他实施例中,扫描、测试,和/或调试逻辑330可以在其中扫描、测试,和/或调试逻辑102并不需要,或希望牺牲扫描、测试,和/或调试逻辑102以便提高安全性的各实施例中,可任选地被省略。

[0050] 图4是具有安全密钥推导和密码逻辑402的处理器400的实施例的框图,安全密钥推导和密码逻辑402可任选地包括物理和/或逻辑安全认证的边界416。安全密钥推导和密码逻辑402包括根密钥生成逻辑404、密码基元逻辑406、密钥推导逻辑408、可选的推导的密钥存储器410、可选的加密/解密逻辑412、本地互连414,以及可选的扫描、测试,和/或调试逻辑430。处理器还包括其他逻辑418。这些组件中的每一个都可以可任选地与图1的处理器100和/或图3的处理器300的相应地命名的组件具有相同或类似的特征。此外,在某些实施例中,根密钥生成逻辑404可以与图2的逻辑相同或类似。为避免模糊描述,下面将主要详细地描述图4的处理器400的不同的和/或附加结构和特征。

[0051] 安全认证的边界416包括安全密钥推导和密码逻辑402。在某些实施例中,经安全认证的边界被联邦政府协会、联邦政府机构、联邦政府管理当局、联邦政府标准化机关等等证明是安全的。在某些实施例中,安全认证的边界被有声望的安全协会、标准化机关等等证明是安全的。在某些实施例中,安全认证的边界可以表示联邦信息处理标准(FIPS)证明的边界。在某些实施例中,FIPS认证的边界可以根据在2001年5月25日发布的并在2002年12月3日更新的FIPS公布140-2而被认证,尽管这不是必需的。

[0052] 在某些实施例中,在安全认证的边界内,所有密码算法(例如,密钥推导算法、加密算法、解密算法、散列算法等等)都可以是被认证安全性的机构所接受的标准化的算法。在某些实施例中,所有或大部分算法都可以是美国国家标准与技术协会(NIST)标准化的算法、联邦信息处理标准(FIPS)标准化的算法,或其他美国联邦政府接受的标准化的算法。在某些实施例中,不是如此标准化的算法中的任何一个都可以是美国国家标准协会(ANSI)、国际标准化组织(ISO)、电气电子工程师学会(IEEE),或类似地标准化的算法。在某些实施例中,在安全认证的边界内,所有密码算法都可以适用于对边界的FIPS认证。

[0053] 在某些实施例中,可不能由安全认证的边界外部的实体来扫描、测试,和/或调试安全认证的边界内的逻辑。在某些实施例中,内部扫描、测试,和/或调试逻辑430可以被包括在边界内。可另选地,在其他实施例中,可以省略内部扫描、测试,和/或调试逻辑。

[0054] 在某些实施例中,所有安全密钥推导和密码逻辑402都可以以管芯上的或处理器上的逻辑(即,潜在地带有某些固件但不带有任何软件的硬件)来实现。在某些实施例中,所有安全密钥派生和密码逻辑都可以主要、几乎完全,或完全以管芯上的和/或处理器上的硬件逻辑来实现。

[0055] 图5是具有安全密钥派生和密码逻辑502的处理器的实施例500的框图,安全密钥派生和密码逻辑502可任选地包括数字随机数生成器逻辑540和相关联的缓冲区。安全密钥推导和密码逻辑502包括根密钥生成逻辑504、密码基元逻辑506、密钥推导逻辑508、可选的推导的密钥存储器510,以及可选的加密/解密逻辑512,以及本地互连514。处理器还包括其他逻辑418。这些组件中的每一个都可以可任选地与图1的处理器的相应地命名的组件具有相同或类似的特征。安全密钥推导和密码逻辑502还包括可选的扫描、测试,和/或调试逻辑530,该逻辑530可以可任选地与图3的扫描、测试,和/或调试逻辑具有相同或类似的特征。

安全密钥推导和密码逻辑502包括在边界516内。边界516可以可任选地与图1的边界和/或图3的边界和/或图4的边界具有相同或类似的特征。在某些实施例中，边界516是FIPS认证的边界，虽然这不是必需的。为避免模糊描述，下面将主要详细地描述图5的处理器的不同的和/或附加结构和特征。

[0056] 数字随机数生成器(DRNG)逻辑540以及其相关联的缓冲区(buffer)542通常被包括在处理器或其他集成电路中。DRNG逻辑以及缓冲区的合适的示例是本领域已知的。作为示例，它们可以被用来生成随机密钥。如图所示，在某些实施例中，DRNG逻辑和缓冲区被包括在安全密钥推导和密码逻辑502中和/或边界516内。将DRNG逻辑和缓冲区包括在逻辑502中和/或边界内的一项优点在于高效率的实现，其允许某些逻辑被DRNG逻辑及逻辑502的其他部分重复使用或共享。逻辑的这种重复使用或共享可以允许降低逻辑的总量(例如，通过避免逻辑的重复)，这可以帮助，例如，降低集成电路的制造成本和功率消耗。

[0057] 例如，在某些实施例中，密码基元逻辑506(例如，块密码逻辑)可以被DRNG逻辑540，和密钥推导逻辑508、密钥推导逻辑508，和/或加密/解密逻辑512中的一个或多个所使用和/或共享。作为另一个示例，在某些实施例中，扫描、测试，和/或调试逻辑530可以被DRNG逻辑540，和密钥推导逻辑508、密钥推导逻辑508，和/或加密/解密逻辑512中的一个或多个所使用和/或共享。在某些实施例中，扫描、测试，和/或调试逻辑可包括逻辑的第一部分544以用于扫描、测试，和/或调试逻辑504-514中的一个或多个，逻辑的第二部分546以用于扫描、测试、和/或调试逻辑540和542。作为再一个示例，在某些实施例中，熵提取逻辑(例如，熵提取逻辑224)可以被基于PUF的根密钥生成系统和DRNG逻辑540两者所使用和/或共享。DRNG逻辑的熵源可能不是完美的，并可以得益于熵提取逻辑以改善或调节熵/随机性。不是去复制逻辑，熵提取逻辑可以被共享或重复使用以用于两种目的。作为再一个示例，在某些实施例中，本地互连514可以被DRNG逻辑540，和密钥推导逻辑508、密钥推导逻辑508，和/或加密/解密逻辑512中的一个或多个所使用和/或共享。作为更进一步的示例，在某些实施例中，总线端点逻辑547，以及时钟和功率(例如，功率选通)逻辑548可以被DRNG逻辑540，和密钥推导逻辑508、密钥推导逻辑508，和/或加密/解密逻辑512中的一个或多个使用和/或共享。

[0058] 图6是具有能够用于提供加密地实施的域分离的安全密钥推导和密码逻辑602的处理器600的计算系统650的框图。安全密钥推导和密码逻辑602可以用于图1、3、4或5的处理器中的任何一个中。可另选地，安全密钥推导和密码逻辑602可以用于类似的或不同的处理器中。此外，图1、3、4、或5的处理器还可以使用与图6的相同，类似或不同的安全密钥推导和密码逻辑。在某些实施例中，安全密钥推导和密码逻辑602可包括图2的基于PUF的根密钥生成逻辑，虽然这不是必需的。

[0059] 处理器和计算系统包括多个域654。如图所示，某些域可以是管芯上的或处理器上的，诸如域1到域X，而其他域可以是管芯之外的或处理器之外的但是在计算系统中，诸如域X+1到域N。对于特定实现，数字X和N可以是任何合适的数字(例如，从几个到数千的量级)。任何常规的处理器级别或系统级别的组件、逻辑，或实体都可以潜在地被用作域。几个代表性的示例包括，但不仅限于，线程、核、硬件单元、密码逻辑、协处理器、图形处理器、图形卡、通信卡、虚拟机、虚拟机监视器、安全引擎、加密模块、传感器中枢、硬件IP块，等等。

[0060] 安全密钥推导和密码逻辑602包括域标识符存储器656。在所示实施例中，域标识

符存储器656可任选地被示为在安全密钥推导和密码逻辑602内,虽然这不是必需的。在其他实施例中,存储器656可以位于别处,诸如,例如,在访问控制逻辑652中。域标识符存储器包括或存储系统中的多个域中的每一个的不同的域标识符(ID)658。例如,可以有域ID 1到域ID N,每一个都对应于N个域中的不同的域。作为示例,不同的域ID中的每一个都可以是该域所特有的、不同的预定的静态比特序列。作为示例,每个域ID,可以有大约从大约五个到大约数十个这样的比特。这些域ID可以表示并被用作域特有的额外信息,以用于对密码算法求值。作为示例,这些域ID可以存储在安全密钥推导和密码逻辑内的非易失性存储器中(例如,在FIPS边界、调试禁用的边界,或围绕安全密钥推导和密码逻辑的其他边界内,如在别处所公开的)。

[0061] 在某些实施例中,各种域654可以通过处理器的可选的访问控制逻辑652来访问安全密钥推导和密码逻辑。通常,这主要是实施域分离(domain separation)和访问控制的硬件逻辑。本领域中已知的常规访问控制逻辑都是合适的。访问控制逻辑可以通过防止一个域获取计划用于另一个域的数据,来帮助实施域分离。例如,如果第一域请求数据,则访问控制逻辑可能向第一域提供数据,而不允许其他域能够获取数据。作为示例,访问控制逻辑可,例如,在域所耦合到的总线的端点处,包括硬件过滤机制,该过滤机制能够用于有选择地过滤掉数据,以防止它被其未打算发往的域获取。在某些实施例中,访问控制逻辑可以如转让给本申请的受让人的在2010年9月24日提出的标题为“METHOD FOR ENFORCING RESOURCE ACCESS CONTROL IN COMPUTER SYSTEMS”的专利申请12/890,040所描述。可另选地,可以使用其他访问控制逻辑。在其他实施例中,可以可任选地省略访问控制逻辑。

[0062] 在某些实施例中,密码逻辑607可以使用域ID来帮助实施域分离。在某些实施例中,密码逻辑607可包括密钥推导逻辑,该密钥推导逻辑能够用于基于和/或依赖于域ID,推导一个或多个密钥。例如,当给定域向逻辑602提供对密钥的请求(例如,提供推导字符串)时,密钥推导逻辑可以生成基于对应于该域的给定域ID的密钥(例如,可以利用所提供的推导字符串和给定域ID,对密钥推导函数求值)。类似地,当其他域请求密钥时,将基于它们的对应的不同的并且唯一的域ID,生成密钥。作为示例,对于来自域X的带有个性化字符串“793”的第一密钥推导请求,不会为其返回和来自域N的带有个性化字符串“793”的第二密钥推导请求相同的推导的密钥,因为域X和N将具有将被包括到对密钥推导函数的求值中的不同的域ID。有利地,这可以被用来提供基于域分离的密钥推导。

[0063] 在某些实施例中,密码逻辑607可包括密钥推导逻辑和/或其他响应生成逻辑,该响应生成逻辑能够用于基于和/或依赖于域ID,提供对质询的响应。例如,来自给定域的质询可以产生基于该给定域的对应的域ID的响应(例如,可以至少部分地基于对应于该域的给定域ID,对密钥推导算法求值)。这可以防止不同的域获得对质询的相同响应。

[0064] 在某些实施例中,密码逻辑607可包括加密和/或解密逻辑,该加密和/或解密逻辑能够用于基于和/或依赖于域ID,加密和/或解密数据。例如,来自一个域的对明文数据的加密的请求可以基于该域的对应的域ID(例如,可以利用明文数据和给定域ID,对加密算法求值)。作为另一个示例,来自一个域的对密文的解密的请求可以基于该域的对应的域ID(例如,可以利用密文和给定域ID,对解密算法求值)。

[0065] 有利地,域ID以及能够用于使用它们的密码逻辑可以允许不同的域共享安全密钥推导和密码逻辑,同时独立地使用其服务,干扰或跨域秘密共享的风险降低。这可以帮助避

免或至少降低隐私和/或受关注的平台序列号类别的风险。在某些实施例中,这样的域ID和加密地实施的域分离的使用可以可任选地由用户配置为启用或禁用。

[0066] 图7是能够用于利用从PUF推导出的密钥来加密和解密熔丝值的熔丝逻辑(fuse logic)760的实施例的框图。熔丝逻辑可以用于图1、3、4、5或6的处理器中的任何一个中。可另选地,熔丝逻辑可以用于类似的或不同的处理器中。此外,图1、3、4、5,或6的处理器还可以使用与图7的相同,类似或不同的熔丝逻辑。

[0067] 熔丝逻辑包括熔丝存储器762(该熔丝存储器762可包括各种类型的随机存取存储器(RAM)中的任何一种),熔丝阵列764(其中,熔丝将被烧断或以别的方式编程),熔丝控制逻辑766,以及安全密钥推导和密码逻辑702。安全密钥推导和密码逻辑702可以与图1、3、4、5,或6的相应地命名的逻辑的那个相同、类似,或不同。

[0068] 要被编程的熔丝值可以存储在熔丝存储器762中。在熔丝控制逻辑将它们编程到熔丝阵列764之前,熔丝值可以被加密。可以将熔丝值提供到安全密钥推导和密码逻辑702。其密码逻辑712可以使用基于来自基于PUF的根密钥生成逻辑705的PUF根密钥的一个或多个密钥,来加密熔丝值。在某些实施例中,可以使用PUF根密钥。在其他实施例中,可以使用从PUF根密钥推导出的密钥。在某些实施例中,基于PUF的根密钥生成逻辑705可以与图2的那个逻辑相同、类似,或不同。可以将经加密的熔丝值提供回至熔丝控制逻辑,该熔丝控制逻辑可以使用编程逻辑767将熔丝编程到熔丝阵列中。当由感应逻辑765从熔丝阵列中读取经加密的熔丝值时(例如,在平台重置之后),熔丝控制器可以将经加密的熔丝值提供到安全密钥推导和密码逻辑702的解密逻辑712。解密逻辑可以解密经加密的熔丝值,并将经解密的熔丝值提供给熔丝控制逻辑。熔丝控制逻辑可以将经解密的熔丝值写入或存储到熔丝存储器762。

[0069] 有利地,对熔丝值的加密可以帮助保护存储在熔丝中的密钥免于物理攻击。即使攻击方能够读取熔丝,他们一般也不能得知所使用的实际密钥,因为所使用的密钥是通过来自基于PUF的安全密钥推导和密码逻辑702加密的。在某些实施例中,熔丝值加密可以对外部实体(例如,测试器和探测器或其他制造环境)不可见。在某些实施例中,熔丝逻辑760可以被包括在FIPS边界、其他安全认证的边界、其他调试禁用的边界,或本文的别处所公开的其他边界中。

[0070] 为避免使描述模糊,示出和描述了相对简单的处理器。在其他实施例中,处理器可以可任选地包括其他已知的组件。这样的组件的示例包括,但不仅限于,指令获取单元、指令调度单元、转移预测单元、指令和数据高速缓存、指令和数据转换后备缓冲器、预取缓冲器、微指令队列、微指令定序符、总线接口单元、第二或较高级别的高速缓存、隐退单元、寄存器重命名单元、处理器中所包括的其他组件,以及其各种组合。各实施例可以具有多个核、逻辑处理器,或执行引擎。有处理器中的组件的很多不同的组合和配置,各实施例不仅限于任何特定组合或配置。处理器可以表示集成电路或一组一个或多个半导体管芯或芯片(例如,单一管芯或芯片,或包括两个或更多管芯或芯片的封装)。在某些实施例中,处理器可以表示片上系统(SOC)。

[0071] 示例性核架构、处理器和计算机架构

[0072] 可以以不同方式、出于不同目的、在不同的处理器中实现处理器核。例如,这样的核的实现可以包括:1)旨在用于通用计算的通用有序核;2)预期用于通用计算的高性能通

用无序核；3)旨在主要用于图形和/或科学(吞吐量)计算的专用核。不同处理器的实现可包括：1)包括旨在用于通用计算的一个或多个通用有序核和/或旨在用于通用计算的一个或多个通用无序核的CPU；以及2)包括旨在主要用于图形和/或科学(吞吐量)的一个或多个专用核的协处理器。这样的不同处理器导致不同的计算机系统架构，其可包括：1)在与CPU分开的芯片上的协处理器；2)在与CPU相同的封装中但分开的管芯上的协处理器；3)与CPU在相同管芯上的协处理器(在该情况下，这样的协处理器有时被称为诸如集成图形和/或科学(吞吐量)逻辑等专用逻辑，或被称为专用核)；以及4)可以将所描述的CPU(有时被称为应用核或应用处理器)、以上描述的协处理器和附加功能包括在同一管芯上的芯片上系统。

[0073] 接着描述示例性核架构，随后描述示例性处理器和计算机架构。

[0074] 示例性核架构

[0075] 有序和无序核框图

[0076] 图8A是示出根据本发明的各实施例的示例性有序流水线和示例性的寄存器重命名的无序发布/执行流水线的框图。图8B是示出根据本发明的各实施例的要包括在处理器中的有序架构核的示例性实施例和示例性的寄存器重命名的无序发布/执行架构核的框图。图8A-B中的实线框示出有序流水线和有序核，而任选增加的虚线框示出寄存器重命名的无序发布/执行流水线和核。给定有序方面是无序方面的子集的情况下，将描述无序方面。

[0077] 在图8A中，处理器流水线800包括取出级802、长度解码级804、解码级806、分配级808、重命名级810、调度(也称为分派或发布)级812、寄存器读取/存储器读取级814、执行级816、写回/存储器写入级818、异常处理级822以及提交级824。

[0078] 图8B示出了处理器核890，包括耦合到执行引擎单元850的前端单元830，执行引擎单元850和前端单元830两者都耦合到存储器单元870。核890可以是精简指令集计算(RISC)核、复杂指令集计算(CISC)核、超长指令字(VLIW)核或混合或替代核类型。作为又一选项，核890可以是专用核，诸如例如网络或通信核、压缩引擎、协处理器核、通用计算图形处理器单元(GPGPU)核、或图形核等等。

[0079] 前端单元830包括耦合到指令高速缓存单元834的分支预测单元832，指令高速缓存单元834耦合到指令转换后备缓冲器(TLB)836，指令转换后备缓冲器836耦合到指令取出单元838，指令取出单元838耦合到解码单元840。解码单元840(或解码器)可解码指令，并生成从原始指令解码出的、或以其他方式反映原始指令的、或从原始指令导出的一个或多个微操作、微代码进入点、微指令、其他指令、或其他控制信号作为输出。解码单元840可使用各种不同的机制来实现。合适的机制的示例包括但不限于查找表、硬件实现、可编程逻辑阵列(PLA)、微代码只读存储器(ROM)等。在一个实施例中，核890包括微代码ROM，或存储某些宏指令的微代码(例如，在解码单元840中或以其它方式在前端单元830内的)的其他介质。解码单元840耦合到执行引擎单元850中的重命名/分配器单元852。

[0080] 执行引擎单元850包括耦合到隐退单元852的重命名/分配器单元854和一组一个或多个调度器单元856。调度器单元856表示任意数量的不同的调度器，包括预留站、中心指令窗口等等。调度器单元856耦合到物理寄存器组单元858。每个物理寄存器组单元858表示一个或多个物理寄存器组，其中不同的物理寄存器组存储一种或多种不同的数据类型，诸如标量整数、标量浮点、紧缩整数、紧缩浮点、向量整数、向量浮点、状态(例如，作为要执行

的下一指令的地址的指令指针)等。在一个实施例中,物理寄存器组单元858包括向量寄存器单元、写掩码寄存器单元和标量寄存器单元。这些寄存器单元可以提供架构向量寄存器、向量掩码寄存器、和通用寄存器。物理寄存器组单元858与引退单元854重叠以示出可以用来实现寄存器重命名和无序执行的各种方式(例如,使用重新排序缓冲器和引退寄存器组;使用将来的文件、历史缓冲器和引退寄存器组;使用寄存器映射和寄存器池等等)。隐退单元854和物理寄存器组单元858耦合到执行群集860。执行群集860包括一组一个或多个执行单元862和一组一个或多个存储器访问单元864。执行单元862可以对各种类型的数据(例如,标量浮点、紧缩整数、紧缩浮点、向量整型、向量浮点)执行各种操作(例如,移位、加法、减法、乘法)。尽管一些实施例可以包括专用于特定功能或功能组的若干个执行单元,但是,其他实施例可以只包括一个执行单元或都执行所有功能的多个执行单元。调度器单元856、物理寄存器组单元858,以及执行群集860被示为可能是多个,因为某些实施例对于某些类型的数据/操作创建单独的流水线(例如,标量整数流水线、标量浮点/紧缩整数/紧缩浮点/向量整数/向量浮点流水线和/或存储器访问流水线,每一个流水线都具有它们自己的调度器单元、物理寄存器组单元和/或执行群集——并且在单独的存储器访问流水线的情况下,实现了其中只有此流水线的执行群集具有存储器访问单元864的某些实施例)。还应该理解,使用单独的流水线,这些流水线中的一个或多个可以是无序发出/执行,其余的是有序的。

[0081] 存储器访问单元864的集合耦合到存储器单元870,该存储器单元包括耦合到数据高速缓存单元874的数据TLB单元872,其中数据高速缓存单元耦合到二级(L2)高速缓存单元876。在一个示例性实施例中,存储器访问单元864可以包括加载单元、存储地址单元以及存储数据单元,其中每一个都耦合到存储器单元870中的数据TLB单元872。指令高速缓存单元834进一步耦合到存储器单元870中的2级(L2)高速缓存单元876。L2高速缓存单元876耦合到一个或多个其他级的高速缓存,并最终耦合到主存储器。

[0082] 作为示例,示例性寄存器重命名的、无序发布/执行核架构可以如下实现流水线800:1)指令取出838执行取出和长度解码级802和804;2)解码单元840执行解码级806;3)重命名/分配器单元852执行分配级808和重命名级810;4)调度器单元856执行调度级812;5)物理寄存器组单元858和存储器单元870执行寄存器读取/存储器读取级814;执行群集860执行执行级816;6)存储器单元870和物理寄存器组单元858执行写回/存储器写入级818;7)各单元可牵涉到异常处理级822;以及8)引退单元854和物理寄存器组单元858执行提交级824。

[0083] 核890可支持一个或多个指令集(例如,x86指令集(具有与较新版本一起添加的一些扩展);加利福尼亚州桑尼维尔市的MIPS技术公司的MIPS指令集;加利福尼州桑尼维尔市的ARM控股的ARM指令集(具有诸如NEON等可选附加扩展)),其中包括本文中描述的各指令。在一个实施例中,核890包括支持紧缩数据指令集合扩展(例如,AVX1、AVX2)的逻辑,由此允许被许多多媒体应用使用的操作将使用紧缩数据来执行。

[0084] 应该理解,核可以支持多线程(执行操作或线程的两个或更多并行组),并可以以各种方式达到这一目的,包括时间切片多线程,同时的多线程(其中,单个物理核为物理核同时正在多线程处理的每一个线程提供一种逻辑核),或其组合(例如,时间切片获取和解码和此后的同时的多线程处理,诸如在Intel®Hyperthreading技术中)。

[0085] 尽管寄存器重命名是在无序执行的上下文中描述的,但是,应该理解,寄存器重命名可以用于有序架构中。尽管所示出的处理器的实施例还包括分开的指令和数据高速缓存单元834/874以及共享L2高速缓存单元876,但替代实施例可以具有用于指令和数据两者的一个内部高速缓存,诸如例如一级(L1)内部高速缓存或多个级别的内部高速缓存。在某些实施例中,系统可以包括内部高速缓存和核和/或处理器外部的外部高速缓存的组合。可另选地,全部高速缓存都可以核和/或处理器外部的。

[0086] 具体的示例性有序核架构

[0087] 图9A-B示出了更为具体的示例性有序核架构的框图,该核将是芯片中的多个逻辑块中的一个(包括相同类型和/或不同类型的其他核)。取决于应用,这些逻辑块通过高带宽的互连网络(例如,环形网络)与一些固定的功能逻辑、存储器I/O接口和其它必要的I/O逻辑通信。

[0088] 图9A是根据本发明的各实施例的单个处理器核的框图,以及其与管芯上的互连网络902的连接以及其第2级(L2)高速缓存904的本地子集。在一个实施例中,指令解码器900支持具有紧缩数据指令集扩展的x86指令集。L1高速缓存906允许对进入标量和向量单元中的高速缓存存储器的低等待时间访问。尽管在一个实施例中(为了简化设计),标量单元908和向量单元910使用分开的寄存器集合(分别为标量寄存器912和向量寄存器914),并且在这些寄存器之间转移的数据被写入到存储器并随后从一级(L1)高速缓存906读回,但是本发明的替代实施例可以使用不同的方法(例如使用单个寄存器集合或包括允许数据在这两个寄存器组之间传输而无需被写入和读回的通信路径)。

[0089] L2高速缓存的本地子集904是全局L2高速缓存的一部分,该全局L2高速缓存被划分成多个分开的本地子集,即每个处理器核一个本地子集。每个处理器核具有到其自己的L2高速缓存904的本地子集的直接访问路径。被处理器核读出的数据被存储在其L2高速缓存子集904中,并且可以与其他处理器核访问其自己的本地L2高速缓存子集并行地被快速访问。被处理器核写入的数据被存储在其自己的L2高速缓存子集904中,并在必要的情况下从其它子集清除。环形网络确保共享数据的一致性。环形网络是双向的,以允许诸如处理器核、L2高速缓存和其它逻辑块之类的代理在芯片内彼此通信。每个环形数据路径为每个方向1012位宽。

[0090] 图9B是根据本发明的实施例的图9A中的处理器核的一部分的展开图。图9B包括L1高速缓存904L1数据高速缓存906A部分,以及关于向量单元910和向量寄存器914的更多细节。具体地说,向量单元910是16宽向量处理单元(VPU)(见16宽ALU 928),该单元执行整型、单精度浮点以及双精度浮点指令中的一个或多个。该VPU通过混合单元920支持对寄存器输入的混合、通过数值转换单元922A-B支持数值转换、并通过复制单元924支持对存储器输入的复制。写掩码寄存器926允许断言所产生的向量写。

[0091] 具有集成存储器控制器和图形器件的处理器

[0092] 图10是根据本发明的各实施例的可能具有一个以上核、可能具有集成存储器控制器、以及可能具有集成图形器件的处理器1000的框图。图10中的实线框示出具有单个核1002A、系统代理1000、一个或多个总线控制器单元1010的集合的处理器1000,而虚线框的可选附加示出具有多个核1002A-N、系统代理单元1016中的一个或多个集成存储器控制器单元1014的集合以及专用逻辑1008的处理器1100。

[0093] 因此,处理器1000的不同实现可包括:1) CPU,其中专用逻辑1008是集成图形和/或科学(吞吐量)逻辑(其可包括一个或多个核),并且核1002A-N是一个或多个通用核(例如,通用有序核、通用无序核、这两者的组合);2) 协处理器,其中核1002A-N是旨在主要用于图形和/或科学(吞吐量)的多个专用核;以及3) 协处理器,其中核1002A-N是多个通用有序核。因此,处理器1000可以是通用处理器、协处理器或专用处理器,诸如例如网络或通信处理器、压缩引擎、图形处理器、GPGPU(通用图形处理单元)、高吞吐量的集成众核(MIC)协处理器(包括30个或更多核)、或嵌入式处理器等。该处理器可以被实现在一个或多个芯片上。处理器1000可以是一个或多个衬底的一部分,和/或可以使用诸如例如BiCMOS、CMOS或NMOS等的多个加工技术中的任何一个技术将处理器200实现在一个或多个衬底上。

[0094] 存储器层次结构包括核内的一级或多级高速缓存,一组或一个或多个共享高速高速缓存单元1006,以及耦合到集成的存储器控制器单元1014组的外部存储器(未示出)。共享高速高速缓存单元1006的集合可以包括一个或多个中级高速缓存,诸如2级(L2)、3级(L3)、4级(L4),或其他级别的高速缓存,末级高速缓存(LLC),和/或其组合。尽管在一个实施例中,基于环的互连单元1012将集成图形逻辑1008、共享高速缓存单元1006的集合以及系统代理单元1010/集成存储器控制器单元1014互连,但替代实施例可使用任何数量的公知技术来将这些单元互连。在一个实施例中,维持一个或多个高速缓存单元1006和核1002-A-N之间的一致性(coherency)。

[0095] 在某些实施例中,核1002A-N中的一个或多个能够多线程处理。系统代理1010包括协调和操作核1002A-N的那些组件。系统代理单元1010可包括例如功率控制单元(PCU)和显示单元。PCU可以是或包括用于管理核1002A-N和集成的图形逻辑1008的电能状态所需的逻辑和组件。显示单元用于驱动一个或多个从外部连接的显示器。

[0096] 核1002A-N在架构指令集方面可以是同构的或异构的;即,这些核1002A-N中的两个或更多个核可能能够执行相同的指令集,而其他核可能能够执行该指令集的仅仅子集或不同的指令集。

[0097] 示例性计算机架构

[0098] 图11-14是示例性计算机架构的框图。本领域已知的对膝上型设备、台式机、手持PC、个人数字助理、工程工作站、服务器、网络设备、网络集线器、交换机、嵌入式处理器、数字信号处理器(DSP)、图形设备、视频游戏设备、机顶盒、微控制器、蜂窝电话、便携式媒体播放器、手持设备以及各种其他电子设备的其他系统设计和配置也是合适的。一般地,能够包含本文中所公开的处理器和/或其他执行逻辑的多个系统和电子设备一般都是合适的。

[0099] 现在请参看图11,所示是根据本发明的一个实施例的系统1100的框图。系统1100可以包括一个或多个处理器1110、1115,这些处理器耦合到控制器中枢1120。在一个实施例中,控制器中枢1120包括图形存储器控制器中枢(GMCH)1190和输入/输出中枢(IOH)1150(其可以在分开的芯片上);GMCH 1190包括存储器和图形控制器,存储器1140和协处理器1145耦合到该存储器和图形控制器;IOH 1150将输入/输出(I/O)设备1160耦合到GMCH 1190。可另选地,存储器和图形控制器中的一个或两者都集成在处理器内(如此处所描述的),存储器1140和协处理器1145利用IOH1150,直接耦合到单个芯片中的处理器1110以及控制器中枢1120。

[0100] 在图11中利用虚线表示额外的处理器1115的可任选的特性。每一处理器1110、

1115可包括本文中描述的处理核中的一个或多个，并且可以是处理器1000的某一版本。

[0101] 存储器1140可以是例如动态随机存取存储器(DRAM)、相变存储器(PCM)或这两者的组合。对于至少一个实施例，控制器中枢1120经由诸如前端总线(FSB)之类的多分支总线、诸如快速通道互连(QPI)之类的点对点接口、或者类似的连接1195与处理器1110、1115进行通信。

[0102] 在一个实施例中，协处理器1145是专用处理器，诸如例如高吞吐量MIC处理器、网络或通信处理器、压缩引擎、图形处理器、GPGPU、或嵌入式处理器等等。在一个实施例中，控制器中枢1120可以包括集成图形加速器。

[0103] 就包括架构、微架构、热的，功率消耗特征等等的一系列优点的度量而言，在物理资源1110、1115之间可能会有各种差异。

[0104] 在一个实施例中，处理器1110执行控制一般类型的数据处理操作的指令。协处理器指令可嵌入在这些指令中。处理器1110将这些协处理器指令识别为应当由附连的协处理器1145执行的类型。因此，处理器1110在协处理器总线或者其他互连上将这些协处理器指令(或者表示协处理器指令的控制信号)发布到协处理器1145。协处理器1145接受并执行所接收的协处理器指令。

[0105] 现在请参看图12，所示是根据本发明的一个实施例的第一更具体的示例性系统1200的框图。如图12所示，多处理器系统1200是点对点互连系统，并包括通过点对点互连1250耦合的第一处理器1270和第二处理器1280。处理器1270和1280中的每一个都可以是处理器1000的某一版本。在本发明的一个实施例中，处理器1270和1280分别是处理器1110和1115，而协处理器1238是协处理器1145。在另一实施例中，处理器1270和1280分别是处理器1110和协处理器1145。

[0106] 处理器1270和1280被示为分别包括集成存储器控制器(IMC)单元1272和1282。处理器1270还包括点对点(P-P)接口1276和1278，作为其总线控制器单元的一部分；类似地，第二处理器1280包括P-P接口1286和1288。处理器1270、1280可以使用点对点(P-P)接口电路1278、1288经由P-P接口1250来交换信息。如图12所示，IMC 1272和1282将处理器耦合到相应的存储器，即，存储器1232和存储器1234，它们可以是本地连接到相应的处理器的主存储器的一部分。

[0107] 处理器1270、1280可各自经由使用点对点接口电路1276、1294、1286、1298的各个P-P接口1252、1254与芯片组1290交换信息。芯片组1290可以可选地经由高性能接口1239与协处理器1238交换信息。在一个实施例中，协处理器1238是专用处理器，诸如例如高吞吐量MIC处理器、网络或通信处理器、压缩引擎、图形处理器、GPGPU、或嵌入式处理器等等。

[0108] 共享高速缓存(未示出)可以被包括在任一处理器之内，或被包括在两个处理器外部但仍经由P-P互连与这些处理器连接，从而如果将某处理器置于低功率模式时，可将任一处理器或两个处理器的本地高速缓存信息存储在该共享高速缓存中。

[0109] 芯片组1290可经由接口1296耦合至第一总线1216。在一个实施例中，第一总线1216可以是外围组件互连(PCI)总线，或诸如PCI Express总线或另一第三代I/O互连总线之类的总线，但本发明的范围并不受此限制。

[0110] 如图12所示，各种I/O设备1214，以及将第一总线1216耦合到第二总线1220的总线桥1218可以耦合到第一总线1216。在一个实施例中，诸如协处理器、高吞吐量MIC处理器、

GPGPU的处理器、加速器(诸如例如图形加速器或数字信号处理器(DSP)单元)、现场可编程门阵列或任何其他处理器的一个或多个附加处理器1215被耦合到第一总线1216。在一个实施例中,第二总线1220可以是低引脚计数(LPC)总线。各种设备可以被耦合至第二总线1220,在一个实施例中这些设备包括例如键盘/鼠标1222、通信设备1227以及诸如可包括指令/代码和数据1228的盘驱动器或其他大容量存储设备的存储单元1230。此外,音频I/O 1224可以被耦合至第二总线1220。请注意,其他架构也是可以的。例如,代替图12的点对点架构,系统可以实现多点分支总线或其他这样的架构。

[0111] 现在请参看图13,所示是根据本发明的一个实施例的第二更具体的示例性系统1300的框图。图12和13中的相同元素带有相同参考编号,从图13省略了图12的某些方面,以便不至于使图13的其他方面变得模糊。

[0112] 图13示出了处理器1270、1280可以分别包括集成的存储器和I/O控制逻辑(“CL”)1272和1282。因此,CL 1272、1282包括集成存储器控制器单元并包括I/O控制逻辑。图13示出了不仅存储器1232,1234耦合到CL 1272,1282,而且I/O设备1314也耦合到控制逻辑1272,1282。传统I/O设备1315被耦合至芯片组1290。

[0113] 现在请参看图14,所示是根据本发明的实施例的SoC 1400的框图。图10中的类似的元素带有相同的参考编号。另外,虚线框是更先进的SoC的可选特征。在图14中,互连单元1402耦合到:应用处理器1410,该应用处理器包括一个或多个核202A-N的集合以及共享高速缓存单元1006;系统代理单元1010;总线控制器单元1016;集成存储器控制器单元1014;一组或一个或多个协处理器1420,其可包括集成图形逻辑、图像处理器、音频处理器和视频处理器;静态随机存取存储器(SRAM)单元1430;直接存储器存取(DMA)单元1432;以及用于耦合至一个或多个外部显示器的显示单元1440。在一个实施例中,协处理器1420包括专用处理器,诸如例如网络或通信处理器、压缩引擎、GPGPU、高吞吐量MIC处理器、或嵌入式处理器等等。

[0114] 本文公开的机制的各实施例可以被实现在硬件、软件、固件或这些实现方法的组合中。本发明的实施例可实现为在可编程系统上执行的计算机程序或程序代码,该可编程系统包括至少一个处理器、存储系统(包括易失性和非易失性存储器和/或存储元件)、至少一个输入设备以及至少一个输出设备。

[0115] 可以将诸如图12中所示出的代码1230之类的程序代码应用于输入指令,以执行此处所描述的功能并生成输出信息。可以按已知方式将输出信息应用于一个或多个输出设备。为了本申请的目的,处理系统包括具有诸如例如数字信号处理器(DSP)、微控制器、专用集成电路(ASIC)或微处理器之类的处理器的任何系统。

[0116] 程序代码可以用高级程序化语言或面向对象的编程语言来实现,以便与处理系统通信。在需要时,也可用汇编语言或机器语言来实现程序代码。事实上,本文中描述的机制不限于任何特定编程语言的范围。在任一情形下,该语言可以是编译语言或解释语言。

[0117] 至少一个实施例的一个或多个方面可以由存储在机器可读介质上的表征性指令来实现,该指令表示处理器中的各种逻辑,该指令在被机器读取时使得该机器制作用于执行本文所述的技术的逻辑。被称为“IP核”的这样的表示可以存储在有形的机器可读介质中,并提供给各种客户或生产设施,以加载到实际制造逻辑或处理器的制造机器中。

[0118] 这样的机器可读存储介质可以包括但不限于通过机器或设备制造或形成的物品

的非瞬态的有形安排,其包括存储介质,诸如:硬盘;任何其它类型的盘,包括软盘、光盘、紧致盘只读存储器(CD-ROM)、紧致盘可重写(CD-RW)以及磁光盘;半导体器件,例如只读存储器(ROM)、诸如动态随机存取存储器(DRAM)和静态随机存取存储器(SRAM)之类的随机存取存储器(RAM)、可擦除可编程只读存储器(EPROM)、闪存、电可擦除可编程只读存储器(EEPROM);相变存储器(PCM);磁卡或光卡;或适于存储电子指令的任何其它类型的介质。

[0119] 因此,本发明的各实施例还包括非瞬态的有形机器可读介质,该介质包含指令或包含设计数据,诸如硬件描述语言(HDL),它定义本文中描述的结构、电路、装置、处理器和/或系统特征。这样的实施例还可以被称为程序产品。

[0120] 仿真(包括二进制变换、代码变形等)

[0121] 在某些情况下,可以使用指令转换器来将指令从源指令集转换为目标指令集。例如,指令转换器可以转换(例如,使用静态二进制转换、包括动态编译的动态二进制转换)、变形、模仿,或以别的方式将指令转换为要由核处理的一个或多个其他指令。指令转换器可以以软件、硬件、固件,或其组合来实现。指令转换器可以在处理器上、在处理器外、或者部分在处理器上且部分在处理器外。

[0122] 图15是根据本发明的各实施例的对照使用软件指令转换器将源指令集中的二进制指令转换成目标指令集中的二进制指令的框图。在所示的实施例中,指令转换器是软件指令转换器,但作为替代,该指令转换器可以用软件、固件、硬件或其各种组合来实现。图15示出了可以使用x86编译器1504来编译高级语言1502的程序以生成x86二进制代码1506,该x86二进制代码1506可以由带有至少一个x86指令集核1416的处理器来原生地(natively)执行。具有至少一个x86指令集核的处理器1516表示任何处理器,这些处理器能通过兼容地执行或以其他方式处理以下内容来执行与具有至少一个x86指令集核的英特尔处理器基本相同的功能:1)英特尔x86指令集核的指令集的本质部分,或2)目标为在具有至少一个x86指令集核的英特尔处理器上运行的应用或其他程序的目标代码版本,以便取得与具有至少一个x86指令集核的英特尔处理器基本相同的结果。x86编译器1504表示能够用于生成x86二进制代码1506(例如,目标代码)的编译器,该二进制代码1506可通过或不通过附加的链接处理在具有至少一个x86指令集核的处理器1516上执行。类似地,图15示出了可以使用替代指令集编译器1502来编译高级语言1508的程序以生成替代指令集二进制代码1510,该替代指令集二进制代码1514可以由没有至少一个x86指令集核1414的处理器(例如,带有执行位于CA的Sunnyvale的MIPS Technologies的MIPS指令集和/或执行位于CA的Sunnyvale的ARM Holdings的ARM指令集的核的处理器)来原生地执行。指令转换器1512被用来将x86二进制代码1506转换成可以由不具有x86指令集核的处理器1514原生执行的代码。该转换后的代码不大可能与替代性指令集二进制代码1510相同,因为能够这样做的指令转换器难以制造;然而,转换后的代码将完成一般操作并由来自替代指令集的指令构成。因此,指令转换器1512通过仿真、模拟或任何其他过程来表示允许不具有x86指令集处理器或核的处理器或其他电子设备执行x86二进制代码1506的软件、固件、硬件或其组合。

[0123] 在说明书和权利要求书中,可能使用了术语“耦合的”和“连接的”及其衍生词。应当理解,这些术语并不旨在作为彼此的同义词。相反,在特定实施例中,可以使用“连接”来表示两个或更多元件彼此处于直接的物理和/或电接触的状态。“耦合的”可表示两个或更多个元件直接物理或电接触。然而,“耦合的”也可表示两个或更多个元件并未彼此直接接

触,但是仍然彼此协作、彼此相互作用。

[0124] 术语“和/或”可能已被使用。如本文中所使用的,术语“和/或”意指一个或其他或两者(例如,A和/或B意指A或B或者A和B两者)。

[0125] 在上面的描述中,出于说明目的,阐述了众多具体细节以便提供对本发明的各实施例的全面理解。然而,对本领域技术人员将显而易见的是,没有这些具体细节中的某些也可实施一个或多个其他实施例。所描述的具体实施例不是为了限制本发明而是为了说明。本发明的范围不是由上面所提供的具体示例确定,而是仅由下面的权利要求确定。所有与附图中所示出的以及说明书中所描述的那些等效的关系都包含在本发明的各实施例内。在其他情况下,以框图形式,而不是详细地示出已知的电路、结构、设备,和操作以便不至于使对描述的理解变得模糊。

[0126] 在认为适宜之处,附图标记和/或附图标记的结尾部分在诸附图当中被重复以指示可选地具有类似特性或相同特征的对应或类似的要素,除非以其他方式来指定或显而易见。在示出和描述了多个组件的一些情况下,它们可被结合到单一组件中。在示出并描述单一组件的其他情况下,它可以可任选地被分离成两个或更多组件。在附图中,箭头表示耦合,双向箭头表示双向耦合。

[0127] 已描述了各种操作和方法。已经以流程图方式以相对基础的方式对一些方法进行了描述,但这些操作可选择地被添加至这些方法和/或从这些方法中移去。另外,尽管流程图示出根据各示例实施例的操作的特定顺序,但可以理解,该特定特定顺序是示例性的。替换实施例可以可任选地以不同方式执行这些操作、组合某些操作、交错某些操作等。可以对方法作出许多修改。设备的此处所描述的组件、特征,以及特定可选细节还可以可任选地应用于此处所描述的方法,在各实施例中,这些方法可以由这样的设备执行和/或利用这样的设备执行。

[0128] 还应该理解,在本说明书中,对,例如,“一个实施例”、“实施例”、“一个或多个实施例”引用,表示特定特征可以被包括在本发明的实施中。类似地,应该理解,在描述中,各种特点有时分组在单一实施例、图形或其描述中,以便简化说明,并帮助理解本发明的各个方面。然而,该公开方法不应被解释成反映本发明需要比每项权利要求中所明确记载的更多特征的意图。相反,如所附权利要求反映的,发明性方面可以在于少于单一公开的实施例的所有特征。因此,所附权利要求因此被明确纳入该说明书中,每一项权利要求独自作为本发明单独的实施例。

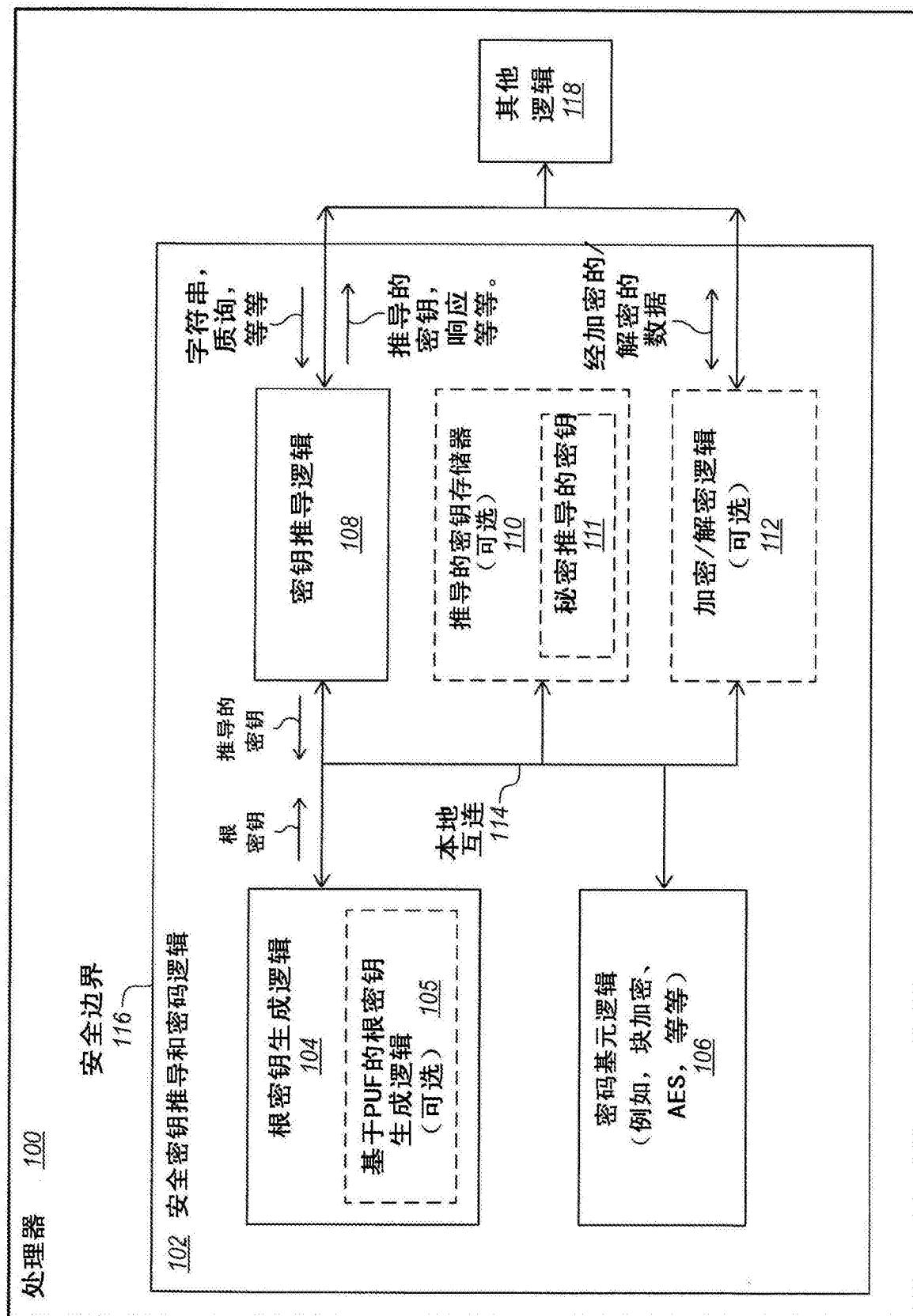


图1

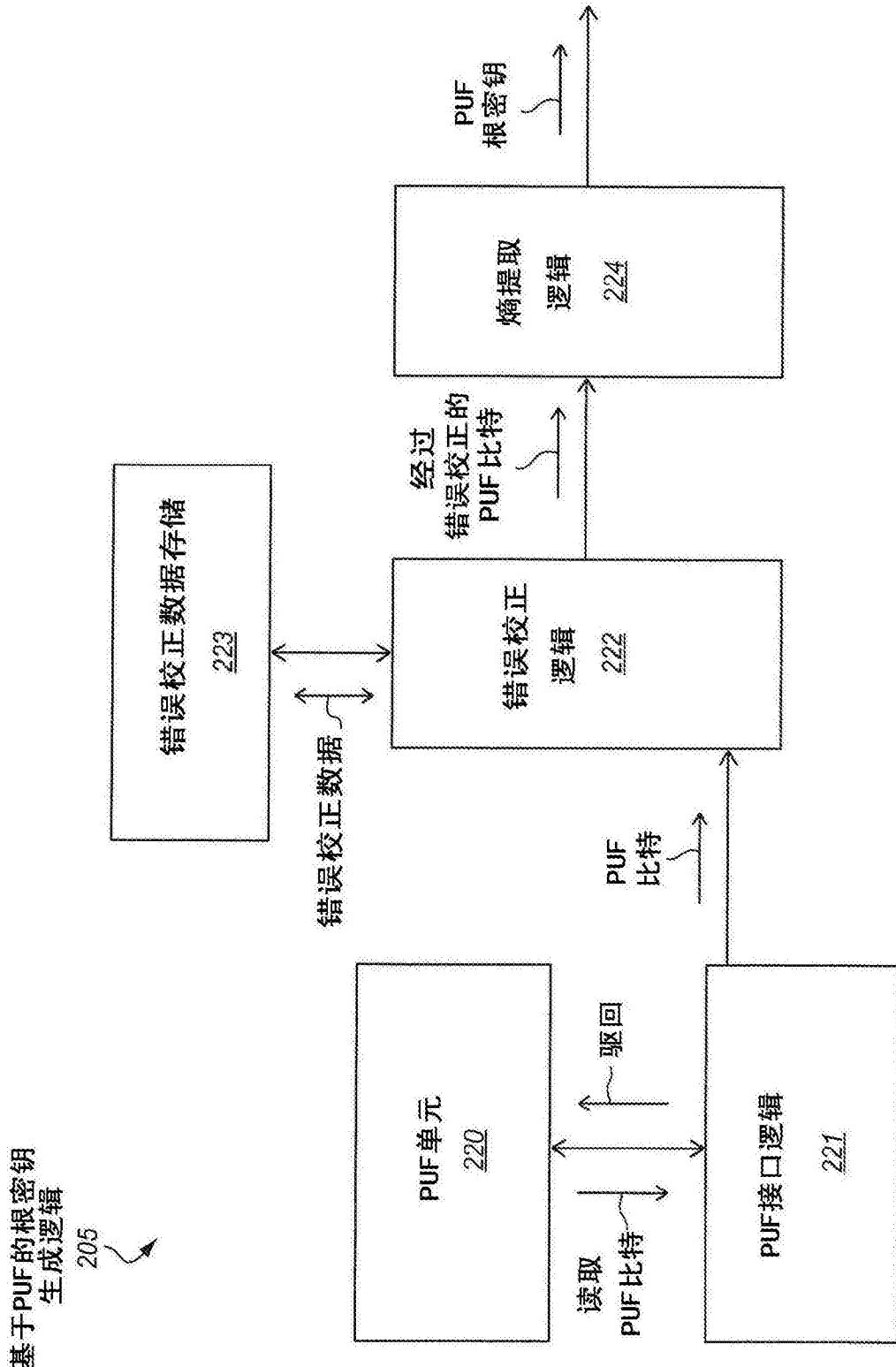


图2

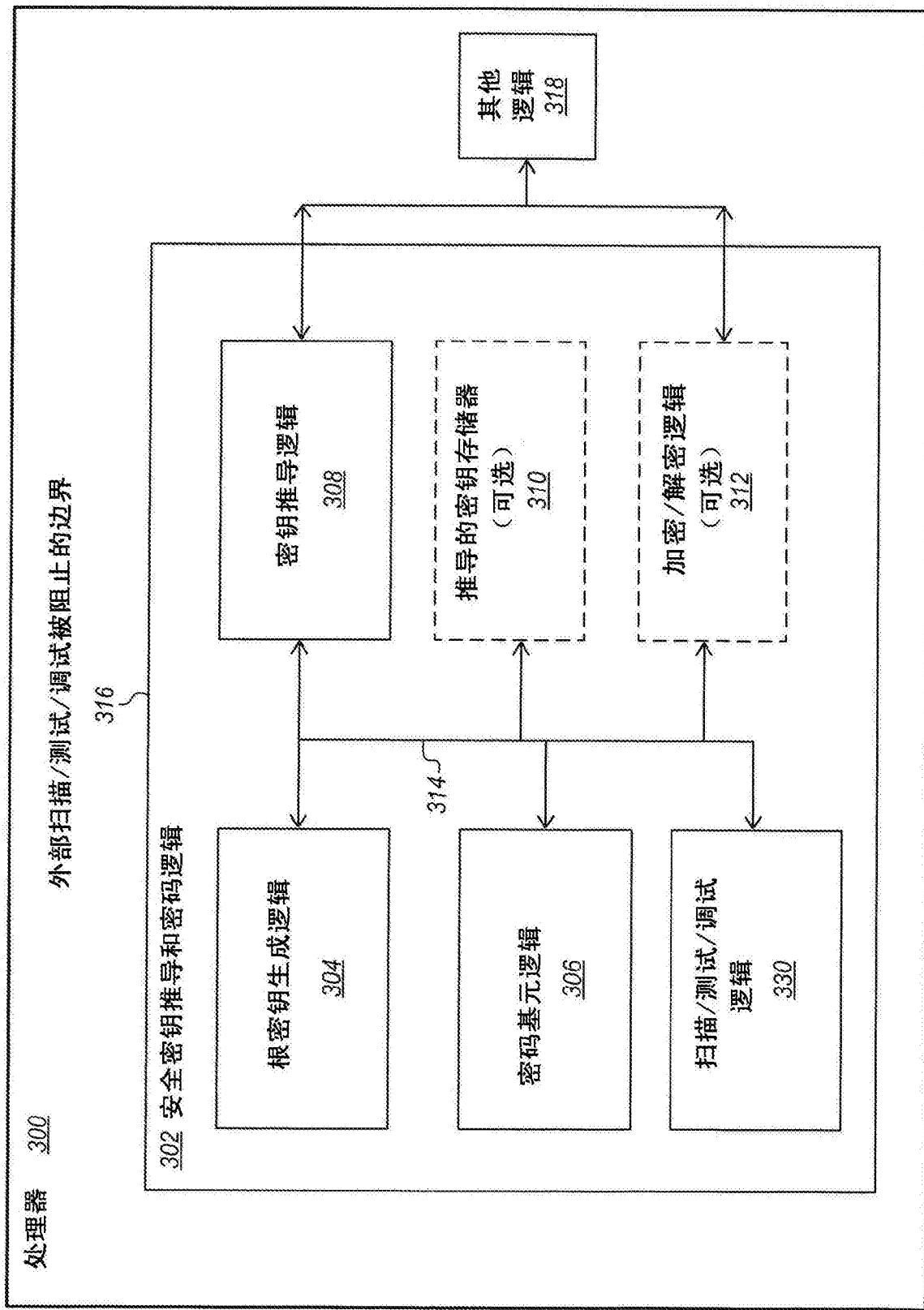


图3

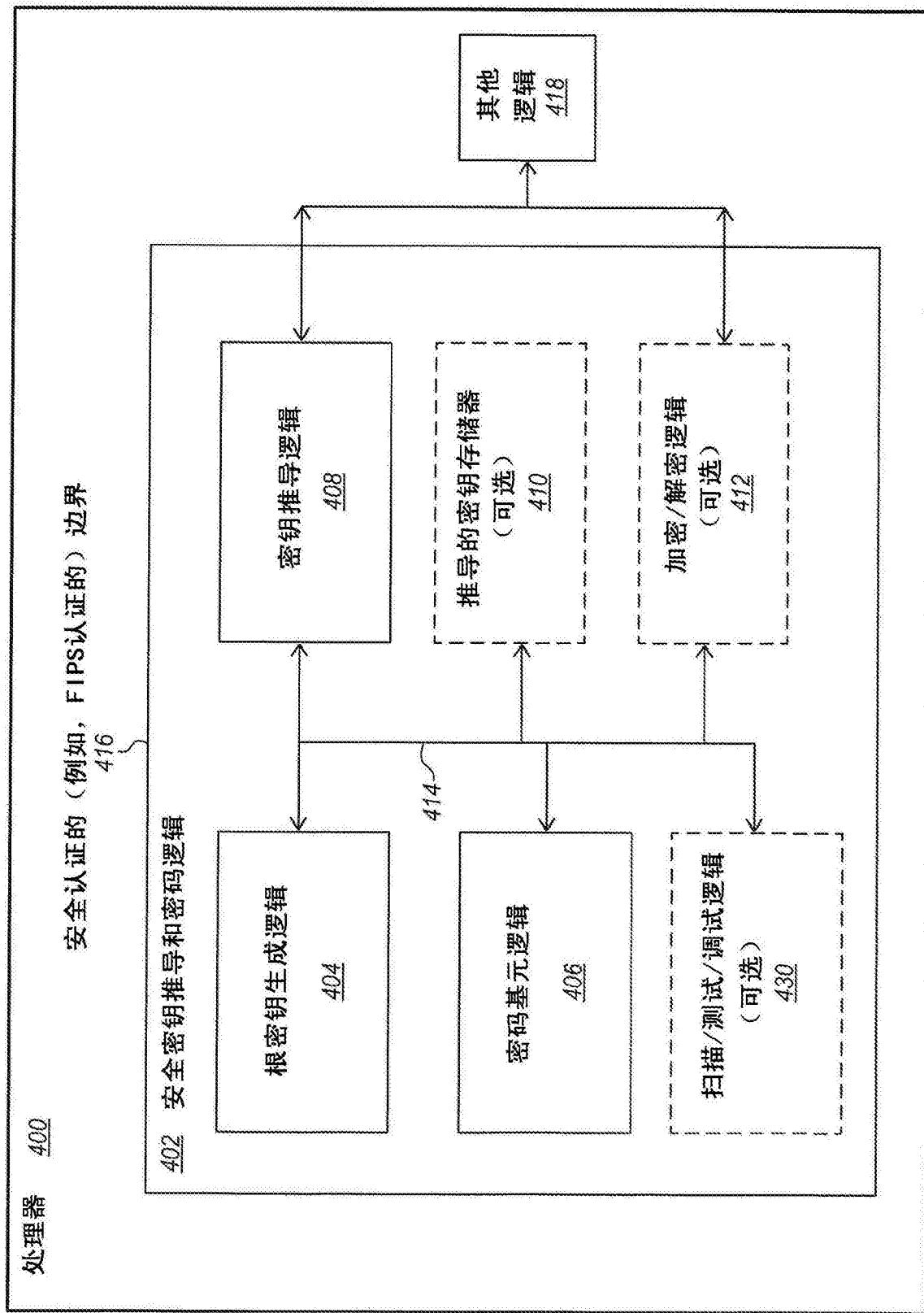


图4

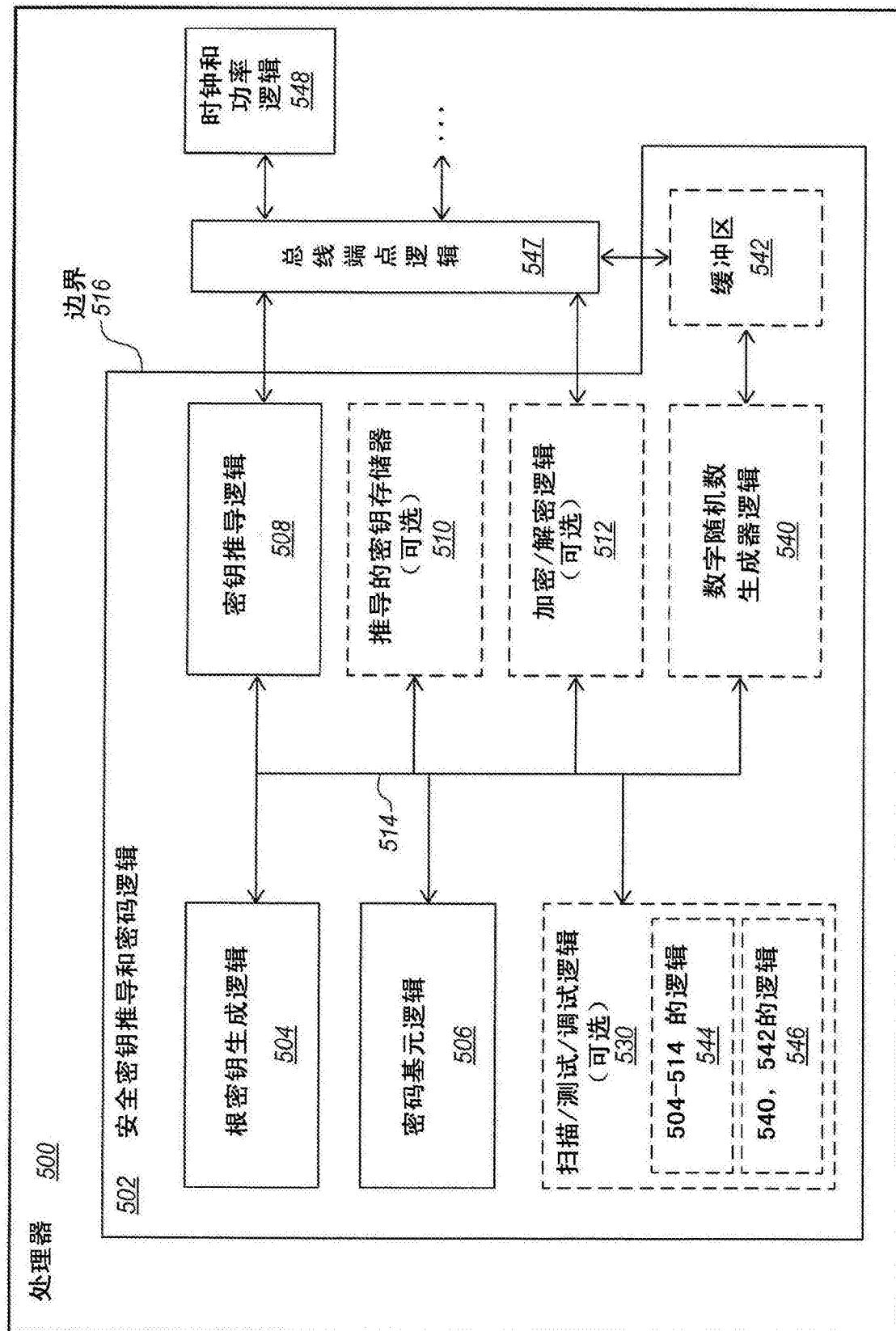


图 5

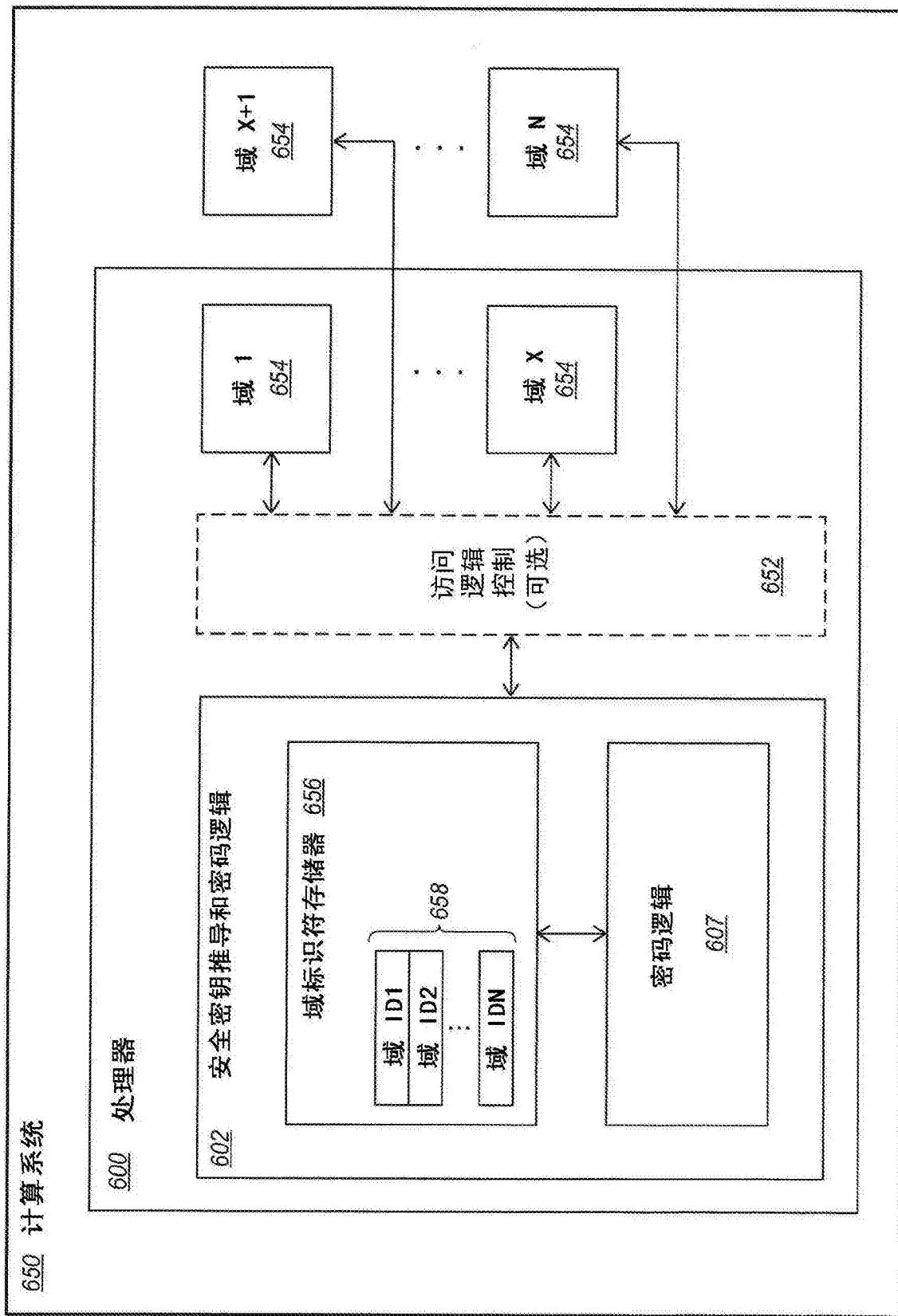


图6

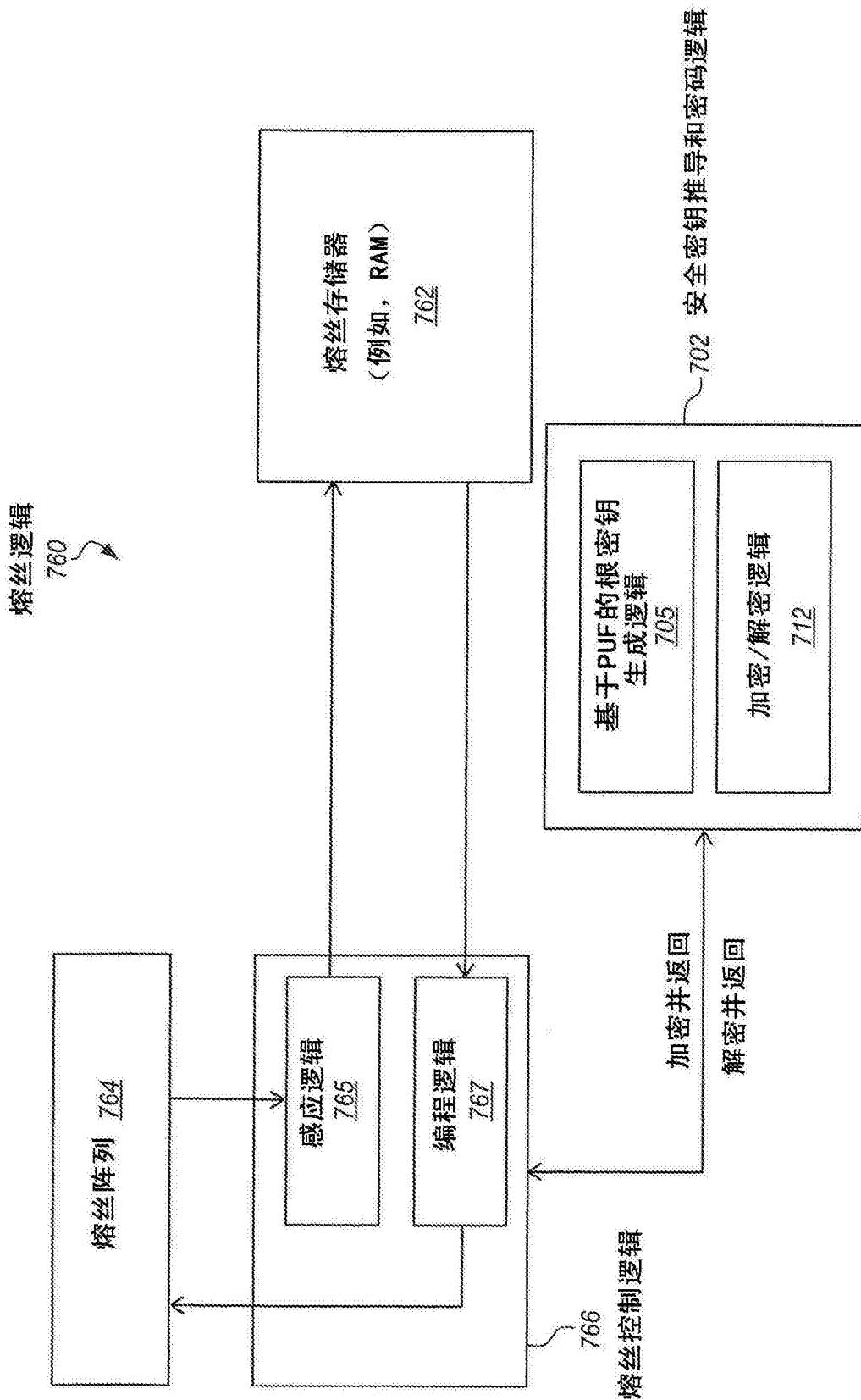


图7

取出 802	长度解码 804	解码 808	分配 808	重命名 810	调度 812	寄存器 读取/ 存储器读取 814	执行级 818	写回/ 存储器写入 818	异常 处理 822	提交 824
-----------	-------------	-----------	-----------	------------	-----------	----------------------------	------------	---------------------	-----------------	-----------

图 8A

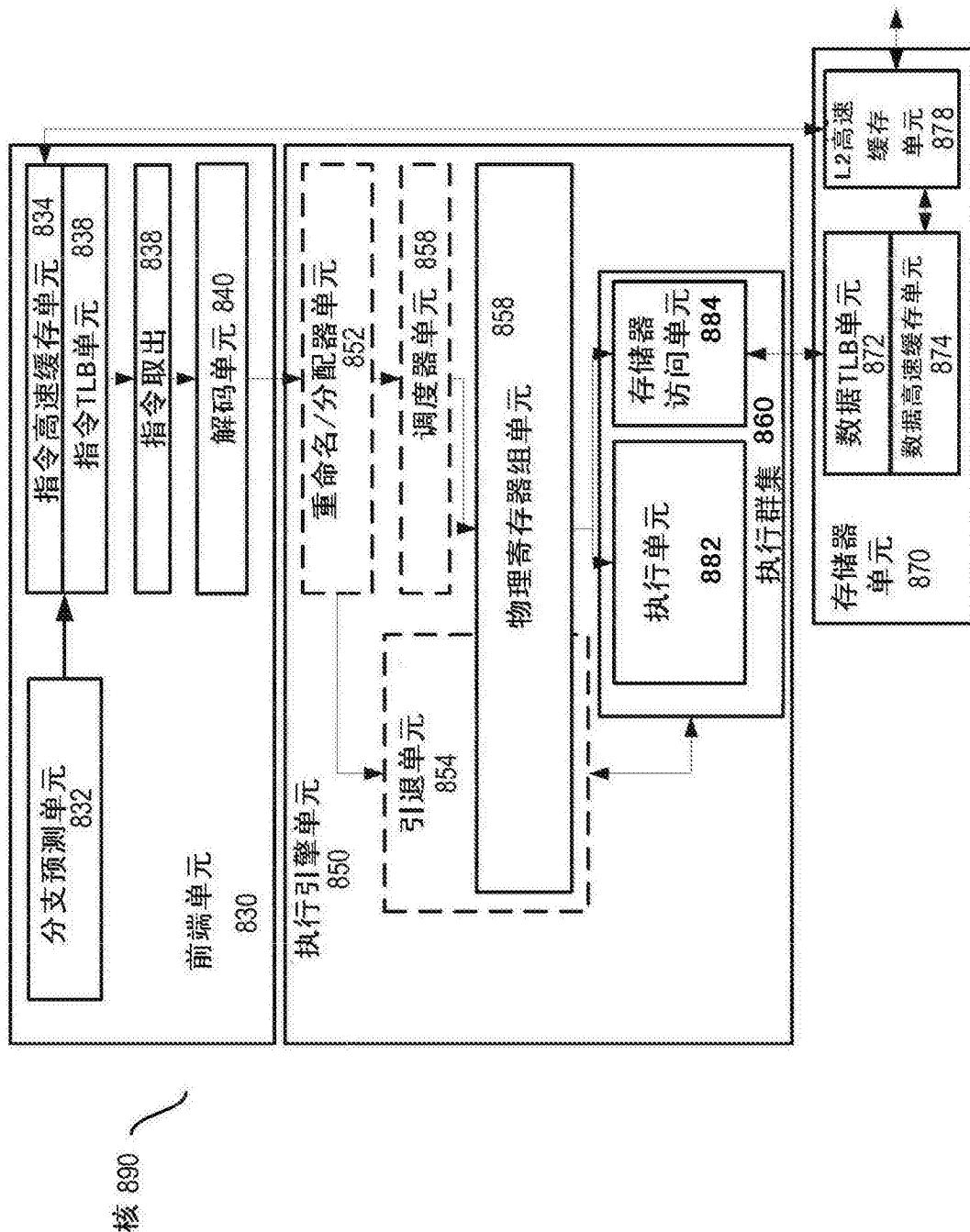


图8B

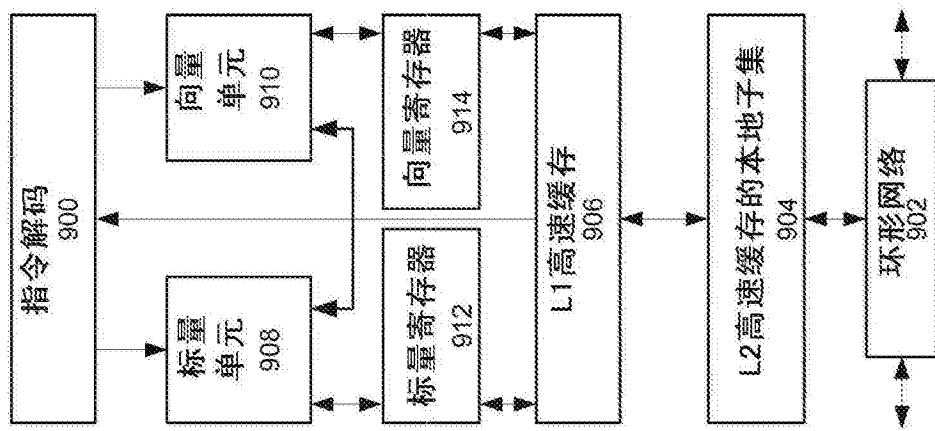


图9A

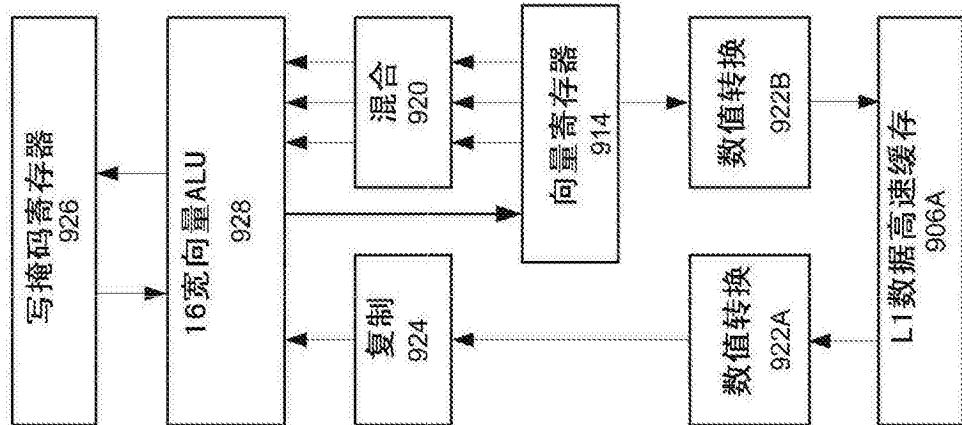


图9B

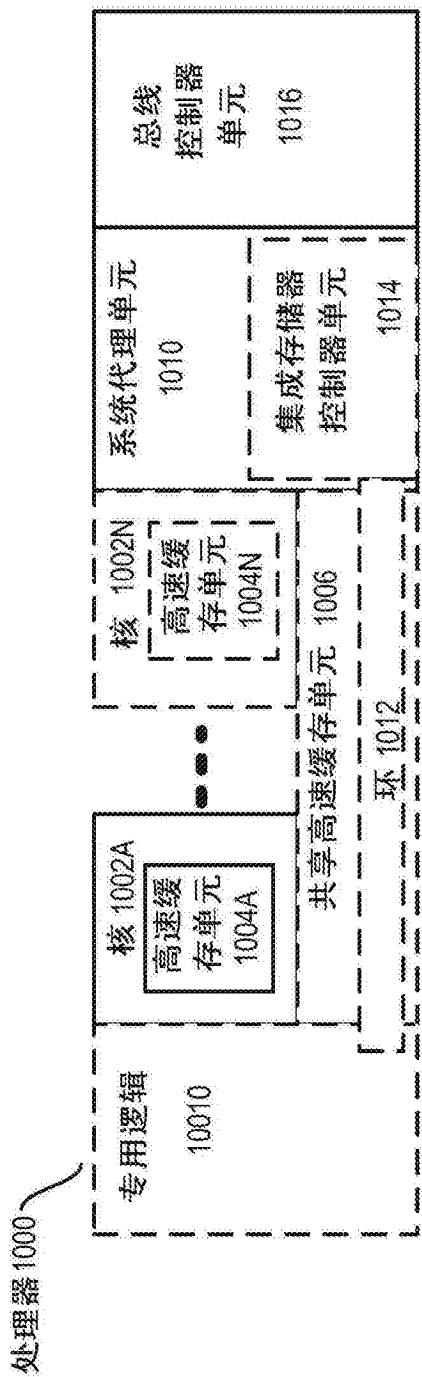


图10

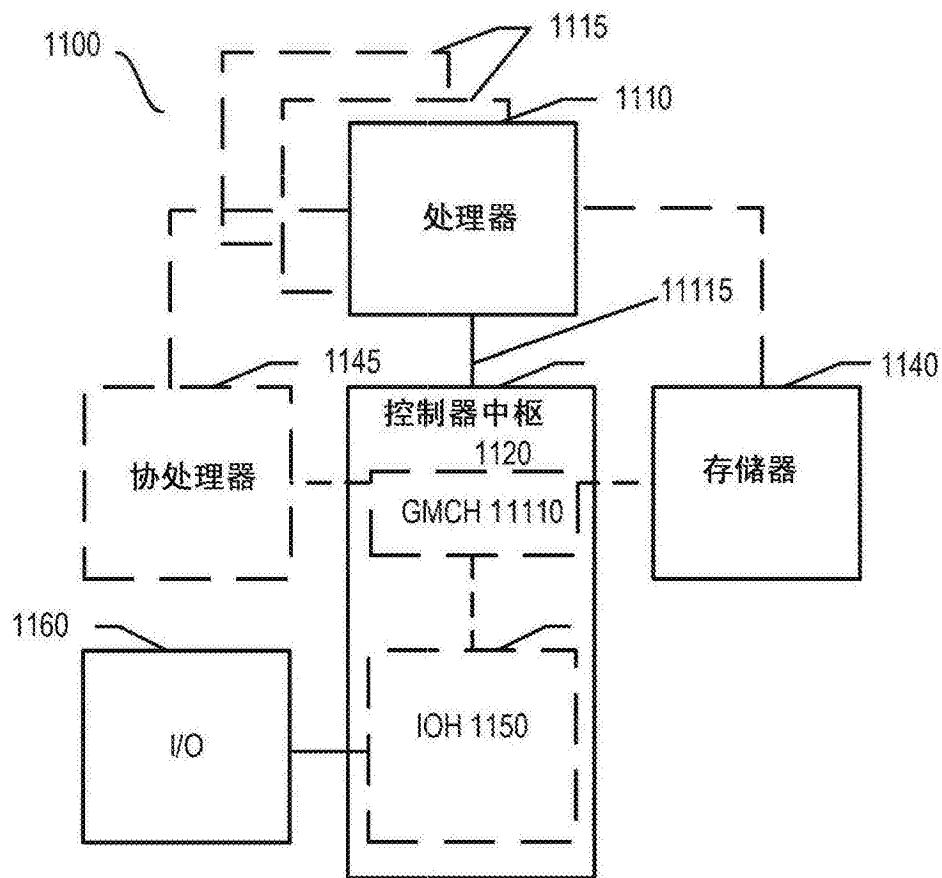


图11

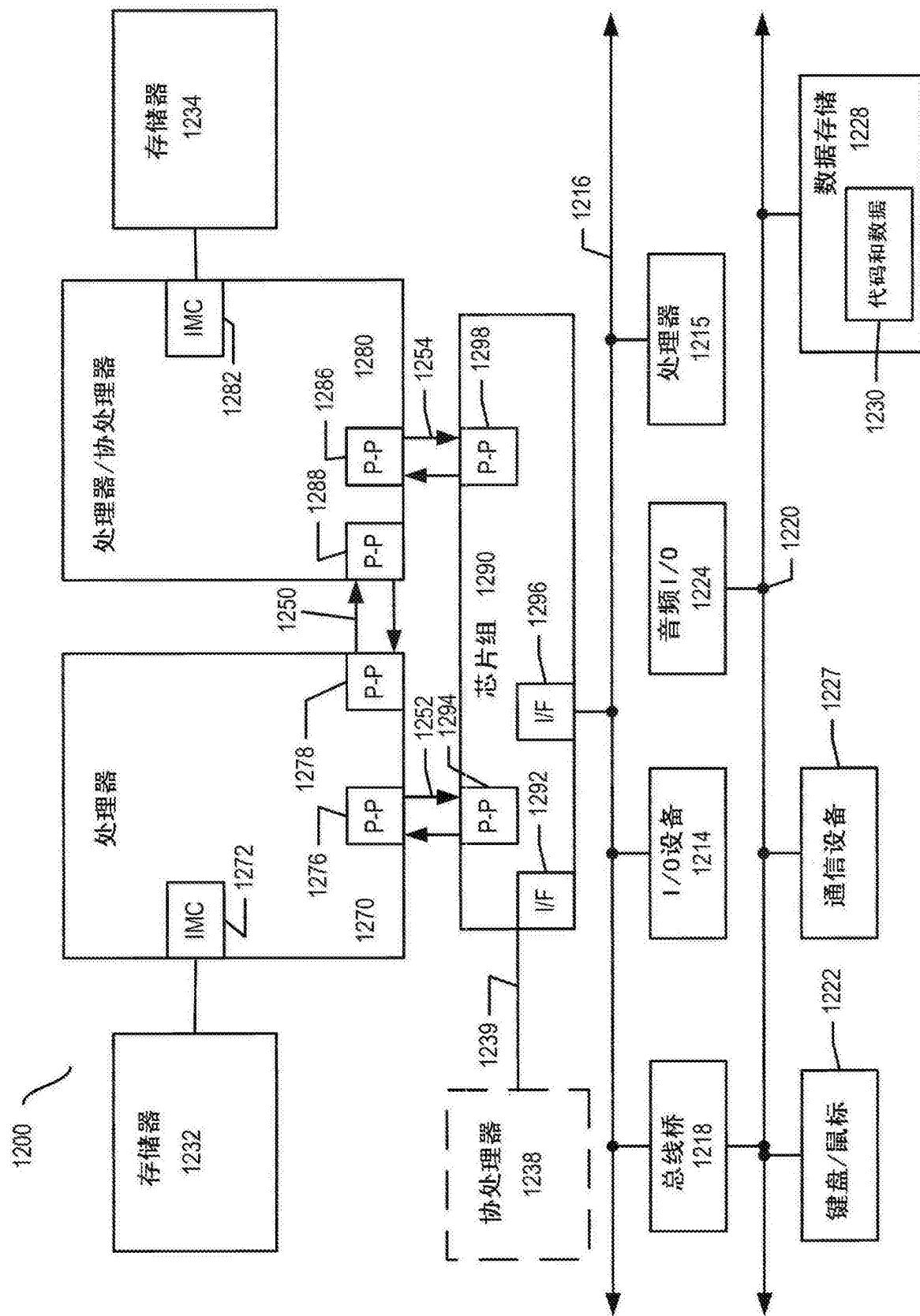


图12

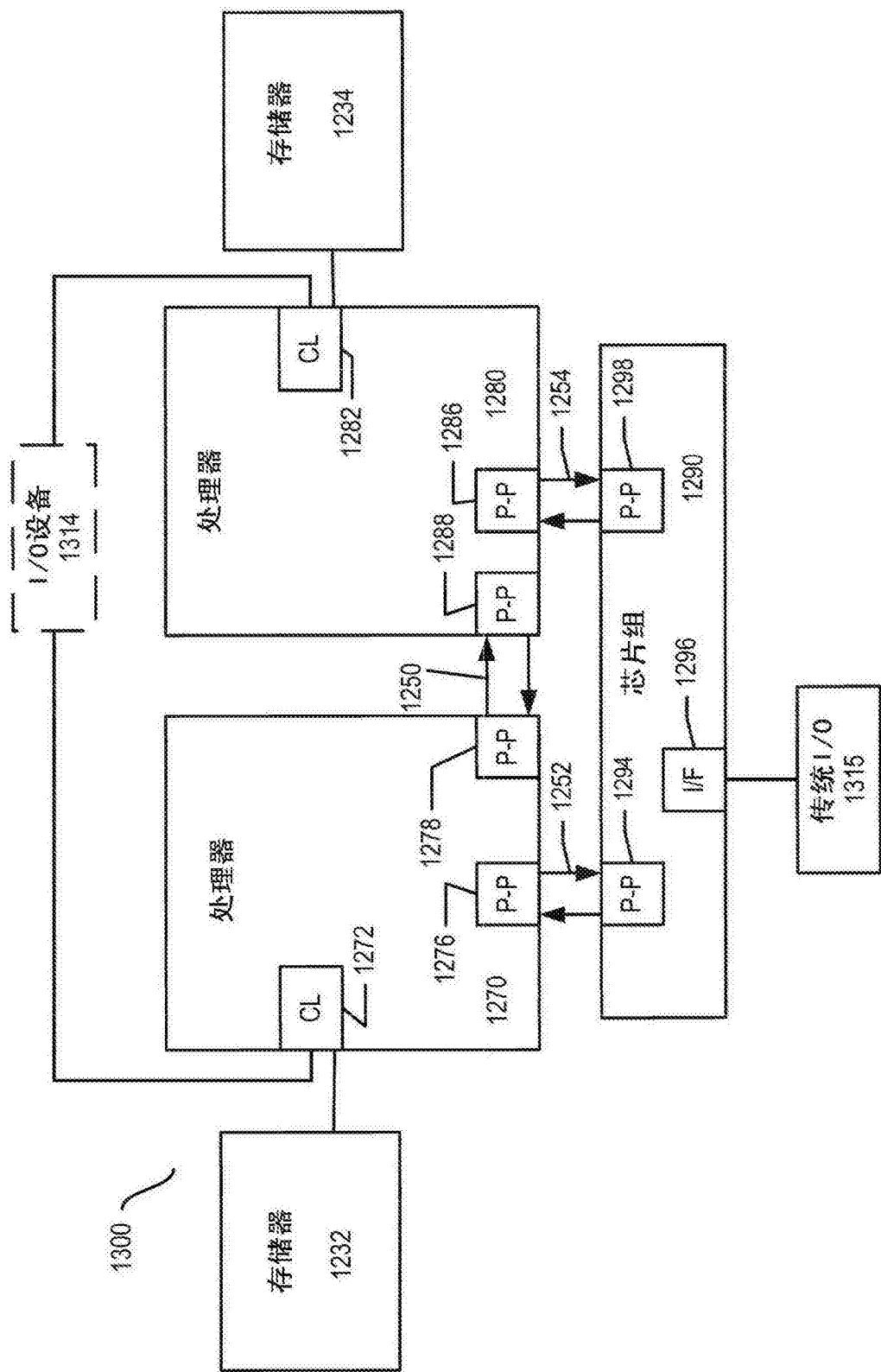


图13

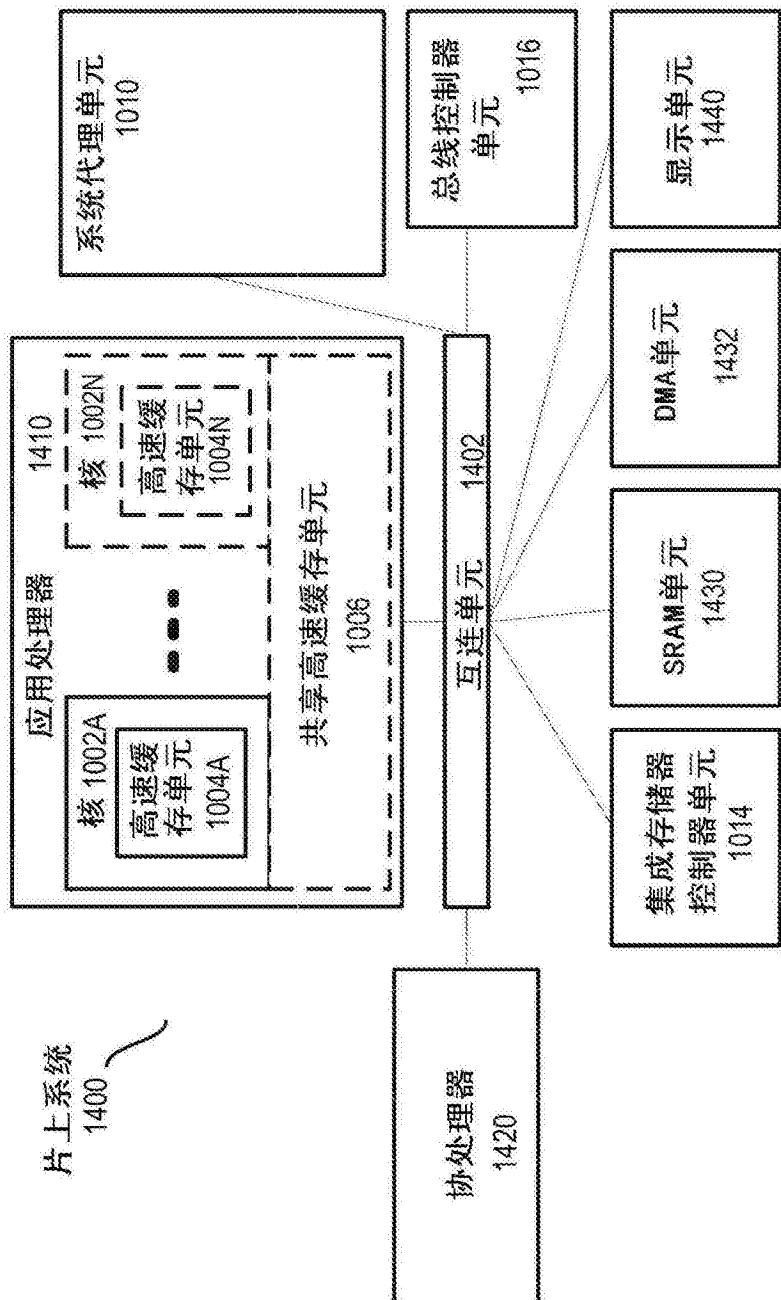


图14

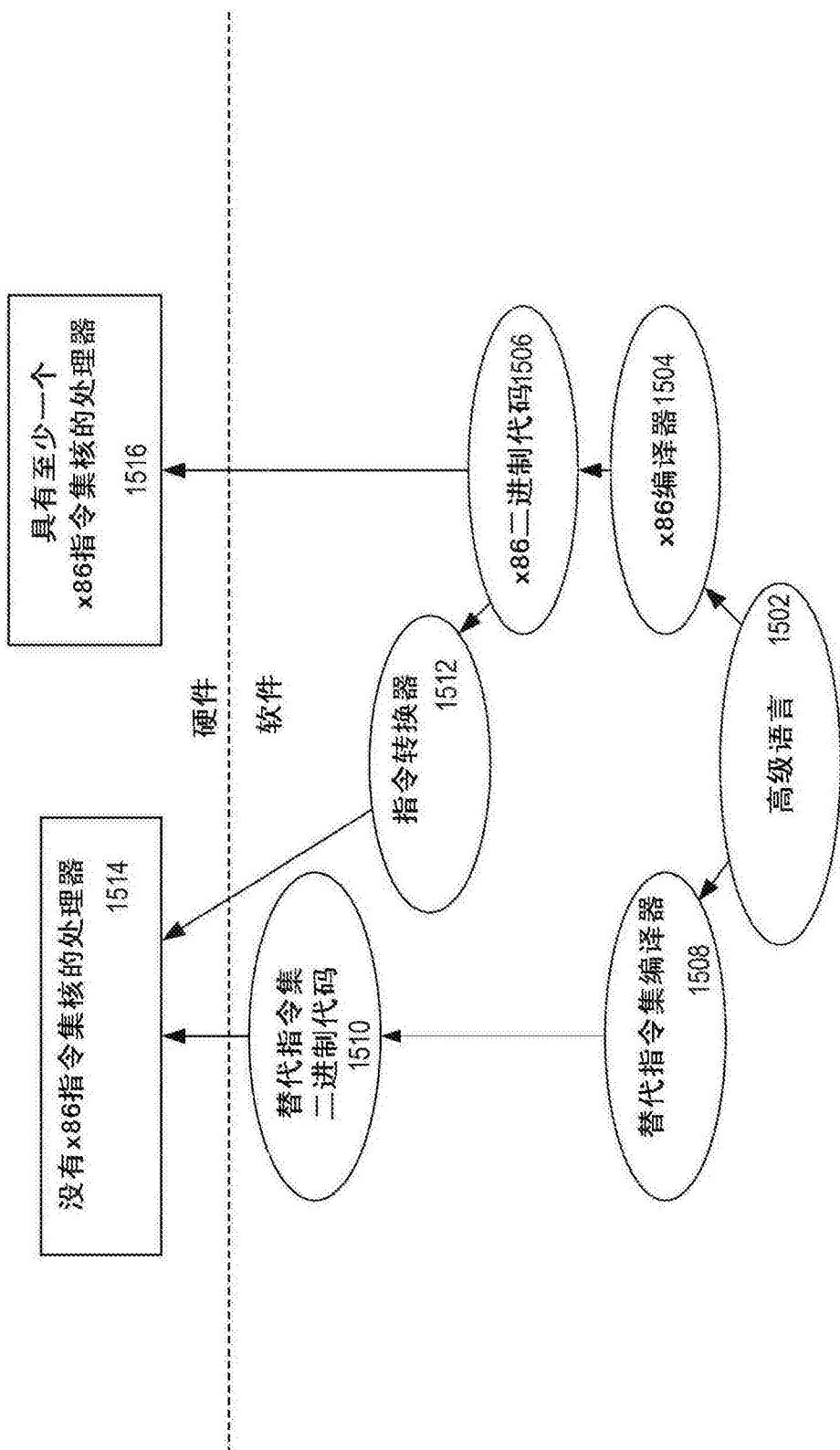


图15