(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103218572 A
(43)申请公布日 2013.07.24

(21)申请号 201310026033.9

(22)申请日 2013.01.23

(30)优先权数据
13/355,806 2012.01.23 US

(71)申请人 国际商业机器公司
地址 美国纽约阿芒克

(72)发明人 A.J. 穆夫 P.E. 沙特 R.A. 希勒
M.R. 塔布斯

(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 邸万奎

(51)Int.Cl.
G06F 21/62(2013.01)
G06F 3/06(2006.01)
G06F 12/10(2006.01)

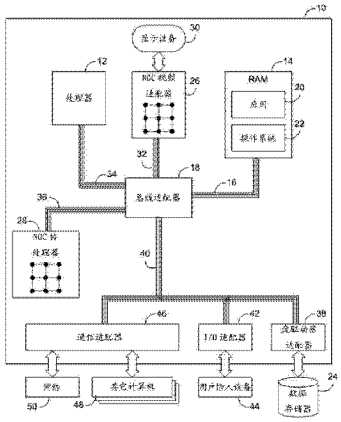
权利要求书3页 说明书19页 附图11页

(54)发明名称

用于在数据处理系统中访问数据的方法和设备

(57)摘要

本发明提供一种方法和电路布置，其基于存储在诸如有效至真实转换(ERAT)或旁路转换缓冲(TLB)的存储地址转换数据结构中的关于加密和/或压缩的页面属性，选择性地将数据流化到加密或压缩引擎。例如可以关于对于存储页面中的数据的存储器访问请求而访问存储地址转换数据结构，使得可以将数据结构中与存储页面相关的属性与处理存储器访问请求相联合地用于控制数据是否被加密/解密和/或压缩/解压缩。



CN 103218572 A

1. 一种在数据处理系统中访问数据的方法，所述方法包括：

响应于由处理核中的线程发起的存储器访问请求，访问存储地址转换数据结构，以对于所述存储器访问请求执行存储地址转换；

访问所述存储地址转换数据结构中的关于加密的页面属性，以确定与所述存储器访问请求相关的存储页面是否被加密；以及

基于与所述存储器访问请求相关的存储页面是否被确定为被加密，而选择性地将所述存储页面中的安全数据流化通过加密引擎，以对所述安全数据执行加密操作。

2. 如权利要求 1 所述的方法，其中所述存储地址转换数据结构包括多个页表条目，每个页表条目包括与相关于所述页表条目的存储页面相关的真实地址以及与这样的存储页面相关的关于加密的页面属性。

3. 如权利要求 1 所述的方法，还包括：响应于由第二线程发起的第二存储器访问请求，通过将与所述第二线程相关的处理标识符与所述存储地址转换数据结构中的访问控制数据进行比较、并且响应于所述比较拒绝所述第二线程对所述存储页面的访问，而限制所述第二线程对所述存储页面的访问。

4. 如权利要求 3 所述的方法，其中所述存储器访问请求包括与由所述处理核执行的第一处理相关的第一存储器访问请求，所述方法还包括：响应于与未被授权访问所述存储页面的第二处理相关的第二存储器访问请求，断言软件异常。

5. 如权利要求 4 所述的方法，还包括：响应于所述第二存储器访问请求，确定所述存储页面是否被加密，并且如果是，则当断言所述软件异常时指示所述页面被加密。

6. 如权利要求 3 所述的方法，其中所述存储器访问请求包括与由所述处理核执行的第一处理相关的第一存储器访问请求，所述方法还包括：响应于与未被授权访问所述存储页面的第二处理相关的第二存储器访问请求，确定所述存储页面是否被加密，并且如果是，则执行系统关闭。

7. 如权利要求 1 所述的方法，其中所述关于加密的页面属性包括指示所述存储页面是否被加密的加密属性。

8. 如权利要求 1 所述的方法，其中所述关于加密的页面属性包括级别属性，其识别多级存储器架构中所述存储页面被加密的至少一个级别。

9. 如权利要求 8 所述的方法，其中选择性地将所述存储页面中的安全数据流化通过所述加密引擎基于所述级别属性是否指示所述存储页面在所述存储器访问请求的源或目标级别中被加密。

10. 如权利要求 1 所述的方法，其中从包括加密所述安全数据和解密所述安全数据的组中选择所述加密操作。

11. 一种在数据处理系统中访问数据的方法，所述方法包括：

响应于由处理核中的线程发起的存储器访问请求，访问存储地址转换数据结构，以对于所述存储器访问请求执行存储地址转换；

访问所述存储地址转换数据结构中的关于压缩的页面属性，以确定与所述存储器访问请求相关的存储页面是否被压缩；以及

基于与所述存储器访问请求相关的存储页面是否被确定为被压缩，而选择性地将所述存储页面中的数据流化通过压缩引擎，以对所述数据执行压缩操作。

12. 如权利要求 11 所述的方法，其中所述存储地址转换数据结构包括多个页表条目，每个页表条目包括与相关于所述页表条目的存储页面相关的真实地址以及与这样的存储页面相关的关于压缩的页面属性。

13. 如权利要求 11 所述的方法，其中所述关于压缩的页面属性包括级别属性，其识别多级存储器架构中所述存储页面被压缩的至少一个级别，并且其中选择性地将所述存储页面中的数据流化通过所述压缩引擎基于所述级别属性是否指示所述存储页面在所述存储器访问请求的源或目标级别中被压缩。

14. 如权利要求 11 所述的方法，其中从包括压缩所述数据和解压所述数据的组中选择所述压缩操作。

15. 一种电路布置，包括：

多核处理器，其包括多个处理核；以及

存储地址转换数据结构，其布置在所述多个处理核之中的第一处理核中，所述存储地址转换数据结构被配置为存储用于存储页面的地址转换数据，其中所述存储地址转换数据结构还被配置为存储用于所述存储页面的关于加密的页面属性；

其中所述多核处理器被配置为响应于由所述第一处理核中的线程发起的、并与所述存储页面相关的存储器访问请求，基于与所述存储器访问请求相关的所述存储页面是否被加密而选择性地将来自所述存储页面的安全数据流化通过加密引擎，以对所述安全数据执行加密操作。

16. 如权利要求 15 所述的电路布置，其中所述存储地址转换数据结构包括多个页表条目，每个页表条目包括与相关于所述页表条目的存储页面相关的真实地址以及与这样的存储页面相关的关于加密的页面属性。

17. 如权利要求 15 所述的电路布置，其中所述存储器访问请求包括与由所述处理核执行的第一处理相关的第一存储器访问请求，并且其中所述第一处理核被配置为：响应于与未被授权访问所述存储页面的第二处理相关的第二存储器访问请求，确定所述存储页面是否被加密，并且如果是，则当断言软件异常时指示所述页面被加密。

18. 如权利要求 15 所述的电路布置，其中所述存储器访问请求包括与由所述处理核执行的第一处理相关的第一存储器访问请求，并且其中所述第一处理核被配置为：响应于与未被授权访问所述存储页面的第二处理相关的第二存储器访问请求，确定所述存储页面是否被加密，并且如果是，则执行系统关闭。

19. 如权利要求 15 所述的电路布置，其中所述关于加密的页面属性包括级别属性，其识别多级存储器架构中所述存储页面被加密的至少一个级别，并且其中所述多核处理器被配置为基于所述级别属性是否指示所述存储页面在所述存储器访问请求的源或目标级别中被加密，而选择性地将所述存储页面中的安全数据流化通过所述加密引擎。

20. 如权利要求 15 所述的电路布置，其中从包括加密所述安全数据和解密所述安全数据的组中选择所述加密操作。

21. 一种电路布置，包括：

多核处理器，其包括多个处理核；以及

存储地址转换数据结构，其布置在所述多个处理核之中的第一处理核中，所述存储地址转换数据结构被配置为存储用于存储页面的地址转换数据，其中所述存储地址转换数据

结构还被配置为存储用于所述存储页面的关于压缩的页面属性；

其中所述多核处理器被配置为：响应于由所述第一处理核中的线程发起的、并与所述存储页面相关的存储器访问请求，基于与所述存储器访问请求相关的所述存储页面是否被压缩而选择性地将来自所述存储页面的数据流化通过压缩引擎，以对所述数据执行压缩操作。

22. 如权利要求 21 所述的电路布置，其中所述存储地址转换数据结构包括多个页表条目，每个页表条目包括与相关于所述页表条目的存储页面相关的真实地址以及与这样的存储页面相关的关于压缩的页面属性。

23. 如权利要求 21 所述的电路布置，其中所述关于压缩的页面属性包括级别属性，其识别多级存储器架构中所述存储页面被压缩的至少一个级别，并且其中所述多核处理器被配置为基于所述级别属性是否指示所述存储页面在所述存储器访问请求的源或目标级别中被压缩，而选择性地将所述存储页面中的数据流化通过所述压缩引擎。

24. 如权利要求 21 所述的电路布置，其中从包括压缩所述数据和解压所述数据的组中选择所述压缩操作。

用于在数据处理系统中访问数据的方法和设备

技术领域

[0001] 本发明一般地涉及数据处理，并且特别涉及通过其中的数据处理系统和处理器存储和 / 或使用的数据的加密和 / 或压缩。

背景技术

[0002] 保护由数据处理系统的处理器存储或使用的安全数据在许多数据处理应用中是至关重要的。通常将加密算法施加于安全数据，以使其在没有解密算法应用的情况下呈现为不可理解的，并且通常以加密格式将安全数据存储在大容量存储器和其它非易失性存储器介质中，在安全数据可以被数据处理系统中的处理器读出和 / 或操控之前需要执行解密。然而，在许多实例中，加密的安全数据的解密导致安全数据以未加密形式存储在数据处理系统中的各种类型的易失性存储器中，例如，在主存储器内，或在用于加速访问频繁使用的数据的各种级别的高速缓存（cache）存储器内。然而，在任何时间将数据以不安全形式存储在数据可能经受未授权访问的数据处理系统的任何存储器中，都有可能损害数据的机密性。

[0003] 然而，加密和解密数据通常需要一定量的处理开销，并且因而甚至在正在处理安全数据的应用中，也期望在数据处理系统中保持其它非安全数据，这样使得该其它数据的处理不经受与加密和解密相关的相同处理开销。

[0004] 此外，由于就时钟速度的增加而言，半导体技术继续愈来愈接近实用极限，所以架构师日益专注于处理器架构的并行度（parallelism）以获得性能改进。在芯片级，通常将多个处理核布置在同一芯片上，其以与独立处理器芯片、或在某种程度上与完全独立的计算机非常相同的方式运作。此外，即使在核内，也通过专门处理某些类型的操作的多个执行单元的使用而采用并行度。在许多实例中也采用流水线技术（pipelining），使得可以采取多个时钟周期来执行的某些操作被拆分为阶段（stage），使得其它操作能够在较早的操作完成之前就开始。也采用多线程（multithreading）以使多个指令流能够被并行地处理，使得在任何给定的时钟周期中能够执行更多整体工作。

[0005] 由于这些增加的并行度，在数据处理系统中维护安全数据的挑战比在以前的非并行数据处理系统中更显著。例如，在仅包括具有单线程的单处理器的数据处理系统中，可以将安全数据以加密形式存储在处理器外部，并且一旦数据被加载到处理器中就在必要时由该单线程解密。然而，当在同一处理器芯片上布置附加线程甚至附加处理核时，可能需要将对安全数据的访问限制到仅芯片上的某些线程或处理核。因而，例如，如果多个线程或处理核共享公共高速缓存存储器，则在该高速缓存存储器中以未加密形式存储任何安全数据可能存在未授权方可以经由除被授权访问该安全数据的线程或处理核之外的线程或处理核而获得对该数据的访问的风险。此外，由于现代片上系统（SOC）处理器设计发展为在处理器芯片上有上百个处理核，所以针对来自甚至同一处理器芯片上的其它处理保护未加密数据也变得日益重要。

[0006] 传统上，已经通过处理器上执行的软件处理加密和解密。然而，因为加密和解密是

处理器密集型任务，所以已经开发了专用的基于硬件的加密引擎，从而以比通常可以由软件实现的方式更快和更有效的方式执行安全数据的加密 / 解密，由此减少与这种操作相关的开销。传统加密引擎通常被布置在处理核外部，例如在处理核和存储控制器之间，或者连接到任何处理核外部的存储器总线。此外，为了帮助确定哪些数据被加密和未被加密，可以将安全数据存储在特定存储地址区域中，使得可以使用过滤来控制加密引擎仅加密 / 解密存储在所识别的存储地址范围内的数据。

[0007] 然而，这种架构可以导致高速缓存（cache）中存在未加密数据并且该未加密数据可以被芯片中的其它线程和 / 或处理核访问。此外，通常需要位于处理器芯片外部的存储控制器，以建立和管理其中存储安全数据的存储地址的范围，从而可以对于涉及安全数据的存储事务选择性地激活加密，这会导致安全数据的低效吞吐量。

[0008] 关于数据压缩也存在类似的挑战。可以将数据压缩用于减少存储数据所需要的存储量；然而，压缩的数据必须在由处理核或线程使用之前被解压缩。数据的压缩和解压缩涉及处理开销，并且因而在软件中实施这种功能通常伴随性能损失。此外，也已经开发了专用压缩引擎来减少与压缩和解压缩数据相关的处理开销；然而，这种引擎通常被布置在处理核的外部（例如，在存储控制器内），并且因此可能需要将压缩数据以解压缩格式存储在不同级别的高速缓存存储器中，其限制了在这种高速缓存存储器中用于存储其它数据的空间量，因而降低了存储系统的性能。另外，由于具有加密数据，可能需要存储控制器建立和管理其中存储压缩数据的存储地址范围，使得可以对于涉及压缩数据的存储事务选择性地激活压缩引擎，从而导致低效的压缩数据吞吐量。

[0009] 因此，对于最小化与访问和管理数据处理系统中的加密和 / 或压缩数据相关的性能开销、以及提供对多线程和 / 或多核处理器芯片以及包含其的数据处理系统内的加密数据的进一步保护的方式，在本领域中继续存在强烈的需要。

发明内容

[0010] 本发明通过提供一种方法和电路布置来解决与在先技术相关的这些和其它问题，所述方法和电路布置基于存储在诸如有效至真实转换（EffectiveTo Real Translation，ERAT）或旁路转换缓冲（Translation Lookaside Buffer，TLB）的存储地址转换数据结构中的关于加密和 / 或压缩的页面属性，选择性地将数据流化（stream）到加密或压缩引擎。例如，可以关于对于存储页面中的数据的存储器访问请求，而访问存储地址转换数据结构，使得可以与处理存储器访问请求相关联地、将与数据结构中的存储页面相关的属性用于控制数据是否被加密 / 解密和 / 或压缩 / 解压缩。

[0011] 因此，符合本发明的一个方面，响应于由处理核中的线程发起的存储器访问请求而访问存储地址转换数据结构，以对于该存储器访问请求执行存储地址转换。此外，访问该存储地址转换数据结构中的关于加密的页面属性，以确定与该存储器访问请求相关的该存储页面是否被加密。接着基于与该存储器访问请求相关的该存储页面是否被确定为被加密而选择性地将该存储页面中的安全数据流化通过加密引擎，以对该安全数据执行加密操作。

[0012] 符合本发明的另一个方面，响应于由处理核中的线程发起的存储器访问请求，访问存储地址转换数据结构，以对于该存储器访问请求执行存储地址转换。此外，访问该存储

地址转换数据结构中的关于压缩的页面属性，以确定与该存储器访问请求相关的该存储页面是否被压缩。接着基于与该存储器访问请求相关的该存储页面是否被确定为被压缩而选择性地将该存储页面中的数据流化通过压缩引擎，以对该数据执行压缩操作。

[0013] 刻画本发明的特征的这些以及其它优势和特性在所附的权利要求书中阐述，并权利要求书形成本发明的另一部分。然而，为了更好地理解本发明以及通过其的使用而实现的优点和目标，应参考其中描述了本发明的示例性实施例的附图和附随描述。

附图说明

[0014] 图 1 是包括在符合本发明实施例的数据处理中有用的示例性计算机的示例性自动计算机器的框图。

[0015] 图 2 是在图 1 的计算机中实施的示例性 NOC 的框图。

[0016] 图 3 是更详细地示出来自图 2 的 NOC 的节点的示例性实施方式的框图。

[0017] 图 4 是示出来自图 2 的 NOC 的 IP 块的示例性实施方式的框图。

[0018] 图 5 是包含符合本发明的基于存储地址转换的数据加密 / 压缩的示例性数据处理系统的框图。

[0019] 图 6 是用于图 5 中引用的 ERAT 的示例性 ERAT 条目格式的框图。

[0020] 图 7 是示出使用符合本发明的支持基于存储地址转换的加密 / 压缩的数据处理系统的示例性存储器访问的框图。

[0021] 图 8 是示出用于访问图 7 的数据处理系统中的数据的操作的示例性序列的流程图。

[0022] 图 9 是示出用于执行图 7 的数据处理系统中的读出总线事务的操作的示例性序列的流程图。

[0023] 图 10 是示出用于执行图 7 的数据处理系统中的写入总线事务的操作的示例性序列的流程图。

[0024] 图 11 是示出用于执行图 7 的数据处理系统中的读出总线事务连同级别选择性加密 / 压缩的操作的交替序列的流程图。

[0025] 图 12 是示出用于执行图 7 的数据处理系统中的写入总线事务连同级别选择性加密 / 压缩的操作的交替顺序的流程图。

[0026] 图 13 是示出使用包含符合本发明的集成加密引擎的另一个数据处理系统的示例性存储器访问的框图。

[0027] 图 14 是示出包含符合本发明的集成加密引擎、并且还包含用于在存储安全数据中使用的独立安全高速缓存的另一个数据处理系统的框图。

具体实施方式

[0028] 符合本发明的实施例基于存储在诸如有效至真实转换（ERAT）或旁路转换缓冲（TLB）的存储地址转换数据结构中的关于加密和 / 或压缩的页面属性，选择性地将数据流化到加密或压缩引擎。例如可以关于对于存储页面中的数据的存储器访问请求一起访问存储地址转换数据结构，使得可以与处理存储器访问请求相关联地，使用数据结构中的与存储页面相关的属性来控制是否加密 / 解密和 / 或压缩 / 解压缩数据。

[0029] 此外，在符合本发明的某些实施例中，可以在多核处理器的处理核内采用集成加密引擎，以关于访问安全数据的存储器访问请求而执行加密操作，即，安全数据的加密和解密。当与诸如有效至真实转换（ERAT）或旁路转换缓冲（TLB）的存储地址转换数据结构（其已被增加有关于加密的页面属性以指示数据结构中识别的存储页面是否被加密）相结合时，可以基于用于与存储器访问请求相关的存储页面的关于加密的页面属性选择性地将与处理核中的存储器访问请求相关的安全数据流化到集成加密引擎。此外，在某些实施例中，可以将集成加密引擎连接到处理核中的 L1 高速缓存，从其存储安全数据的角度，L1 高速缓存是有效地安全的。可以将 L1 高速缓存配置为存储安全和非安全数据二者，或者替代地，L1 高速缓存可以专用于安全数据，并且可以将第二非安全 L1 高速缓存用于缓存非安全数据。在二者中的任一实例中，可以仅在处理核中解密安全数据，并且在安全数据位于处理核外部的任何时间加密安全数据，因而提供了超过传统设计的增强的安全性。

[0030] 本领域普通技术人员将认识到其它变化和修改。因而，本发明不限定于这里讨论的具体的实施方式。

[0031] 硬件和软件环境

[0032] 现在转到附图，其中遍及几个视图的相似数字表示相似部件。图 1 示出包括在符合本发明实施例的数据处理中有用的示例性计算机 10 的示例性自动计算机器。图 1 的计算机 10 包括至少一个计算机处理器 12 或“CPU”、以及随机存取存储器 14（“RAM”），其通过高速存储器总线 16 和总线适配器 18 连接到处理器 12 以及计算机 10 的其它组件。

[0033] RAM 14 中存储的是应用程序 20，其为用户层计算机程序指令模块，用于执行特定数据处理任务，诸如例如文字处理、电子数据表、数据库操作、视频游戏、股票市场模拟、原子量子过程模拟、或者其它用户层应用。RAM 14 中还存储操作系统 22。连同本发明实施例一起有用的操作系统包括 UNIX™、Linux™、Microsoft Windows XP™、AIX™、IBM 的 i5/OS™、以及本领域技术人员将会使用的其它操作系统。图 1 的示例中的操作系统 22 和应用程序 20 在 RAM 14 中示出，但是这种软件的许多组件通常也被存储在非易失性存储器中，例如在盘驱动器 24 上。

[0034] 如下面将变得更显而易见的，可以在片上网络（NOC）集成电路器件或芯片内实施符合本发明的实施例，并且因而，示出了包括两个示例性 NOC（视频适配器 26 和协处理器 28）的计算机 10。可以被替代地称为图形适配器的 NOC 视频适配器 26 是专门设计用于向诸如显示屏幕或计算机显示器的显示设备 30 输出图形的 I/O 适配器的示例。将 NOC 视频适配器 26 通过高速视频总线 32、总线适配器 18、和前端总线 34（其也是高速总线）连接到处理器 12。将 NOC 协处理器 28 通过总线适配器 18、以及前端总线 34 和 36（其也是高速总线）连接到处理器 12。图 1 的 NOC 协处理器可以被优化为例如在主处理器 12 的命令下加速特定数据处理任务。

[0035] 图 1 的示例性 NOC 视频适配器 26 和 NOC 协处理器 28 各自包括 NOC，其包括集成处理器（“IP”）块、路由器、存储器通信控制器、以及网络接口控制器，下面将连同图 2-3 更详细地讨论其细节。针对使用并行处理并且还需要快速随机访问共享存储器的程序分别优化 NOC 视频适配器和 NOC 协处理器。然而，受益于本即时公开的本领域技术人员将认识到，可以在除 NOC 器件和器件架构之外的器件和器件架构中实施本发明。因此本发明不限定于 NOC 器件中的实施方式。

[0036] 图 1 的计算机 10 包括通过扩展总线 40 和总线适配器 18 连接到处理器 12 及计算机 10 的其它组件的盘驱动器适配器 38。盘驱动器适配器 38 以盘驱动器 24 的形式将非易失性数据存储器连接到计算机 10，并且可以例如使用电子集成驱动器（“IDE”）适配器、小型计算机系统接口（“SCSI”）适配器、以及本领域技术人员将会见到的其它适配器实施。也可以将非易失性计算机存储器实施为如本领域技术人员将会见到的光盘驱动器、电可擦除可编程只读存储器（所谓的“EEPROM”或“Flash”存储器）、RAM 驱动器等。

[0037] 计算机 10 也包括一个或多个输入 / 输出（“I/O”）适配器 42，其例如通过用于控制到诸如计算机显示屏幕的显示设备的输出、以及从诸如键盘和鼠标的用户输入设备 44 的用户输入的软件驱动器和计算机硬件来实施面向用户的输入 / 输出。此外，计算机 10 包括用于与其它计算机 48 进行数据通信、以及用于与数据通信网络 50 进行数据通信的通信适配器 46。可以通过 RS-232 连接、通过诸如通用串行总线（“USB”）的外部总线、通过诸如 IP 数据通信网络的数据通信网络、以及本领域技术人员将会使用的其它方式而串行地执行这样的数据通信。通信适配器实施数据通信的硬件层，一台计算机通过其将数据通信直接地或者通过数据通信网络而发送到另一台计算机。适合于在计算机 10 中使用的通信适配器的示例包括用于有线拨号通信的调制解调器、用于有线数据通信网络通信的以太网（IEEE 802.3）适配器、以及用于无线数据通信网络通信的 802.11 适配器。

[0038] 为了进一步说明，图 2 阐述了根据本发明实施例的示例 NOC 102 的功能框图。在“芯片”100 上（即，在集成电路上）实施图 2 中的 NOC。NOC 102 包括集成处理器（“IP”）块 104、路由器 110、存储器通信控制器 106、以及分组为相互连接的节点的网络接口控制器 108。每个 IP 块 104 通过存储器通信控制器 106 和网络接口控制器 108 适配于路由器 110。每个存储器通信控制器控制 IP 块和存储器之间的通信，并且每个网络接口控制器 108 通过路由器 110 控制 IP 块间（inter-IP block）通信。

[0039] 在 NOC 102 中，每个 IP 块表示被用作 NOC 内的数据处理的构成块（building block）的同步或异步逻辑设计的可重复使用的单元。术语“IP 块”有时被展开为“知识产权块”，其有效地将 IP 块指定为由一方拥有的设计（其为一方的知识产权），且其被许可给半导体电路的其他用户或设计者。然而在本发明的范围内，不存在 IP 块受制于任何特定所有权的需要，所以该术语在本说明书中总是被展开为“集成处理器块”。如这里所规定的，IP 块是逻辑、单元（cell）或芯片布局设计的可重复使用的单元，其可以是或不是知识产权的主题。IP 块是可以形成为 ASIC 芯片设计或 FPGA 逻辑设计的逻辑核。

[0040] 通过类比描述 IP 块的一种方法是：IP 块对于 NOC 设计，正如程序库对于计算机编程、或者分离的集成电路组件对于印刷电路板设计。在符合本发明实施例的 NOC 中，可以将 IP 块实施为通用门网络表（generic gate netlists）、实施为完整的专用或通用微处理器、或者本领域技术人员将会使用的其它方式。网络表是 IP 块的逻辑功能的布尔代数（Boolean-algebra）表示（门、标准单元），类似用于高级程序应用的汇编代码（assembly-code）列表。NOC 也可以被实施为例如在诸如 Verilog 或 VHDL 的硬件描述语言中描述的可综合（synthesizable）形式。除网络表和可综合实施方式之外，NOC 也可以以更低层的物理描述表达。可以以诸如 GDSII 的晶体管布局格式发布诸如 SERDES、PLL、DAC、ADC 等的模拟 IP 块元件。有时也以布局格式提供 IP 块的数字元件。也将认识到的是，符合本发明所实施的 IP 块以及其它逻辑电路可以以计算机数据文件的形式发布，所述计算机

数据文件以各种详细级别定义实施这样的逻辑的电路布置的功能性和 / 或布局，例如为逻辑限定程序代码。因此，虽然已经在并且后面还将在以完全功能集成电路器件实施的电路布置、使用这种器件的数据处理系统、以及其它有形物理硬件电路的上下文中描述本发明，但受益于该即时公开的本领域普通技术人员将认识到的是，本发明也可以在程序产品内实施，并且无论被用于发布程序产品的计算机可读存储介质的特定类型，本发明同样适用。计算机可读存储介质的示例包括但不限于物理可记录类型的介质，诸如易失和非易失性存储设备、软盘、硬盘驱动器、CD-ROM、和 DVD（以及其它）。

[0041] 图 2 的示例中的每个 IP 块 104 通过存储器通信控制器 106 适配于路由器 110。每个存储器通信控制器是适配于提供 IP 块和存储器之间的数据通信的同步和异步逻辑电路的集合（aggregation）。IP 块和存储器之间的这种通信的示例包括存储器加载指令和存储器存储指令。下面参考图 3 更详细地描述存储器通信控制器 106。每个 IP 块 104 也通过网络接口控制器 108 适配于路由器 110，所述网络接口控制器 108 控制 IP 块 104 之间通过路由器 110 的通信。IP 块之间的通信示例包括并行应用中和流水线式应用中的 IP 块之间的、携带数据和用于处理数据的指令的消息。

[0042] 路由器 110 和其间的相应链路 118 实施 NOC 的网络操作。链路 118 可以是在连接所有路由器的物理并行线总线上实施的包结构。即，可以在宽度足够同时容纳包括所有报头信息和有效负载数据的整个数据交换包的线总线上实施每个链路。如果包结构包括 64 比特，例如包括 8 字节的报头和 56 字节的有效负载数据，则构成（subtending）每个链路的线总线为 64 字节宽，512 条线。此外，每个链路可以是双向的，从而如果链路数据包结构包括 64 字节，则线总线在每个路由器和其在网络中的邻居之间实际上包含 1024 条线。在这样的实施方式中，消息可以包括多于一个包，但是每个包将精确地适合于线总线的宽度。在替代方案中，可以在宽度仅足够容纳包的一部分的线总线上实施链路，使得数据包将被拆分为多拍（beat），例如使得如果链路的宽度被实施为 16 字节、或 128 条线，则 64 字节的包可以被分成四拍。将会认识到的是，不同的实施方式可以基于实际物理限制和期望的性能特征而使用不同总线宽度。如果路由器和线总线的每个部分之间的连接被称为端口，那么每个路由器包括五个端口，其中，网络上数据传输的四个方向的每一个各使用一个端口，以及第五个端口用于通过存储器通信控制器和网络接口控制器将路由器适配于特定 IP 块。

[0043] 每个存储器通信控制器 106 控制 IP 块和存储器之间的通信。存储器可以包括片外主 RAM 112、通过存储器通信控制器 106 直接连接到 IP 块的存储器 114、启用为 IP 块的片上存储器 116、以及片上高速缓存（cache）。在 NOC 102 中，例如可以作将片上存储器 114、116 中的任一个实施为片上高速缓存存储器。存储器的所有这些形式可以被布置在相同的地址空间（物理地址或虚拟地址）中，甚至对于直接附接到 IP 块的存储器也是如此的。因此，存储器寻址的（addressed）消息关于 IP 块可以是完全双向的，因为可以从网络上任何位置上的任何 IP 块直接寻址这样的存储器。IP 块上的存储器 116 可以从该 IP 块或从 NOC 中的任何其它 IP 块来寻址。可以由通过存储器通信控制器而适配于该网络的 IP 块寻址直接附接到该存储器通信控制器的存储器 114，并且也可以从 NOC 中的任何位置上的任何其它 IP 块寻址直接附接到该存储器通信控制器的存储器 114。

[0044] NOC 102 包括两个存储器管理单元（“MMU”）120、122，例示了符合本发明实施例的用于 NOC 的两个替代存储器架构。在 IP 块内实施 MMU 120，其允许 IP 块内的处理器在虚拟

存储器中操作，同时允许NOC的整个剩余架构在物理存储地址空间中操作。在片外实施MMU 122，其通过数据通信端口 124 连接到NOC。端口 124 包括用于在NOC和MMU之间传输信号所需的引脚和其它互连，并且足够智能以将消息包从NOC包格式转换为外部MMU 122所需的总线格式。MMU的外部位置意味着：NOC的所有IP块中的所有处理器可以在虚拟存储地址空间中操作，由片外MMU 112处理向片外存储器的物理地址的所有转换。

[0045] 除了通过MMU 120、122的使用示出的两个存储器架构之外，数据通信端口126还示出能够在本发明实施例中采用的、在NOC中有用的第三存储器架构。端口126提供NOC 102的IP块和片外存储器112之间的直接连接。在处理路径中没有MMU的情况下，该架构提供NOC的所有IP块对物理地址空间的利用。在双向共享该地址空间中，通过直接连接到端口126的IP块引导的存储器寻址的消息（包括加载和存储），NOC的所有IP块可以访问地址空间中的存储器。端口126包括在NOC和片外存储器112之间传输信号所需的引脚和其它互连，并且充分智能以将消息包从NOC包格式转换为片外存储器112所需的总线格式。

[0046] 在图2的示例中，一个IP块被指派主机接口处理器128。主机接口处理器128提供NOC和安装了该NOC的主机计算机10之间的接口，并且还提供到NOC上的其它IP块的数据处理服务，例如包括接收和在NOC的IP块之间调度来自主机计算机的数据处理请求。NOC例如可以在较大型的计算机10上实施视频图形适配器26或协处理器28，如以上参考图1所述的。在图2的示例中，主机接口处理器128通过数据通信接口130连接到该较大型的计算机。端口130包括在NOC和主机计算机之间传输信号所需的引脚和其它互连，并且充分智能以将消息包从NOC转换为主机计算机10所需的总线格式。在图1的计算机中的NOC协处理器的示例中，这样的端口将提供NOC协处理器28的链路结构、与NOC协处理器28和总线适配器18之间的前端总线36所需的协议之间的数据通信格式转换。

[0047] 图3接着示出一功能框图，其更详细地示出NOC 102中的IP块104、存储器通信控制器106、网络接口控制器108以及路由器110内实施的组件，其以132集体表示。IP块104包括计算机处理器134和I/O功能136。在该示例中，由IP块104中的随机存取存储器（“RAM”）的片段表示计算机存储器。如以上参考图2所述的存储器可以占用物理地址空间的片段，其在每个IP块上的内容是可从NOC中的任何IP块寻址和访问的。每个IP块中的处理器134、I/O能力136、以及存储器138有效地将IP块实施为一般可编程微计算机。然而，如上所述，在本发明的范围内，IP块一般表示被用作NOC内的数据处理的构成块的同步或异步逻辑的可重复使用的单元。因此，本发明不局限于将IP块实施为一般可编程微计算机，尽管其为对于说明的目的有用的一般实施例。

[0048] 在图3的NOC 102中，每个存储器通信控制器106包括多个存储器通信执行引擎140。每个存储器通信执行引擎140被使得能够执行来自IP块104的存储器通信指令，包括网络和IP块104之间的双向存储器通信指令流141、142、144。由存储器通信控制器执行的存储器通信指令不仅可以源于通过特定存储器通信控制器适配于路由器的IP块，而且也可以源于NOC 102中的任何位置上的任何IP块104。即，NOC中的任何IP块可以生成存储器通信指令并且将该存储器通信指令通过NOC的路由器发送到与另一个IP块相关的另一个存储器通信控制器，用于执行该存储器通信指令。这样的存储器通信指令例如可以包括旁路转换缓冲控制指令、高速缓存控制指令、阻碍（barrier）指令、以及存储器加载和存储指令。

[0049] 每个存储器通信控制器 140 被使得能够独立并且与其它存储器通信执行引擎并行地执行完整的存储器通信指令。存储器通信执行引擎实施针对存储器通信指令的并行吞吐量而优化的可扩展的存储事务处理器。存储器通信控制器 106 支持多个存储器通信执行引擎 140，所有存储器通信执行引擎 140 并行运行以同时执行多个存储器通信指令。通过存储器通信控制器 106 将新的存储器通信指令分配到存储器通信引擎 140，并且存储器通信执行引擎 140 可以同时接受多个响应事件。在该示例中，所有存储器通信执行引擎 140 是相同的。因此，可以通过增减存储器通信执行引擎 140 的数目来实施可以通过存储器通信控制器 106 同时处理的存储器通信指令的数目的增减。

[0050] 在图 3 的 NOC 102 中，每个网络接口控制器 108 被使得能够将通信指令从命令格式转换为网络包格式，用于通过路由器 110 在 IP 块 104 之间传输。通信指令可以由 IP 块 104 或存储器通信控制器 106 表示为命令格式，并被以命令格式提供到网络接口控制器 108。命令格式可以是符合 IP 块 104 和存储器通信控制器 106 的架构寄存器文件的原生格式。网络包格式通常是通过网络的路由器 110 传输所需的格式。每个这样的消息由一个或多个网络包组成。网络接口控制器中的这种从命令格式转换为包格式的通信指令的示例包括 IP 块和存储器之间的存储器加载指令和存储器存储指令。这种通信指令也可以包括并行应用中和流水线式应用中的、在 IP 块之间发送消息的通信指令，该消息携带数据和用于在 IP 块之间处理数据的指令。

[0051] 在图 3 的 NOC 102 中，每个 IP 块被使得能够通过该 IP 块的存储器通信控制器向和从存储器发送基于存储地址的通信，并且接着还通过其网络接口控制器将该基于存储地址的通信发送到网络。基于存储地址的通信是由 IP 块的存储器通信控制器的存储器通信执行引擎执行的存储器访问指令，诸如加载指令或存储指令。这种基于存储地址的通信通常起源于 IP 块中，以命令格式表示，并且为了执行而被转移(hand off)到存储器通信控制器。

[0052] 以消息流量执行许多基于存储地址的通信，因为要被访问的任何存储器可以位于物理存储地址空间中的任何地方、在片上或片外、直接附接到 NOC 中的任何存储器通信控制器、或者最终通过NOC的任何IP块访问——而无论哪个IP块引起任何特定的基于存储地址的通信。因此，在 NOC 102 中，所有以消息流量执行的基于存储地址的通信从存储器通信控制器传递到相关的网络接口控制器，用于从命令格式转换为包格式并且通过网络在消息中传输。在到包格式的转换中，网络接口控制器还识别根据要由基于存储地址的通信访问的存储地址或多个存储地址的、用于数据包的网络地址。基于存储地址的消息与存储地址一起被寻址。每个存储地址由网络接口控制器映射到网络地址，通常是负责物理存储地址的某一范围的存储器通信控制器的网络位置。存储器通信控制器 106 的网络位置自然也是该存储器通信控制器的相关路由器 110、网络接口控制器 108、以及 IP 块 104 的网络位置。每个网络接口控制器内的指令转换逻辑 150 被使得能够为了通过 NOC 的路由器发送基于存储地址的通信的目的而将存储地址转换为网络地址。

[0053] 在从网络的路由器 110 接收消息流量时，每个网络接口控制器 108 检查用于存储器指令的每个包。包含存储器指令的每个包被交给与接收网络接口控制器相关的存储器通信控制器 106，其在将包的剩余有效负载发送到 IP 块用于进一步的处理之前执行存储器指令。以这种方式，总是准备存储内容，以在 IP 块开始执行来自消息的依赖于特定存储内容

的指令之前支持通过 IP 块的数据处理。

[0054] 在图 3 的 NOC 102 中，每个 IP 块 104 被使得能够绕过其存储器通信控制器 106 并且将 IP 块间的网络寻址通信 146 通过 IP 块的网络接口控制器 108 直接发送到网络。网络寻址通信是由网络地址指引到另一个 IP 块的消息。这样的消息在流水线式应用中发送工作数据、在 SIMD 应用中的 IP 块之间发送用于单个程序处理的多个数据，等等，如本领域技术人员将会遇到的。这样的消息与基于存储地址的通信的区别在于它们从一开始就通过知道该消息通过 NOC 的路由器要被指引到的网络地址的起源 IP 块而网络寻址。这样的网络寻址通信由 IP 块通过 I/O 功能 136 以命令格式直接传递到 IP 块的网络接口控制器，接着被网络接口控制器转换为包格式并通过 NOC 的路由器发送到另一个 IP 块。这样的网络寻址通信 146 是双向的，其可能进入以及来自 NOC 的每个 IP 块，取决于其在任何特定应用中的使用。然而，每个网络接口控制器被使得能够将这样的通信发送到相关路由器、以及从相关路由器接收这样的通信，并且每个网络接口控制器被使得能够绕过相关存储器通信控制器 106 而将这样的通信直接发送到相关 IP 块、以及从相关 IP 块直接接收这样的通信。

[0055] 图 3 的示例中的每个网络接口控制器 108 也被使得能够实施网络上的虚拟信道，通过类型刻画网络包的特征。每个网络接口控制器 108 包括虚拟信道实现逻辑 148，其将每个通信指令分类、并且在将包格式的指令转移到路由器 110 以在 NOC 上传输之前在网络包格式的字段中记录指令的类型。通信指令类型的示例包括 IP 块间的基于网络地址的消息、请求消息、请求消息的响应、指引到高速缓存的无效消息；存储器加载和存储消息；以及存储器加载消息的响应等。

[0056] 图 3 的示例中的每个路由器 110 包括路由逻辑 152、虚拟信道控制逻辑 154、以及虚拟信道缓冲器 156。路由逻辑通常被实施为在由路由器 110、链路 118、以及路由器之间的总线线形成的网络中实施用于数据通信的数据通信协议栈的同步和异步逻辑的网络。路由逻辑 152 包括本领域的读者可能在片外网络中将其与路由表相关联的功能，至少某些实施例中的路由表被认为对于 NOC 中的使用过于缓慢和繁琐。被实施为同步和异步逻辑的网络的路由逻辑可以被配置为快到在单个时钟周期中做出路由决定。该示例中的路由逻辑通过选择用于转发在路由器中接收的每个包的端口来路由包。每个包包括该包将要被路由到的网络地址。

[0057] 在上述基于存储地址的通信中，每个存储地址被描述为由网络接口控制器映射到网络地址，即，存储器通信控制器的网络位置。存储器通信控制器 106 的网络位置自然也是该存储器通信控制器的相关路由器 110、网络接口控制器 108、和 IP 块 104 的网络位置。因此，在 IP 块间、或基于网络地址的通信中，应用层数据处理通常查看作为由 NOC 的路由器、链路和总线线形成的网络中的 IP 块的位置的网络地址。图 2 示出这种网络的一个组织是行和列的网状结构，其中每个网络地址可以例如实施为用于网状结构的相关路由器、IP 块、存储器通信控制器、和网络接口控制器的每个集合的唯一标识符，或者网状结构中的每个这种集合的 x、y 坐标。

[0058] 在图 3 的 NOC 102 中，每个路由器 110 实施两个或更多个虚拟通信信道，其中每个虚拟通信信道通过通信类型刻画特征。通信指令类型以及因此虚拟信道类型包括上述那些类型：IP 块间的基于网络地址的消息、请求消息、请求消息的响应、指引到高速缓存的无效消息；存储器加载和存储消息；以及存储器加载消息的响应；等等。为了支持虚拟信道，图

3 的示例中的每个路由器 110 也包括虚拟信道控制逻辑 154 和虚拟信道缓冲器 156。虚拟信道控制逻辑 154 针对所分配的通信类型而检查每个接收的包，并且将每个包放置在针对该通信类型的输出(outgoing)虚拟信道缓冲器中，用于通过端口传输到 NOC 上的相邻路由器。

[0059] 每个虚拟信道缓冲器 156 具有有限的存储空间。当在短时间内接收许多包时，虚拟信道缓冲器可能充满——从而不能将更多包放入缓冲器中。在其它协议中，到达其缓冲器已满的虚拟信道的包将被丢弃(drop)。然而，该示例中的每个虚拟信道缓冲器 156 被使得能够用总线线的控制信号，通过虚拟信道控制逻辑通知(advise)周围的路由器暂停虚拟信道中的传输，即暂停特定通信类型的包的传输。当这样暂停一个虚拟信道时，所有其它虚拟信道不受影响，并且可以继续以全容量操作。通过每个路由器将控制信号一路传回到每个路由器的相关网络接口控制器 108。每个网络接口控制器被配置为在接收到这样的信号时拒绝从其相关地存储器通信控制器 106 或从其相关的 IP 块 104 接受用于暂停的虚拟信道的通信指令。以这种方式，虚拟信道的暂停影响实施虚拟信道的所有硬件，一路直到发源 IP 块。

[0060] 在虚拟信道中暂停包传输的一个效果是永不丢弃包。当路由器遇到在诸如例如因特网协议的某些不可靠协议下可能丢弃数据包的情况时，图 3 的示例中的路由器可以通过它们的虚拟信道缓冲器 156 和它们的虚拟信道控制逻辑 154 来暂停虚拟信道中的所有包的传输，直至缓冲器空间再次可用为止，从而消除了丢弃包的任何需要。因此，图 3 的 NOC 可以用非常薄的硬件层实施高可靠性的网络通信协议。

[0061] 图 3 的示例 NOC 也可以被配置为保持片上和片外高速缓存存储器二者之间的高速缓存一致性。每个 NOC 可以支持多个高速缓存，其中的每个针对相同的底层存储地址空间而操作。例如，高速缓存可以通过 IP 块、通过存储器通信控制器、或者通过 NOC 外部的高速缓存控制器来控制。图 2 的示例中的片上存储器 114、116 中的任何一个也可以被实施为片上高速缓存，并且在本发明的范围内，高速缓存存储器也可以被实施为片外高速缓存。

[0062] 图 3 中示出的每个路由器 110 包括五个端口，其中包括通过总线线 118 连接到其它路由器的四个端口 158A-D、以及通过网络接口控制器 108 和存储器通信控制器 106 将每个路由器连接到其相关的 IP 块 104 的第五个端口 160。如从图 2 和图 3 中的图示可见，NOC 102 的路由器 110 和链路 118 形成具有连接每个路由器中的垂直和水平端口的垂直和水平链路的网状网络。例如，在图 3 的图示中，端口 158A、158C 和 160 被称为垂直端口，且端口 158B 和 158D 被称为水平端口。

[0063] 图 4 接着以另一种方式示出符合本发明的 IP 块 104 的一个示例性实施方式，其被实施为被分区(partition)为发布(issue)或指令单元(IU)162、执行单元(XU)164、以及辅助执行单元(AXU)166 的处理元件。在所示的实施方式中，IU162 包括多个指令缓冲器 168，其从 L1 指令高速缓存(iCACHE)170 接收指令。每个指令缓冲器 168 是多个(例如，四个)对称多线程(SMT)硬件线程中的一个所专用的。有效至真实转换单元(iERAT)172 连接到 iCACHE170，并且被用于将来自多个取线程序列发生器(thread fetch sequencer)174 的取指令请求(instruction fetch request)转换为用于从较低阶存储器检索指令的真实地址。每个取线程序列发生器 174 是特定硬件线程专用的，并且被用于确保要被相关线程执行的指令被取到 iCACHE，用于调度到适当的执行单元。如图 4 中还示出的，取到指令缓冲器

168 的指令也可以通过分支预测逻辑 176 监测，所述分支预测逻辑 176 将提示（hint）提供给每个取线程序列发生器 174，以最小化由执行线程中的分支引起的指令高速缓存未命中（miss）。

[0064] IU 162 还包括每个硬件线程专用的依赖性（dependency）/ 发布逻辑块 178，并且依赖性（dependency）/ 发布逻辑块 178 被配置为解决依赖性并控制从指令缓冲器 168 到 XU 164 的指令发布。此外，在所示实施例中，在 AXU 166 中提供独立的依赖性 / 发布逻辑 180，因而使独立指令能够被不同线程同时发布到 XU 164 和 AXU 166。在替代实施例中，逻辑 180 可以被布置在 IU 162 中，或者可以完全被省略，使得逻辑 178 将指令发布到 AXU166。

[0065] XU 164 被实施为定点执行单元，包括连接到定点逻辑 184 的通用寄存器（GPR）的集合 182、分支逻辑 186 以及加载 / 存储逻辑 188。加载 / 存储逻辑 188 通过由 dERAT 逻辑 192 提供的有效至真实转换，连接到 L1 数据高速缓存（dCACHE）190。XU 164 可以被配置为实际实施任何指令集合，例如 32b 或 64b PowerPC 指令集合的全部或部分。

[0066] AXU 166 操作为包括专用的依赖性 / 发布逻辑 180 以及一个或多个执行块 194 的辅助执行单元。AXU 166 可以包括任意数目的执行块，并且可以实际实施任何类型的执行单元，例如 :浮点单元，或者诸如加密 / 解密单元、协处理器、矢量处理单元、图形处理单元、XML 处理单元等的一个或多个专门的执行单元。在所示实施例中，AXU 166 包括到 XU 164 的高速辅助接口，例如用以支持 AXU 架构状态和 XU 架构状态之间的直接迁移（move）。

[0067] 可以经由连接到 NOC 102 的网络接口控制器 108，以以上连同图 2 所讨论的方式管理与 IP 块 104 的通信。可以与基于消息的通信一起提供基于地址的通信，例如，用以访问 L2 高速缓存存储器。例如，每个 IP 块 104 可以包括专用的收件箱（in box）和 / 或发件箱（out box），以便处理 IP 块之间的节点间通信。

[0068] 可以在以上连同图 1-4 所说明的硬件和软件环境内实施本发明的实施例。然而，受益于本即时公开的本领域普通技术人员将认识到，本发明可以在诸多不同环境中实施，并且可以在不违背本发明的精神和范围的情况下对上述硬件和软件环境做出其它修改。因此，本发明不限于此处公开的特定硬件和软件。

[0069] 基于存储地址转换的数据加密 / 压缩

[0070] 保护未加密的安全数据不被在安全线程之外访问在许多数据处理应用中极为重要。一般地，该数据仅在芯片外部被保持安全，但是由于 SOC 在芯片上增长到几百个处理器，所以保护未加密数据甚至不被同一芯片上的其它处理访问也越来越重要。传统地，为加密所识别的某些地址范围被过滤、并且由于数据被传递到存储控制器 / 从存储控制器传递而被加密 / 不加密。然而，这可以导致未加密数据存在于高速缓存中，并且对于其它线程是可访问的。

[0071] 另一方面，符合本发明的关于加密的实施例能够通过将一个或多个关于加密的页面属性添加到用于执行虚拟和真实存储地址之间的存储地址转换的存储地址转换数据结构，来保护存储页面仅被授权线程访问。例如，在一个实施例中，可以将一个或多个关于加密的页面属性添加到处理核的有效至真实转换（ERAT）表的页表条目（PTE），使得将仅允许具有访问存储页面许可的安全线程（如由该页面的 PTE 指定）访问该页面。如果另一个线程试图访问该页面，则将会引起安全中断，并且通过系统管理程序（hypervisor）或其它主管级软件（supervisor-level software）以期望的方式处理安全中断。此外，页面属性还被

用于通过识别需要被发送到加密引擎的存储器访问而简化硬件加密 / 解密。响应于 L1 高速缓存上的未命中，可以将该页面属性例如发送到 L2 高速缓存或其它较低级存储器，作为必须针对加载或存储而将重载（reload）数据流化通过加密引擎以适当地解密 / 加密该数据的指示。

[0072] 通过将关于加密的页面属性合并入处理器的存储地址转换功能，在降低了对性能的影响的情况下增强了安全数据在整个数据处理系统中的安全性，特别是在具有许多 SMT 处理核的 SOC 中。此外，在许多实例中，PTE 不捆绑到特定处理标识符，因此在本发明的某些实施例中，不同处理可以被授权访问同一加密数据。

[0073] 同样地，关于在数据处理系统中压缩的数据，传统的数据处理系统使用一系列寄存器控制压缩的存储区域以配置存储范围，其通常需要附加硬件以及复杂的软件配置，其二者都可能对系统性能产生不利影响。

[0074] 另一方面，符合本发明的关于压缩的实施例将一个或多个关于压缩的页面属性添加到用于执行虚拟和真实存储地址之间的存储地址转换的存储地址转换数据结构，来简化压缩 / 解压过程并减少与访问压缩数据相关的等待时间（latency）。例如，在一个实施例中，可以将一个或多个关于压缩的页面属性添加到处理核的有效至真实转换（ERAT）的页表条目（PTE），使得当针对包括压缩数据的存储页面初始化 PTE 时，可以当执行加载时将相应的关于压缩的页面属性转发到存储器子系统，从而使得来自存储器或较高级高速缓存的数据将直接流化通过硬件压缩引擎以被解密。相反过程也如此，即，任何存储数据将在被发送到第一级的压缩存储之前流化通过压缩引擎。这简化了管理压缩数据的过程，并且减少了与管理压缩数据相关的支持硬件以及性能开销的量。

[0075] 此外，在符合本发明的某些实施例中，页面属性可以包括级别属性，其可以用于将页面配置为在存储系统的各种级别中（例如，在主存储器或在像 L1、L2 或 L3 的较高级的高速缓存中）被选择性地加密和 / 或压缩。因而，例如，某些页面可以被在 L2 或 L3 高速缓存中加密 / 压缩，同时其它页面可以被在存储器中加密 / 压缩，但是被在 L2 或 L3 高速缓存中解密 / 解压。这提供了更大的灵活性并且可以提高性能，特别是当期望加速针对保存在较高级的存储系统中的频繁访问的数据的存储访问性能时。此处实施的压缩可以有效地增加高速缓存的大小而不需要相应增加存储器带宽，以及具有其它益处。

[0076] 例如，图 5 示出符合本发明的适合于实施基于存储地址转换的数据加密 / 压缩的示例性数据处理系统 200。系统 200 被例示为具有存储器总线 202，其将多个处理核 204 与存储器管理单元（MMU）206 连接到一起。尽管图 5 中仅示出两个处理核 204，但将认识到的是，在本发明的不同实施例中可以采用任何数目的处理核。

[0077] 每个处理核 204 是包括多个（N 个）硬件线程 208、以及有效至真实转换（ERAT）表 210 和集成的 L1 高速缓存 212 的 SMT 核。如本领域所理解的，ERAT 表 210 用作用于存储地址转换数据（例如，PTE）的高速缓存，并且通常与较低级数据结构（例如，布置在 MMU 206 中或可被 MMU 206 访问的旁路转换缓冲（TLB）214）相关联。TLB 214 也可以用作用于较大的页面表的高速缓存，较大的页面表通常被存储在存储器 216 中。

[0078] 存储系统可以包括多级存储器和高速缓存，并且这样，示出的数据处理系统 200 包括连接到 MMU 206 并且由处理核 204 共享的 L2 高速缓存 218，然而，将认识到的是，在本发明的其它实施例中可以采用各种替代存储器架构。例如，可以使用例如 L3 高速缓存的附

加级别的高速缓存存储器，并且在某些实施例中(例如，在基于非一致存储器访问(NUMA)的数据处理系统中)，可以将存储器 216 分区。此外，例如，可以对特定处理核专用附加的高速缓存级别，使得每个处理核包括专用的 L2 高速缓存，其可以被集成到处理核或者连接在处理核和存储器总线之间。在某些实施例中，L2 或 L3 高速缓存可以直接连接到存储器总线，而不是经由专用接口连接到 MMU。

[0079] 此外，将会认识到的是，图 5 中示出的组件可以被集成到同一集成电路器件、或芯片上，或者可以被布置在多个这样的芯片中。例如，在一个实施例中，每个处理核被实施为 NOC 布置中的 IP 块，并且总线 202、MMU 206 和 L2 高速缓存 218 被集成到同一芯片上作为 SOC 布置中的处理核。在其它实施例中，总线 202、MMU 206、L2 高速缓存 218、和 / 或存储器 216 各自可以被集成到同一芯片上或来自所述处理核的不同芯片中，并且在某些实例中处理核可以被布置在独立芯片上。

[0080] 鉴于可以采用本发明的各种已知的处理器和存储器架构，因此将认识到的是，本发明不限于此处所示的特定存储器架构。

[0081] 为了实施符合本发明的基于存储地址转换的数据加密 / 压缩，数据处理系统 200 包括加密引擎 220 和压缩引擎 222，其连接到总线 202 并因此可被访问用于加密 / 解密和压缩 / 解压在总线 202 上通信的数据。尽管引擎 220 和 222 被分别称为加密和压缩引擎，但将认识到的是，引擎 220 通常包括加密和解密逻辑二者，并且引擎 222 通常包括压缩和解压逻辑二者，而不论此处所使用的名称。将认识到的是，在某些实施例中，可以在独立的“引擎”中布置加密和解密逻辑，也可以在独立的“引擎”中布置压缩和解压逻辑。然而，为了本发明的目的，加密引擎可以被认为是能够执行数据的加密和 / 或解密的硬件逻辑的任何集合，并且压缩引擎可以被认为是能够执行数据的压缩和 / 或解压的硬件逻辑的任何集合。

[0082] 为了便于讨论本发明的目的，数据处理系统 200 被描述为包括加密和压缩功能两者。然而，在许多实施例中，可能期望仅支持加密或仅支持压缩功能，并且这样，符合本发明的实施例不需要支持数据加密和数据压缩二者。因而，在某些实施例中可以省略引擎 220、222 中的任一，并且在某些实施例中，用于指示存储页面是否被加密或压缩的页面属性可以仅包括关于加密的属性或关于压缩的属性。此外，尽管示出的引擎 220、222 附接到总线 202，但引擎 220、222 中的任何一个或这二者可以连接到 MMU206 或集成于其中。

[0083] 如上所述，基于存储地址转换的数据加密 / 压缩可以通过将一个或多个页面属性添加到例如页表条目(PTE)的存储地址转换数据结构而实施。例如，图 6 示出能够在 ERAT 210 中保持并扩展为包括各种页面属性 232-238 以支持基于存储地址转换的数据加密 / 压缩的示例性 PTE 230。对于加密，可以使用加密属性 232 (例如，1 比特标志)来指示页面中的数据是否被加密。同样地，对于压缩，可以使用压缩属性 234 (例如，1 比特标志)来指示页面中的数据是否被压缩。

[0084] 此外，在某些实施例中，可能期望选择性地指定数据在页面中被加密和 / 或压缩的级别，使得将在存储器架构中的任何更高级别(或者可选地，以指定级别)处解密 / 解压数据，并且将在存储器架构中的指定级别和任何更低级别处加密 / 压缩数据。例如，可以提供 2 比特级别属性(例如，用于加密的级别属性 236 和用于压缩的级别属性 238)，以编码到四个存储器级别，例如，L1=“00”、L2=“01”、L3=“10”以及存储器 =“11”。替代地，每个存储器级别可以具有与其相关联的独立 1 比特标识。此外，在某些实施例中，加密和 / 或压缩

属性 232、234 可以与相关级别属性联合。例如，如果不支持 L3 高速缓存，则在 2 比特级别属性中编码的四个状态中的一个(例如，“00”)可以表示该页面是未加密或未压缩的。

[0085] 类似于传统 PTE，PTE 230 也存储附加数据。例如，可以在 PTE 中包括附加页面属性 240，诸如指示页面是否是可缓存的、被保护的(guarded)、或者只读的，是否需要存储器一致性或写通(write-through)，endian 模式比特等的属性；也可以包括分配到用户模式数据 242 的一个或多个比特，用于软件一致性或对高速缓存锁定选项的控制。提供访问控制页面属性 244，以例如通过指定与被授权访问该页面的处理(process)相关的处理标识符(PID)、或者匹配和 / 或掩蔽(mask)数据的组合(选择性地)、或者适合于指定被授权访问存储页面的处理的集合的其它数据，来控制允许什么处理访问存储页面。例如，访问控制属性可以从 PID 掩蔽掉的一个或多个 LSB，使得与访问控制属性中的 MSB 匹配的任何 PID 将被允许访问相应的存储页面。ERAT 页面属性 246 为 PTE 存储有效至真实转换数据，其通常包括与被用于访问该 PTE 的有效 / 虚拟地址相应的真实地址、以及该有效 / 虚拟地址，其也被用于经由 CAM 功能索引 ERAT。

[0086] 将认识到的是，PTE 230 的格式也可以被用在位于存储器架构中的 TLB 214 和任何其它页面表中。替代地，存储在存储器架构的不同级别中的 PTE 可以基于存储器架构的该特定级别的需要包括其它数据或者省略某些数据。此外，将认识到的是，尽管此处讨论的实施例采用术语 ERAT 和 TLB 来描述将存储地址转换信息存储或高速缓存在处理器或处理核中的各种硬件逻辑，但这种硬件逻辑可以被称为其它命名，因此本发明不限于与 ERAT 和 TLB 一起使用。此外，可以使用其它 PTE 格式，并且因此本发明不限于图 6 中示出的特定 PTE 格式。

[0087] 通过在 PTE 中存储关于加密和关于压缩的属性，页面是否被加密和 / 或压缩的判定以及这样的数据的实际解密和解压被限制到仅那些被数据处理系统中的否则控制到页面本身的访问的功能授权的处理，以及代表其执行的硬件线程。因此，可以将基于页面的访问控制扩展为支持数据处理系统中存储的加密和 / 或压缩数据的管理，所述基于页面的访问控制传统上被用于阻止数据处理系统中执行的处理访问或破坏位于数据处理系统中的其它处理的存储。

[0088] 如在本领域中公知的，例如运行在固件、内核(kernel)、分区管理器或操作系统中的系统管理程序或其它主管级软件传统地被用于将存储页面分配到特定处理并且处理访问违规(violation)，否则如果处理试图访问未被授权访问的存储页面，则可能发生所述访问违规。这样的主管级软件例如可以使用数据处理系统中的用于高速缓存来自 TLB 214 和 ERAT 210 中的页面表的 PTE 的专用硬件管理用于数据处理系统的整个页面表。符合本发明的实施例因此能够充分利用现有的主管级访问控制来限制对加密和 / 或压缩数据的访问。例如，在许多实例中，处理可能甚至不能确定存储页面是否被加密或压缩，因为对 PTE 的访问被主管级软件限制。因而，例如，如果一个处理核上执行的处理都没有被授权访问已经被分配到另一处理核上的处理的存储页面，则前一个处理核中的处理将甚至都不被允许从该存储页面检索(retrieve)数据，即使以加密或压缩的形式。

[0089] 例如，为了示出符合本发明的采用基于存储地址转换的数据加密 / 压缩的示例性存储器访问的目的，图 7 示出示例性数据处理系统 250，并且特别地，示出其中的示例性处理核。(例如在处理核的加载 / 存储单元中提供的)地址生成逻辑 252 可以例如响应于由处

理核中执行的硬件线程(未示出)所执行的指令,生成存储器访问请求以访问来自特定存储页面的数据(例如,高速缓存线(cache line))。将存储器访问请求并行地发布到ERAT253和L1高速缓存254二者,前者执行地址转换操作、以及确定存储器访问请求对于与请求硬件线程相关的PID是否被授权,并且后者确定由存储器访问请求指定的高速缓存线当前是否被高速缓存在L1高速缓存中。在图7所示的实施例中,ERAT 253被指派"dERAT"并且L1高速缓存254被指派"dCache",以指示这些组件与数据访问相关,并且指示可以提供相应的iERAT和iCache组件以处理指令访问(未示出)。

[0090] ERAT 253响应于存储器访问请求而访问由存储器访问请求指定的存储页面的PTE 256。系统管理程序保护异常处理程序(handler)逻辑258将用于存储器访问请求的PID与PTE中的访问控制比特进行比较,并且如果由于PID未被授权访问该存储页面而导致发生访问违规,则逻辑258通过将软件异常抛到主管级软件而发出中断信号,如260所表示的。在存储器访问请求被授权但在L1高速缓存上发生未命中的事件中,存储器访问请求被转发到加载/未命中队列262,其将该请求发布到例如L2高速缓存264的更低级存储器。

[0091] 图8更详细地示出可以响应于代表数据处理系统250中的处理的硬件线程发布的存储器访问请求而执行的操作序列270。例如处理程序逻辑258的保护逻辑访问ERAT 253,以确定PTE 256是否指示请求线程有权访问与存储器访问请求相关的页面(块272)。如果被授权(块274),则进行请求是否能够由L1高速缓存254实现的判定(块276)。如果存储器访问请求在L1高速缓存254上命中,则请求由L1高速缓存254实现(块278),并且完成存储器访问请求的处理。

[0092] 然而,如果请求在L1高速缓存254上未命中,则请求被路由到加载/未命中队列262,以在队列中添加对应于该请求的条目。此外,可能期望在条目中设置指示符,以指示请求与被加密和/或压缩的数据相关联。接着,在将请求发布到更低级存储器(例如通过存储器总线发布到L2高速缓存或更低级存储器)之前,在块282中进行页面是否被指示为被加密和/或压缩的判定,如从PTE 256中的页面属性所确定。如果不是,则块284中对于存储器访问请求发布总线事务。另一方面,如果该页面被加密和/或压缩,则在块286中与来自PTE 256的附加的关于加密/压缩的边带(sideband)数据一起发布总线事务。

[0093] 关于加密/压缩的边带数据可以以符合本发明的多种方式,通过存储器总线进行通信。例如,可以在总线架构中提供附加控制线,以指定总线事务是否与加密和/或压缩数据相关联,使得可以基于一个或多个控制线的状态确定数据是否被加密或压缩。替代地,事务类型可以与加密和/或压缩数据相关,使得可以简单地基于总线事务的事务类型而做出判定。特别地,在后者的实例中,将不需要加密引擎或压缩引擎侦探(snoop)特定存储范围,而是可以仅仅寻找某些事务类型。

[0094] 返回到块274,在请求线程未被授权访问所请求的页面的情况中,控制转到块288以处理访问违规。与传统访问违规处理不同,在某些实施例中可能期望对于与加密数据相关的访问违规执行替代或增强的操作,以反映与试图访问安全数据的未授权访问相关联的额外考虑。这样,在检测到访问违规时,块288通过访问页面的关于加密的页面属性来确定页面是否被加密。如果未被加密,则断言(assert)软件异常(块290)并且以传统方式处理。另一方面,如果页面被加密,则控制转到块292以确定是否关闭数据处理系统。在某些高安全性实施例中,例如,在数据处理系统中可能保持高度机密信息的情况中,可能期望在潜在

攻击的事件中立即关闭系统。这样，可能期望提供访问违规导致系统立即关闭的可配置模式。这样，如果可配置模式被设置，则块 292 将控制转到块 294 以关闭系统。否则，块 292 将控制转到块 296，以类似于传统异常地断言软件异常，除了具有指示正被发出异常信号的页面被加密的指示符。例如内核或其它管理级程序的软件可以接着执行增强的异常处理，诸如记录访问加密页面的尝试、通知中央处理器、在网络上将消息发送到外部设备、擦除存储器和状态内容、或者基于数据和 / 或数据处理系统的安全性要求所期望的任何其它操作。

[0095] 图 9 和图 10 接着分别示出与执行上面连同图 8 所讨论的响应于存储器访问请求而发布的读取和写入总线事务相关的操作序列。例如，图 9 示出用于处理读取总线事务的操作序列 300。读取总线事务最初导致通过总线（例如，如果数据已经被高速缓存，则通过 L2 或 L3 高速缓存，或者如果数据未被高速缓存则从存储器）实现请求（块 302）。接着，例如通过针对与加密或压缩事务相关的事务类型、或者总线上被断言的控制线而分别侦探总线的加密和压缩引擎，进行事务是否指示数据被加密（块 304）或被压缩（块 306）的判定。

[0096] 如果数据未被加密或压缩，则数据以传统方式返回（块 308）。然而，如果数据被加密，则在将数据返回到请求处理核之前将返回数据流化通过加密引擎（例如，图 5 的加密引擎 220），以将数据解密（块 310）。同样地，如果数据被压缩，则在数据返回到请求处理核之前将返回数据流化（块 312）通过压缩引擎（例如，图 5 的压缩引擎 222）。

[0097] 如图 10 中所示，经由操作序列 320 以类似于读取总线事务的方式处理写入总线事务。写入总线事务包括要被写入到较低级存储器的高速缓存线；然而，在例如通过 L2 或 L3 高速缓存或主存储器将数据转发到适当目的地之前，例如通过针对与加密或压缩事务相关联的事务类型、或者总线上被断言的控制线而分别侦探总线的加密和压缩引擎，进行事务是否指示数据被加密（块 322）或被压缩（块 324）的判定。

[0098] 如果数据未被加密或压缩，则数据以传统方式被写入到适当目的地（块 326）。然而，如果数据未被加密，则在写入数据之前首先将写入数据流化通过加密引擎（例如图 5 的加密引擎 220），以加密数据（块 328）。同样地，如果数据被压缩，则在写入数据之前首先将写入数据流化（块 330）通过压缩引擎（例如图 5 的压缩引擎 222）。

[0099] 图 11 和图 12 接着分别示出与执行上面连同图 8 所讨论的响应于存储器访问请求而发布的读取和写入总线事务相关的操作序列，但是针对在 PTE 中支持级别属性以控制存储页面被在多级存储器架构中的哪个级别中加密 / 压缩的实施方式。

[0100] 例如，图 11 示出用于处理读取总线事务的操作序列 350。读取总线事务最初导致通过总线（例如，如果数据已经被高速缓存，则通过 L2 或 L3 高速缓存，或者如果数据未被高速缓存则从存储器）实现请求（块 352）。接着，例如通过针对与加密或压缩事务相关的事务类型、或者总线上被断言的控制线而分别侦探总线的加密和压缩引擎，进行事务是否指示数据在数据来源的存储器架构级别上被加密（块 354）或被压缩（块 356）的判定。在所示实施例中，如果与所请求的数据相关的级别属性等于或高于数据来源的存储器级别，则所请求的数据将被加密或压缩（例如，如果级别属性指示 L2 高速缓存，则将在 L2 高速缓存、在任意 L3 或更低级高速缓存中、以及主存储器中加密 / 压缩数据）。

[0101] 如果数据未被在数据来源的级别上加密或压缩，则数据以传统方式返回（块 358）。然而，如果数据被在来源级别上加密，则在将数据返回到请求处理核之前将返回数据流化通过加密引擎（例如图 5 的加密引擎 220），以解密数据。同样地，如果数据被在来源级别上

压缩，则在将数据返回到请求处理核之前将返回数据流化（块 362）通过压缩引擎（例如图 5 的压缩引擎 222）。

[0102] 如图 12 所示，以与读出总线事务类似的方式，经由操作序列 370 处理写入总线事务。写入总线事务包括要被写入到较低级存储器中的高速缓存线；然而，在（例如通过 L2 或 L3 高速缓存或主存储器）将数据转发到适当目的地之前，例如通过针对与加密或压缩事务相关的事务类型、或者总线上被断言的控制线而分别侦探总线的加密和压缩引擎，进行事务是否指示数据被在事务的目标级别上加密（块 372）或被压缩（块 374）的判定。

[0103] 如果数据未被在目标级别上加密或压缩，则将数据以传统方式写入到适当目的地（块 376）。然而，如果数据被在目标级别上加密，则在写入数据之前首先写入数据流化通过加密引擎（例如图 5 的加密引擎 220），以加密数据（块 378）。同样地，如果数据被在目标级别上压缩，则在写入数据之前首先将写入数据流化（块 380）通过压缩引擎（例如图 5 的压缩引擎 222）。

[0104] 因而，在图 11-12 中所示实施例中，通过可能对不同存储页面指定不同选择的级别，将仅在存储器分级结构（hierarchy）中的选择的级别上加密和 / 或压缩数据。因此，对于不同应用可以提供高度的灵活性。然而，将认识到的是，不需要在所有实施方式中实施该级别属性，并且此外，在支持加密和压缩两个功能的实施方式中不需要对加密和压缩二者都支持级别属性。

[0105] 可以在符合本发明的某些实施例中支持附加存储事务。例如，每当更新加密或压缩的高速缓存线中的数据时，可以使用读取 - 修改 - 写入事务（read modify write transaction）来加密和 / 或重新压缩整个高速缓存线。

[0106] 集成加密引擎

[0107] 如上所述，在某些实施例中，加密引擎可以被集成到处理核中，以在处理核内有效地提供安全高速缓存，因而在整个存储系统中提供对安全数据的进一步保护。可以使用上述关于加密的页面属性来访问该安全高速缓存，以阻止安全数据在任何时候以未加密形式离开处理核，从而该安全数据在位于处理核外部的任何时间都被加密。集成加密引擎通常被配置为执行诸如加密和解密的加密操作。

[0108] 例如，在一个实施例中，可以与标准（非安全）L1 高速缓存一起提供独立的安全 L1 高速缓存，使得安全 L1 高速缓存可以与标准高速缓存并行地被访问，并在访问后被多路复用到流水线。在安全高速缓存中未命中时，可以将加载发送到存储系统的下一级别，例如 L2 高速缓存。可以在加载未命中队列中保持关于加密的页面属性，使得当返回加密数据时其将通过用于解密的集成加密引擎转发到安全高速缓存，因而确保仅被授权的安全线程可以访问未加密数据。然而，在另一个实施例中，可以不使用独立的安全 L1 高速缓存，由此可以使用集成加密引擎来基于与该数据相关的关于加密的页面属性选择性地加密和解密存储在 L1 高速缓存中的数据。

[0109] 例如，为了示出采用符合本发明的集成加密引擎的示例性存储器访问的目的，图 13 示出示例性数据处理系统 400，并且特别地，示出其中的示例性处理核。例如，响应于由处理核中执行的硬件线程（未示出）所执行的指令，例如在处理核的加载 / 存储单元中提供的地址生成逻辑 402 可以生成存储器访问请求，以访问来自特定存储页面的数据（例如高速缓存线）。存储器访问请求被并行地发布到 ERAT 406 和 L1 高速缓存 404 二者，前者确定

对于与请求硬件线程相关的PID，存储器访问请求是否被授权，并且后者确定由存储器访问请求指定的高速缓存线当前是否被高速缓存在L1高速缓存中。在图13所示的实施例中，ERAT 406被指派“dERAT”并且L1高速缓存404被指派“dCache”，以指示这些组件与数据访问相关，并且指示可以提供相应的iERAT和iCache组件以处理指令访问(未示出)。

[0110] ERAT 406响应于存储器访问请求而访问用于由存储器访问请求指定的存储页面的PTE 408。系统管理程序保护异常处理程序逻辑410将用于存储器访问请求的PID与PTE中的访问控制比特进行比较，并且如果由于PID未被授权访问该存储页面而导致发生访问违规，则逻辑410通过将软件异常抛到主管级软件而发出中断信号，如412所表示的。在存储器访问请求被授权但在L1高速缓存上发生未命中的情况中，存储器访问请求被转发到加载/未命中队列414，其将请求发布到例如L2高速缓存416的较低级存储器。

[0111] 此外，L1高速缓存404包括与其连接并集成到处理核中的加密引擎418。例如，可以使用加密引擎418来加密被写出处理核的数据(例如来自L1高速缓存404)，或者解密被处理核使用并从较低级存储器或者从L1高速缓存接收的数据。例如，加密引擎418可以被配置为经由总线422将解密的数据流化到寄存器文件420，或者将解密的数据流化回到L1高速缓存404。此外，加密引擎418可以连接到旁路网络(bypass network)424，以绕过寄存器文件420并将解密的数据直接提供到执行单元(未示出)。

[0112] 利用图13中所示的配置，可以实施许多场景。例如，可以使用L1高速缓存404以存储加密和未加密的数据二者。替代地，L1高速缓存404可以是安全高速缓存，独立的L1高速缓存(未示出)被用于存储非安全数据。此外，在某些实施方式中，L1高速缓存404可以存储解密的数据，使得加密引擎418被用于解密从较低级存储器所接收的数据、以及加密被写出到较低级存储器的数据。在替代方案中，特别是对于安全性敏感的应用，可能期望在L1高速缓存404中以加密形式保持所有安全数据，由此可以使用加密引擎418来将从L1高速缓存404检索的数据解密到寄存器文件420中，并且将从寄存器文件420写回到L1高速缓存404的数据加密。

[0113] 在某些实施例中，可能期望在与针对存储器的安全页面的存储器访问请求相关的加载/未命中队列中的条目中联合地设置指示符或属性。通过这样做，当返回所请求的数据时，可以访问该指示符以确定数据被加密。这样，例如如果在处理核中提供两个独立的安全和非安全L1高速缓存，则可以使用该指示符来将返回数据路由到适当的L1高速缓存。替代地，如果在L1高速缓存中以未加密格式存储安全数据，则可以使用该指针来导致在将返回数据存储在L1高速缓存中之前将返回数据流化通过加密引擎。

[0114] 接着，转到图14，如上所述，在某些实例中，可能期望在处理核中采用独立的安全和非安全L1高速缓存，例如使得非安全数据不经受与加密和解密数据相关的性能开销。例如，处理器450合并了与加载/存储单元454连接的加密引擎452。加载/存储单元454中的ERAT456将关于加密的以及其它的页面属性提供给非安全L1高速缓存458和安全L1高速缓存460二者。每个L1高速缓存458、460可以具有独立的加载/未命中队列462、464以及连接到存储器总线470的总线连接466、468，或者替代地可以共享同一加载/未命中队列和总线连接。

[0115] 集成加密引擎的实施方式可以在不同实施例中变化。例如，可以在某些实施例中将集成加密引擎实施为诸如上面连同图4所讨论的AXU。然而，本发明不限于这样的实施方

式。

[0116] 结论

[0117] 符合本发明的实施例相对于传统设计提供许多优势。例如，页面属性的使用通常消除开销，所述开销否则是存储器管理单元用于设置和管理加密或压缩数据的存储器区域或范围所需的开销。此外，在多个处理核被连接到同一总线的情况中，其它处理核将通常不能访问用于另一处理核的加密或压缩数据，因为在每个处理核内保持用于加密或压缩数据的页面属性，并且加密或压缩数据被限制为仅由授权处理访问。

[0118] 也可以做出符合本发明的各种修改。例如，如同集成加密引擎，可能期望在处理核内提供集成压缩引擎，或者期望提供能够采用页面属性以基于其选择性地处理数据的其它类型的加速器引擎。

[0119] 可以在不偏离本发明的精神和范围的情况下对所公开的实施例做出其它修改。因此，本发明在于所附权利要求书中。

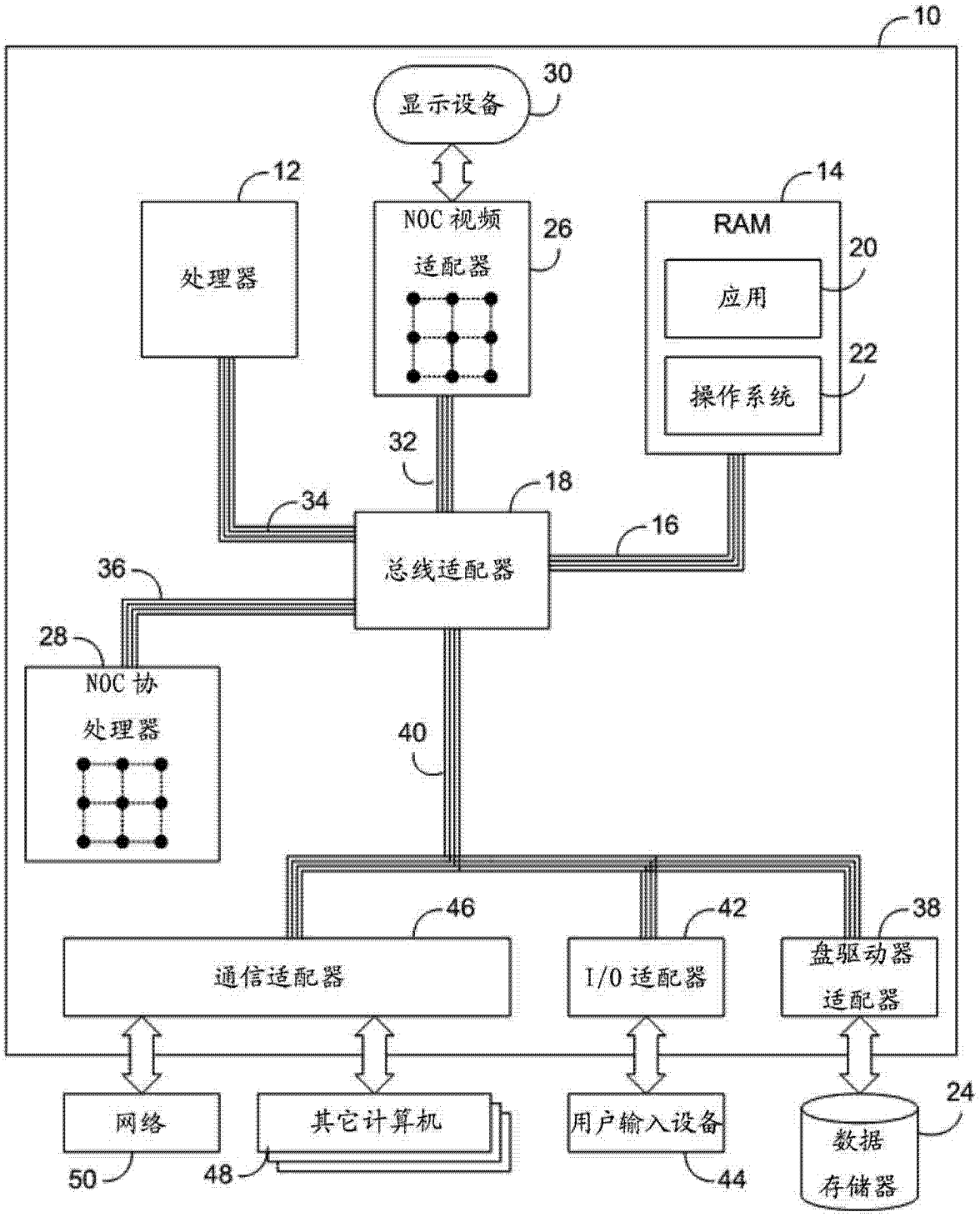

10
显示设备
30
12
处理器
NOC 视频
适配器
26
14
RAM
20
应用
22
操作系统
32
18
34
16
总线适配器
36
28
NOC 协
处理器
40
46
42
38
通信适配器
I/O 适配器
盘驱动器
适配器
网络
其它计算机
用户输入设备
24
数据
存储器
50
48
44

图 1

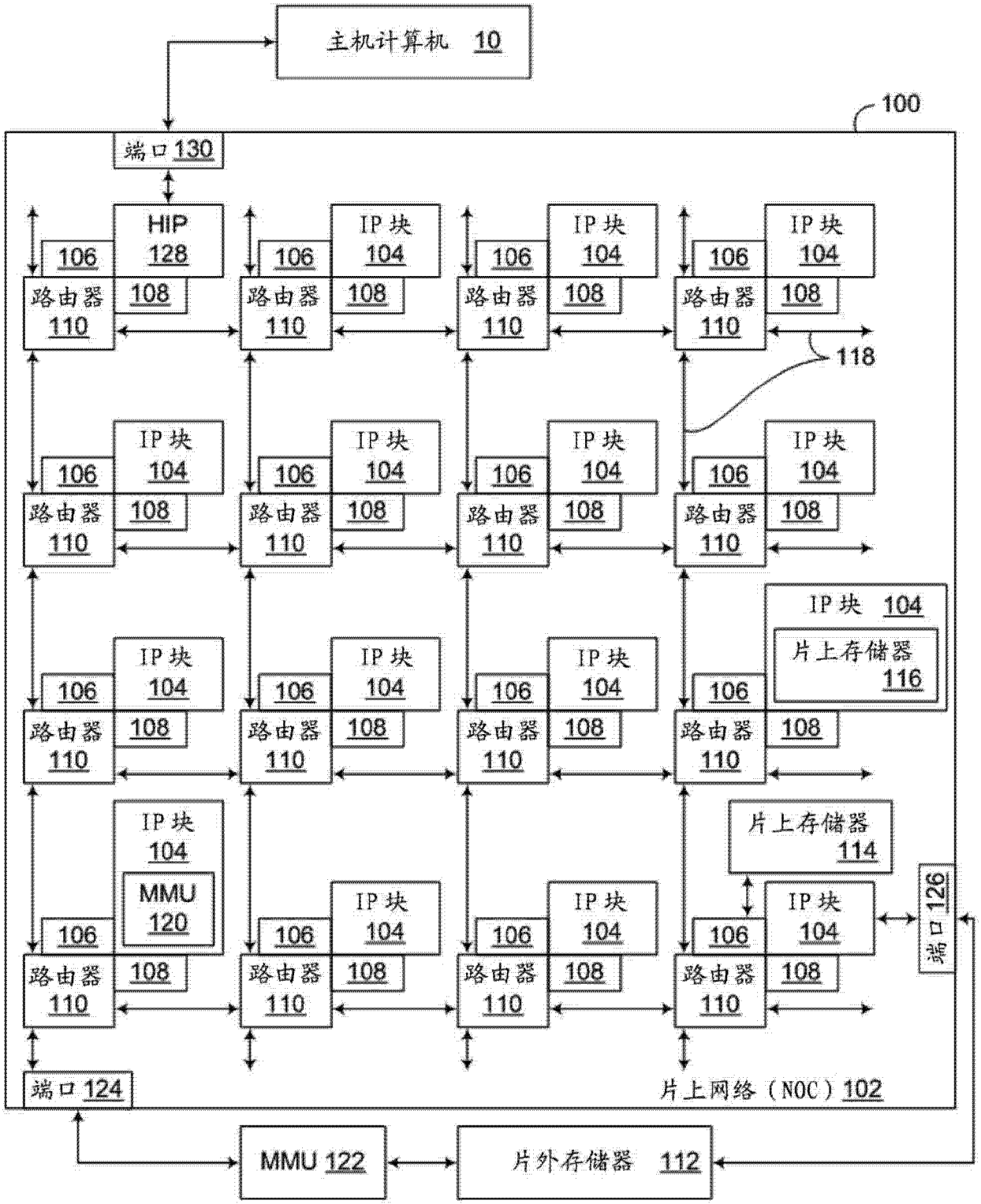

主机计算机 10
100
端口 130
HIP 128
IP 块 104
106
路由器 110
108
118
片上存储器 116
片上存储器 114
MMU 120
端口 126
端口 124
片上网络（NOC）102
MMU 122
片外存储器 112

图 2

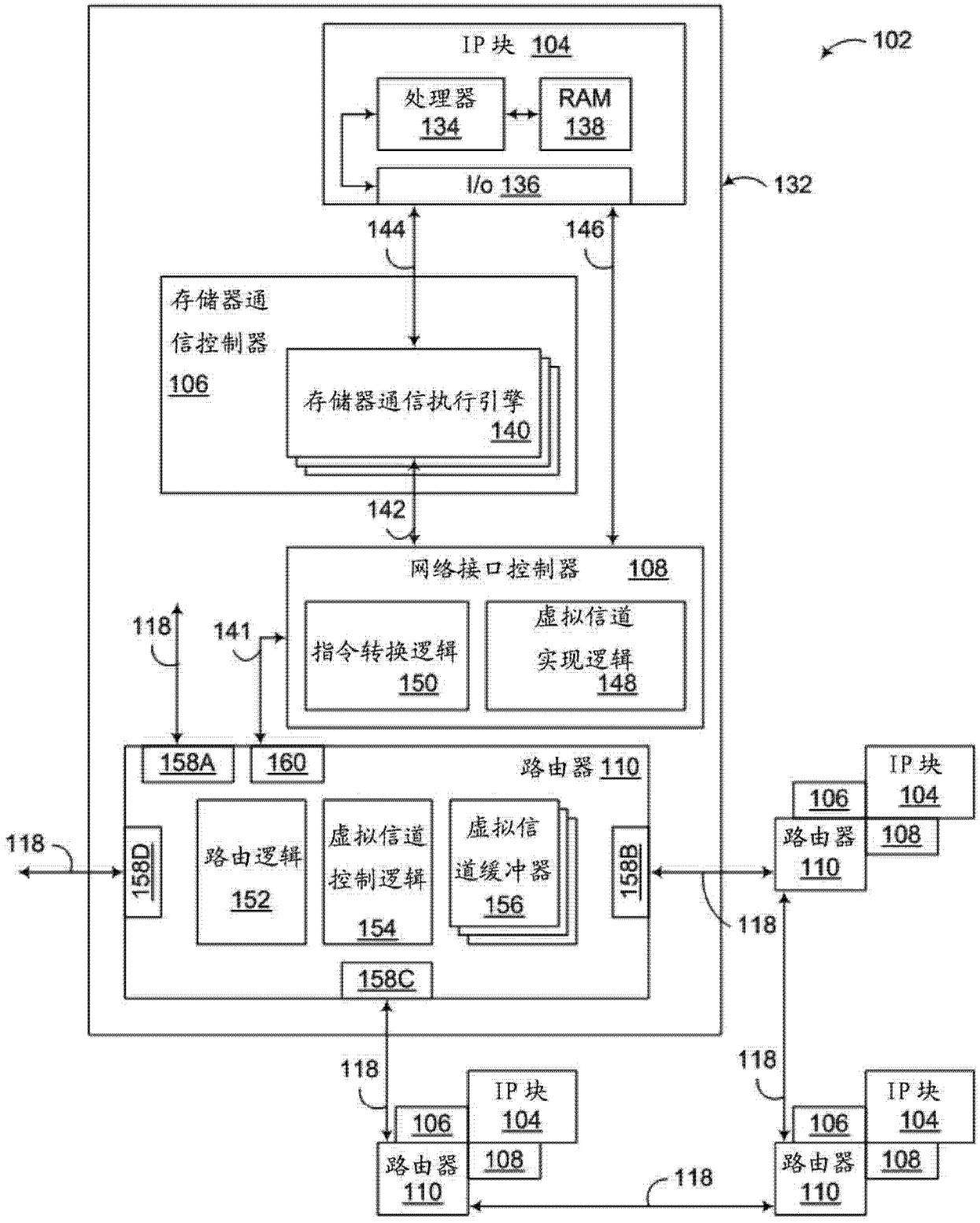

102
IP 块 104
处理器 134
RAM 138
I/o 136
132
144
146
存储器通信控制器 106
存储器通信执行引擎 140
142
网络接口控制器 108
118
141
指令转换逻辑 150
虚拟信道实现逻辑 148
158A
160
路由器 110
118
158D
路由逻辑 152
虚拟信道控制逻辑 154
虚拟信道缓冲器 156
158B
158C
118
IP 块 104
106
路由器 110
108
118
118
IP 块 104
106
路由器 110
108
118
IP 块 104
106
路由器 110
108

图 3

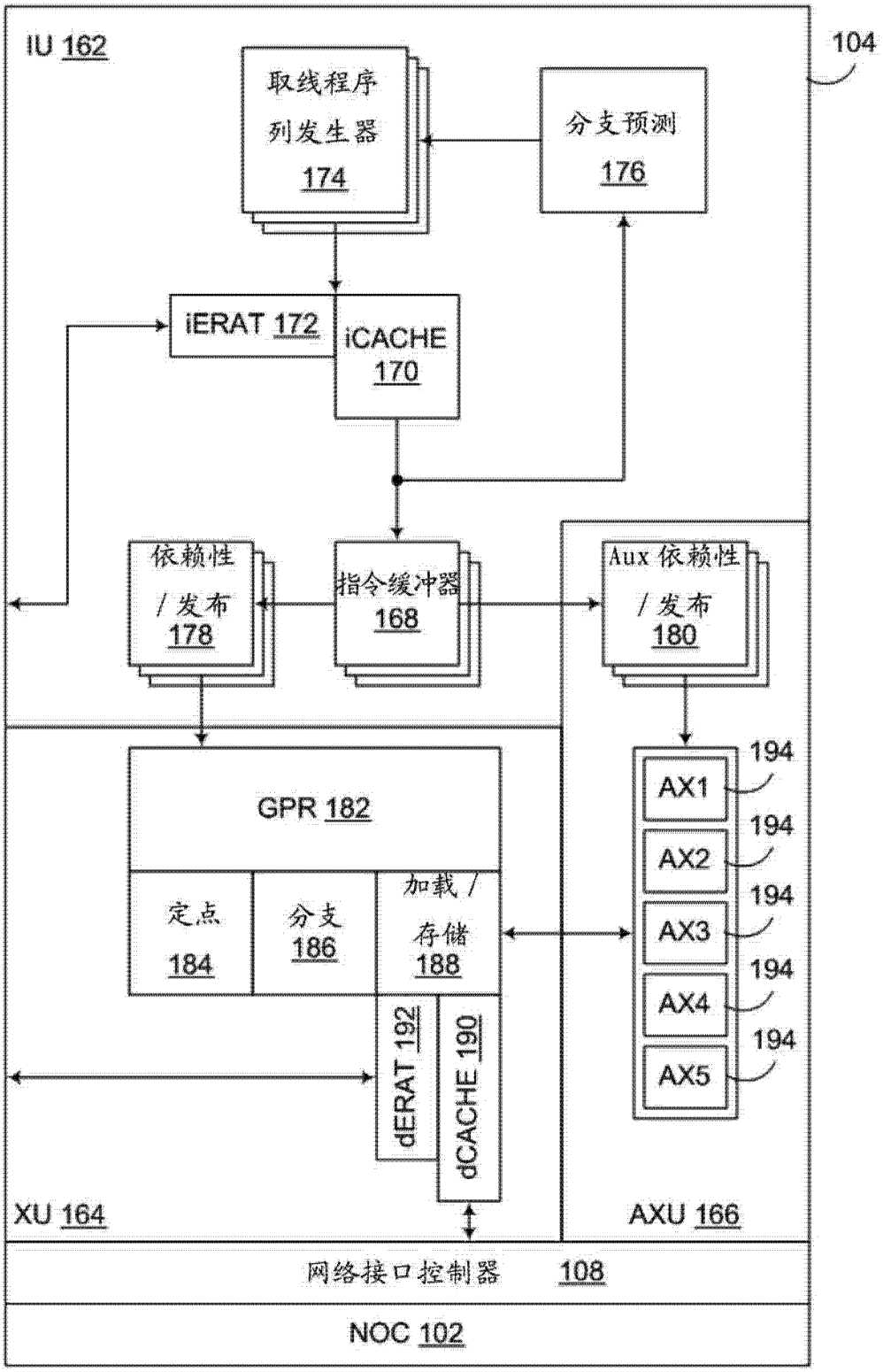

IU 162
104
取线程序列发生器 174
分支预测 176
iERAT 172
iCACHE 170
依赖性 / 发布 178
指令缓冲器 168
Aux 依赖性 / 发布 180
GPR 182
定点 184
分支 186
加载 / 存储 188
dERAT 192
dCACHE 190
AX1
AX2
AX3
AX4
AX5
194
XU 164
AXU 166
网络接口控制器 108
NOC 102

图 4

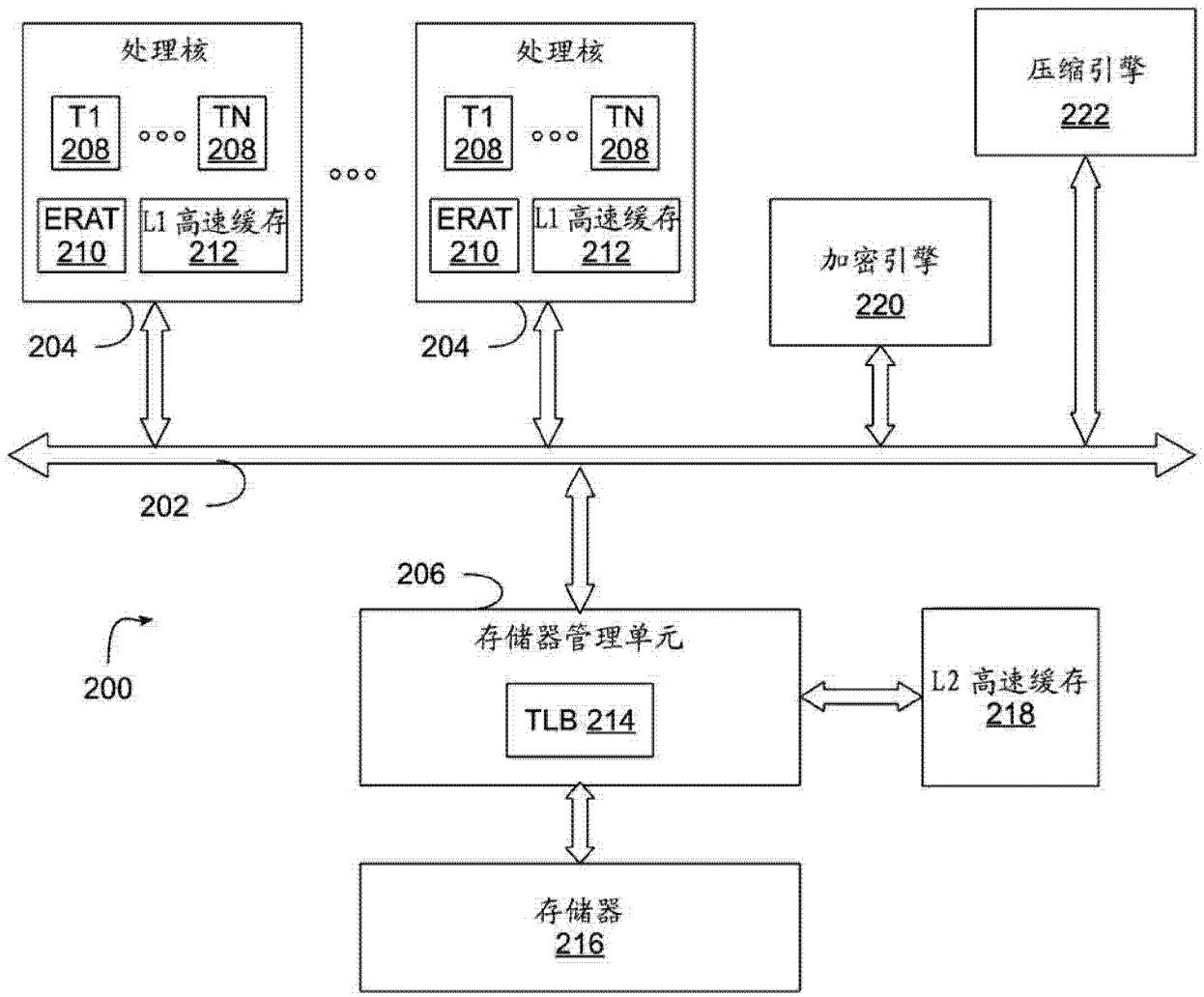

处理核
T1 208
TN 208
ERAT 210
L1 高速缓存 212
处理核
T1 208
TN 208
ERAT 210
L1 高速缓存 212
压缩引擎 222
加密引擎 220
204
204
202
206
200
存储器管理单元
TLB 214
L2 高速缓存 218
存储器 216

图 5


236
238
ENC
ENC LVL
CMP
CMP LVL
属性
用户
访问控制
ERAT
230
232
234
240
242
244
246

图 6

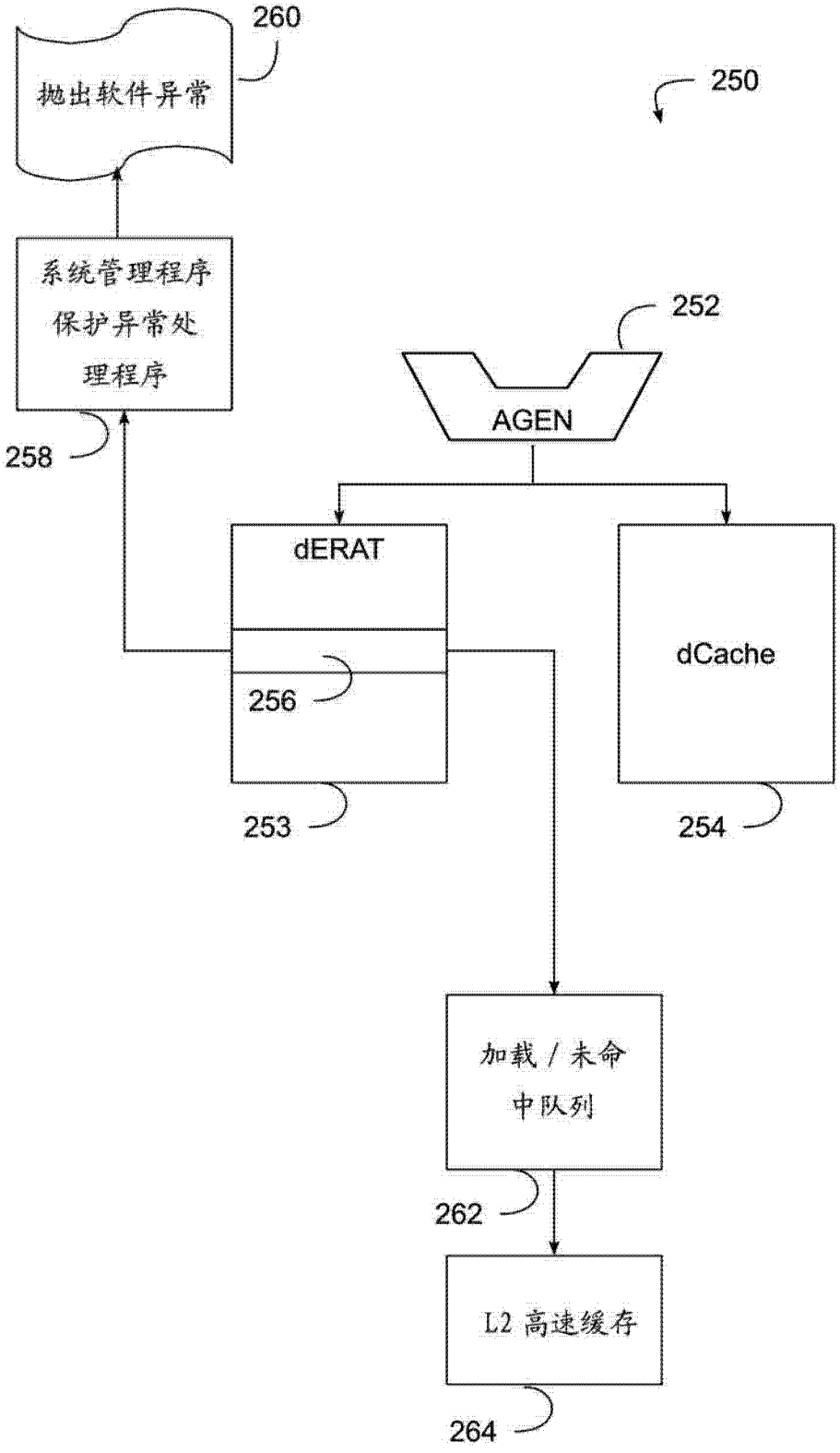

260
抛出软件异常
250
系统管理程序
保护异常处
理程序
252
AGEN
258
dERAT
dCache
256
253
254
加载 / 未命
中队列
262
L2 高速缓存
264

图 7

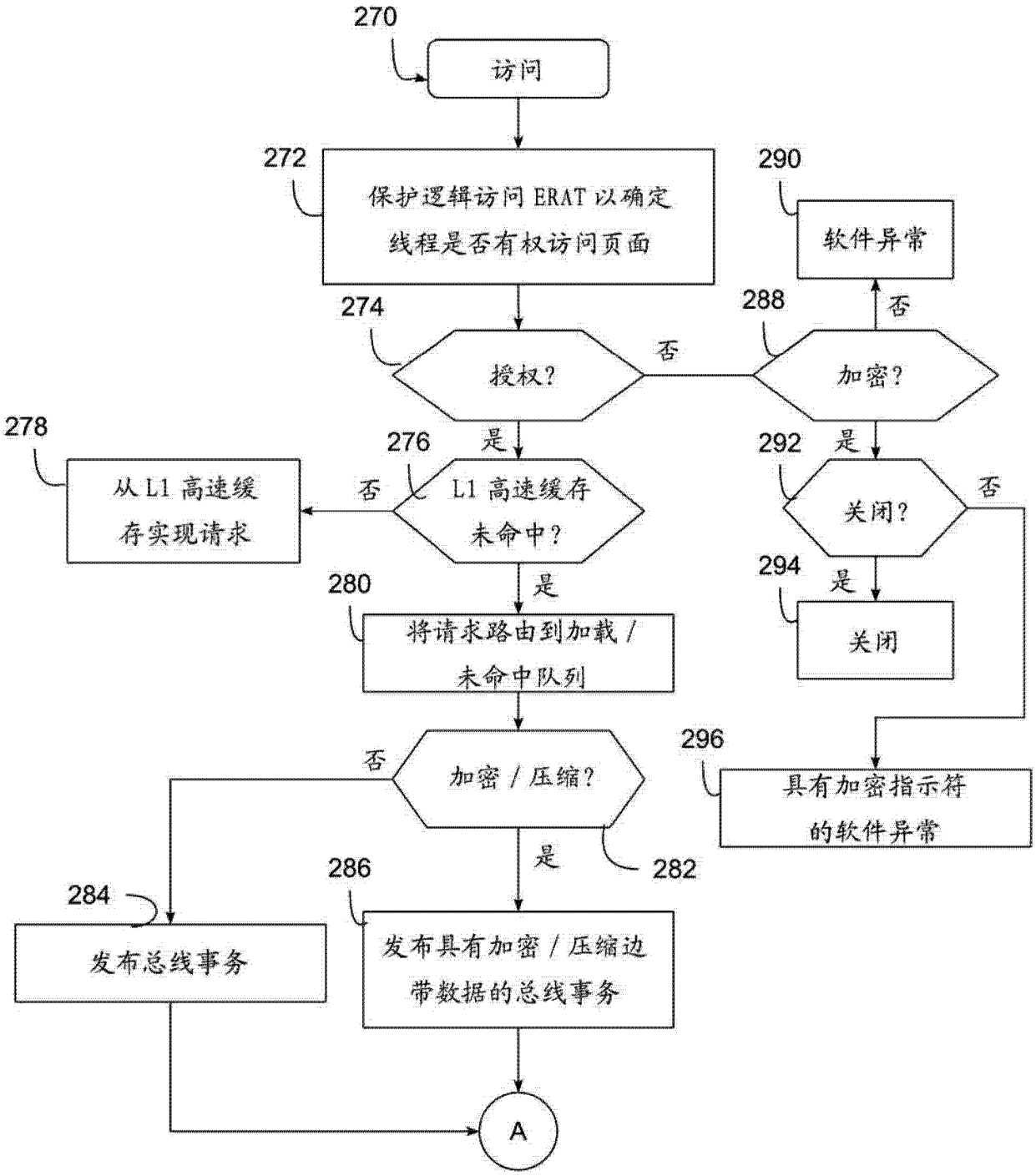

270
访问
272
保护逻辑访问 ERAT 以确定线程是否有权访问页面
274
授权?
否
是
278
从 L1 高速缓存实现请求
否
276
L1 高速缓存未命中?
是
280
将请求路由到加载 / 未命中队列
否
加密 / 压缩?
是
282
284
发布总线事务
286
发布具有加密 / 压缩边带数据的总线事务
A
290
软件异常
否
288
加密?
是
292
关闭?
否
是
294
关闭
296
具有加密指示符的软件异常

图 8

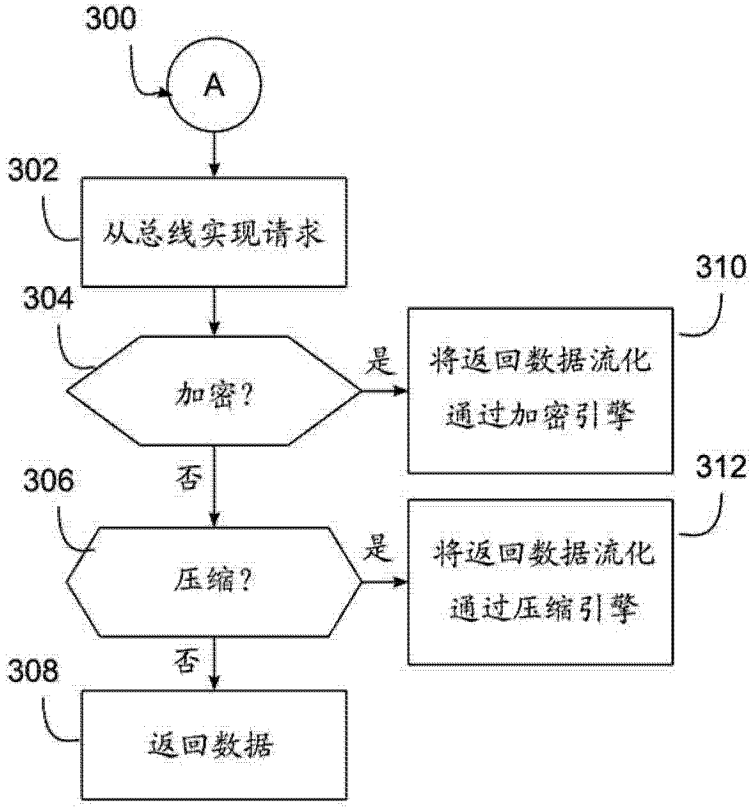

300
A
302
从总线实现请求
304
加密?
是
310
将返回数据流化
通过加密引擎
否
306
压缩?
是
312
将返回数据流化
通过压缩引擎
否
308
返回数据

图 9

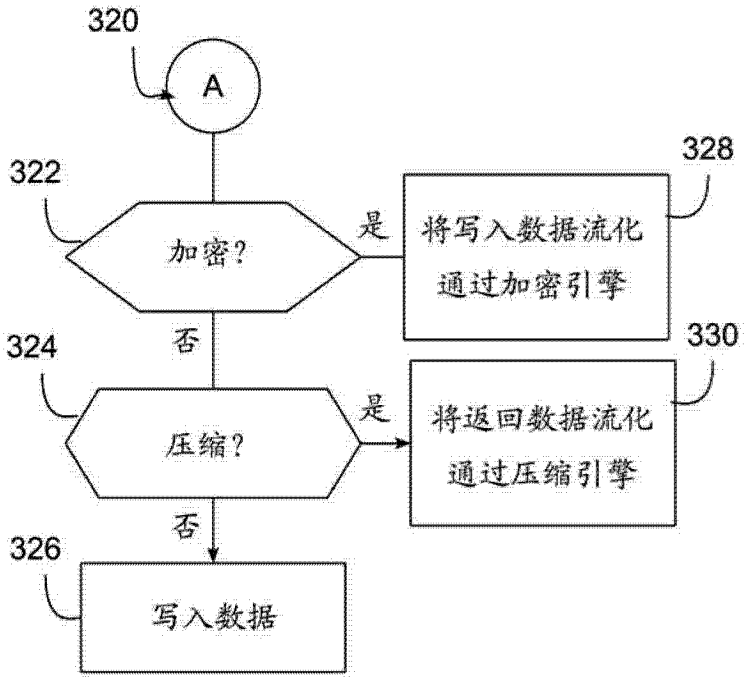

320
A
322
加密?
是
328
将写入数据流化
通过加密引擎
否
324
压缩?
是
330
将返回数据流化
通过压缩引擎
否
326
写入数据

图 10

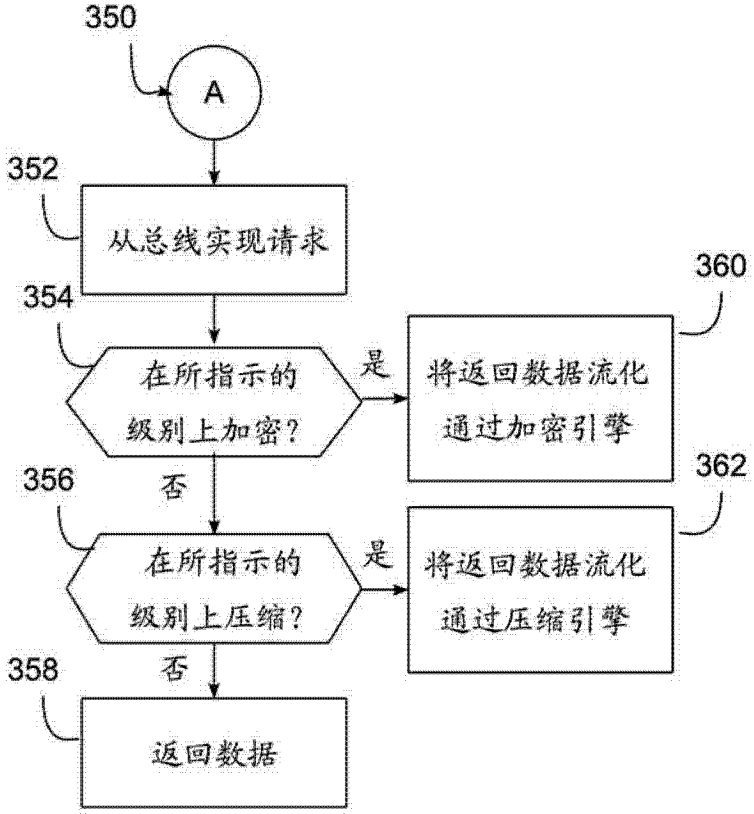

350
A
352
从总线实现请求
354
在所指示的级别上加密？
是
360
将返回数据流化通过加密引擎
否
356
在所指示的级别上压缩？
是
362
将返回数据流化通过压缩引擎
否
358
返回数据

图 11

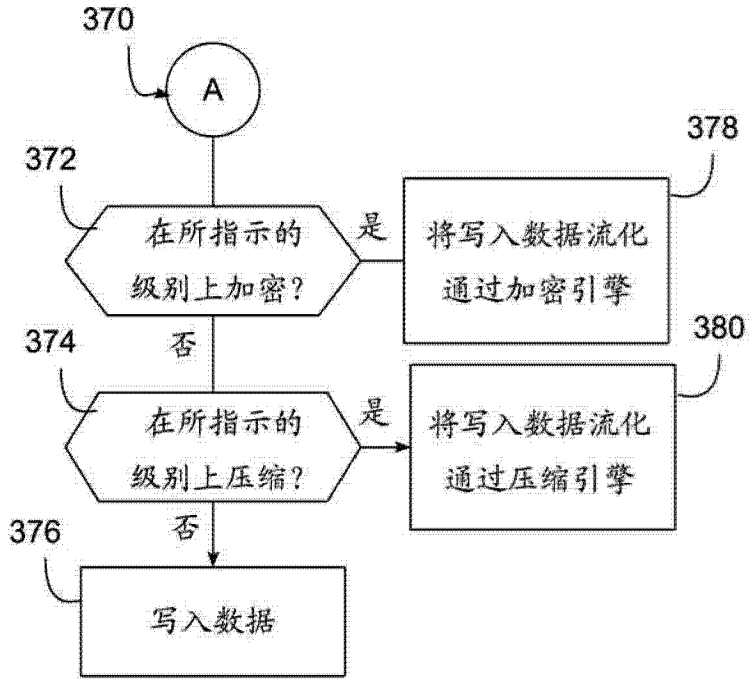

370
A
372
在所指示的级别上加密？
是
378
将写入数据流化通过加密引擎
否
374
在所指示的级别上压缩？
是
380
将写入数据流化通过压缩引擎
否
376
写入数据

图 12

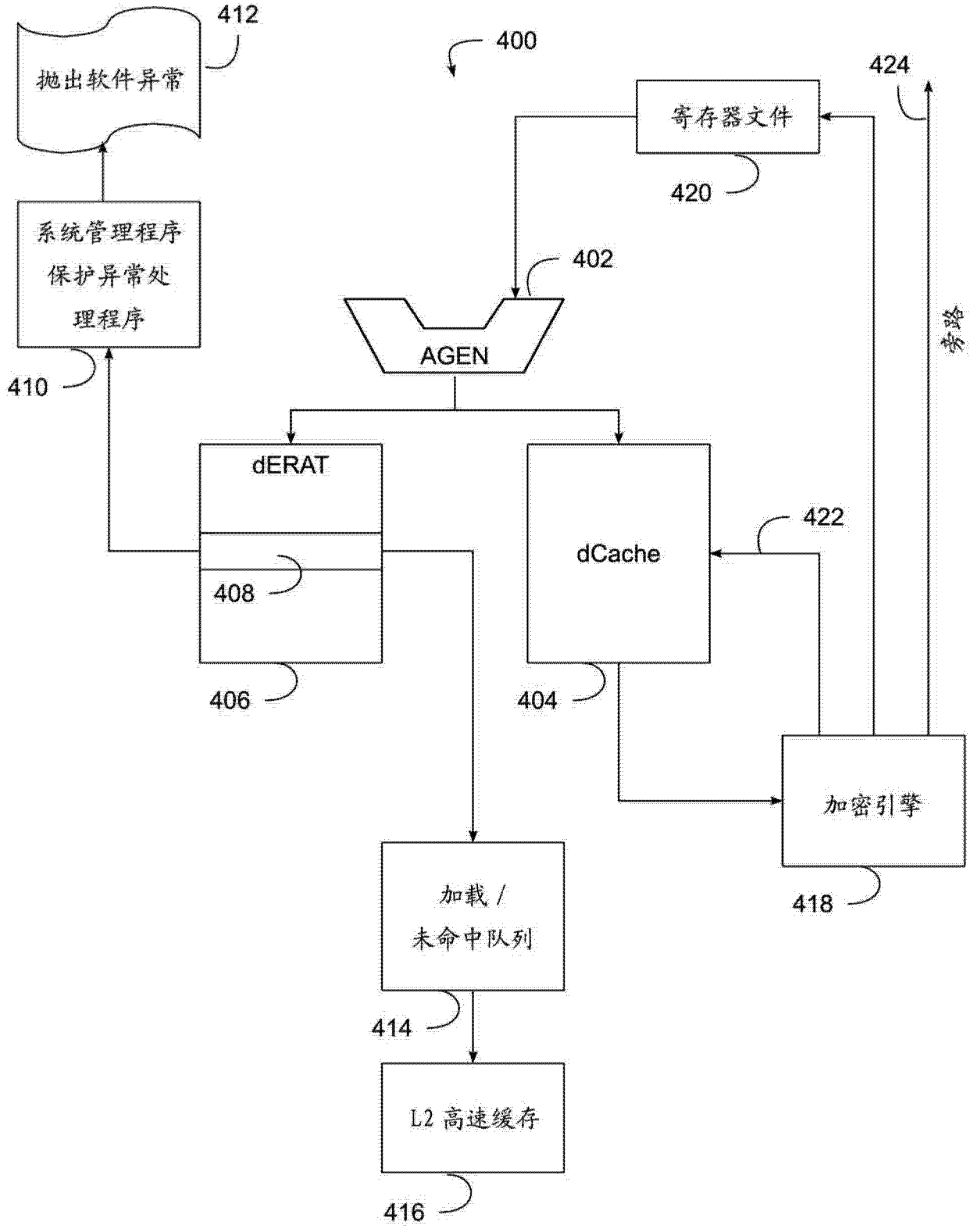

412
抛出软件异常
400
424
寄存器文件
420
系统管理程序
保护异常处
理程序
402
AGEN
旁路
410
dERAT
dCache
422
408
406
404
加密引擎
418
加载 /
未命中队列
414
L2 高速缓存
416

图 13

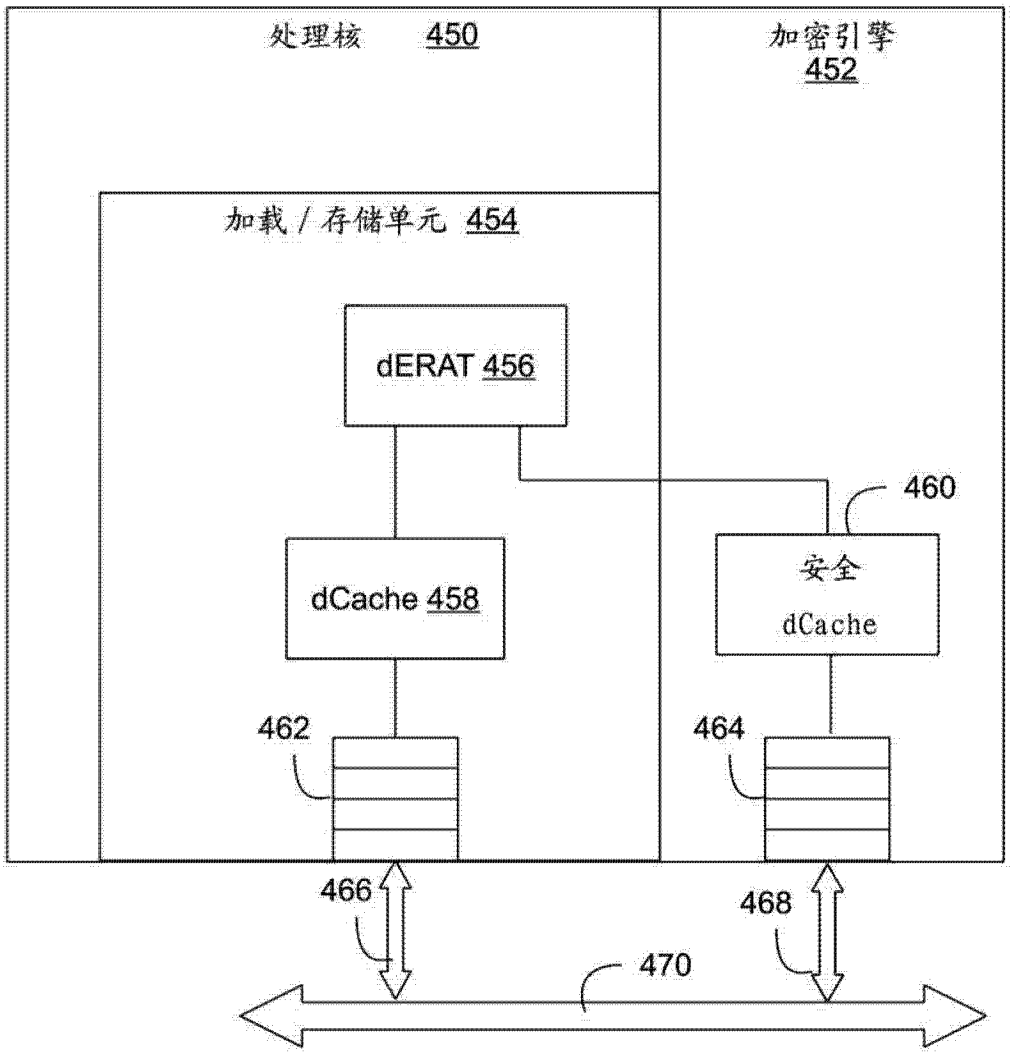

处理核 450
加密引擎 452
加载 / 存储单元 454
dERAT 456
dCache 458
460
安全 dCache
462
464
466
468
470

图 14