



(12) 发明专利

(10) 授权公告号 CN 102193886 B

(45) 授权公告日 2014. 07. 30

(21) 申请号 201010185538. 6

W0 9917212 A1, 1999. 04. 08, 全文.

(22) 申请日 2010. 05. 19

CN 101317166 A, 2008. 12. 03, 说明书第 4 页第 15 行至第 24 行、第 9 页第 6 行至第 12 页第 25 行、第 15 页第 27 行至第 17 页第 23 行、第 34 页第 17 行至第 35 页第 9 行, 图 2、3.

(30) 优先权数据

12/718, 279 2010. 03. 05 US

JP 63-18452 A, 1988. 01. 26, 全文.

(73) 专利权人 LSI 公司

US 2007088864 A1, 2007. 04. 19, 全文.

地址 美国加利福尼亚

US 6898646 B1, 2005. 05. 24, 全文.

(72) 发明人 G·皮奇里洛 D·M·奥尔森

审查员 李乐

(74) 专利代理机构 中国国际贸易促进委员会专利商标事务所 11038

代理人 申发振

(51) Int. Cl.

G06F 13/28 (2006. 01)

(56) 对比文件

US 6161165 A, 2000. 12. 12, 说明书第 9 栏第 15 行至第 11 栏第 67 行, 图 4.

US 2009248910 A1, 2009. 10. 01, 全文.

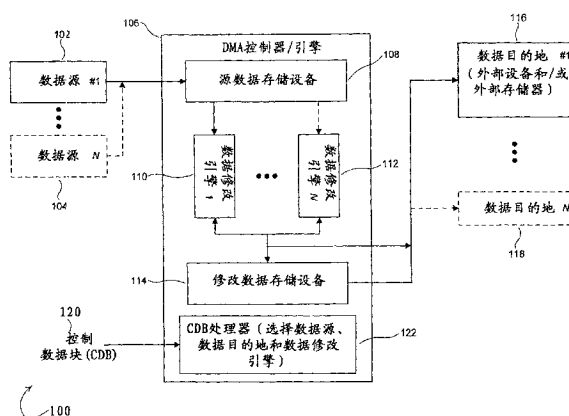
权利要求书3页 说明书11页 附图10页

(54) 发明名称

能够进行并发数据操纵的 DMA 引擎

(57) 摘要

本公开涉及能够进行并发数据操纵的 DMA 引擎。公开了在通过以 DMA 控制器 / 引擎管理的 DMA 通道传输的数据上并发地执行多个数据操纵操作的方法和设备。可以由 DMA 控制器获取控制从何处检索数据、将数据传递到何处和如何执行多个数据操纵操作的控制数据块 (CDB)。工作于 DMA 控制器内的 CDB 处理器可以读取 CDB, 并根据 CDB 的内容建立数据读、数据操纵操作和数据写。数据被从一个或多个源提供, 并且数据 / 修改后的数据可被传递到一个或多个目的地。在数据被通过 DMA 控制器传递时, DMA 控制器可以对数据并发地执行多个数据操纵操作。执行数据操纵操作的数据修改引擎可实现在 DMA 控制器上, 从而在数据操纵操作过程中使用本地 RAM 以避免在数据操纵操作过程中访问外部存储器。



1. 一种对通过直接存储器访问 DMA 通道传输的原始数据执行多个数据修改操作的方法,所述 DMA 通道由 DMA 控制器管理,该方法包括以下步骤:

在所述 DMA 控制器内提供多个数据修改引擎,每个数据修改引擎执行各种数据修改操作中的至少一种;

由所述 DMA 控制器获取控制数据块 CDB,所述 CDB 包含用于从至少一个数据源读所述原始数据、对所述原始数据执行所述多个数据修改操作、以及将包括所述原始数据和通过所述多个数据修改操作修改后的数据的数据写到至少一个目的地的指令;

由所述 DMA 控制器根据编码在所述 CDB 内的所述指令从至少一个数据源检索所述原始数据;

根据编码在所述 CDB 内的所述指令,使用为每种数据修改操作在所述 DMA 控制器上提供的所述多个数据修改引擎之一,在所述 DMA 控制器中对由所述 DMA 控制器接收的所述原始数据执行所述多个数据修改操作中的每个数据修改操作,从而由所述 DMA 控制器并发地执行所述多个数据修改操作中的至少两个,并且所述多个数据修改引擎中的第一数据修改引擎创建第一数据结果,所述第一数据结果用作所述多个数据修改引擎中的第二数据修改引擎的计算的基础以便执行所述多个数据修改操作中的至少一个数据修改操作,所述第一数据修改引擎和所述第二数据修改引擎是在所述 DMA 控制器内提供的所述多个数据修改引擎中的任何数据修改引擎;和

根据编码在所述 CDB 内的所述指令,由所述 DMA 控制器将所述多个数据修改操作的结果发送到至少一个目的地。

2. 如权利要求 1 所述的方法,其中所述从所述至少一个数据源检索所述原始数据的步骤进一步包括:

根据编码在所述 CDB 内的所述指令,在所述 DMA 控制器内创建读命令,以便从所述至少一个数据源读取所述原始数据;

将所述读命令从所述 DMA 控制器发送到所述至少一个数据源;和

在所述 DMA 控制器处接收由所述至少一个数据源根据所述读命令发送的来自所述至少一个数据源的所述原始数据。

3. 如权利要求 1 所述的方法,其中所述将所述多个数据修改操作的所述结果发送到至少一个目的地的步骤进一步包括:

根据编码在所述 CDB 内的所述指令,在所述 DMA 控制器内创建写命令,以便将所述多个数据修改操作的所述结果写到所述至少一个目的地;和

根据所述写命令,由所述 DMA 控制器将所述多个数据修改操作的所述结果发送到所述至少一个目的地。

4. 如权利要求 1 所述的方法,还包括以下步骤:将由所述 DMA 控制器检索到的所述原始数据存储在本地地包括在所述 DMA 控制器内的电子存储器内。

5. 如权利要求 1 所述的方法,还包括以下步骤:将所述多个数据修改操作的所述结果存储在本地地包括在所述 DMA 控制器内的电子存储器内。

6. 如权利要求 1 所述的方法,其中所述多个数据修改引擎中的每个数据修改引擎执行一组数据修改操作中的至少一个数据修改操作,所述一组数据修改操作包括:散列、散列消息鉴别码 HMAC、散列 /HMAC 组合、填充模式、线性反馈移位寄存器 LFSR、端到端数据保护

EEDP 检查、EEDP 添加、EEDP 更新、EEDP 去除、异或 XOR、加密和解密。

7. 如权利要求 1 所述的方法,其中所述多个数据修改引擎包括至少两个单一类型的数据修改操作,从而能够在所述 DMA 控制器上并发地执行所述单一类型的数据修改操作。

8. 如权利要求 1 所述的方法,其中所述将所述多个数据修改操作的所述结果发送到至少一个目的地的步骤还将由所述 DMA 控制器检索到的所述原始数据发送到所述至少一个数据目的地。

9. 一种直接存储器访问 DMA 控制器,该 DMA 控制器对通过直接存储器访问 DMA 通道传输的原始数据执行多个数据修改操作,所述 DMA 通道由所述 DMA 控制器管理,所述 DMA 控制器包括:

控制数据块 CDB 处理器子系统,该 CDB 处理器子系统获取控制数据块 CDB,所述 CDB 包含用于从至少一个数据源读所述原始数据、对所述原始数据执行所述多个数据修改操作、以及将包括所述原始数据和通过所述多个数据修改操作修改后的数据的数据写到至少一个目的地的指令;

填充子系统,该填充子系统根据编码在所述 CDB 内的所述指令从至少一个数据源检索所述原始数据;

所述 DMA 控制器内的多个数据修改引擎,每个数据修改引擎根据编码在所述 CDB 内的所述指令,对由所述填充子系统接收的所述原始数据执行针对所述多个数据修改操作中的每个数据修改操作的各种数据修改操作中的至少一种,从而由所述 DMA 控制器并发地执行所述多个数据修改操作中的至少两个,并且所述多个数据修改引擎中的第一数据修改引擎创建第一数据结果,所述第一数据结果用作所述多个数据修改引擎中的第二数据修改引擎的计算的基础以便执行所述多个数据修改操作中的至少一个数据修改操作,所述第一数据修改引擎和所述第二数据修改引擎是在所述 DMA 控制器内提供的所述多个数据修改引擎中的任何数据修改引擎;和

排出子系统,排出子系统根据编码在所述 CDB 内的所述指令,将所述多个数据修改操作的结果发送到至少一个目的地。

10. 如权利要求 9 所述的 DMA 控制器,其中所述填充子系统通过根据编码在所述 CDB 内的所述指令创建读命令来从所述至少一个数据源读取所述原始数据,将所述读命令发送到所述至少一个数据源,以及接收由所述至少一个数据源根据所述读命令发送的来自所述至少一个数据源的所述原始数据,而从所述至少一个数据源检索所述原始数据。

11. 如权利要求 9 所述的 DMA 控制器,其中所述排出子系统通过根据编码在所述 CDB 内的所述指令创建写命令来将所述多个数据修改操作的所述结果写到至少一个目的地,以及根据所述写命令将所述多个数据修改操作的所述结果发送到所述至少一个目的地,而将所述多个数据修改操作的所述结果发送到所述至少一个目的地。

12. 如权利要求 9 所述的 DMA 控制器,还包括存储由所述填充子系统检索到的所述原始数据的源数据存储设备,所述源数据存储设备是本地地包括在所述 DMA 控制器内的电子存储器。

13. 如权利要求 9 所述的 DMA 控制器,还包括存储所述多个数据修改操作的所述结果的至少一个修改数据存储设备,所述至少一个修改数据存储设备是本地地包括在所述 DMA 控制器内的电子存储器。

14. 如权利要求9所述的DMA控制器,其中所述多个数据修改引擎中的每个数据修改引擎执行一组数据修改操作中的至少一个数据修改操作,所述一组数据修改操作包括:散列、散列消息鉴别码 HMAC、散列/HMAC 组合、填充模式、线性反馈移位寄存器 LFSR、端到端数据保护 EEDP 检查、EEDP 添加、EEDP 更新、EEDP 去除、异或 XOR、加密和解密。

15. 如权利要求9所述的DMA控制器,其中所述多个数据修改引擎包括至少两个单一类型的数据修改操作,从而能够在所述DMA控制器上并发地执行所述单一类型的数据修改操作。

16. 如权利要求9所述的DMA控制器,其中所述排出子系统还将由所述DMA控制器检索到的所述原始数据发送到所述至少一个数据目的地。

17. 一种直接存储器访问DMA控制器,该DMA控制器对通过直接存储器访问DMA通道传输的原始数据执行多个数据修改操作,所述DMA通道由所述DMA控制器管理,所述DMA控制器包括:

用于在所述DMA控制器内提供多个数据修改引擎的装置,每个数据修改引擎执行各种数据修改操作中的至少一种;

用于由所述DMA控制器获取控制数据块CDB的装置,所述CDB包含用于从至少一个数据源读所述原始数据、对所述原始数据执行所述多个数据修改操作、以及将包括所述原始数据和通过所述多个数据修改操作修改后的数据的数据写到至少一个目的地的指令;

用于由所述DMA控制器根据编码在所述CDB内的所述指令从至少一个数据源检索所述原始数据的装置;

用于根据编码在所述CDB内的所述指令使用为每种数据修改操作在所述DMA控制器上提供的所述多个数据修改引擎之一在所述DMA控制器中对由所述DMA控制器接收的所述原始数据执行所述多个数据修改操作中的每个数据修改操作的装置,从而由所述DMA控制器并发地执行所述多个数据修改操作中的至少两个,并且所述多个数据修改引擎中的第一数据修改引擎创建第一数据结果,所述第一数据结果用作所述多个数据修改引擎中的第二数据修改引擎的计算的基础以便执行所述多个数据修改操作中的至少一个数据修改操作,所述第一数据修改引擎和所述第二数据修改引擎是在所述DMA控制器内提供的所述多个数据修改引擎中的任何数据修改引擎;和

用于由所述DMA控制器根据编码在所述CDB内的所述指令将所述多个数据修改操作的结果发送到至少一个目的地的装置。

能够进行并发数据操纵的 DMA 引擎

技术领域

[0001] 本发明涉及直接存储器访问 (DMA)，更具体地涉及能够进行并发数据操纵的 DMA 引擎。

背景技术

[0002] 直接存储器访问 (DMA) 是现代计算机的一种基本特征。DMA 允许计算机的特定硬件子系统能够独立于中央处理单元 (CPU) 读访问和 / 或写访问系统存储器。可以使用 DMA 的某些示例硬件系统包括但不限于硬盘驱动器控制器、芯片上的 RAID (独立磁盘冗余阵列) (ROC) 控制器、图形卡、网卡和声卡。DMA 还可用于多核心处理器上的芯片内数据传输。通常由 DMA 控制器执行 DMA 通道的管理和实现。很多时候 DMA 控制器配备有本地存储器，从而 DMA 控制器向 / 从本地 DMA 存储器和外部主存储器传输数据。由于 DMA 控制器而不是 CPU 管理数据传输，使用 DMA 的数据传输使用很少的计算机 CPU 处理时间，因此增加了具有 DMA 控制器的计算机的有效计算能力。不使用 DMA，与外设的通信或多核心系统的核心之间的通信可能在整个读 / 写操作过程中完全占据 CPU，这使得 CPU 不能用于执行其它计算任务。使用 DMA，CPU 可以启动传输，并且然后在进行传输的同时进行其它操作，并且一旦操作完成，从 DMA 控制器接收中断。因为与外设通信通常比与常规系统随机访问存储器 (RAM) 慢，从而如果不使用由 DMA 控制器管理的 DMA 通道，在与外设通信过程中不能使用 CPU 的时间将会更长，所以 CPU 摆脱执行与外设的数据传输尤其重要。

[0003] 典型的 DMA 传输将一个存储器块从一个设备拷贝到另一个设备。CPU 启动数据传输，但是 CPU 自身不执行数据传输。对于工业标准体系结构 (ISA) 总线，由 DMA 控制器执行数据传输，DMA 控制器通常被结合为计算机主板芯片组的一部分。外设组件互连 (PCI) 总线使用总线主控 DMA，其中外设接管总线控制并且自己执行传输。

[0004] 嵌入式处理器可以在芯片内包括 DMA 引擎 / 控制器，以便允许处理元件发起数据传输，同时在数据传输期间继续执行其它任务。当 DMA 控制器被结合在芯片内时，DMA 控制器通常被称为 DMA 引擎。嵌入多核心的处理器通常包括一个或更多个 DMA 引擎和本地 DMA 存储器的组合作为芯片多核心处理器芯片内的子系统。

发明内容

[0005] 本发明的实施例可以包括一种对通过直接存储器访问 (DMA) 通道传输的数据执行多个数据操纵操作的方法，所述 DMA 通道由 DMA 控制器管理。该方法包括以下步骤：在 DMA 控制器中提供多个数据操纵引擎，每个数据操纵引擎执行各种数据操纵操作中的至少一种；由 DMA 控制器获取控制数据块 (CDB)，CDB 包含用于从至少一个数据源读数据、对该数据执行多个数据操纵操作、以及将数据写到至少一个目的地的指令；由 DMA 控制器根据编码在 CDB 内的指令从至少一个数据源检索数据；根据编码在 CDB 内的指令，使用为每种数据操纵操作在 DMA 控制器上提供的多个数据操纵引擎之一在 DMA 控制器中对由 DMA 控制器接收的数据执行多个数据操纵操作中的每个数据操纵操作，从而由 DMA 控制器并发地执行多

个数据操纵操作中的至少两个 ;和根据编码在 CDB 内的指令,由 DMA 控制器将多个数据操纵操作的结果发送到至少一个目的地。

[0006] 本发明的实施例还可以包括一种直接存储器访问 (DMA) 控制器,该 DMA 控制器对通过直接存储器访问 (DMA) 通道传输的数据执行多个数据操纵操作,所述 DMA 通道由 DMA 控制器管理。该 DMA 控制器包括 :控制数据块 (CDB) 处理器子系统,该 CDB 处理器子系统获取控制数据块 (CDB),CDB 包含用于从至少一个数据源读数据、对数据执行多个数据操纵操作、以及将数据写到至少一个目的地的指令 ;填充子系统,该填充子系统根据编码在 CDB 内的指令从至少一个数据源检索数据 ;DMA 控制器内的多个数据操纵引擎,每个数据操纵引擎根据编码在 CDB 内的指令,对由填充子系统接收的数据执行针对多个数据操纵操作中的每个数据操纵操作的各种数据操纵操作中的至少一种,从而由 DMA 控制器并发地执行多个数据操纵操作中的至少两个 ;和排出子系统,排出子系统根据编码在 CDB 内的指令,将多个 DMA 控制器的结果发送到至少一个目的地。

[0007] 本发明的实施例还可以包括一种直接存储器访问 (DMA) 控制器,该 DMA 控制器对通过直接存储器访问 (DMA) 通道传输的数据执行多个数据操纵操作,所述 DMA 通道由所述 DMA 控制器管理。所述 DMA 控制器包括 :用于在 DMA 控制器内提供多个数据操纵引擎的装置,每个数据操纵引擎执行各种数据操纵操作中的至少一种 ;用于由 DMA 控制器获取控制数据块 (CDB) 的装置,CDB 包含用于从至少一个数据源读数据、对该数据执行多个数据操纵操作、以及将数据写到至少一个目的地的指令 ;用于由 DMA 控制器根据编码在 CDB 内的指令从至少一个数据源检索数据的装置 ;用于根据编码在 CDB 内的指令使用为每种数据操纵操作在 DMA 控制器上提供的多个数据操纵引擎之一在 DMA 控制器中对以 DMA 控制器接收的数据执行多个数据操纵操作中的每个数据操纵操作的装置,从而由 DMA 控制器并发地执行多个数据操纵操作中的至少两个 ;和用于由 DMA 控制器根据编码在 CDB 内的指令将多个数据操纵操作的结果发送到至少一个目的地的装置。

附图说明

[0008] 在附图中 :

[0009] 图 1 是具有被根据控制数据块 (CDB) 管理的多个数据修改引擎的直接存储器访问 (DMA) 控制器系统的实施例的框图 ;

[0010] 图 2 是具有端到端数据保护 (EEDP) 检查和更新周期的 DMA 控制器系统的实施例的框图 ;

[0011] 图 3 是对数据加密并且执行端到端数据保护 (EEDP) 检查的 DMA 控制器系统的实施例的框图 ;

[0012] 图 4 是对数据加密并且具有端到端数据保护 (EEDP) 检查和更新周期的 DMA 控制器系统的实施例的框图 ;

[0013] 图 5 是对数据加密、对数据执行异或 (XOR)、并且执行端到端数据保护 (EEDP) 检查的 DMA 控制器系统的实施例的框图 ;

[0014] 图 6 是对数据加密、对数据执行异或 (XOR)、并且具有端到端数据保护 (EEDP) 检查和更新周期的 DMA 控制器系统的实施例的框图 ;

[0015] 图 7 是对数据执行异或 (XOR)、并且具有端到端数据保护 (EEDP) 检查和更新周期

的 DMA 控制器系统的实施例的框图；

[0016] 图 8 是包括 DMA 引擎 / 控制器的芯片上的 RAID (独立磁盘冗余阵列) (ROC) 系统的实施例的框图；

[0017] 图 9 是 DMA 控制器的实施例的控制数据块 (CDB) 处理的实施例的框图；

[0018] 图 10 是 CDB 处理器实施例的 CDB 处理的有限状态机 (FSM) 操作的框图。

具体实施方式

[0019] 随着现代计算机系统变得更为复杂,期望对在计算机系统内传输的数据和被传输到连接到计算机系统的外设的数据执行附加的数据操纵操作。例如,作为 RAID (独立磁盘冗余阵列) (ROC) 系统操作的一部分,可以期望芯片上的 ROC 系统对正被传输的数据执行若干数据操纵操作。ROC 系统通常包括直接存储器访问 (DMA) 引擎 / 控制器作为 ROC 体系结构内的子系统。某些典型的数据操纵操作包括:散列、散列消息鉴别码 (HMAC)、散列 /HMAC 组合、填充模式、线性反馈移位寄存器 (LFSR)、端到端数据保护 (EEDP) 检查 / 添加 / 更新 / 去除、异或 (XOR)、加密和解密。一个实施例的 DMA 引擎 / 控制器可以合并对在 DMA 引擎 / 控制器内传输的数据执行的各种数据操纵操作,以便减少对计算机系统内的其它系统 / 子系统的处理要求。另外,一个实施例的 DMA 引擎 / 控制器可以并发地对正被传输的数据执行多种数据操纵,从而比以串行方式每次一个地执行多个数据操纵操作更快地执行多种数据操纵。因此,一个实施例允许以尽可能少的中断将数据自由地从源移动到目的地。对正被一个实施例的 DMA 引擎 / 控制器传输的数据执行的数据操纵操作还可以在位于 DMA 引擎 / 控制器的存储器内存储和检索数据 / 修改后的数据 (即,本地存储器),从而避免访问 DMA 引擎 / 控制器之外的存储器 (即,外部存储器) 的不必要的开销。

[0020] 例如,一个实施例的 DMA 引擎 / 控制器可以使用 DMA 引擎 / 控制器上的加密引擎加密并且在位于本地 DMA 存储器内的第一数据存储器内存储数据。在加密数据被置于第一数据缓冲器内的同时,DMA 引擎 / 控制器上的 XOR 引擎可以对加密数据并发地执行 XOR 操作,并且在位于本地 DMA 存储器内的第二数据缓冲器内存储 XOR 数据,而不会对到第一数据缓冲器的加密数据的数据流产生任何中断。通过在产生加密数据的同时对加密数据执行 XOR 操作,一个实施例可以避免在执行 XOR 操作之前等待加密数据被完整写入数据缓冲器的延迟。另外,一个实施例的 DMA 引擎 / 控制器还可以与加密和 XOR 数据操纵操作并行地执行 EEDP 检查、添加和 / 或去除操作,从而进一步减少在正被 DMA 引擎 / 控制器传输的数据上执行多个数据操纵操作所涉及的延迟。

[0021] 可以在具有 DMA 引擎 / 控制器的各种计算机系统和电子设备中的 DMA 引擎 / 控制器上实现各种实施例。如果希望,如果需要多个 DMA 通道,计算机系统和 / 或电子设备可以包括多个 DMA 引擎 / 控制器。另外,某些 DMA 引擎 / 控制器可以提供多个 DMA 通道,从而在系统中包括多个 DMA 引擎 / 控制器导致乘法器效果,其中可获得的 DMA 通道的数目是 DMA 引擎 / 控制器的数目乘以每个 DMA 引擎 / 控制器上可获得的 DMA 通道的数目。一个实施例的 DMA 控制器可被实现为单独系统,这些单独系统可被作为单独的专用计算机“卡”、单独的专用芯片或作为单独的专用电子设备结合在计算机系统和 / 或电子设备内。然而,DMA 控制器通常被结合为集成在计算机系统或电子设备内的较大多功能芯片、卡或电路的子系统或引擎。因此,一个实施例的 DMA 控制器可被实现为被包括在较大多功能芯片、卡或电路内的

DMA 引擎子系统。通常,当 DMA 控制器被作为较大多功能芯片、卡或电路的子系统包括时,DMA 控制器被称为 DMA 引擎。因此,在本文档中,术语 DMA 控制器和 DMA 引擎被可互换地使用,从而当指代 DMA 控制器时也指代 DMA 引擎,并且反之亦然。

[0022] 图 1 是具有被根据控制数据块 (CDB) 120 管理的多个数据修改引擎 110-112 的 DMA 控制器系统 106 的实施例的框图 100。在图 1 所示的实施例中,由 DMA 控制器 / 引擎 106 在 DMA 控制器 106 的 CDB 处理器 122 子系统处获取和接收 CDB120。CDB 处理器子系统 122 在 DMA 控制器 106 上操作,并且被实现为 DMA 控制器 106 的一部分。CDB 包含用于数据传输的指令,且包括正被传输 / 修改的数据的一个或更多数据源 102-104、希望对正被传输的数据进行的数据操纵操作、和正被传输的数据的一个或更多数据目的地 116-118。CDB 处理器 122 读取编码在 CDB120 内的指令,并且根据编码在 CDB120 内的指令指挥 DMA 控制器 106 建立并且进行数据传输。基于 CDB,DMA 控制器 106 可以准备并且向一个或更多数据源 102-104 发出读命令,以便使得一个或更多数据源 102-104 向 DMA 控制器 106 发送所希望的数据。来自一个或更多数据源 102-104 的数据被 DMA 控制器 106 接收,并且被存储在位于 DMA 控制器 106 上的源数据存储设备 108 内。基于 CDB120,DMA 控制器 106 可以选择多个数据修改引擎 110-112,以便对正被通过 DMA 控制器 106 传输的数据执行所希望的数据修改操作。来自数据修改引擎 110-112 的结果数据可被直接传递到数据目的地 116-118,或结果数据可被存储在位于 DMA 控制器 106 上的修改数据存储设备 114 内。存储在 DMA 控制器 106 的修改数据存储设备 114 内的来自数据修改引擎 110-112 的结果数据可被其它数据修改引擎 110-112 或原始数据修改引擎 (例如,执行与过去结果相关的操作的数据修改引擎) 使用,从而可由数据修改引擎 110-112 读 / 写修改数据存储设备 114。一旦完成了所希望的数据修改操作,DMA 控制器可将数据 / 修改后的数据传递到所希望的数据目的地 116-118。在某些情况下,数据修改引擎 110-112 的结果可被仅仅作为由其它数据修改引擎 110-112 使用的中间结果。根据编码在 CDB120 中的指令实现数据修改引擎 110-112 的配置,以及从数据修改引擎 110-112 传递哪些结果数据。

[0023] 对于一个实施例,一个或更多数据源 102-104 可以是与典型的 DMA 控制器 / 引擎 106 兼容的任何数据源 102-104。一个实施例可以从单个数据源 102 或从多个数据源 102-104 收集 / 读取数据。例如,可以从外部存储器 (即, DMA 控制器 106 之外的存储器) 读取来自单个存储器区域的源数据,并且将其传输到各种数据目的地 116-118。类似地,可以从外部读取来自多个存储器区域的源数据,并且在一个数据传输操作中将其传输到各种数据目的地 116-118。

[0024] 可以创建数据修改引擎,以便对正被通过 DMA 控制器 / 引擎 106 传输的数据执行所希望的数据修改。例如,某些数据修改引擎可以包括 (但不限于): 散列、HMAC、散列 / HMAC 组合、填充模式、线性反馈移位寄存器 (LFSR)、端到端数据保护 (EEDP) 检查 / 添加 / 更新 / 去除、异或 (XOR)、加密和解密。支持的散列算法的某些类型包括可由散列引擎单独或并发执行的标准安全散列算法 (SHA)-224、SHA-256、SHA-384 和 SHA-512 算法。用于数据修改引擎 110-112 的输入数据可被指定为是来自另一个数据修改引擎 110-112 的中间数据结果,以便允许组合在单个结果中的多个数据操纵。如果希望,中间结果还可被编码在 CDB120 内发送到目的地位置 116-118 之一。然而,中间数据结果不是必须被发送到数据目的地 116-118。当工作于来自另一个数据操纵引擎的数据时,第二个数据操纵引擎可以与第

一个数据操纵引擎并发地操作,但是第二数据操纵可以略微晚于第一数据操纵引擎开始,以便允许在开始第二数据操纵引擎计算之前,第一数据操纵引擎开始流化发送(stream)中间结果数据。对于各种实施例,当不同数据操纵引擎 110-112 并行地在相同时刻对相同数据(中间结果和/或原始接收数据)执行操作时,也可以发生数据修改引擎 110-112 的并发操作。另外,各种实施例可以将原始数据和修改后的数据结果两者传递到数据目的地 116-118。

[0025] 各种实施例可以将 DMA 控制器 106 内的本地源数据存储设备 108 和本地修改数据存储设备 114 实现为一个或更多个本地电子存储器电路。电子存储器还可被称为计算机可读存储器,虽然计算机可读(即,电子)存储器可被包括在需要存储器存储设备的电子设备(但是通常不被认为是计算机)内。电子存储器的典型实现为提供可被 DMA 控制器 106 划分为源数据存储设备 108 和修改数据存储设备 114 的一组电子随机访问存储器(RAM)。电子存储器可以是用于源数据存储设备 108 和修改数据存储设备 114 的固定部分,但是 DMA 控制器还可以按照源数据存储设备和修改数据存储设备 114 的需要动态地分配电子存储器 RAM,从而可以更有效地利用整个电子存储器。另外,修改数据存储设备 114 可被细分为多个段,以便存储来自多个数据修改引擎 110-112 的数据。

[0026] 一个实施例的数据目的地 116-118 可以是与典型的 DMA 控制器/引擎 106 兼容的任意数据目的地 116-118。各种实施例可以向多个数据目的地 116-118 或向单个数据目的地 116 写数据和/或修改后的数据结果。各个数据结果可被发送到所有可获得的数据目的地 116-118,或仅被发送到可获得的数据目的地 116-118 的子组。即一个结果可被发送到数据目的地 116-118,而其它结果可被仅仅发送到单个目的地 116,或被发送到目的地 116-118 的子集。典型的数据目的地 116-118 可以包括但不限于:盘驱动器、计算机外设、单独的外部存储器段和/或其它外部设备。

[0027] 图 2 是具有端到端数据保护(EEDP)检查和更新周期的 DMA 控制器系统的实施例的框图 200。当接收来自一个或更多数据源 202 的数据时,该数据被存储在 DMA 控制器的源数据存储设备 204 内。第一 EEDP 引擎 206 和第二 EEDP 引擎 210 并行地对从一个或更多数据源 202 接收的数据执行操作。第一 EEDP 引擎 206 是检查 EEDP 引擎,它检查嵌入在接收的数据内的数据保护信息(DPI),以便验证该 DPI 并且向 DMA 控制器报告发现 208 的任意错误。由第一 EEDP 检查引擎 206 发现的错误可被 DMA 控制器记录和/或如果启用(enable)了中断功能则可以产生中断。第二 EEDP 引擎 210 实际上组合两个 EEDP 功能,以便产生用于传输数据流的更新的 DPI 或将 DPI 添加到传输数据流。来自第二 EEDP 添加/更新引擎 210 的更新的 DPI 被与来自一个或更多数据源 202 的原始数据流组合 212,并且被发送到所希望的一个或更多目的地 214。可以作为按照以可标识方式在数据流中插入数据包,或作为数据目的地所期望的其它处理,通过多路复用器执行组合处理 212。各个实施例可以根据由 CDB 定义的块大小在传输数据流内的块边界处插入 DPI。当对于被发送到一个或更多目的地 214 的数据来说已经达到了数据流的数据字节计数时,插入 DPI。对于给定的 CDB 可以移动多个数据块,从而对于单个数据流 DPI 被多次插入传输数据流。

[0028] 图 3 是对数据加密并且执行端到端数据保护(EEDP)检查的 DMA 控制器系统的实施例的框图 300。当接收来自一个或更多数据源 302 的数据时,该数据被存储在 DMA 控制器的源数据存储设备 304 内。EEDP 306 和加密引擎 310 并行地对从一个或更多数据源 302 接

收的数据执行操作。EEDP 引擎 306 是检查 EEDP 引擎,它检查嵌入在接收的数据内的 DPI,以便验证该 DPI 并且向 DMA 控制器报告发现的任意错误 308。由 EEDP 引擎 306 发现的错误可被 DMA 控制器记录和 / 或如果启用了中断功能可以产生中断。加密引擎 310 加密正被发送到一个或更多数据目的地 314 的数据,并且将结果存储在本地修改数据存储设备 312 内。加密的数据被从修改数据存储设备 312 传递到一个或更多数据目的地 314。还可以在类似于图 3 所示的实施例 300 的实施例中使用其它数据修改引擎。例如,一个实施例可以用散列或散列 /HMAC 组合引擎取代图 3 所示的加密引擎 310。类似地,各种实施例还可以用不同的数据修改引擎取代图 4-7 所示的实施例的数据修改引擎。另外,图 2-7 所示的实施例仅仅表示可以在 DMA 控制器上实现的大量各种数据操纵引擎配置的子集。传递给 DMA 控制器的每个 CDB 定义了 DMA 控制器上配置的数据操纵引擎的配置,从而 DMA 控制器可被针对由 DMA 控制器接收的每个 CDB 重新配置为大量各种数据操纵引擎配置中的任意一种。对于取代加密引擎 310 使用的某些数据操纵引擎,可能必须在本地的修改数据存储设备 312 中存储某些修改后的数据,从而数据操纵引擎(例如,异或(XOR)引擎)可以访问已被计算的数据。

[0029] 图 4 是对数据加密并且具有端到端数据保护(EEDP)检查和更新周期的 DMA 控制器系统的实施例的框图 400。当接收来自一个或更多数据源 402 的数据时,该数据被存储在 DMA 控制器的源数据存储设备 404 内。第一 EEDP 引擎 406、加密引擎 410 和第二 EEDP 引擎 414 并行地执行操作。第一 EEDP 引擎 406 是检查 EEDP 引擎,它检查嵌入在接收的数据内的 DPI,以便验证该 DPI 并且向 DMA 控制器报告发现的任意错误 408。由第一 EEDP 引擎 406 发现的错误可被 DMA 控制器系统记录和 / 或如果启用了中断功能可以产生中断。加密引擎 410 加密正被发送到一个或更多数据目的地 418 的数据,并且将结果存储在本地修改数据存储设备 412 内。加密的数据被从修改数据存储设备 412 传递到一个或更多数据目的地 418,并且被传递到第二 EEDP 添加 / 更新引擎 414。第二 EEDP 引擎 414 实际上组合两个 EEDP 功能,以便产生用于传输数据流的更新的 DPI 或将 DPI 添加到传输数据流。来自第二 EEDP 添加 / 更新引擎 414 的更新的 DPI 被与来自修改数据存储设备 412 的加密数据流组合 416,并且被发送到所希望的一个或更多目的地 418。可以作为按照以可标识方式在数据流中插入数据包,或作为数据目的地所期望的其它处理,通过多路复用器执行组合处理 416。各个实施例可以根据由 CDB 定义的块大小在传输数据流内的块边界处插入 DPI。当对于正被发送到一个或更多目的地 418 的数据来说已经达到了数据流的数据字节计数时,插入 DPI。对于给定的 CDB 可以移动多个数据块,从而对于单个数据流 DPI 被多次插入传输数据流。

[0030] 图 5 是对数据加密,对数据执行异或(XOR),并且执行端到端数据保护(EEDP)检查的 DMA 控制器系统的实施例的框图 500。当接收来自一个或更多数据源 502 的数据时,该数据被存储在 DMA 控制器的源数据存储设备 504 内。EEDP 引擎 506、加密引擎 510 和第一 514 和第二 518 异或(XOR)引擎并行地执行操作。EEDP 引擎 506 是检查 EEDP 引擎,它检查嵌入在接收的数据内的 DPI,以便验证该 DPI 并且向 DMA 控制器报告发现的任意错误 508。由 EEDP 引擎 506 发现的错误可被 DMA 控制器系统记录和 / 或如果启用了中断功能可以产生中断。加密引擎 510 加密正被发送到一个或更多数据目的地 524 的数据,并且将结果存储在本地修改数据存储设备 #1(512) 内。加密的数据被从修改数据存储设备 512 传递到一个或更多数据目的地 524,并且被传递到第一 XOR 引擎 514 和第二 XOR 引擎 518。每个 XOR 引擎

514 和 518 可能需要访问已经计算的数据,以正确地完成计算。因此,第一 XOR 引擎 514 可以在第二修改数据存储设备 516 内存储结果数据。类似地,第二 XOR 引擎 518 可以在第三修改数据存储设备 520 内存储结果数据。XOR 结果数据被从第二修改数据存储设备 516 和第三修改数据存储设备 520 传递到一个或更多数据目的地 524。来自第一修改数据存储设备 512 的加密数据,来自第二修改数据存储设备 516 的第一 XOR 数据和来自第三修改数据存储设备 520 的第二 XOR 数据被组合 522,并且传递到一个或更多数据目的地 524。可以作为按照以可标识方式在数据流中插入数据包,或作为数据目的地 524 所期望的其它处理,通过多路复用器执行组合处理 522。

[0031] 对于图 5 所示的实施例 500,可以从每个数据源 502 检索固定的数据部分(例如,1kb 数据),每次检索一个固定数据部分。然后,每个固定的数据部分被发送到加密引擎 510,并且加密结果被发送到指定的一个或更多数据目的地 524。当加密数据被从加密引擎 510 发送到加密数据的一个或更多数据目的地 524 时,执行 XOR 处理。一旦已经从每个数据源 502 检索了固定的数据部分(例如,从每个数据源 502 检索的 1kb 数据),并且加密数据已被发送到了加密引擎 510 的指定的一个或更多数据目的地 524,第一 XOR 引擎 514 的结果被发送到第一 XOR 引擎 514 的指定的一个或更多数据目的地 524。在第一 XOR 引擎 514 的结果被发送到第一 XOR 引擎 514 的指定的一个或更多数据目的地 524 之后,第二 XOR 引擎 518 的结果被发送到第二 XOR 引擎 518 的指定的一个或更多数据目的地 524。重复来自每个数据源 502 的固定数量数据的处理,直到已经检索了最后的固定数量数据(或更少)为止。

[0032] 图 6 是对数据加密,对数据执行异或(XOR),并且具有端到端数据保护(EEDP)检查和更新周期的 DMA 控制器系统的实施例的框图 600。当接收来自一个或更多数据源 602 的数据时,该数据被存储在 DMA 控制器的源数据存储设备 604 内。第一 EEDP 引擎 606、加密引擎 610、第一 614 和第二 618 异或(XOR)引擎和第二 EEDP 引擎 624 并行地执行操作。第一 EEDP 引擎 606 是检查 EEDP 引擎,它检查嵌入在接收的数据内的 DPI,以便验证该 DPI 并且向 DMA 控制器报告发现的任意错误 608。由第一 EEDP 引擎 606 发现的错误可被 DMA 控制器系统记录和/或如果启用了中断功能可以产生中断。加密引擎 610 加密正被发送到一个或更多数据目的地 628 的数据,并且将结果存储在本地修改数据存储设备 #1(612)内。加密的数据被从第一修改数据存储设备 612 传递到一个或更多数据目的地 628,传递到第二 EEDP 添加/更新引擎 624,并且传递到第一 XOR 引擎 614 和第二 XOR 引擎 618。每个 XOR 引擎 614 和 618 可能需要访问已经计算的数据,以正确地完成计算。因此,第一 XOR 引擎 614 可以在第二修改数据存储设备 616 内存储结果数据。类似地,第二 XOR 引擎 618 可以在第三修改数据存储设备 620 内存储结果数据。XOR 结果数据被从第二修改数据存储设备 616 和第三修改数据存储设备 620 传递到一个或更多数据目的地 618,并且传递到第二 EEDP 添加/更新引擎 624。来自第一修改数据存储设备 612 的加密数据,来自第二修改数据存储设备 616 的第一 XOR 数据和来自第三修改数据存储设备 620 的第二 XOR 数据被组合并且传递到第二 EEDP 引擎 624。可以作为按照以可标识方式在数据流中插入数据包,或作为第二 EEDP 引擎 624 所期望的其它处理,通过多路复用器执行组合处理 622。第二 EEDP 引擎 624 实际上组合两个 EEDP 功能,以便产生用于加密并且异或的数据流的更新的 DPI,或将 DPI 添加到当前传输数据流。来自第二 EEDP 添加/更新引擎 624 的更新的 DPI 被与来自第一修改数据存

储设备 612 的加密数据流、来自第二修改数据存储设备 616 的第一 XOR 数据和来自第三修改数据存储设备 620 的第二 XOR 数据组合 626, 并且被发送到所希望的一个或更多目的地 628。可以作为按照以可标识方式在数据流中插入数据包, 或作为数据目的地所期望的其它处理, 通过多路复用器执行组合处理 626。

[0033] 在图 6 所示的实施例的数据修改引擎的多源移动的示例操作中, DMA 引擎/控制器从源检索数据, 并且通过加密引擎 610 发送该数据。同时, 相同的数据还被发送到第一 EEDP 检查引擎 606, 以便验证 DPI 数据是正确的。在已经加密了数据流的每 16 字节之后, 除了被发送到数据目的地 628 之外, 加密的数据被同时发送到三个不同的数据修改引擎 614、618、624。加密的数据被发送到第一 XOR 引擎 614、第二 XOR 引擎 618 和第二 EEDP 添加/更新引擎 624。XOR 数据被存储在相关的存储器缓冲器 616 和 620 内, 从而当检索到来自不同源 602 的数据时可以累积 XOR 结果。第二 EEDP 添加/更新引擎 624 创建当数据被发送到第二 EEDP 添加/更新引擎 624 时所需的新 DPI。当达到块边界时 (例如, 在 16 字节之后), 产生的 DPI 被置于被发送到目的地 628 的数据流内。上述处理继续, 直到来自所有所需源 602 的数据被检索、加密并且被异或在一起为止。因此, 5 个不同的数据修改引擎同时在 DMA 引擎/控制器上并发地操作。各种实施例可以根据由 CDB 定义的块大小在传输数据流内的块边界处插入 DPI。当对于正被发送到一个或更多目的地 628 的数据来说已经达到了数据流的数据字节计数时, 插入 DPI。对于给定的 CDB 可以移动多个数据块, 从而对于单个数据流 DPI 被多次插入传输数据流。

[0034] 实现图 6 所示的配置 600 的实施例可以从每个数据源 602 检索固定的数据部分 (例如, 1kb 数据), 每次检索一个固定数据部分。然后, 每个固定的数据部分被发送到加密引擎 610, 并且加密结果被发送到第二 EEDP 引擎 624, 并且然后被发送到加密数据的指定的一个或更多数据目的地 524。当加密数据被从加密引擎 610 发送到第二 EEDP 引擎 624 并且被发送到加密引擎 610 的一个或更多数据目的地 628 时, 执行 XOR 处理。一旦已经从每个数据源 602 检索了固定的数据部分 (例如, 从每个数据源 602 检索的 1kb 数据), 并且加密数据已被通过第二 EEDP 引擎 624 发送到了加密引擎 610 的指定的一个或更多数据目的地 628, 第一 XOR 引擎 614 的结果被发送到第二 EEDP 引擎 624, 并且被发送到第一 XOR 引擎 614 的指定的一个或更多数据目的地 628。在第一 XOR 引擎 614 的结果被发送到第一 XOR 引擎 614 的指定的一个或更多数据目的地 628 之后, 第二 XOR 引擎 618 的结果被发送到第二 EEDP 引擎 624, 并且被发送到第二 XOR 引擎 618 的指定的一个或更多数据目的地 628。重复来自每个数据源 602 的固定数量数据的处理, 直到已经检索了最后的固定数量数据 (或更少) 为止。为向第二 EEDP 引擎 624 流化发送数据选择的本地修改数据存储设备 612、616、620 与当前传输到指定一个或更多目的地 628 的数据流匹配 (加密引擎 610 的结果、第一 XOR 引擎 614 或第二 XOR 引擎结果 618)。

[0035] 图 7 是对数据执行异或 (XOR), 并且具有端到端数据保护 (EEDP) 检查和更新周期的 DMA 控制器系统的实施例的框图 700。当接收来自一个或更多数据源 702 的数据时, 该数据被存储在 DMA 控制器的源数据存储设备 704 内。第一 EEDP 引擎 706、第一 XOR 引擎 710 和第二 XOR 引擎 714 和第二 EEDP 引擎 720 并行地执行操作。第一 EEDP 引擎 606 是检查 EEDP 引擎, 它检查嵌入在接收的数据内的 DPI, 以便验证该 DPI 并且向 DMA 控制器报告发现的任意错误 708。由第一 EEDP 引擎 706 发现的错误可被 DMA 控制器系统记录和 / 或如果启

用了中断功能可以产生中断。每个 XOR 引擎 710 和 714 可能需要访问已经计算的数据,以正确地完成任务。因此,第一 XOR 引擎 710 可以在第一修改数据存储设备 712 内存储结果数据。类似地,第二 XOR 引擎 714 可以在第二修改数据存储设备 716 内存储结果数据。XOR 结果数据被从第一修改数据存储设备 712 和第二修改数据存储设备 716 传递到一个或更多数据目的地 724,并且传递到第二 EEDP 添加 / 更新引擎 720。来自第一修改数据存储设备 712 的第一 XOR 数据和来自第二修改数据存储设备 716 的第二 XOR 数据被组合并且传递到第二 EEDP 引擎 720。可以作为按照以可标识方式在数据流中插入数据包,或作为第二 EEDP 引擎 720 所期望的其它处理,通过多路复用器执行组合处理 718。第二 EEDP 引擎 720 实际上组合两个 EEDP 功能,以便产生用于异或数据流的更新的 DPI,或将 DPI 添加到当前传输数据流。来自第二 EEDP 添加 / 更新引擎 720 的更新的 DPI 被与来自第一修改数据存储设备 712 的第一 XOR 数据和来自第二修改数据存储设备 716 的第二 XOR 数据组合 722,并且然后被发送到所希望的一个或更多目的地 724。可以作为按照以可标识方式在数据流中插入数据包,或按照数据目的地所期望的其它处理,通过多路复用器执行组合处理 722。各种实施例可以根据由 CDB 定义的块大小在传输数据流内的块边界处插入 DPI。当对于正被发送到一个或更多目的地 724 的数据来说已经达到了数据流的数据字节计数时,插入 DPI。对于给定的 CDB 可以移动多个数据块,从而对于单个数据流 DPI 被多次插入传输数据流。

[0036] 对于图 7 所示的实施例 700,可以从每个数据源 702 检索固定的数据部分(例如,1kb 数据),每次检索一个固定数据部分。然后,每个固定的数据部分被同时发送到第一 XOR 引擎 710 和第二 XOR 引擎 714。当从一个或更多数据源 702 检索数据时,每个 XOR 引擎 710、714 以当前检索到的数据在 XOR 引擎 710、714 各自的本地修改数据存储设备 712、716 内执行异或操作。一旦已经从每个数据源 702 检索了固定的数据部分(例如,从每个数据源 702 检索的 1kb 数据),来自第一 XOR 引擎 710 的结果被发送到第二 EEDP 引擎 720,并且被发送到第一 XOR 引擎 710 的指定的一个或更多数据目的地 724。在第一 XOR 引擎 710 的结果被发送到第一 XOR 引擎 710 的指定的一个或更多数据目的地 724 之后,第二 XOR 引擎 714 的结果被发送到第二 EEDP 引擎 720,并且被发送到第二 XOR 引擎 714 的指定的一个或更多数据目的地 724。重复来自每个数据源 702 的固定数量数据的处理,直到已经检索了最后的固定数量数据(或更少)为止。为向第二 EEDP 引擎 720 流化发送数据所选择的本地修改数据存储设备 712、716 与当前传输到指定一个或更多目的地 724 的数据流匹配(第一 XOR 引擎结果 710 或第二 XOR 引擎结果 714)。

[0037] 图 8 是包括 DMA 引擎 / 控制器 812 的芯片上的 RAID(独立磁盘冗余阵列)(ROC)系统 802 的实施例的框图 800。图 8 所示的 ROC802 实施例提供了在多功能芯片上包括 DMA 引擎 / 控制器 812 的例子。在图 8 所示的示例 ROC802 实施例中,ROC802 由几个子系统组成,包括:串行附加 SCSI(小型计算机系统接口)(SAS)核心 810、DMA 引擎 / 控制器 812、存储器队列和存储器控制器 813、时钟配置 / 重置子系统(CCR)814、外设组件互连(PCI)特快(Express)(PCE)核心 818、输入 / 输出(I/O)子系统 820、处理器本地总线(PLB)静态 RAM(SRAM)822 和外设总线访问模块(PBAM)824。ROC802 的各种子系统通过处理器本地总线(PLB)816 和 / 或通过直接连接 830-836 彼此通信。DMA 引擎 / 控制器 812 具有与存储器队列和存储器控制器 813 的直接连接 836,以便控制到外部连接的双数据速率(DDR)RAM 存储器 806 的直接存储器访问通道 808。存储器队列和存储器控制器 813 可被与 DMA 引擎 / 控制

器 812 结合在一起,以便形成 ROC802 的单个逻辑元件,但是它们被分别示出,以便清楚地示出 DMA 引擎 / 控制器 812 的位置。SAS 核心 810 在允许传输 SAS 协议通信 804 的通信路径上提供与 SAS 设备之间的往来外部通信。类似地,PCE 核心 818 在允许传输 PCI-Express 协议通信 826 的通信路径上提供与 PCI-Express 设备之间的往来外部通信。PBAM 在允许传输串行数据 (SD)828 的通信路径上提供与串行数据 (SD) 外设之间的往来外部通信。CCR814、I/O 处理器子系统 820 和 PLB SRAM 822 提供互连各种外部通信子系统 810、812-813、818 和 824 的支持功能。

[0038] 在图 8 所示的 ROC 实施例 802 中,DMA 引擎 / 控制器 812 通过到存储器队列和存储器控制器 813 的直接连接 836 访问外部 DDRAM 存储器 806。另外,DMA 引擎 / 控制器 812 具有通过 PLB 816 到 ROC 802 上的其它系统 810、814、818、820、822、824 的连接。DMA 引擎 / 控制器 812 被通过 PLB 816 连接到 ROC 实施例 802 的其余子系统。为了提高通信速度,特别是对于大量的数据移动,除了 PLB816 连接之外,可以结合 ROC 802 的子系统之间的某些直接通信连接 830-836。例如,在图 8 所示的实施例中,PCE 核心 818 和 SAS 核心 810 之间存在直接通信连接 830。在图 8 所示的实施例中,还存在 DMA 引擎 / 控制器 812 和 PCE 核心 818 之间的直接通信连接 832,以及存储器队列和存储器控制器 813 和 SAS 核心 810 之间的直接通信连接 834。通过 SAS 核心 810、PCE 核心 818 和 PBAM824、DMA 引擎 / 控制器 812 可以与 SAS 设备 804、PCI-Express 设备 826 和串连外设 828 通信。如果系统设计者需要,ROC 802 或其它可以使用 DMA 的设备 / 系统的各种实施例可以结合更少或更多的外部数据通信连接。

[0039] 图 9 是 DMA 控制器的实施例的 CDB 处理的实施例的框图 900。CDB 获取引擎 904 可以获得 CDB902,并且将该 CDB 存储在预取本地 RAM906 中。CDB 获取引擎 906 分析嵌入 CDB902 内的指令,并且指挥 CDB 处理器 908 执行必须的动作,以便执行编码在 CDB902 内的指令。CDB 处理器 908 还可以与存储器管理器 910 交互,以便确定 DMA 控制器的本地 RAM912 中存在多少可用空间。CDB 处理器 908 还可以与 DMA 控制器的本地 RAM912 交互,以便当传输数据并且执行所希望的多种数据操纵需要时,存储数据和修改后的数据。CDB 处理器 908 可以从 DMA 控制器上可以获得的数据修改引擎 916-932 中选择适当的数据修改引擎 914。对于图 9 所示的实施例,可获得的数据修改引擎包括:散列 /HMAC 引擎 916、填充模式引擎 918、LFSR 引擎 920、EEDP 产生引擎 922、EEDP 检查引擎 924、第一 (#1) XOR 引擎 926、第二 (#2) XOR 引擎 928、解密引擎 930 和加密引擎 932。对于各种实施例,散列 /HMAC 引擎 916 可以是,一起或单独地,散列、HMAC 或散列 /HMAC 组合引擎。填充模式引擎 918 可用于以固定的数据模式填充修改存储设备。因此,由于所写的模式是固定的数据模式,填充模式引擎 918 不需要从数据源读数据。LFSR 引擎 920 也不需要从数据源读数据来执行 LFSR 引擎 920 的操作。类似地,某些数据操纵引擎不向目的地写数据,而是仅仅对从源接收的数据执行检查,诸如 EEDP 检查引擎 924。因此,对于某些数据操纵引擎(例如,填充模式引擎 918 和 / 或 LFSR 引擎 920),根据 CDB 从数据源读取可能不需要从数据源读取任何实际数据。类似地,对于某些数据操纵引擎(例如,EEDP 检查引擎 924),根据 CDB 向数据目的地写数据可能不需要将实际数据写到数据目的地。CDB 处理器 908 处理将数据写到数据目的地,并且针对 CDB 追踪数据操纵引擎的执行。一旦 CDB 处理器 908 完成了给定 CDB902 的所有数据移动,CDB 完成引擎 934 将 CDB 完成结果写到完成队列 936。完成队列 936 可以位于 DMA 控制器 /

引擎之外的存储器内。当第一 CDB 的数据操纵接近完成或完成时,各种实施例还可以允许排出 FSM1012 处理第一 CDB 的写数据,同时填充 FSM1006 可以为第二(即,下一个)CDB 开始检索数据的处理,以便提高整体处理速度。

[0040] 图 10 是 CDB 处理器实施例的 CDB 处理的有限状态机 (FSM) 操作的框图 1000。CDB 获取引擎有限状态机 (FSM) 1002 分析嵌入 CDB 内的指令,并且可以与 4 个有限状态机 1006-1012 交互来执行必须的动作,以便执行嵌入由 DMA 控制器接收的 CDB 内的指令。填充 FSM1006 从 CDB 获取引擎 FSM1002 接收指令,并且控制读命令 1004 的创建和发出,以便从一个或更多数据源检索数据。修改 FSM1008 根据嵌入 CDB 内的指令使用 DMA 控制器数据修改引擎 1018 控制数据操纵操作的执行。排出命令 (CMD) FSM1010 创建写命令,该写命令指挥系统将数据和 / 或修改后的数据从数据修改引擎 1018 发送到所希望的一个或更多数据目的地 1016。排出数据 FSM1012 执行在排出命令 FSM1010 中创建的写命令,以便将数据和 / 或修改后的数据写到所希望的一个或更多数据目的地 1016。在某些情况下,排出数据 FSM1012 还可以使用数据修改引擎 1018 执行某些数据修改操作,诸如用于不需要在本地 DMA 控制器 RAM 中缓存数据以便执行数据操纵计算的 EEDP 产生的数据修改操作。

[0041] 各种实施例可以通过操作在计算机系统、现场可编程门阵列 (FPGA)、专用集成电路 (ASIC)、其它可编程逻辑电路或其它电子设备上的应用,提供此处详述的控制和管理功能。实施例可被作为可被包括在计算机可读或计算机可用介质计算机程序产品提供,所述介质上存储有可用于对计算机(或其它电子设备)编程 / 操作计算机(或其它电子设备)或计算机系统,以便执行根据本发明的一个或更多处理的指令。计算机可读介质可以包括但不限于硬盘驱动器、软盘、光盘、压缩盘只读存储器 (CD-ROM)、数字通用盘 ROM (DVD-ROM)、通用串行总线 (USB) 存储器棒、磁光盘、ROM、随机访问存储器 (RAM)、可擦除可编程 ROM (EPROM)、电可擦除可编程 ROM (EEPROM)、磁光卡、闪存或适合于存储电子指令的其它类型的介质 / 机器可读介质。计算机程序指令可以驻留在并且运行在单个计算机 / 电子设备 / 电子电路上,或各种部分可以分布在构成计算机系统的多个计算机 / 设备 / 电子电路上。另外,实施例还可以被作为计算机程序产品下载,其中程序可被借助于包含在载波内的数据信号或其它传播介质通过通信链路(例如,调制解调器或网络连接,包括有线 / 电缆线和无线连接两者)从远程计算机传输到做出请求的计算机。

[0042] 已经出于说明和描述的目的给出了本发明的上面描述。它不旨在是穷尽性的或将本发明局限于公开的精确形式,并且根据上面的教导可以有其它修改和变形。选择并且描述实施例,以便最好地解释本发明的原理和其实际应用,从而使得本领域的其它技术人员能够在适合于构想的特定用途的各种实施例和各种修改中最好地利用本发明。所附的权利要求旨在被认为包括本发明的除了受到现有技术限制之外的其它替换实施例。

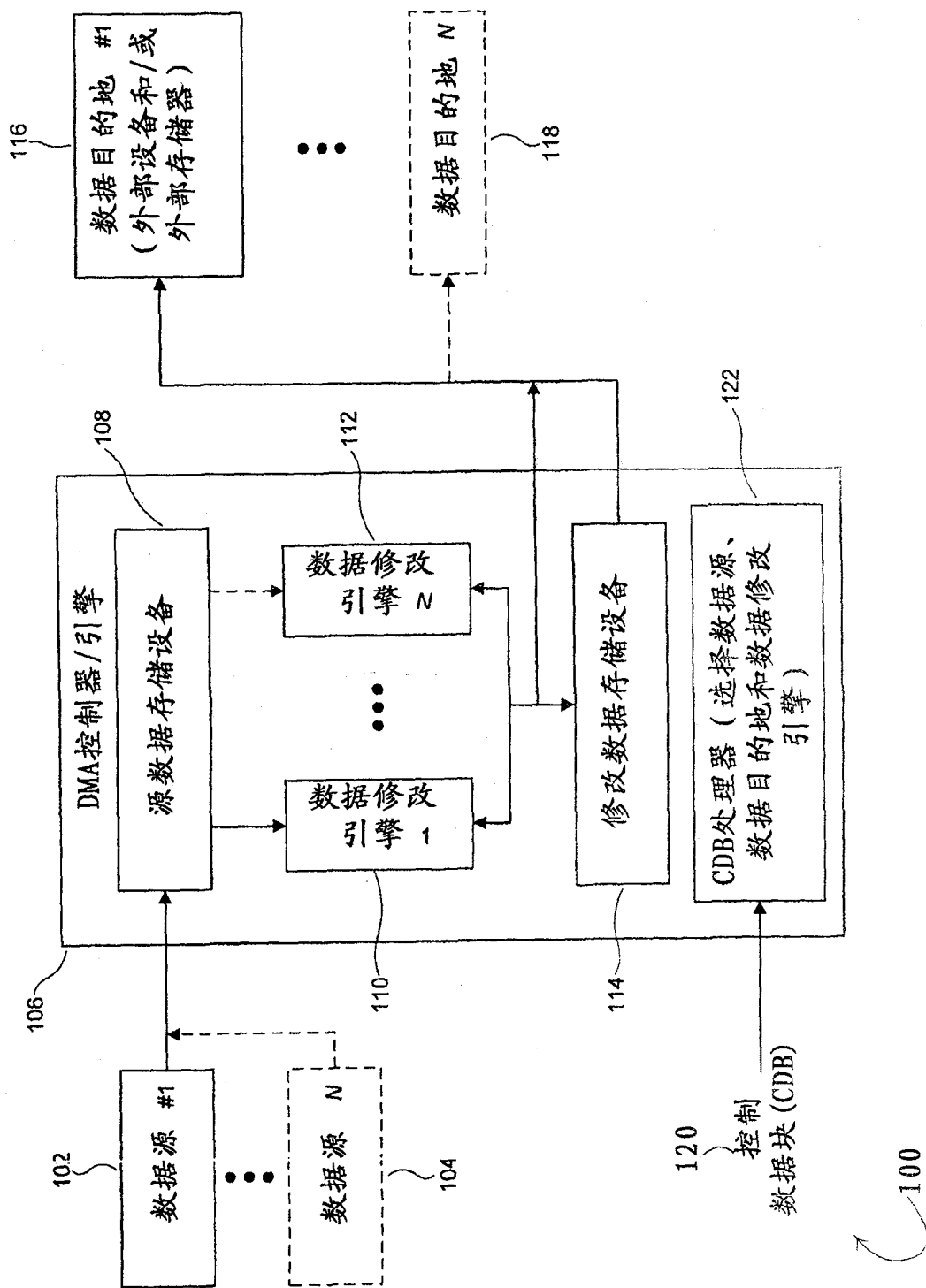


图 1

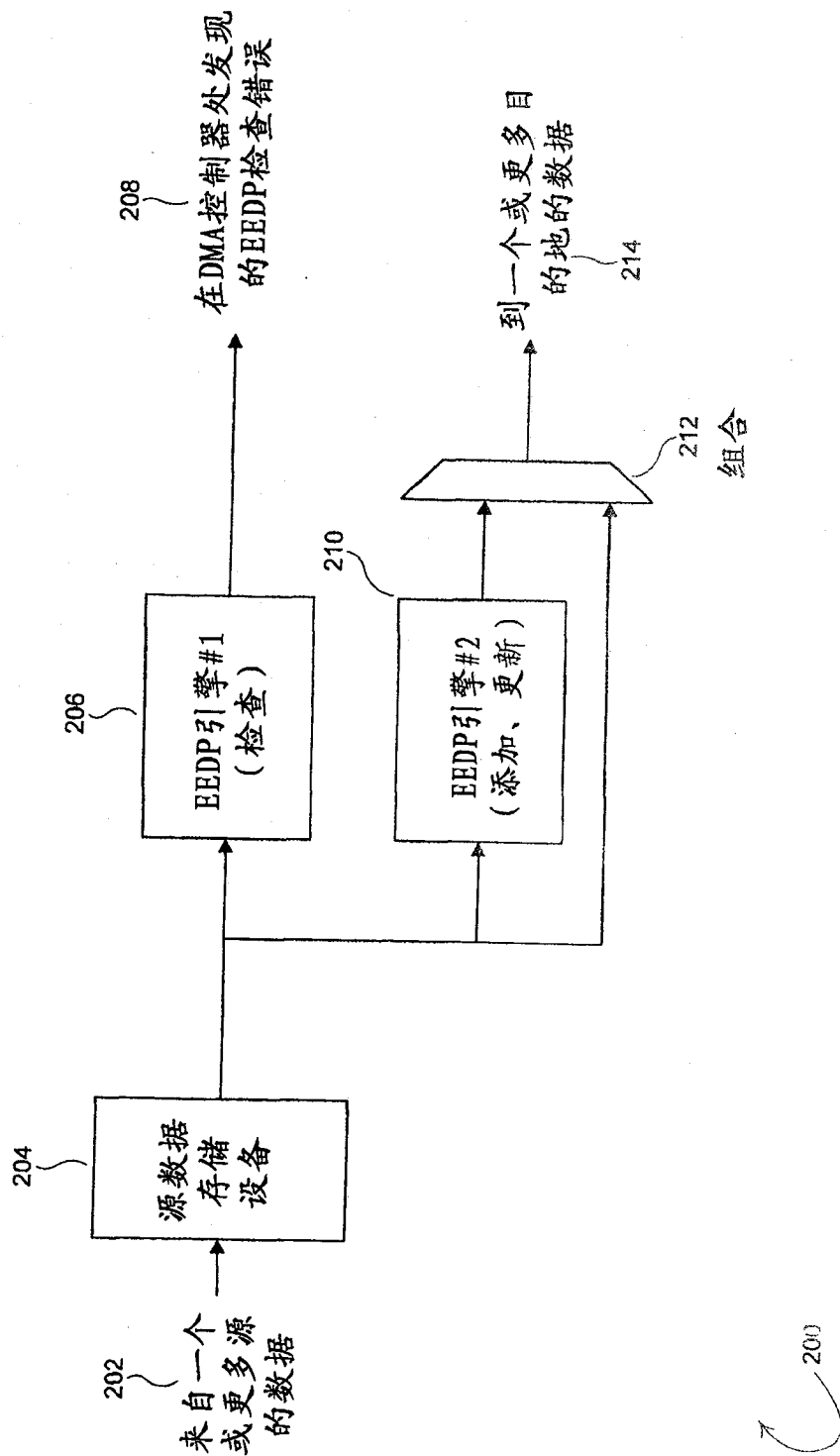


图 2

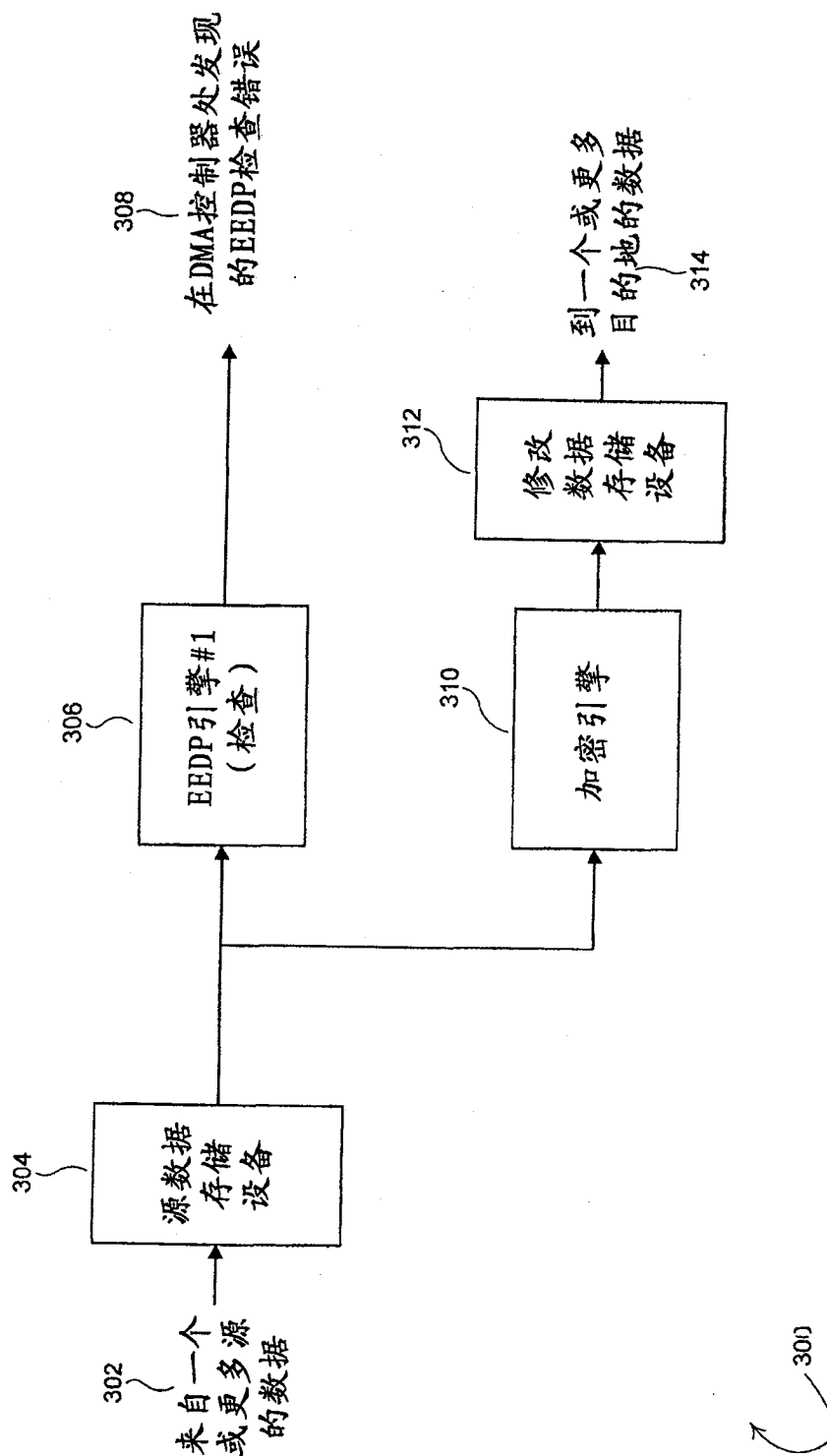


图 3

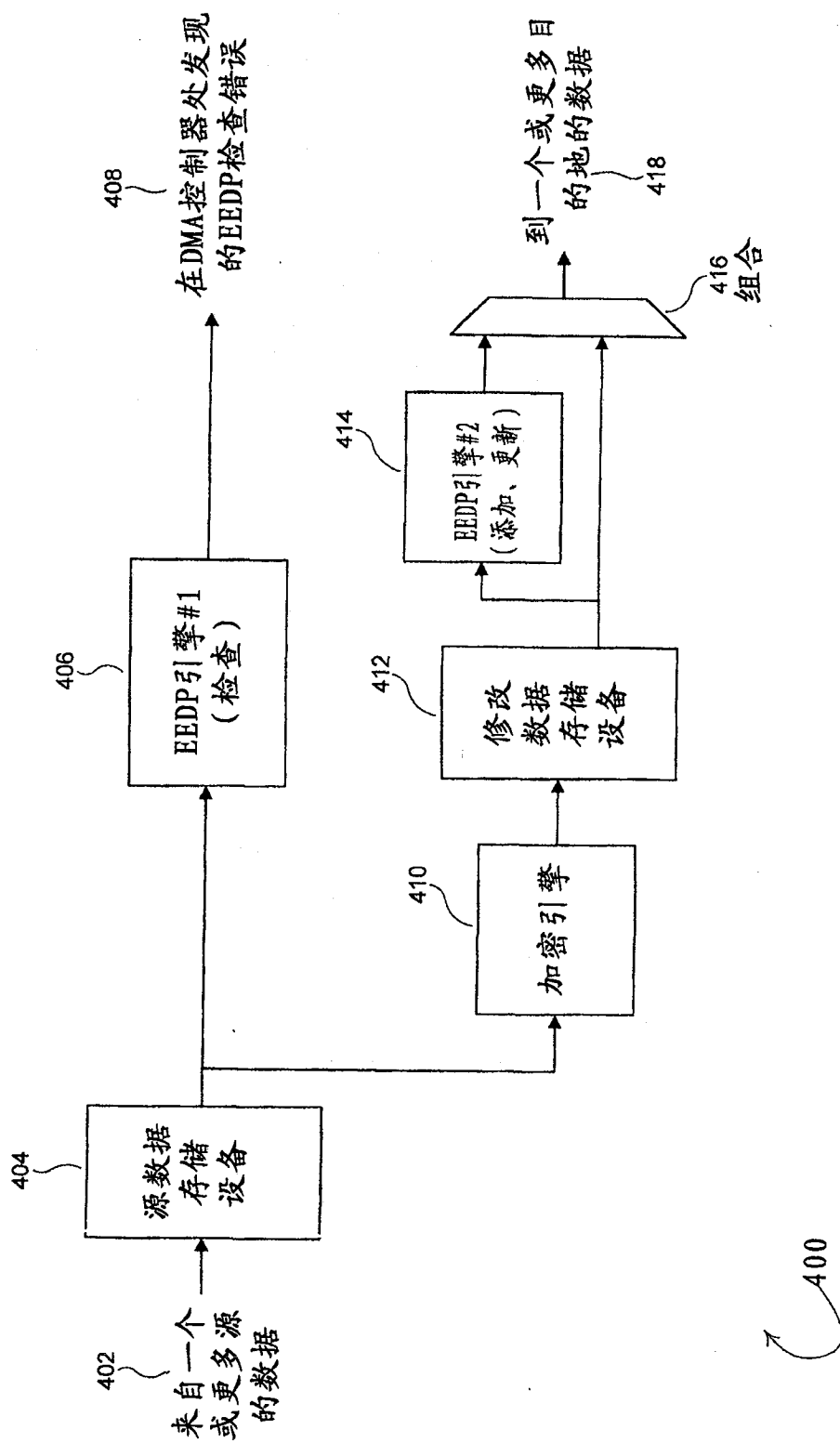


图 4

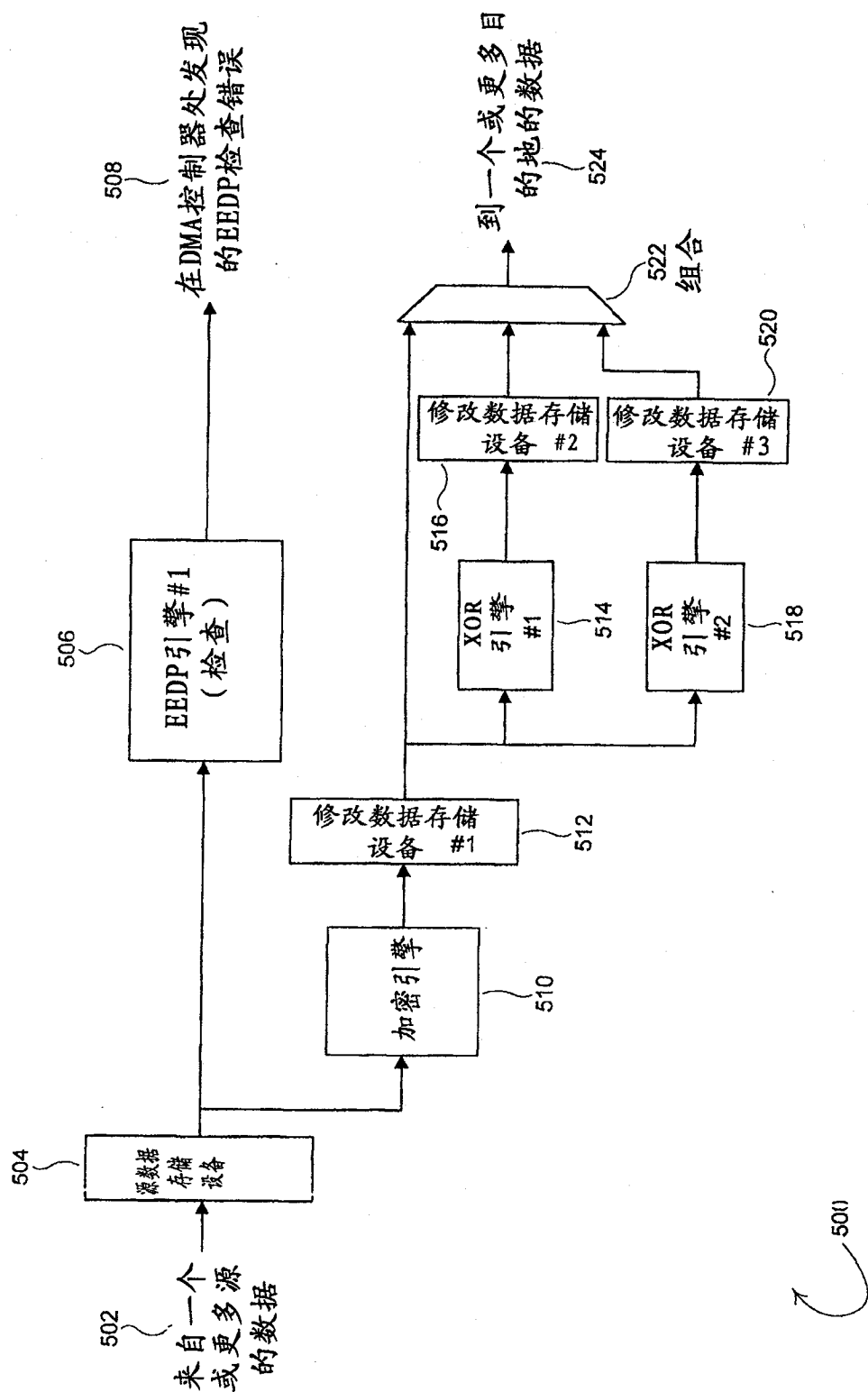


图 5

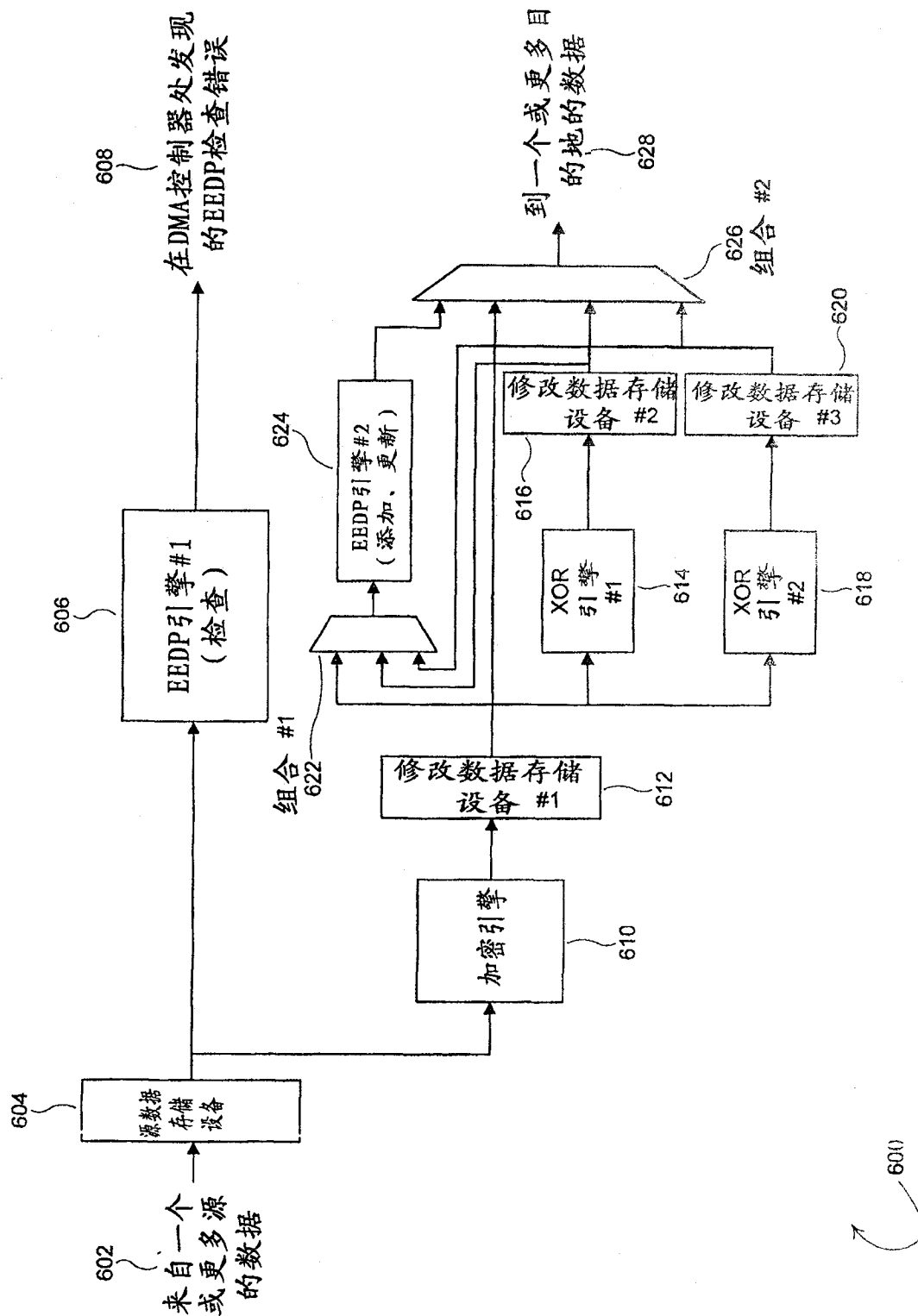


图 6

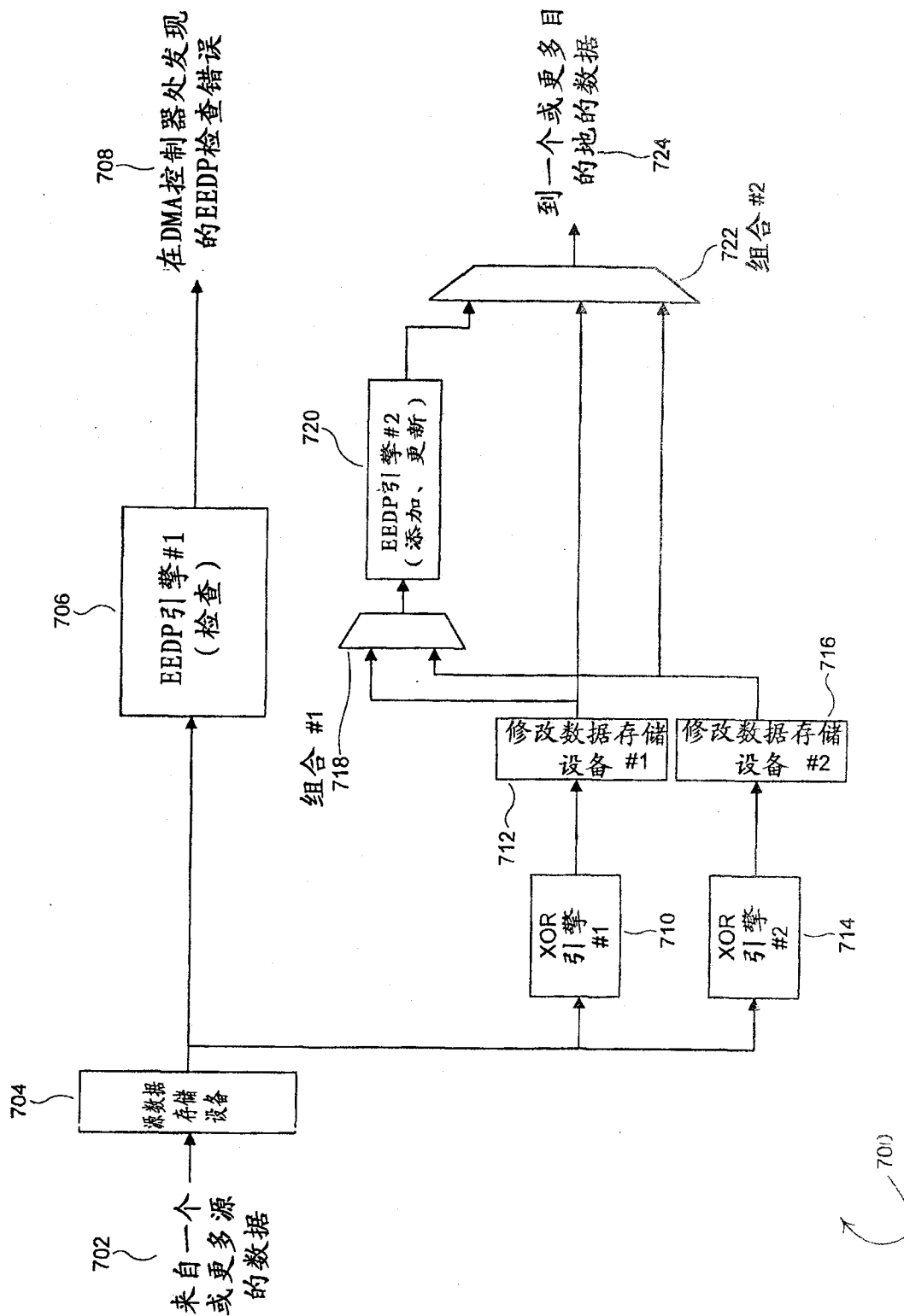


图 7

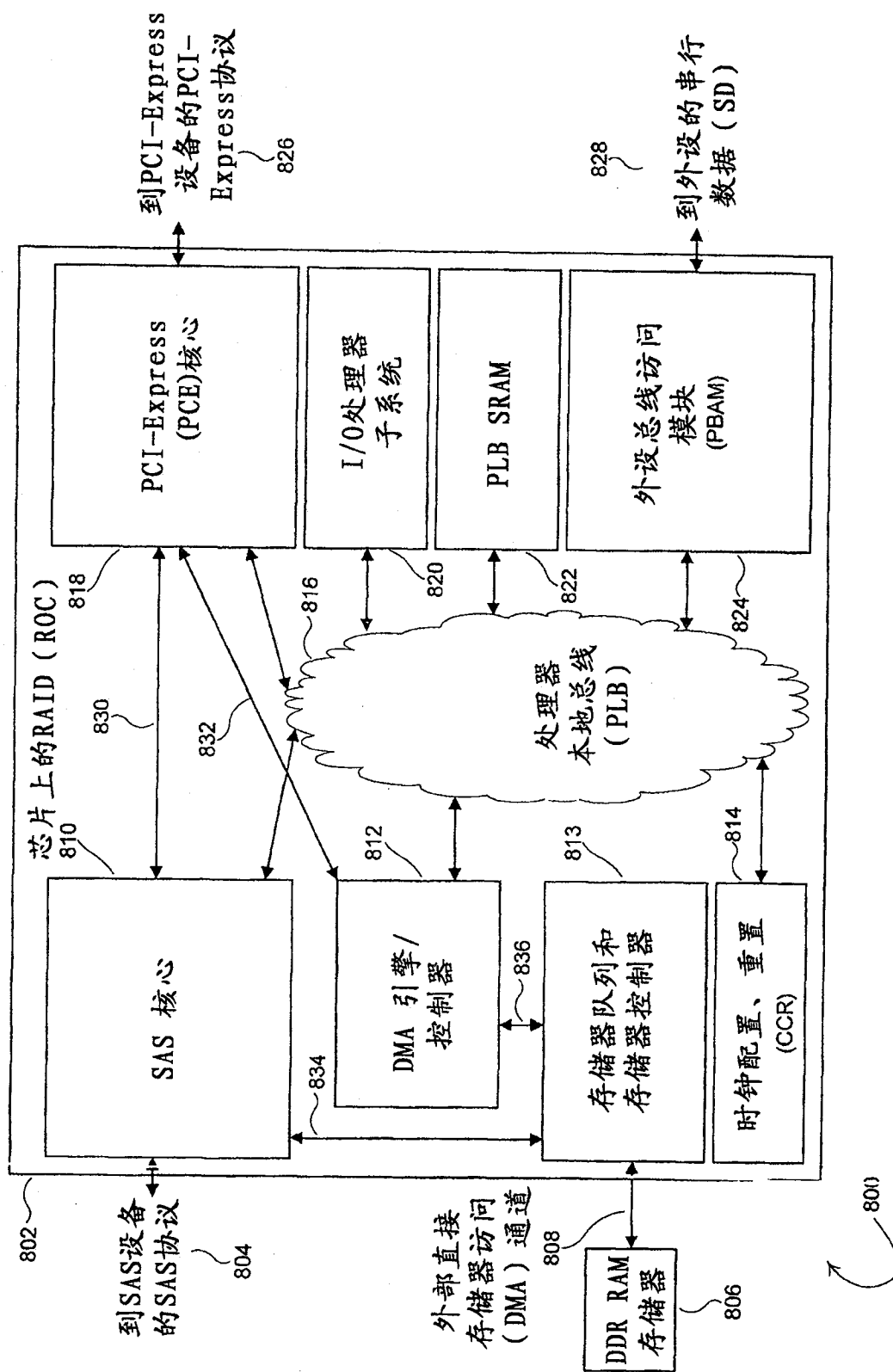


图 8

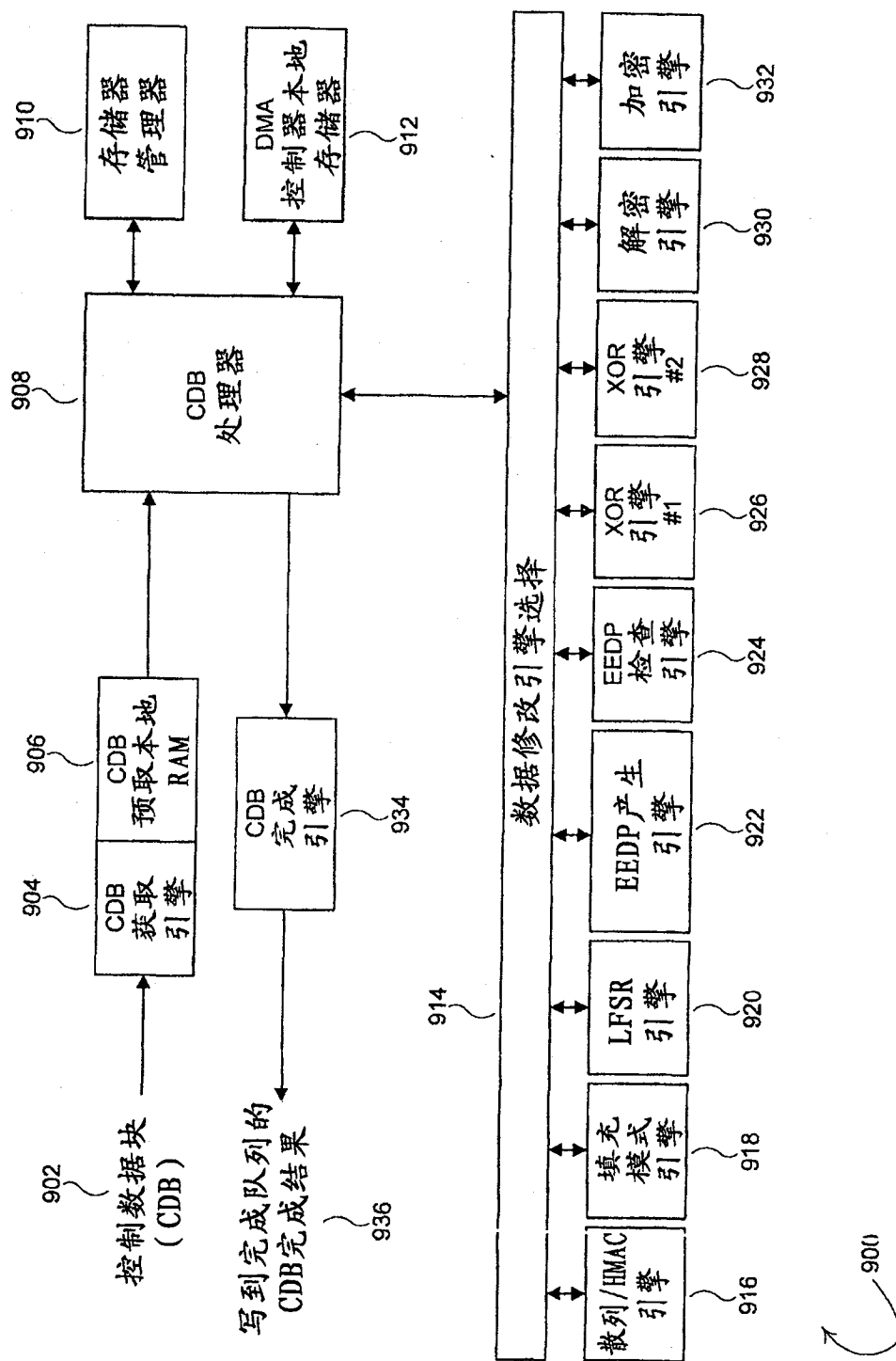


图 9

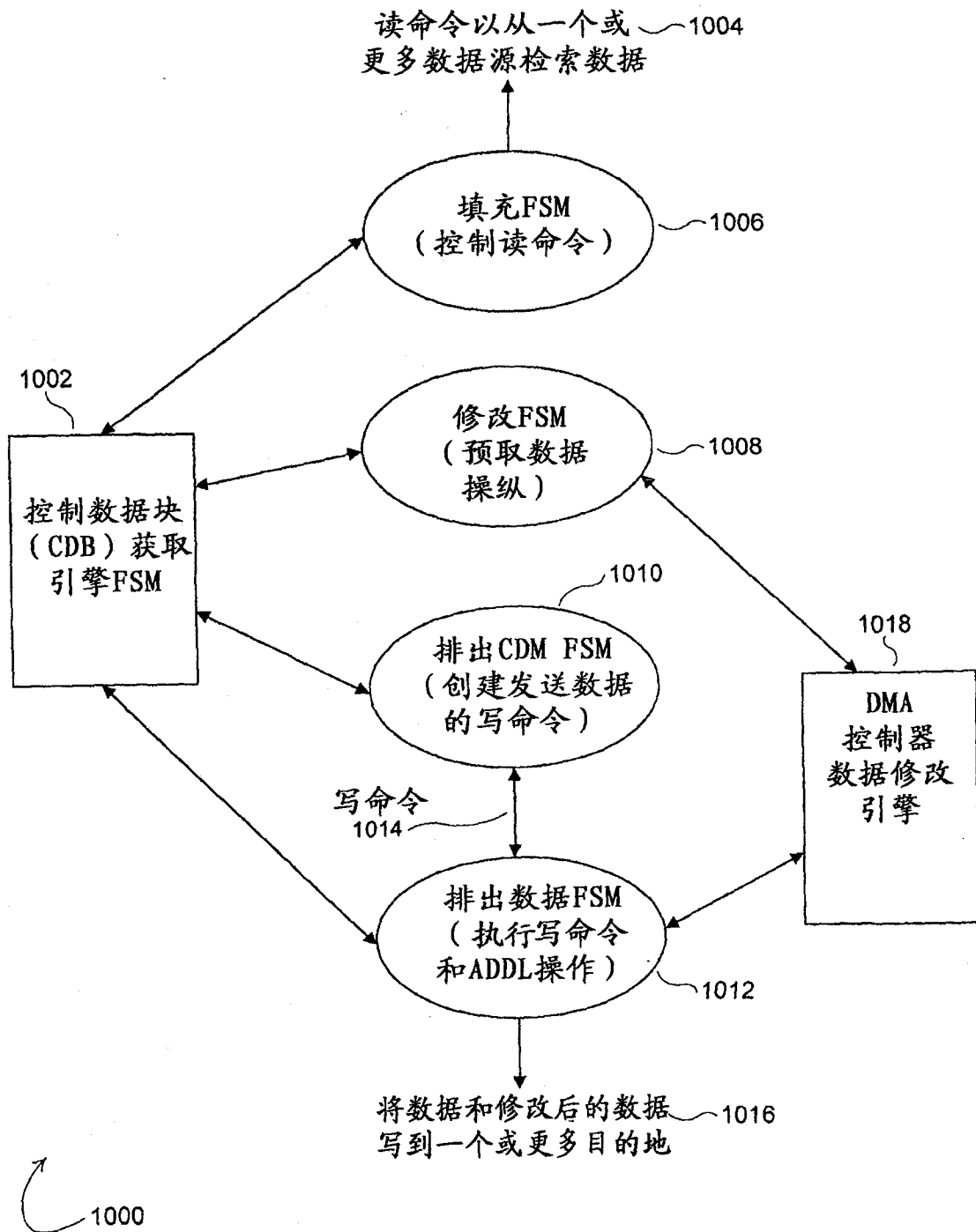


图 10