



(12)发明专利申请

(10)申请公布号 CN 109155146 A

(43)申请公布日 2019.01.04

(21)申请号 201680085880.4

(22)申请日 2016.09.22

(30)优先权数据

15/168,045 2016.05.28 US

(85)PCT国际申请进入国家阶段日

2018.11.19

(86)PCT国际申请的申请数据

PCT/US2016/053138 2016.09.22

(87)PCT国际申请的公布数据

WO2017/209781 EN 2017.12.07

(71)申请人 超威半导体公司

地址 美国加利福尼亚州

(72)发明人 凯文·M·布朗德

(74)专利代理机构 上海胜康律师事务所 31263

代理人 李献忠 张华

(51)Int.Cl.

G11C 29/00(2006.01)

G11C 11/4093(2006.01)

G11C 29/44(2006.01)

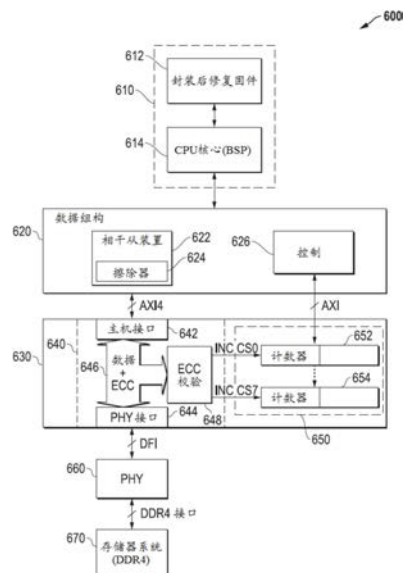
权利要求书3页 说明书10页 附图7页

(54)发明名称

一体式封装后修复装置

(57)摘要

一种封装后修复系统,包括存储器通道控制器、第一错误计数器、擦除器和数据处理器。所述存储器通道控制器将数据访问请求转换为对应存储器访问,并且响应于从存储器接口接收到的响应而将返回的数据提供给主机接口,其中所述响应包括返回的数据和多个错误校正码(ECC)位。所述第一错误计数器对所述返回数据中的错误进行计数,并且响应于达到预定状态而提供控制信号。所述擦除器控制所述存储器通道控制器以从存储器系统的多个地址顺序地且周期性地读取数据,并且响应于检测到可校正的错误,重写校正的数据。所述数据处理器对所述控制信号作出响应,以响应于所述控制信号利用所述存储器系统执行封装后修复操作。



1. 一种封装后存储器修复系统(600),其包括:

存储器通道控制器(640),所述存储器通道控制器用于将从主机接口(642)接收到的数据存取请求转换为提供给存储器接口(644)的对应存储器存取,并且响应于从所述存储器接口(644)接收到的响应而将返回的数据提供给所述主机接口(642),其中所述响应包括返回的数据和多个错误校正码(ECC)位;

耦接到所述存储器通道控制器(640)的第一错误计数器(652),所述第一错误计数器用于对所述返回数据中的错误进行计数,并且响应于达到预定状态而提供控制信号;

耦接到所述存储器通道控制器(640)的擦除器(624),所述擦除器用于控制所述存储器通道控制器(640)以从存储器系统(670)的多个地址顺序地且周期性地读取数据,并且响应于检测到可校正的错误,重写校正的数据;以及

数据处理器(610),所述数据处理器耦接到所述存储器通道控制器(640),并且对所述控制信号作出响应,以响应于所述控制信号利用所述存储器系统(670)执行封装后修复操作。

2. 根据权利要求1所述的封装后存储器修复系统(600),其中所述数据处理器(610)包括:

耦接到所述存储器通道控制器(640)的中央处理单元(CPU)核心(614);以及

耦接到所述CPU核心(614)的固件存储器(612),所述固件存储器用于存储多个指令,所述多个指令在由所述CPU核心(614)执行时,使所述CPU核心(614)控制所述存储器通道控制器(640)以执行所述封装后修复操作。

3. 根据权利要求2所述的封装后存储器修复系统(600),其中所述固件存储器(612)存储基本输入-输出系统(BIOS),并且所述多个指令是所述BIOS的一部分。

4. 根据权利要求2所述的封装后存储器修复系统(600),其中所述数据处理器(610)响应于所述存储器通道控制器(640)检测到预定水平的错误而增加所述擦除器的擦除率。

5. 根据权利要求1所述的封装后存储器修复系统(600),其中:

所述擦除器(624)从预定大小的存储器中读取数据元素,所述预定大小小于所述存储器系统(670)中对应行的大小;并且

所述存储器通道控制器(670)具有所述预定大小的读取/写入缓冲区,所述读取/写入缓冲区用于临时存储所述数据元素。

6. 根据权利要求1所述的封装后存储器修复系统(600),其中当所述第一错误计数器(652)达到终端计数时,达到所述预定状态。

7. 根据权利要求1所述的封装后存储器修复系统(600),其中所述存储器通道控制器(640)包括:

ECC校验电路(646),所述ECC校验电路对所述返回数据和所述多个ECC位作出响应以检测所述返回数据中的错误并确定所述错误是否为可校正的。

8. 根据权利要求7所述的封装后存储器修复系统(600),其还包括:

多个附加错误计数器(654),

其中响应于检测到存储器存取中的ECC错误,所述ECC校验电路(646)还提供以下各项中的一者:提供给所述第一错误计数器(652)的第一递增信号,以及提供给根据所述存储器系统(670)中所述存储器存取的区域选择的多个对应附加错误计数器(654)的多个附加递

增信号。

9. 根据权利要求1所述的封装后存储器修复系统(600),其中所述存储器接口(644)适于耦接到与双倍数据速率(DDR)-物理接口(DFI)规范兼容的接口。

10. 根据权利要求1所述的封装后存储器修复系统(600),其中所述擦除器(624)被实现为硬件电路。

11. 根据权利要求1所述的封装后存储器修复系统(600),其中所述擦除器(624)由在所述数据处理器(610)上执行的软件实现。

12. 一种存储器控制器(630),其包括:

存储器通道控制器(640),所述存储器通道控制器用于将从主机接口(642)接收到的数据存取请求转换为提供给存储器接口(644)的对应存储器存取,并且响应于从所述存储器接口(644)接收到的响应而将返回的数据提供给所述主机接口(642),其中所述响应包括返回的数据和多个错误校正码(ECC)位;

其中所述存储器通道控制器(640)通过基于对应于所述返回数据的所述多个错误校正码(ECC)位执行错误校验来检测所述返回数据中的错误,并且响应于检测到所述返回数据中的所述错误而提供计数信号;以及

耦接到所述存储器通道控制器(640)的第一错误计数器(652),所述第一错误计数器用于响应于所述计数信号而对错误进行计数,并且响应于达到预定状态而提供控制信号。

13. 根据权利要求12所述的存储器控制器(600),其中所述控制信号包括机器校验架构中断。

14. 根据权利要求12所述的存储器控制器(630),其中所述存储器通道控制器(640)包括:

ECC校验电路(646),所述ECC校验电路对所述返回数据和所述多个ECC位作出响应以检测所述返回数据中的所述错误并确定所述错误是否为可校正的。

15. 根据权利要求14所述的存储器控制器(630),其还包括:

多个附加错误计数器(654),

其中所述ECC校验电路(646)还响应于检测到存储器存取中的ECC错误,将从所述计数信号中的选定计数信号提供给所述第一错误计数器(652),并将多个附加递增信号提供给多个对应的附加错误计数器(654),其中所述选定的计数信号针对所述存储器存取根据存储器系统(670)中的所述存储器存取的区域进行选择。

16. 根据权利要求12所述的存储器控制器(630),其中所述存储器接口(644)适于耦接到与双倍数据速率(DDR)-物理接口(DFI)规范兼容的接口。

17. 一种方法,其包括:

将从主机接口(642)接收到的数据存取请求转换为提供给存储器接口(644)的对应存储器存取;

响应于从所述存储器接口(644)接收到的响应,将返回的数据提供给所述主机接口(642),其中所述响应包括所述返回数据和多个错误校正码(ECC)位;

根据所述返回数据和所述多个ECC位来擦除存储器系统(670)中的错误,并修复可校正的错误;以及

响应于在所述擦除期间所述存储器系统(670)的区域中的预定数量的错误,在所述存

存储器系统上执行封装后修复操作。

18. 根据权利要求17所述的方法,其中所述转换包括:

将所述数据存取请求存储在队列(520)中;以及

在存储在所述队列(520)中的多个数据存取请求之间进行仲裁,以在由预定标准确定的时间处提供所述存储器存取。

19. 根据权利要求17所述的方法,其中所述擦除包括:

向所述存储器系统(670)中的下一个地址提供读取命令;

接收来自所述存储器系统(670)的所述响应;

根据所述返回数据和所述多个ECC位执行ECC校验;以及

如果所述执行所述ECC校验指示错误,则使错误计数器(652)递增。

20. 根据权利要求19所述的方法,其还包括:

响应于所述错误计数器(652)达到终端计数,在所述存储器系统(670)上执行所述封装后修复操作。

21. 根据权利要求17所述的方法,其中所述擦除包括:

周期性地且顺序地擦除所述存储器系统(670)中的每个地址的所述可校正错误。

22. 根据权利要求17所述的方法,其还包括:

检测不可校正的错误;以及

将所述不可校正的错误报告给基本输入输出系统(BIOS)。

## 一体式封装后修复装置

### 技术领域

[0001] 本公开总体涉及数据处理系统,并且更具体地涉及具有高可靠性存储器的数据处理系统。

### 背景技术

[0002] 现代动态随机存取存储器 (DRAM) 由于其低成本、高密度和随机存取时间而被用于当今的大多数个人计算机系统和服务中。DRAM 基于小存储器单元,该小存储器单元在电容器中存储电荷以指示存储器单元的状态。电容存储装置是动态的,电容器会随时间推移而失去电荷。因此,必须定期刷新存储器单元。此外,读取操作具有破坏性,因为它会耗尽电容器上的电荷。在访问特定行上的存储器位置之前,通过将行的内容存储在大页面缓冲区内来“激活”该行,该大页面缓冲区的大小可为例如八千字节 (8kB)。在另一行中读取或写入存储器位置之前,必须通过将页面缓冲器内容沿着行重写回存储器单元对当前在页面缓冲区中的存储器行进行“预充电”,这会将电容器充电回到其初始状态。

[0003] 由于大小较小,DRAM 存储器单元易受软错误的影响。软错误是由诸如穿过电容器的 $\alpha$ 粒子、电磁干扰等随机电事件的发生引起的数据错误。因此,软错误不反映电路中的任何根本错误或缺陷。为了校正软错误,存储器制造商通常通过每组八个 DRAM 芯片包括一个额外的 DRAM 芯片来采用所谓的错误校正码 (ECC)。ECC 是与数据一起存储的额外位,其可以允许例如校正一组位中的单位错误,并且检测但不校正多位错误。ECC 允许校正单位错误,因为 ECC 码包含足够的信息来识别失效位的位置,使得在随后的预充电操作期间将位重写到存储器阵列之前可以反转逻辑状态。

[0004] 在读取或写入访问期间,使用 ECC 位实时检测软错误很困难。因此,存储器控制器有时使用“擦除器”以针对软错误执行存储器单元的后台检查。擦除器定期检查存储器线是否存在 ECC 错误。如果擦除器发现可校正的错误,则它会校正错误,从而降低在实际读取或写入访问期间发生错误的概率。擦除器定期检查整个物理存储器空间中的所有存储器位置是否存在此类错误,例如每天一次。

[0005] 另一方面,存储器单元偶尔会经历电路缺陷或“硬”错误,这些错误随时间推移变得更严重,直到存储器单元或一组相邻存储器单元失效。通常,DRAM 在出厂时进行测试以检测硬错误,并通过用冗余行或列代替失效的行或列来校正。然而,在制造之后变得有缺陷的存储器单元的检测和校正变得更加困难。通常,在启动时测试存储器是否存在硬错误,并从系统存储器映射中删除经历过硬错误的存储器部分。然而,如果在启动后发生硬错误,则运行程序可能会崩溃,从而给用户带来不便或数据丢失。此外,没有已知的策略来检测和校正随时间推移而缓慢发展的硬错误,以及由于 ECC 位的可用性而不会导致程序故障或数据丢失的硬错误。

[0006] 为了校正出厂测试后出现的硬错误,联合电子设备工程委员会 (JEDEC) 规定的双倍数据速率第四版 (DDR4) 存储器采用了被称为封装后修复的特征。然而,没有已知的系统能够使用封装后修复特征在程序故障或数据丢失之前简单有效地校正硬错误。

## 附图说明

- [0007] 图1以方框图形式示出根据一些实施例的数据处理系统；
- [0008] 图2以方框图形式示出适合用于图1的数据处理系统中的加速处理单元 (APU) ；
- [0009] 图3以方框图形式示出根据一些实施例的适合用于图2的APU中的存储器控制器和相关联的物理接口 (PHY) ；
- [0010] 图4以方框图形式示出根据一些实施例的适合用于图2的APU中的另一存储器控制器和相关联的PHY；
- [0011] 图5以方框图形式示出根据一些实施例的存储器控制器；
- [0012] 图6以方框图形式示出根据一些实施例的具有一体式封装后修复装置的数据处理系统；
- [0013] 图7示出根据一些实施例的与图6的数据处理系统中的封装后修复装置相关的操作的流程图；并且
- [0014] 图8示出根据一些实施例的封装后修复程序的流程图。
- [0015] 在以下描述中，不同图中使用的相同的附图标记指示相似或相同的项目。除非另外指出，否则单词“耦接”及其相关联的动词形式包括直接连接和通过本领域中已知手段进行的间接电连接，并且除非另外指出，否则对直接连接的任何描述也意味着使用适当形式的间接电连接的替代性实施例。

## 具体实施方式

- [0016] 如下文将以一种形式所述，封装后修复系统包括存储器通道控制器、擦除器、第一错误计数器和数据处理器。存储器通道控制器将从主机接口接收到的数据访问请求转换为提供给存储器接口的对应存储器访问，并且响应于从存储器接口接收到的响应而将返回的数据提供给主机接口，其中响应包括返回的数据和多个错误校正码 (ECC) 位。第一错误计数器耦接到存储器通道控制器并对返回数据中的错误进行计数，并且响应于达到预定状态而提供控制信号。擦除器耦接到存储器通道控制器，该存储器通道控制器用于控制存储器通道控制器以从存储器系统的多个地址顺序地且周期性地读取数据，并且响应于检测到可校正的错误，重写校正的数据。数据处理器耦接到存储器通道控制器，并且对控制信号作出响应，以响应于控制信号利用存储器系统执行封装后修复操作。
- [0017] 在另一种形式中，存储器控制器适合用于在存储器系统上执行封装后修复的系统。存储器控制器包括存储器通道控制器和第一错误计数器。存储器通道控制器将从主机接口接收到的数据访问请求转换为存储器接口上的对应存储器访问，并且响应于从存储器接口接收到的响应而将返回的数据提供给主机接口。响应包括返回的数据和多个错误校正码 (ECC) 位。存储器通道控制器通过基于对应于返回数据的多个错误校正码 (ECC) 位执行错误校验来检测返回数据中的错误，并且响应于检测到返回数据中的错误而提供计数信号。第一错误计数器耦接到存储器通道控制器以响应于计数信号对错误进行计数，并且响应于达到预定状态而提供控制信号，例如机器校验架构 (MCA) 异常。MCA异常允许另一个代理 (例如运行封装后修复固件的数据处理器) 执行封装后修复操作。
- [0018] 在另一种形式中，将从主机接口接收到的数据访问请求转换为提供给存储器接口的对应存储器访问。响应于从存储器接口接收到的响应，将返回的数据提供给主机接口，其

中响应包括返回的数据和多个错误校正码 (ECC) 位。根据返回的数据和多个ECC位擦除存储器系统的错误,并且修复可校正的错误。响应于擦除期间存储器系统的区域中预定数量的错误,在存储器系统上执行封装后修复操作。

[0019] 图1以方框图形式示出根据一些实施例的数据处理系统100。数据处理系统100通常包括呈加速处理单元 (APU) 形式的数据处理器110、存储器系统120、外围部件快速互连 (PCIe) 系统150、通用串行总线 (USB) 系统160和磁盘驱动器170。数据处理器110充当数据处理系统100的中央处理单元 (CPU),并且提供可用于现代计算机系统中的各种总线和接口。这些接口包括两个双倍数据速率 (DDR<sub>x</sub>) 存储器通道、用于连接到PCIe链路的PCIe根复合体、用于连接到USB网络的USB控制器以及通向串行高级技术附件 (SATA) 大容量存储装置的接口。

[0020] 存储器系统120包括存储器通道130和存储器通道140。存储器通道130包括连接到DDR<sub>x</sub>总线132的一组双列直插存储器模块 (DIMM),所述DIMM包括在该实例中对应于单独存储排 (rank) 的代表性DIMM 134、136和138。同样,存储器通道140包括连接到DDR<sub>x</sub>总线142的一组DIMM,所述DIMM包括代表性DIMM 144、146和148。

[0021] PCIe系统150包括连接到数据处理器110中的PCIe根复合体的PCIe交换机152、PCIe装置154、PCIe装置156和PCIe装置158。PCIe装置156继而连接到系统基本输入/输出系统 (BIOS) 存储器157。系统BIOS存储器157可以是多种非易失性存储器类型中的任一种,例如只读存储器 (ROM)、快闪电可擦除可编程ROM (EEPROM) 等。

[0022] USB系统160包括连接到数据处理器110中的USB主控的USB集线器162,以及各自连接到USB集线器162的代表性USB装置164、166和168。USB装置164、166和168可以是诸如键盘、鼠标、快闪EEPROM端口等的装置。

[0023] 磁盘驱动器170经由SATA总线连接到数据处理器110,并且为操作系统、应用程序、应用文件等提供大容量存储装置。

[0024] 数据处理系统100通过提供存储器通道130和存储器通道140而适合用于现代计算应用中。每个存储器通道130和140可连接到最先进的DDR存储器,例如DDR第四版 (DDR4)、低功率DDR4 (LPDDR4)、图形DDR第五版 (gDDR5) 和高带宽存储器 (HBM),并且可适于未来的存储器技术。这些存储器提供高总线带宽和高速操作。同时,这些存储器还提供低功率模式以节省例如膝上型计算机等电池供电的应用的功率,并且还提供内置的热监测。

[0025] 图2以方框图形式示出适合用于图1的数据处理系统100中的APU 200。APU 200通常包括中央处理单元 (CPU) 核心复合体210、图形核心220、一组显示器引擎230、存储器管理集线器240、数据组构250、一组外围控制器260、一组外围总线控制器270、系统管理单元 (SMU) 280和一组存储器控制器290。

[0026] CPU核心复合体210包括CPU核心212和CPU核心214。在该实例中,CPU核心复合体210包括两个CPU核心,但在其他实施例中,CPU核心复合体可包括任意数量的CPU核心。CPU核心212和214中的每一个均双向地连接到会形成控制组构的系统管理网络 (SMN) 以及数据组构250,并且能够将存储器访问请求提供给数据组构250。CPU核心212和214中的每一个可为单一核心,或可另外为具有共享例如高速缓存等特定资源的两个或更多个单一核心的核心复合体。

[0027] 图形核心220为高性能图形处理单元 (GPU),其能够以高度集成且并行的方式执行

图形操作,例如顶点处理、片段处理、明暗处理、纹理混合等。图形核心220双向地连接到SMN和数据组250,并且能够将存储器访问请求提供给数据组250。就此而言,APU 200可支持其中CPU核心复合体210和图形核心220共享同一存储器空间的统一存储器架构,或其中CPU核心复合体210和图形核心220共享存储器空间的一部分同时图形核心220还使用CPU核心复合体210不可访问的专用图形存储器的存储器架构。

[0028] 显示器引擎230渲染并光栅化通过图形核心220生成的对象以显示在监视器上。图形核心220和显示器引擎230双向地连接到共同的存储器管理集线器240以统一转变成存储器系统120中的适合的地址,并且存储器管理集线器240双向地连接到数据组250以生成此类存储器访问并接收从存储器系统返回的读取数据。

[0029] 数据组250包括用于在任何存储器访问代理与存储器控制器290之间路由存储器访问请求和存储器响应的交叉交换机。所述数据组还包括通过BIOS限定的系统存储器映射以用于基于系统配置来确定存储器访问的目的地,以及用于每一虚拟连接的缓冲器。

[0030] 外围控制器260包括USB控制器262和SATA接口控制器264,所述控制器中的每一个均双向地连接到系统集线器266和SMN总线。这两个控制器仅举例示出可在APU 200中使用的外围控制器。

[0031] 外围总线控制器270包括系统控制器或“南桥”(SB) 272和PCIe控制器274,所述控制器中的每一个均双向地连接到输入/输出(I/O)集线器276和SMN总线。I/O集线器276还双向地连接到系统集线器266并连接到数据组250。因此,例如,CPU核心可通过访问来对USB控制器262、SATA接口控制器264、SB 272或PCIe控制器274中的寄存器进行编程,数据组250通过I/O集线器276来路由所述访问。

[0032] SMU 280为本地控制器,其控制APU 200上的资源的操作并同步其间的通信。SMU 280管理APU 200上的各种处理器的加电定序,并经由重置、启用和其他信号来控制多个芯片外装置。SMU 280包括图2中未示出的一个或多个时钟源,例如锁相环路(PLL),以便为APU 200的部件中的每一个提供时钟信号。SMU 280还管理各种处理器和其他功能块的电源,并且可从CPU核心212和214以及图形核心220接收所测量的功耗值以确定适合的电源状态。

[0033] APU 200还实施各种系统监测和节电功能。确切地,一个系统监测功能为热监测。例如,如果APU 200变热,则SMU 280可减小CPU核心212和214和/或图形核心220的频率和电压。如果APU 200变得过热,则可完全关闭所述APU 200。热事件还可通过SMU 280经由SMN总线从外部传感器接收,并且SMU 280可减小时钟频率和/或电源电压以作为响应。

[0034] 图3以方框图形式示出根据一些实施例的适合用于图2的APU 200中的存储器控制器300和相关联的物理接口(PHY) 330。存储器控制器300包括存储器通道310和功率引擎320。存储器通道310包括主机接口312、存储器通道控制器314和物理接口316。主机接口312经由可扩展数据端口(SDP)将存储器通道控制器314双向地连接到数据组250。物理接口316经由遵守DDR-PHY接口规范(DFI)的总线将存储器通道控制器314双向地连接到PHY 330。功率引擎320经由SMN总线双向地连接到SMU 280,经由高级外围总线(APB)双向地连接到PHY 330,并且还双向地连接到存储器通道控制器314。PHY 330具有通向存储器通道(例如图1的存储器通道130或存储器通道140)的双向连接。存储器控制器300为用于使用单一存储器通道控制器314的单一存储器通道的存储器控制器的实例化,并且具有功率引擎320以按下文将进一步描述的方式控制存储器通道控制器314的操作。

[0035] 图4以方框图形式示出根据一些实施例的适合用于图2的APU 200中的另一存储器控制器400和相关联的PHY 440和450。存储器控制器400包括存储器通道410和420以及功率引擎430。存储器通道410包括主机接口412、存储器通道控制器414和物理接口416。主机接口412经由SDP将存储器通道控制器414双向地连接到数据组构250。物理接口416将存储器通道控制器414双向地连接到PHY 440,并遵守DFI规范。存储器通道420包括主机接口422、存储器通道控制器424和物理接口426。主机接口422经由另一SDP将存储器通道控制器424双向地连接到数据组构250。物理接口426将存储器通道控制器424双向地连接到PHY 450,并遵守DFI规范。功率引擎430经由SMN总线双向地连接到SMU 280,经由APB双向地连接到PHY 440和450,并且还双向地连接到存储器通道控制器414和424。PHY 440具有通向存储器通道(例如图1的存储器通道130)的双向连接。PHY 450具有通向存储器通道(例如图1的存储器通道140)的双向连接。存储器控制器400为具有两个存储器通道控制器的存储器控制器的实例化,并且使用共享的功率引擎430以下文将进一步描述的方式控制存储器通道控制器414和存储器通道控制器424两者的操作。

[0036] 图5以方框图形式示出根据一些实施例的存储器控制器500。存储器控制器500通常包括存储器通道控制器510和功率控制器550。存储器通道控制器510通常包括接口512、队列514、命令队列520、地址生成器522、内容可寻址存储器(CAM) 524、回放队列530、刷新逻辑块532、定时块534、页表536、仲裁器538、错误校正码(ECC) 校验块542、ECC生成块544和数据缓冲区(DB) 546。

[0037] 接口512具有经由外部总线到数据组构250的第一双向连接,并且具有输出。在存储器控制器500中,此外部总线与由英国剑桥的ARM Holdings, PLC指定的高级可扩展接口第四版(被称为“AXI4”)相容,但在其他实施例中可为其他类型的接口。接口512将存储器访问请求从被称为FCLK(或MEMCLK)域的第一时钟域转变为存储器控制器500内部的被称为UCLK域的第二时钟域。类似地,队列514将存储器访问从UCLK域提供到与DFI接口相关联的DFICLK域。

[0038] 地址生成器522将经由AXI4总线从数据组构250接收到的存储器访问请求的地址进行解码。存储器访问请求包括以标准化格式表示的物理地址空间中的访问地址。地址生成器522将规范化地址转换成可用于对存储器系统120中的实际存储器装置进行寻址以及用于有效地调度相关访问的格式。此格式包括区域识别符,所述区域识别符使存储器访问请求与特定存储排、行地址、列地址、存储体地址以及存储体群组相关联。在启动时,系统BIOS查询存储器系统120中的存储器装置以确定所述存储器装置的大小和配置,并对与地址生成器522相关联的一组配置寄存器进行编程。地址生成器522使用存储在配置寄存器中的配置将规范化地址转变成适合的格式。命令队列520为从数据处理系统100中的存储器访问代理(例如CPU核心212和214以及图形核心220)接收到的存储器访问请求的队列。命令队列520存储通过地址生成器522解码的地址字段,以及允许仲裁器538有效地选择存储器访问的其他地址信息,包括访问类型和服务质量(QoS) 识别符。CAM 524包括用以实施排序规则的信息,例如写后写(WAW)和写后读(RAW)排序规则。

[0039] 回放队列530是用于存储由仲裁器538选取的等待响应的存储器访问的临时队列,所述响应诸如地址和命令奇偶校验响应、DDR4DRAM的写循环冗余校验(CRC) 响应或gDDR5DRAM的写入和读取CRC响应。回放队列530访问ECC校验块542以确定返回的ECC是正确

的还是指示错误。回放队列530允许访问在这些循环中的一个的奇偶校验错误或CRC错误的情况下回放。

[0040] 刷新逻辑532包括用于各种掉电、刷新和终止电阻 (ZQ) 校准周期的状态机,其与从存储器访问代理接收到的正常读取和写入存储器访问请求分开生成。例如,如果存储排(memory rank)处于预充电掉电,则所述存储排必须定期被唤醒以运行刷新周期。刷新逻辑532周期性地生成刷新命令,以防止由DRAM芯片中的存储器单元的存储电容器的电荷泄漏引起的数据错误。另外,刷新逻辑532定期校准ZQ以防止由于系统中的热变化导致的片内终结电阻中的失配。

[0041] 仲裁器538双向地连接到命令队列520,并且是存储器通道控制器510的心脏。所述仲裁器通过智能地调度访问以提高存储器总线的利用率来提高效率。仲裁器538使用定时块534以通过基于DRAM定时参数来确定命令队列520中的特定访问是否适合于发出而实施恰当的定时关系。例如,每个DRAM在激活命令之间具有最小指定时间,称为“ $t_{RC}$ ”。定时块534保持一组基于在JEDEC规范中指定的该定时参数和其他定时参数来确定合格性的计数器,并且双向连接到回放队列530。页表536保持关于仲裁器538的存储器通道的每一存储体和存储排中的活动页的状态信息,并且双向地连接到回放队列530。

[0042] 响应于从接口512接收到的写入存储器访问请求,ECC生成块544根据写入数据来计算ECC。DB 546针对接收到的存储器访问请求存储写入数据和ECC。当仲裁器538选取对应的写入访问以分派到存储器通道时,所述DB将组合的写入数据/ECC输出到队列514。

[0043] 功率控制器550通常包括通向高级可扩展接口、第一版 (AXI) 的接口552、APB接口554和功率引擎560。接口552具有通向SMN的第一双向连接,包括用于接收在图5中单独地示出的标记为“事件\_n”的事件信号的输入端,以及输出端。APB接口554具有连接到接口552的输出端的输入端,以及用于经由APB连接到PHY的输出端。功率引擎560具有连接到接口552的输出端的输入端,以及连接到队列514的输入端的输出端。功率引擎560包括一组配置寄存器562、微控制器( $\mu C$ ) 564、自刷新控制器 (SLFREF/PE) 566和可靠的读取/写入定时引擎 (RRW/TE) 568。配置寄存器562经由AXI总线来编程,并且存储配置信息以控制存储器控制器500中的各种块的操作。因此,配置寄存器562具有连接到图5中未详细示出的这些块的输出端。自刷新控制器566为引擎,所述引擎允许除通过刷新逻辑532自动地产生刷新外的刷新的手动产生。可靠的读取/写入定时引擎568为存储器或I/O装置提供连续的存储器访问流,以用于诸如DDR接口最大读取延迟 (MRL) 训练和环回测试之类的目的。

[0044] 存储器通道控制器510包括允许所述存储器通道控制器选取存储器访问以分派到相关联的存储器通道的电路。为做出期望的仲裁决定,地址生成器522将地址信息解码成预解码信息,包括存储器系统中的存储排、行地址、列地址、存储体地址和存储体群组,并且命令队列520存储预解码信息。配置寄存器562存储配置信息以确定地址生成器522如何对接收到的地址信息进行解码。仲裁器538使用经解码的地址信息、由定时块534指示的定时合格性信息以及由页表536指示的活动页信息来有效地调度存储器访问,同时遵守其他标准,例如QoS要求。例如,仲裁器538实施对打开页进行访问的偏好以避免改变存储器页所需的预充电和激活命令的开销,并通过将对一个存储体的开销访问与对另一个存储体的读取和写入访问进行交错来隐藏开销访问。特别是在正常操作期间,仲裁器538通常在不同的存储体中保持页面打开,直到在选择不同的页面之前需要对这些存储体进行预充电。

[0045] 图6以方框图形式示出根据一些实施例的具有一体式封装后修复装置的数据处理系统600。数据处理系统600通常包括数据处理器610、数据组构620、存储器控制器630、PHY 660和存储器系统670。数据处理器610包括存储器612,该存储器存储封装后修复固件和CPU核心614。存储器612对应于图1的系统BIOS 157,而CPU核心614对应于CPU核心复合体210的CPU核心,例如CPU核心212或CPU核心214。在图1的多核架构中,CPU核心中的一者被称为BIOS服务处理器(BSP)并且连接到系统BIOS 157。在数据处理器610中,CPU核心614是BSP并且双向连接到存储器612以读取和执行包括封装后修复固件的BIOS指令。CPU核心614还双向连接到数据组构620,以向存储器系统发出读取和写入命令并从存储器系统接收读取响应,并且发出并响应于其自身与其他CPU核心复合体之间的对可存储在CPU核心的内部高速缓存中的共享数据的探测。

[0046] 数据组构620包括相干从装置622和控制块626。相干从装置622是响应于读取和写入请求以生成访问并将所述访问提供给适当的存储器控制器(例如存储器控制器630)的引擎。相干从装置622还包括擦除器624。擦除器624是硬件状态机,其用于通过循环存储器中的每个位置的存储器地址来执行擦除操作,包括每个存储器芯片的每个存储体和系统中的存储器芯片的每个存储排。在擦除操作中,相干从装置622向存储器系统670中的连续位置发出读取-修改-写入周期,使得可以检测存储器错误并且一些错误被校正。另一方面,控制块626跟踪报告的错误,对于不可校正的存储器错误,发信号通知机器校验中断。控制块626还发信号通知CPU核心614需要使用存储在存储器612中的封装后修复来执行封装后修复操作。

[0047] 存储器控制器630可使用图5的存储器控制器500来实现。然而,如图6所示的存储器控制器630仅包括存储器控制器的用于实现存储器错误检测、校正和封装后修复确定的那些部分。如图所示,存储器控制器630包括存储器通道控制器640和功率引擎650。存储器通道控制器640包括主机接口642、PHY接口644、内部数据路径646和ECC校验电路648。内部数据路径646用于传导数据和ECC信号,并在内部连接主机接口642、PHY接口644和ECC校验电路648。主机接口642具有连接到AXI4总线的双向端口,以及用于连接到内部数据路径646的双向端口。PHY接口644具有连接到内部数据路径646的双向端口,以及用于连接到DFI总线的双向端口。ECC校验电路648具有:连接到内部数据路径646的双向端口;以及一组八个输出端,其中图6示出分别对应于存储排0和存储排7的标记为“INC\_CS0”和“INC\_CS7”的两个代表性输出端。功率引擎650包括一组八个计数器,包括代表性计数器652和654,它们分别接收信号INC\_CS0和INC\_CS7中的相应信号并且对应于存储排0和存储排7。每个计数器具有用于经由AXI总线提供计数器溢出信号的输出端,以控制数据组构620中的块626。

[0048] PHY 660具有经由DFI总线连接到存储器控制器630的双向端口,以及连接到标记为“DDR4接口”的存储器接口的双向端口,所述存储器接口用于连接到存储器系统670。PHY 660可以如上文关于图3的DDR<sub>x</sub> PHY 330以及图5的DDR<sub>x</sub> PHY 440和450所述那样实现。

[0049] 存储器系统670是DDR4存储器系统,并且经由DDR4接口连接到PHY 660。DDR4存储器支持封装后修复,因此以下文将更详细地描述的方式与数据处理系统600交互。

[0050] 在操作中,数据处理系统600使用可用资源有效地执行存储器可靠性校验,修复可校正的ECC错误,报告不可校正的ECC错误以及封装后修复。它还利用后台擦除操作来检测硬错误,并使用一些DRAM(例如DDR4 DRAM)的封装后修复特征在发生不可校正的错误之前

校正它们。

[0051] 数据处理系统600在数据处理器610、数据组构620和存储器控制器630之间有效地分发封装后检测和校正任务。数据组构620包括擦除器624,以执行存储器系统670中的所有存储器位置的后台擦除。由于擦除是低带宽后台操作,因此可将单个擦除器与多个存储器控制器结合使用,以仅用少量电路区域擦除整个存储器空间。

[0052] 擦除器624具有两种操作模式。首先,它具有控制模式,在此模式下执行后台擦除。控制模式下的后台擦除涉及读取每个存储器位置以确定是否存在ECC错误。如果特定存储器位置没有ECC错误,则擦除器624在继续读取存储器空间中的后续存储器位置之前重写数据。在一个实施例中,无论是否检测到可校正的ECC错误,擦除器624都将重写数据,因为一些DRAM实现片上ECC,并且后续的写入周期确保在DRAM中被检测到而校正并因此未被擦除器624发现的可校正ECC错误将导致更新的数据被重写到页面缓冲区。然后在后续的预充电命令中,校正的数据将被写入存储器阵列以校正错误。然而,如果没有后续的写入操作,该错误则永远不会在存储器阵列中校正。在替代性实施例中,擦除器624执行读取周期,但仅重写关于检测到的错误的错误的数据。如果擦除器624检测到可校正的错误,则它将错误报告为机器校验架构(MCA)异常。

[0053] 擦除器624以特定速率擦除存储器系统670中的所有存储器位置。例如,它可以遵循每天对所有存储器位置执行一次擦除的规则。在所示实施例中,擦除器624读取大小等于CPU核心的高速缓存行(通常为64字节)的存储器位置,其可以使用长度为8的突发在单个突发中从存储器系统670有效地读取。因此,它包括64字节缓冲区以及ECC检测和校正电路。然而,在其他实施例中,擦除器624可以读取各种其他大小的存储器位置,例如区域或页面。

[0054] 其次,擦除器624具有重定向模式。如上所述,响应于遇到ECC错误的系统访问,擦除器624将错误报告给系统BIOS。例如,DDR<sub>x</sub> DIMM的通用ECC机制支持单位错误检测和校正以及多位错误检测。ECC错误可能是由 $\alpha$ 粒子改变单个存储器单元的内容引起的,且因此表示不需要永久校正的随机错误。响应于报告可校正的ECC错误,系统BIOS重定向擦除器624以执行存储器位置的读写操作。然而,如果擦除器624检测到不可校正的错误,例如多位错误,则它还使用MCA异常报告该错误,允许数据处理器610采取不同的校正措施。例如,软件错误处理程序可以停止正在使用损坏数据的软件应用程序,或者重置数据处理器610。

[0055] 擦除器624将每个读取请求发给存储器控制器630。存储器控制器630以根据正常操作仲裁确定的顺序发出读取请求。在接收到所请求的数据时,存储器控制器630执行其自己的ECC校验并确定所计算的ECC是否与从存储器返回的ECC匹配。如果其未检测到错误,则擦除器624前进至下一个存储器位置。然而,如果其检测到错误,则其指示通过激活递增信号为引擎650供电的错误。最后,如果错误计数值超过阈值,则计数器向控制块626报告错误。然后,控制块626采取适当的补救措施,该补救措施可包括封装后修复操作。

[0056] 通过将擦除器624集成到封装后修复流程中,数据处理系统600使用现有电路块来解锁DDR4封装后修复特征的能力。通过使用存储器控制器630中可用的复制和重写操作,避免了对大型读取/写入缓冲区的需要。例如,对于当前存储器组织和密度,行的大小可为8千字节(kB)并利用读取/写入操作执行这种类型的操作,并且计数器将需要8kB缓冲区来评估整行存储器。擦除器624还允许将数据从失效行迁移为已发生的后台任务。

[0057] 该架构具有若干优点。首先,该架构是有效的,因为它利用了可能已可用于检测和

校正软错误的后台擦除。其次,它使用少量附加硬件,例如ECC校验电路648和计数器组,以跟踪可能最终触发封装后修复操作的错误。第三,它使用能够检测高速率可校正错误的计数器,因此在实际程序失效或数据丢失之前会触发封装后修复操作。

[0058] 图7示出根据一些实施例的与图6的数据处理系统600中的封装后修复装置相关的操作的流程图700。操作从动作框710开始,其中系统BIOS或操作系统例如在启动之后的某个时间启动擦除器624。在动作框720中,擦除器624将读取请求作为读取-修改-写入(RMW)周期的一部分发给下一个地址。在动作框730中,存储器控制器630接收具有返回的读取数据和对应ECC位的读取响应。在动作框740中,存储器控制器630使用ECC校验电路638对数据和对应的ECC位执行ECC校验。然后,流程前进至子流程750。

[0059] 在子流程750中,决策框751确定ECC校验电路648是否检测到数据错误。如果是,则在动作框752处,存储器控制器630使错误计数器递增。决策框753确定错误计数器是否已达到终端计数。检测计数器是否已达到终端计数可以有效地实现为递增计数器的溢出位,或者预加载和递减计数器的下溢位。如果计数器已达到终端计数,则流程前进至封装后修复子流程754。如果否,则流程前进至决策框755。

[0060] 决策框755确定ECC错误是否可校正,例如它是否为单位错误。如果否,则流程前进至动作框756,其中存储器控制器630通过机器校验架构(MCA)异常向操作系统报告不可校正的错误。这种不可校正的错误将导致操作系统采取适当的动作,例如终止进程或程序,重新启动系统等。如果错误是可校正的,则在动作框757中,存储器控制器630将该错误报告给擦除器624。然后在动作框758中,擦除器624重定向存储器控制器630以重写校正的数据而不使计数器递增。

[0061] 在动作框760中,擦除器624使其地址递增以准备下一个擦除周期,然后在动作框770中,擦除器624等待一段时间,然后流程返回到动作框720。在一些实施例中,擦除器624能够以特定速率擦除,例如每天对整个物理存储器进行一次擦除,但是可以例如响应于高水平的错误而加快擦除速率。

[0062] 图8示出根据一些实施例的封装后修复子流程的流程图800。流程图800假设存储器系统670包括DDR4 DRAM,但是可以根据本文所述的操作为同样支持封装后修复的不同DRAM定义对应的流程。流程从框810开始。在动作框820中,存储器控制器630对相关存储器芯片的所有存储体进行预充电,从而允许更新它们的模式寄存器。在动作框830中,存储器控制器630通过写入由DDR4标准定义的某些模式寄存器位来禁用相关DRAM的数据总线反转(DBI)和循环冗余校验(CRC)模式。在动作框840中,存储器控制器630通过将模式寄存器4的位13(MR4[13])设置为1来使相关存储器芯片进入封装后修复模式。注意,在动作框830和840之间,一些DRAM在启用封装后修复之前需要扩展的解锁序列,以防止随机软件错误引起无意的封装后修复操作。在动作框850中,相关DRAM执行封装后修复操作。在动作框860中,存储器控制器630通过将MR4[13]设置为0,使相关存储器芯片退出封装后修复模式。在动作框870中,流程返回到流程图700所示的主流程,具体讲返回到动作框760。

[0063] 因此,封装后修复系统有效地提供增强的系统可靠性。当系统确定需要封装后修复时,它选择性地在DRAM上执行封装后修复,使得在封装后修复期间引入的任何错误仍然可由系统ECC码校正。典型的DRAM存储排包括使用18乘四(x4)DRAM的72位存储器总线。系统600具有x4“chipkill”校正码,允许一次修复一个x4装置。执行封装后修复固件将破坏单个

x4 DRAM的对应页面的当前内容。然而,在封装后修复操作之后,这些错误将保持可校正的错误,并且可以在正常访问这些存储器位置期间在线校正,而不会影响请求方。因此,擦除器624的RMW访问(通过重定向或定期擦除)将重新生成ECC和数据的完整DRAM内容,包括ECC访问所破坏的位。该操作在不需要大缓冲区的情况下进行,以保持DRAM中的数据副本被修复或者长时间停止来自请求方的存储器访问。在其他实施例中,系统600可以将擦除器624实现为系统固件或软件,作为在进行存储器访问请求的处理器上执行的程序。

[0064] 图7和图8的方法以及数据处理系统600的一些操作可以利用硬件和软件的各种组合来实现。例如,ECC校验可以用在控制器上运行的固件中的硬件电路来完成。此外,虽然在所示实施例中,擦除器624被实现为驻留在数据组构620中的硬件电路,但在其他实施例中,擦除器624也可以是以系统固件或软件实现的软件引擎。软件部件中的一些可存储在计算机可读存储介质中,以便由至少一个处理器执行。此外,图7和图8所示的方法中的一些或全部还可以由存储在计算机可读存储介质中并且由至少一个处理器执行的指令来管理。图7和图8所示的每个操作可以对应于存储在非暂时性计算机存储器或计算机可读存储介质中的指令。在各种实施例中,非暂时性计算机可读存储介质包括:磁盘或光盘存储装置、例如快闪存储器的固态存储装置或者一个或多个其他非易失性存储器装置。存储在非暂时性计算机可读存储介质上的计算机可读指令可以是源代码、汇编语言代码、目标代码或者可由一个或多个处理器解译和/或执行的其他指令格式。

[0065] 图6的数据处理系统600或其任何部分可以由呈数据库或其他可由程序读取并直接或间接地用于制造集成电路的数据结构形式的计算机可访问数据结构来描述或表示。例如,该数据结构可以是采用诸如Verilog或VHDL的高级设计语言(HDL)对硬件功能的行为级描述或寄存器传输级(RTL)描述。所述描述可以通过合成工具来读取,所述合成工具可以对描述进行合成,以便从合成库产生包括一系列门的网表。所述网表包括门集合,所述门还表示包括集成电路的硬件的功能性。随后可放置和布线网表以产生描述将应用于掩模的几何形状的数据集。随后可在各种半导体制造步骤中使用掩模来产生集成电路。替代地,计算机可访问存储介质上的数据库根据需要可为网表(具有或不具有合成库)或数据集或图形数据系统(GDS) II数据。

[0066] 虽然对具体实施例进行了描述,但是对这些实施例的各种修改对于本领域的技术人员来说是显而易见的。例如,从DDR4 DRAM角度描述了封装后修复操作,但该操作也可容易地适于当前可能存在的或可在将来被指定为支持封装后修复的其他存储器。此外,虽然后台擦除器被方便地置于数据组构中,因此它可以控制多个存储器控制器,而在其他实施例中,它可以位于存储器控制器中。此外,虽然所公开的实施例使用每字节一个额外ECC位来支持单位检测和校正以及多位检测,但在其他实施例中,可采用使用更多位并且能够检测和校正更多位的其他ECC方案。

[0067] 因此,所附权利要求意图涵盖落入所公开实施例的范围内的对所公开实施例的所有修改。

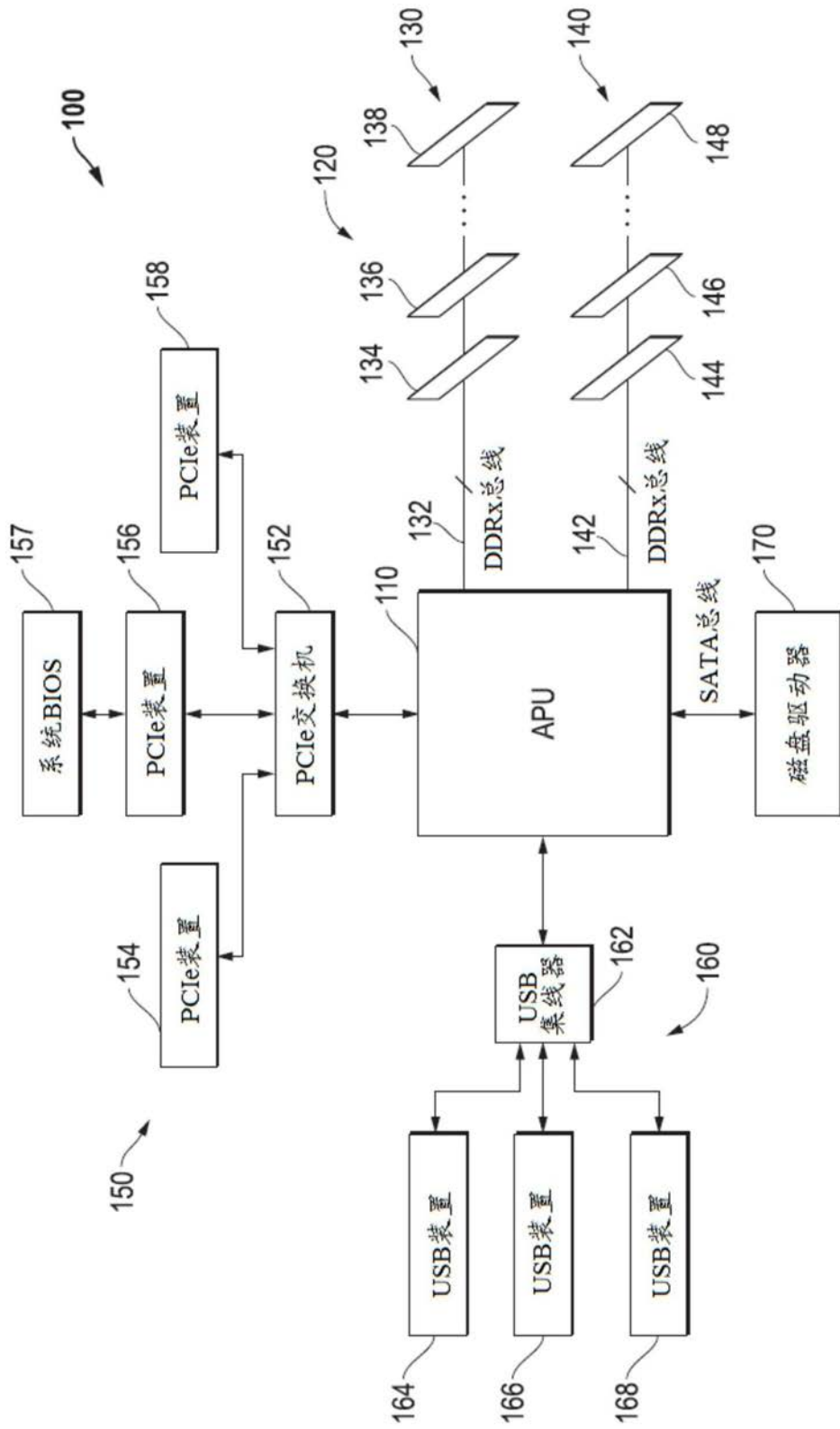


图1

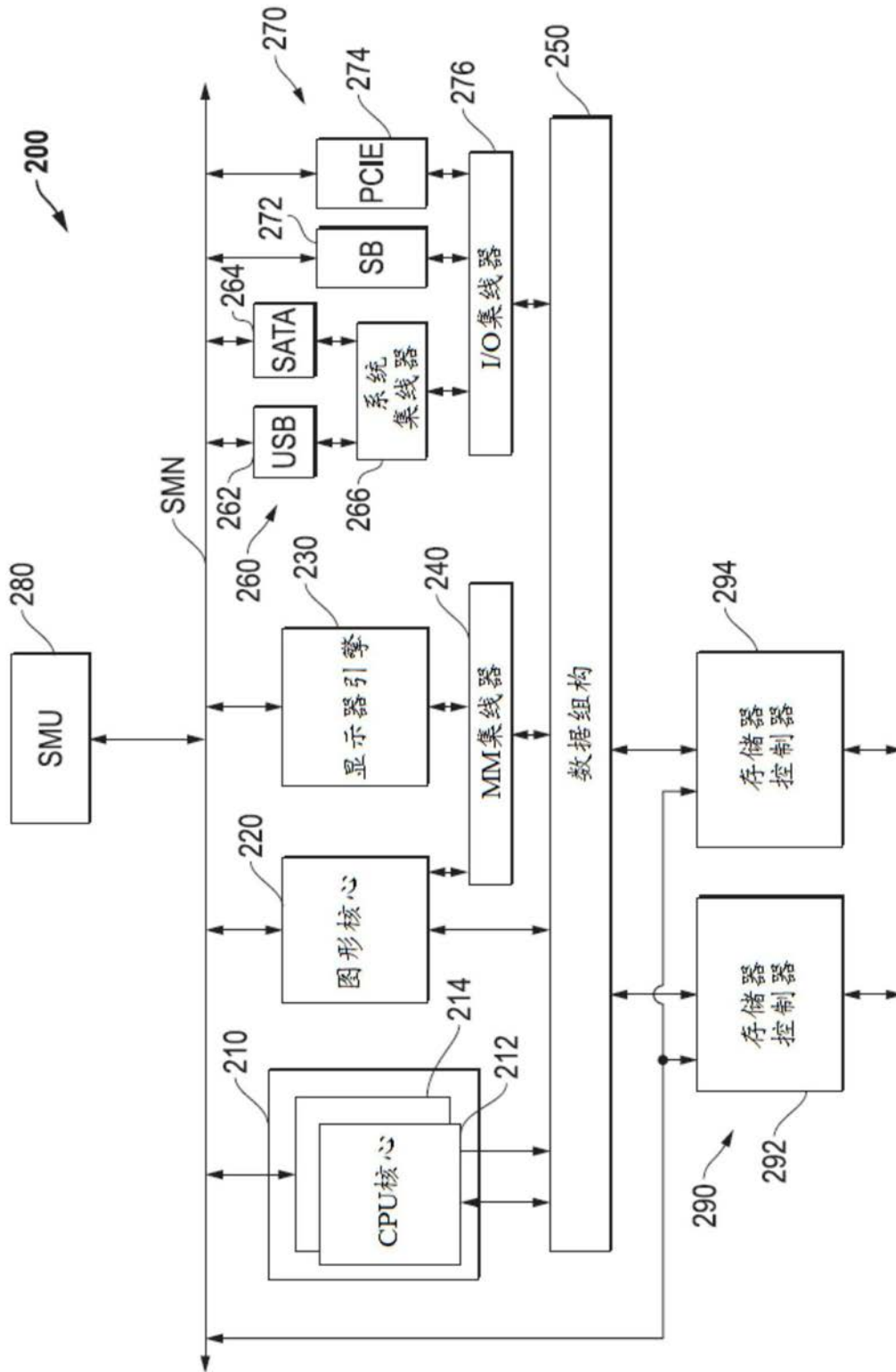


图2

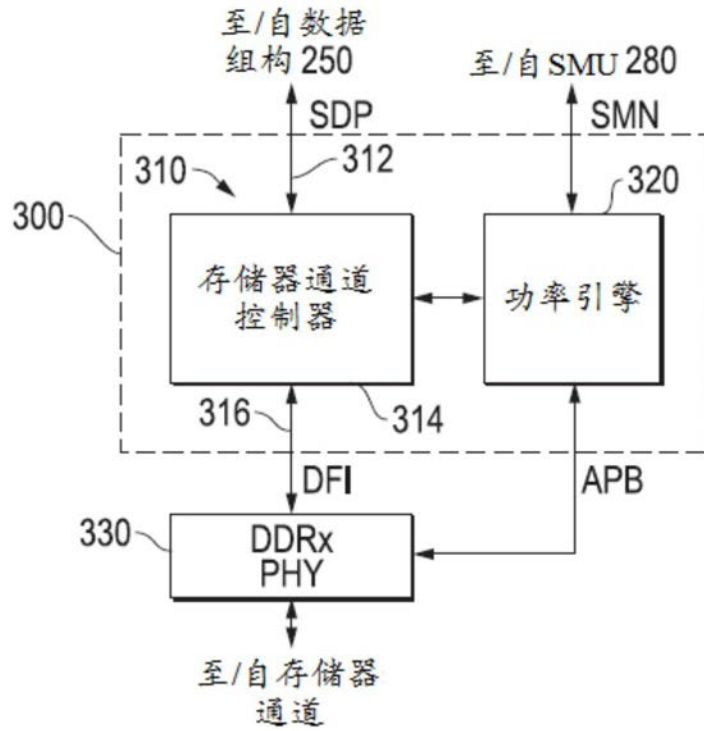


图3

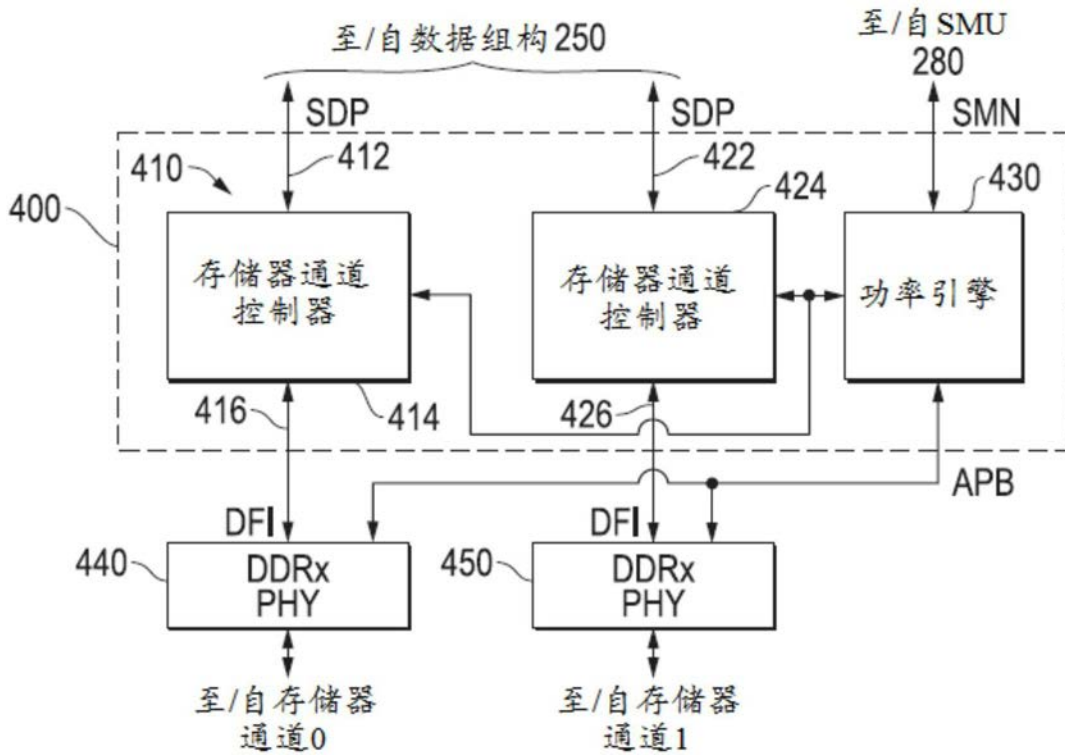


图4

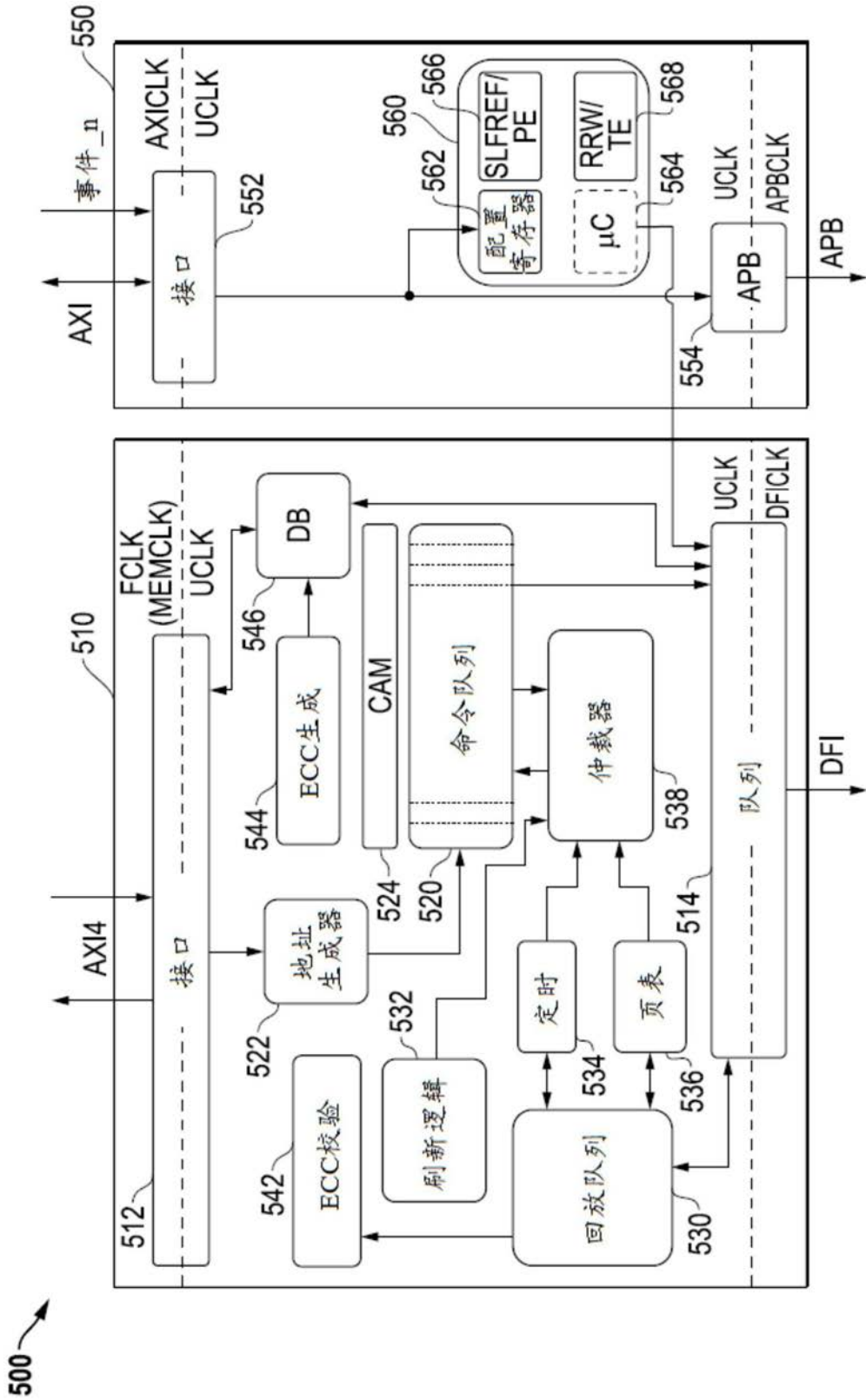


图5

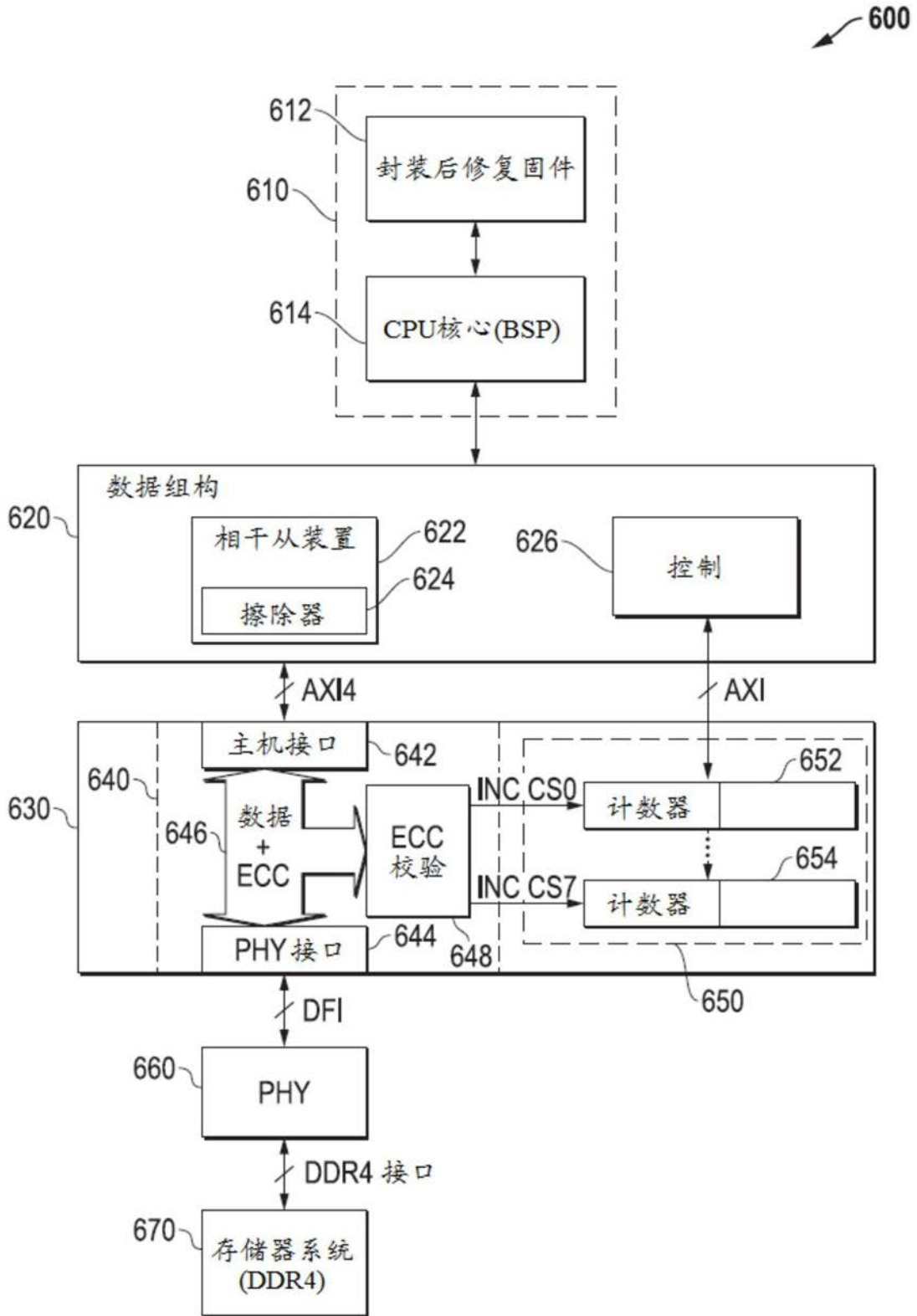


图6

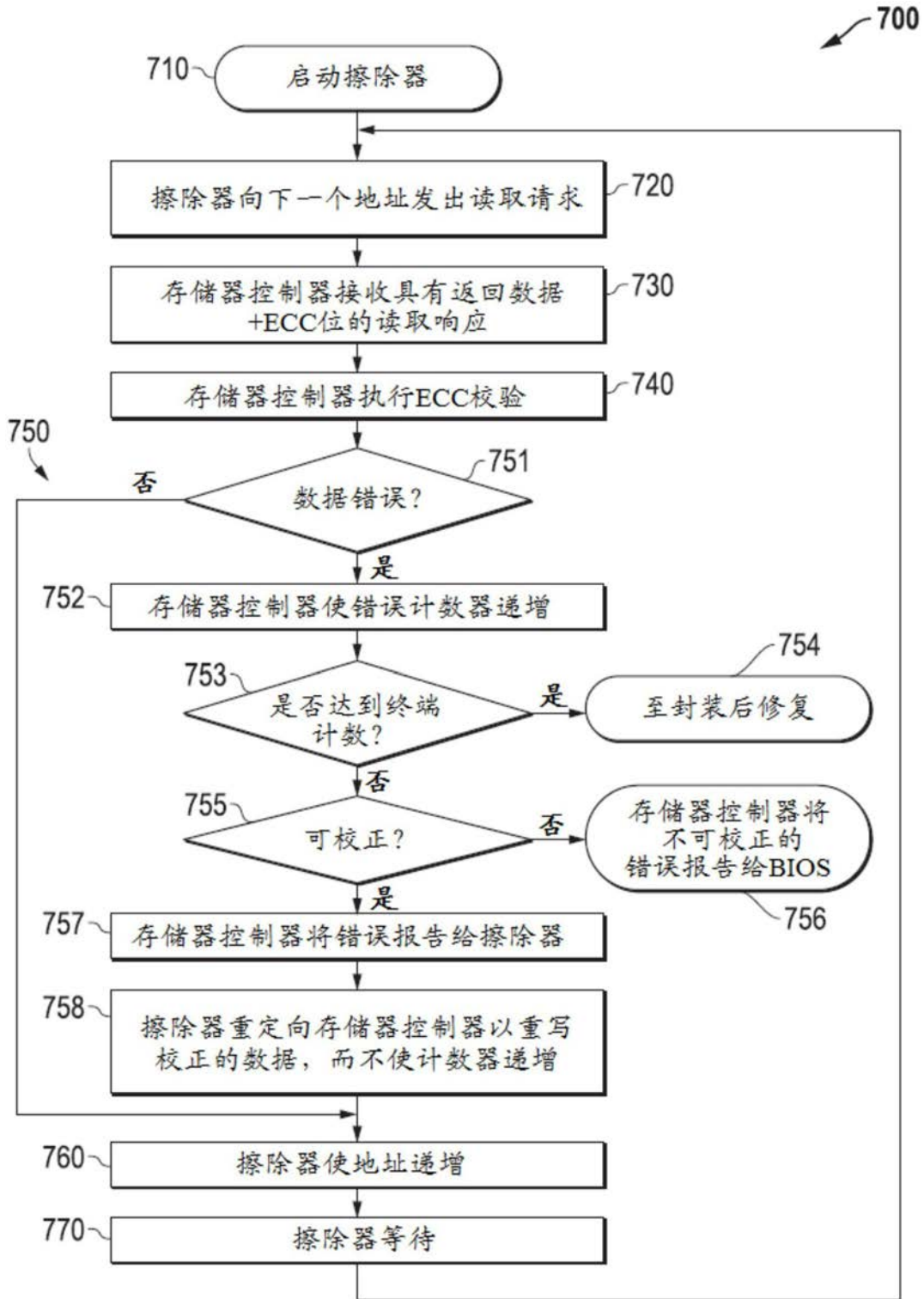


图7

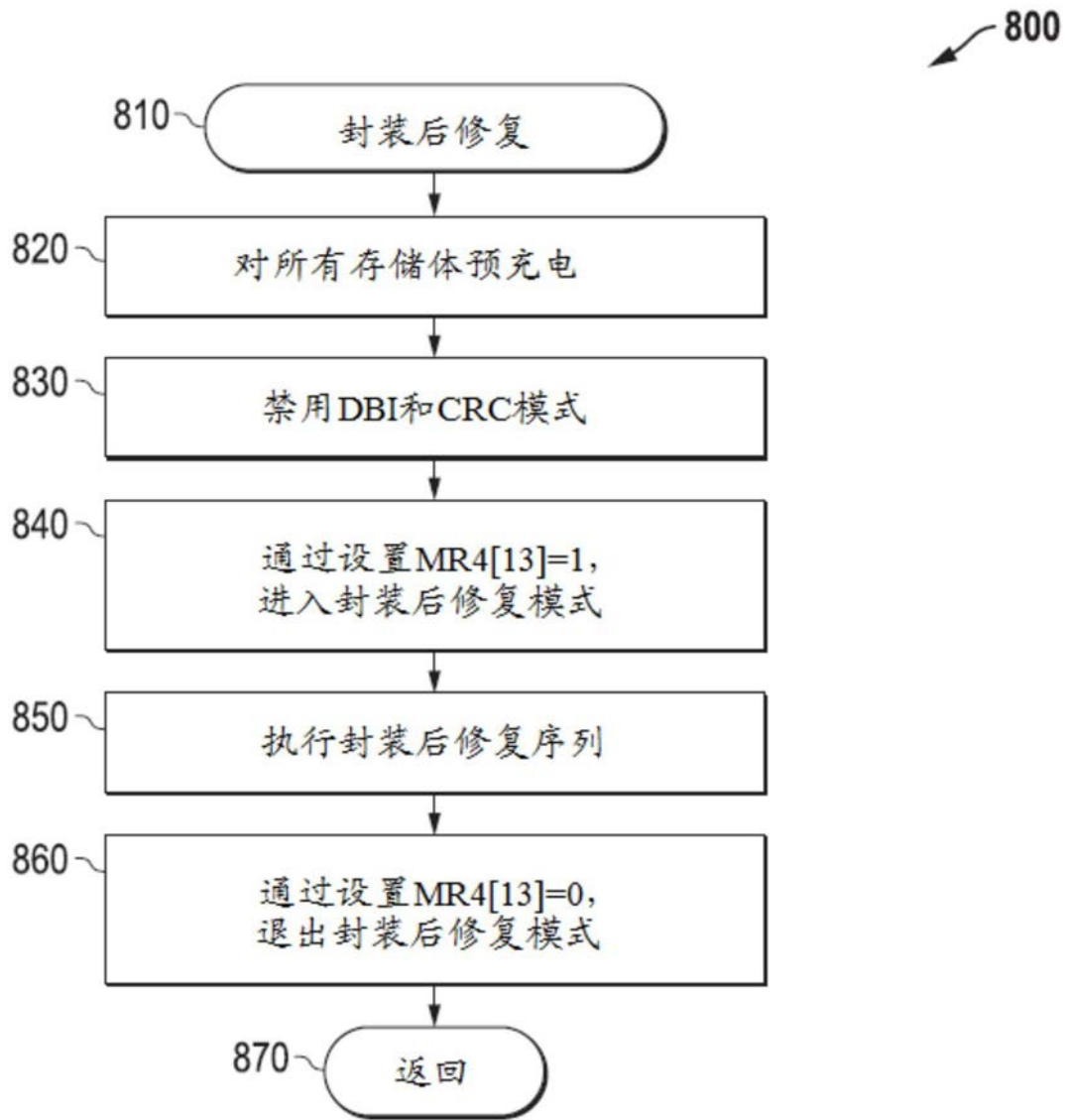


图8