



(12) 发明专利

(10) 授权公告号 CN 108877862 B

(45) 授权公告日 2021. 10. 08

(21) 申请号 201710326110.0

G06F 12/00 (2006.01)

(22) 申请日 2017.05.10

(56) 对比文件

(65) 同一申请的已公布的文献号  
申请公布号 CN 108877862 A

CN 102346694 A, 2012.02.08

CN 108153482 A, 2018.06.12

CN 101101570 A, 2008.01.09

(43) 申请公布日 2018.11.23

CN 102819408 A, 2012.12.12

(73) 专利权人 北京忆芯科技有限公司  
地址 100085 北京市海淀区创业中路32号  
楼32-1-1-167

CN 103034458 A, 2013.04.10

CN 103827806 A, 2014.05.28

CN 104156174 A, 2014.11.19

CN 104991738 A, 2015.10.21

(72) 发明人 王祎磊 任艳伟 沈飞

US 2013227346 A1, 2013.08.29

(74) 专利代理机构 北京卓特专利代理事务所  
(普通合伙) 11572

US 9183085 B1, 2015.11.10

US 2012054413 A1, 2012.03.01

US 2017003880 A1, 2017.01.05

代理人 段宇

审查员 许金明

(51) Int. Cl.

G06F 3/06 (2006.01)

G06F 11/10 (2006.01)

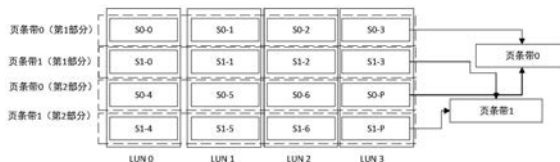
权利要求书2页 说明书17页 附图6页

(54) 发明名称

页条带的数据组织以及向页条带写入数据的方法与装置

(57) 摘要

公开了页条带的数据组织以及向页条带写入数据的方法与装置。所公开的方法包括将第一页条带的第一部分数据写入来自多个逻辑单元的多个第一物理页;将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页;将第一页条带的第二部分数据写入来自所述多个逻辑单元的多个第三物理页;其中第一物理页与第三物理页在非易失存储器的物理空间上不相邻。从而,使用少量的存储空间计算多个页条带的校验数据,并同时向多个页条带写入数据。



1. 一种向非易失存储器写入数据的方法,包括:

将第一页条带的第一部分数据写入来自多个逻辑单元的多个第一物理页;

将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页;

将第一页条带的第二部分数据写入来自所述多个逻辑单元的多个第三物理页;

其中第一物理页与第三物理页在非易失存储器的物理空间上不相邻;

将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页之后,将第一页条带的第三部分用户数据写入来自所述多个逻辑单元的多个第三物理页之前,还将第三页条带的第一部分数据写入来自所述多个逻辑单元的多个第四物理页;

其中第二物理页与第四物理页在非易失存储器的物理空间上相邻。

2. 根据权利要求1所述的方法,其中

第一物理页与第三物理页在非易失存储器的物理空间上不相邻包括,第一物理页与第三物理页不属于相同字线、不属于3D存储器的相同层和/或不属于3D存储器的相邻层。

3. 根据权利要求1或2所述的方法,其中所述多个逻辑单元的每个提供所述多个第一物理页中的一个物理页;所述多个逻辑单元的每个提供所述多个第二物理页中的一个物理页;以及所述多个逻辑单元的每个提供所述多个第三物理页中的一个物理页。

4. 根据权利要求1或2所述的方法,还包括:

备份第一存储器中的根据第一页条带的第一部分数据生成的第一校验数据;以及

在所述第一存储器中缓存根据第二页条带的第一部分数据生成的第二校验数据。

5. 根据权利要求4所述的方法,其中:

在所述第一存储器中根据第一页条带的第一部分数据生成的第一校验数据后,备份第一校验数据;以及

响应于备份了第一校验数据,在所述第一存储器中缓存根据第二页条带的第一部分数据生成的第二校验数据。

6. 根据权利要求4所述的方法,其中:

在将第一页条带的第一部分数据写入来自多个逻辑单元的多个第一物理页后,备份第一校验数据;以及

在将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页过程中,在所述第一存储器中缓存根据第二页条带的第一部分数据生成的第二校验数据。

7. 根据权利要求4所述的方法,还包括:

备份所述第一存储器中的根据第二页条带的第一部分数据生成的第二校验数据;

将所述第一校验数据传送到所述第一存储器;以及

在所述第一存储器中缓存根据所述第一校验数据以及第一页条带的第二部分数据生成的所述第一页条带的校验数据。

8. 根据权利要求7所述的方法,其中:

在所述第一存储器中根据第二页条带的第一部分数据生成的第二校验数据后,备份第二校验数据;以及

响应于备份了第二校验数据,将所述第一校验数据传送到所述第一存储器。

9. 一种介质接口控制器,包括校验数据计算器以及第一存储器,所述校验数据计算器对编程命令对应的数据计算校验数据,校验数据存储在第一存储器;介质接口控制器还耦

合到CPU与第二存储器;校验数据计算器也耦合到CPU与第二存储器;响应于页条带的第一部分数据的校验数据计算完成,介质接口控制器或校验数据计算器将第一存储器中的校验数据存储到第二存储器;

其中响应于CPU提供的编程命令,介质接口控制器将第一页条带的第一部分数据写入来自多个逻辑单元的多个第一物理页;响应于CPU提供的编程命令,介质接口控制器将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页;响应于CPU提供的编程命令,介质接口控制器第一页条带的第二部分数据写入来自所述多个逻辑单元的多个第三物理页;其中第一物理页与第三物理页在非易失存储器的物理空间上不相邻;

其中响应于CPU提供的编程命令,介质接口控制器将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页之后,将第一页条带的第三部分用户数据写入来自所述多个逻辑单元的多个第三物理页之前,还将第三页条带的第一部分数据写入来自所述多个逻辑单元的多个第四物理页;其中第二物理页与第四物理页在非易失存储器的物理空间上相邻。

## 页条带的数据组织以及向页条带写入数据的方法与装置

### 技术领域

[0001] 本申请涉及固态存储设备(Solid Storage Device,SSD),更具体地,本申请涉及在存储介质上组织数据以提升可靠性。

### 背景技术

[0002] 参看图1,展示了存储设备的框图。固态存储设备102同主机相耦合,用于为主机提供存储能力。主机同固态存储设备102之间可通过多种方式相耦合,耦合方式包括但不限于通过例如SATA(Serial Advanced Technology Attachment,串行高级技术附件)、SCSI(Small Computer System Interface,小型计算机系统接口)、SAS(Serial Attached SCSI,串行连接SCSI)、IDE(Integrated Drive Electronics,集成驱动器电子)、USB(Universal Serial Bus,通用串行总线)、PCIe(Peripheral Component Interconnect Express,PCIe,高速外围组件互联)、NVMe(NVM Express,高速非易失存储)、以太网、光纤通道、无线通信网络等连接主机与固态存储设备102。主机可以是能够通过上述方式同存储设备相通信的信息处理设备,例如,个人计算机、平板电脑、服务器、便携式计算机、网络交换机、路由器、蜂窝电话、个人数字助理等。存储设备102包括接口103、控制部件104、一个或多个NVM芯片105以及DRAM(Dynamic Random Access Memory,动态随机访问存储器)110。

[0003] NAND闪存、相变存储器、FeRAM(Ferroelectric RAM,铁电存储器)、MRAM(Magnetic Random Access Memory,磁阻存储器)、RRAM(Resistive Random Access Memory,阻变存储器)等是常见的NVM。

[0004] 接口103可适配于通过例如SATA、IDE、USB、PCIe、NVMe、SAS、以太网、光纤通道等方式与主机交换数据。

[0005] 控制部件104用于控制在接口103、NVM芯片105以及DRAM 110之间的数据传输,还用于存储管理、主机逻辑地址到闪存物理地址映射、擦除均衡、坏块管理等。控制部件104可通过软件、硬件、固件或其组合的多种方式实现,例如,控制部件104可以是FPGA(Field-programmable gate array,现场可编程门阵列)、ASIC(Application Specific Integrated Circuit,应用专用集成电路)或者其组合的形式。控制部件104也可以包括处理器或者控制器,在处理器或控制器中执行软件来操纵控制部件104的硬件来处理IO(Input/Output)命令。控制部件104还可以耦合到DRAM 110,并可访问DRAM 110的数据。在DRAM可存储FTL表和/或缓存的IO命令的数据。

[0006] 控制部件104包括闪存接口控制器(或称为介质接口控制器、闪存通道控制器),闪存接口控制器耦合到NVM芯片105,并以遵循NVM芯片105的接口协议的方式向NVM芯片105发出命令,以操作NVM芯片105,并接收从NVM芯片105输出的命令执行结果。已知的NVM芯片接口协议包括“Toggle”、“ONFI”等。

[0007] 存储器目标(Target)是闪存颗粒105封装内的共享芯片使能(CE,Chip Enable)信号的一个或多个逻辑单元(Logic Unit),每个逻辑单元具有逻辑单元号(LUN,Logic Unit Number)。NAND闪存封装内可包括一个或多个管芯(Die)。典型地,逻辑单元对应于单一的管

芯。逻辑单元可包括多个平面(Plane)。逻辑单元内的多个平面可以并行存取,而NAND闪存芯片内的多个逻辑单元可以彼此独立地执行命令和报告状态。在[http://www.onfi.org/~media/ONFI/specs/ONFI\\_3\\_2%20Gold.pdf](http://www.onfi.org/~media/ONFI/specs/ONFI_3_2%20Gold.pdf)获得的“Open NAND Flash Interface Specification (Revision 3.2)”中,提供了关于目标(target)、逻辑单元、LUN、平面(Plane)的含义,以及也提供了操作NVM芯片的命令。

[0008] 存储介质上通常按页来存储和读取数据。而按块来擦除数据。块包含多个页。存储介质上的页(称为物理页)具有固定的尺寸,例如17664字节。物理页也可以具有其他的尺寸。物理页中可以包括多个数据帧(data frame),数据帧具有指定的尺寸,例如4096或4416字节。

[0009] 在固态存储设备中,利用FTL(Flash Translation Layer,闪存转换层)来维护从逻辑地址到物理地址的映射信息。逻辑地址构成了操作系统等上层软件所感知到的固态存储设备的存储空间。物理地址是用于访问固态存储设备的物理存储单元的地址。在现有技术中还可利用中间地址形态实施地址映射。例如将逻辑地址映射为中间地址,进而将中间地址进一步映射为物理地址。

[0010] 其中,存储了从逻辑地址到物理地址的映射信息的表结构被称为FTL表。FTL表是固态存储设备中的重要元数据。通常FTL表的数据项记录了固态存储设备中以数据页为单位的地址映射关系。

[0011] FTL表包括多个FTL表条目(或称表项)。在一个例子中,每个FTL表条目中记录了一个逻辑页地址与一个物理页的对应关系。在另一个例子中,每个FTL表条目中记录了连续的多个逻辑页地址与连续的多个物理页的对应关系。在又一个例子中,每个FTL表条目中记录了逻辑块地址与物理块地址的对应关系。在依然又一个实施例,FTL表中记录逻辑块地址与物理块地址的映射关系,和/或逻辑页地址与物理页地址的映射关系。

[0012] 在处理来自主机的读命令时,固态存储设备利用读命令中携带的逻辑地址从FTL表中获得对应的物理地址,并依据物理地址向NVM芯片发出读请求,并接收NVM芯片响应于读请求输出的数据。在处理来自主机的写命令时,固态存储设备为写命令分配物理地址,在FTL表中记录写命令的逻辑地址与分配的物理地址的对应关系,并依据分配的物理地址向NVM芯片发出写请求。

[0013] 固态存储设备中包括多个NVM芯片。每个NVM芯片包括一个或多个管芯(DIE)或逻辑单元(LUN, Logic UNit)。管芯或逻辑单元之间可以并行响应读写操作。在同一管芯或逻辑单元上的多个读、写或擦除操作顺序执行。

[0014] 图2示出了大块的示意图。大块包括来自多个逻辑单元的每个的物理块。优选的,每个逻辑单元为大块提供一个物理块。作为举例,在每16个逻辑单元(LUN)上构造大块。每个大块包括16个物理块,来自16个逻辑单元(LUN)的每一个。在图2的例子中,大块0包括来自16个逻辑单元(LUN)的每个的物理块0,而大块1包括来自每个逻辑单元(LUN)的物理块1。也可以有多种其他方式来构造大块。

[0015] 作为一种可选的方式,在大块中构造页条带,每个逻辑单元(LUN)内相同物理地址的物理页构成了“页条带”。图2中,物理页0-0、物理页0-1……与物理页0-x构成了页条带0,其中物理页0-0、物理页0-1……物理页0-14用于存储用户数据,而物理页0-15用于存储根据条带内的所有用户数据计算得到的校验数据。类似地,图2中,物理页2-0、物理页2-1……

与物理页2-x构成了页条带2。可选地,用于存储校验数据的物理页可以位于页条带中的任意位置。

[0016] 大块中的一个或多个物理块可能是坏块。不应向坏块写入数据。因而页条带中可容纳的用户数据量,依赖于页条带所在的大块的坏块数量。物理页包括一个或多个数据单元。页条带的有效数据单元数量,由指页条带中可容纳用户数据的数据单元的数量。作为举例,将页条带中由坏块提供的物理页去除,以及去除用于存储校验数据的物理页,剩下的物理页的数据单元数量,为页条带的有效数据单元数量。

[0017] 为向页条带写入数据,固态存储设备的控制部件(104)(参看图1)提供校验数据计算器。以利用异或操作计算校验数据为例,对于包括N+1个物理页的页条带,对N个物理页的用户数据计算异或(例如, $(P0-1) \text{ XOR } (P0-1) \text{ XOR } (P0-2) \text{ XOR } \dots \text{ XOR } (P0-15)$ ),并将计算结果写入为页条带存储校验数据的物理页(例如P0-X)。可选地,在控制部件(104)中提供多个校验数据计算器(例如,M个),以同时向M个页条带写入数据。

## 发明内容

[0018] 校验数据计算单元需要存储大量的校验数据计算结果,并且为页条带计算校验数据需要页条带的所有用户数据参与计算。因而为同时向多个页条带写入数据,需要使用更多的高速存储资源,这将导致集成电路芯片的高成本与高静态功耗。而若使用较少量的存储资源,又限制了向页条带写入数据的并发性。需要有效的技术手段来利用较少量的存储资源支持向多个页条带并发写入数据时的校验数据计算。

[0019] 根据本申请的第一方面,提供了根据本申请第一方面的第一向非易失存储器写入数据的方法,包括:将第一页条带的第一部分数据写入来自多个逻辑单元的多个第一物理页;将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页;将第一页条带的第二部分数据写入来自所述多个逻辑单元的多个第三物理页;其中第一物理页与第三物理页在非易失存储器的物理空间上不相邻。

[0020] 根据本申请第一方面的第一向非易失存储器写入数据的方法,提供了根据本申请第一方面的第二向非易失存储器写入数据的方法,其中第一物理页与第三物理页在非易失存储器的物理空间上不相邻包括,第一物理页与第三物理页不属于相同字线、不属于3D存储器的相同层和/或不属于3D存储器的相邻层。

[0021] 根据本申请第一方面的第一或第二向非易失存储器写入数据的方法,提供了根据本申请第一方面的第三向非易失存储器写入数据的方法,其中第一物理页与第二物理页在非易失存储器的物理空间上相邻。

[0022] 根据本申请第一方面的第一至第三向非易失存储器写入数据的方法之一,提供了根据本申请第一方面的第四向非易失存储器写入数据的方法,其中第二物理页与第三物理页在非易失存储器的物理空间上相邻。

[0023] 根据本申请第一方面的第一至第三向非易失存储器写入数据的方法之一,提供了根据本申请第一方面的第五向非易失存储器写入数据的方法,还包括:将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页之后,将第一页条带的第三部分用户数据写入来自所述多个逻辑单元的多个第三物理页之前,还将第三页条带的第一部分数据写入来自所述多个逻辑单元的多个第四物理页;其中第二物理页与第四物理页在非易

失存储器的物理空间上相邻。

[0024] 根据本申请第一方面的第五向非易失存储器写入数据的方法,提供了根据本申请第一方面的第六向非易失存储器写入数据的方法,其中第三物理页与第四物理页在非易失存储器的物理空间上相邻。

[0025] 根据本申请第一方面的第一至第六向非易失存储器写入数据的方法之一,提供了根据本申请第一方面的第七向非易失存储器写入数据的方法,其中所述多个逻辑单元的每个为相同页条带提供的两个或更多物理页彼此在非易失存储器的物理空间上不相邻。

[0026] 根据本申请第一方面的第一至第七向非易失存储器写入数据的方法之一,提供了根据本申请第一方面的第八向非易失存储器写入数据的方法,其中所述多个逻辑单元的每个提供所述多个第一物理页中的一个物理页;所述多个逻辑单元的每个提供所述多个第二物理页中的一个物理页;以及所述多个逻辑单元的每个提供所述多个第三物理页中的一个物理页。

[0027] 根据本申请第一方面的第八向非易失存储器写入数据的方法,提供了根据本申请第一方面的第九向非易失存储器写入数据的方法,其中所述多个逻辑单元的每个为页条带提供两个或更多物理页。

[0028] 根据本申请第一方面的第一至第九向非易失存储器写入数据的方法之一,提供了根据本申请第一方面的第十向非易失存储器写入数据的方法,还包括:备份第一存储器中的根据第一页条带的第一部分数据生成的第一校验数据;以及在所述第一存储器中缓存根据第二页条带的第一部分数据生成的第二校验数据。

[0029] 根据本申请第一方面的第十向非易失存储器写入数据的方法,提供了根据本申请第一方面的第十一向非易失存储器写入数据的方法,还包括:备份所述第一存储器中的根据第二页条带的第一部分数据生成的第二校验数据;以及在所述第一存储器中缓存根据第一页条带的第二部分数据生成的第三校验数据。

[0030] 根据本申请第一方面的第十一向非易失存储器写入数据的方法,提供了根据本申请第一方面的第十二向非易失存储器写入数据的方法,还包括:根据所述第三校验数据得到所述第一页条带的校验数据,并写入来自所述多个逻辑单元的物理页。

[0031] 根据本申请第一方面的第十至第十二向非易失存储器写入数据的方法之一,提供了根据本申请第一方面的第十三向非易失存储器写入数据的方法,其中:在第一存储器中根据第一页条带的第一部分数据生成的第一校验数据后,备份第一校验数据;以及响应于备份了第一校验数据,在所述第一存储器中缓存根据第二页条带的第一部分数据生成的第二校验数据。

[0032] 根据本申请第一方面的第十至第十三向非易失存储器写入数据的方法之一,提供了根据本申请第一方面的第十四向非易失存储器写入数据的方法,其中:在将第一页条带的第一部分数据写入来自多个逻辑单元的多个第一物理页后,备份第一校验数据;以及在将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页过程中,在所述第一存储器中缓存根据第二页条带的第一部分数据生成的第二校验数据。

[0033] 根据本申请第一方面的第十向非易失存储器写入数据的方法,提供了根据本申请第一方面的第十一向非易失存储器写入数据的方法,还包括:备份所述第一存储器中的根据第二页条带的第一部分数据生成的第二校验数据;将所述第一校验数据传送到所述第一

存储器;以及在所述第一存储器中缓存根据所述第一校验数据以及第一页条带的第二部分数据生成的所述第一页条带的校验数据。

[0034] 根据本申请第一方面的第十五向非易失存储器写入数据的方法,提供了根据本申请第一方面的第十六向非易失存储器写入数据的方法,还包括:将所述第一页条带的校验数据,写入来自所述多个逻辑单元的物理页。

[0035] 根据本申请第一方面的第十五至第十六向非易失存储器写入数据的方法之一,提供了根据本申请第一方面的第十七向非易失存储器写入数据的方法,其中:在第一存储器中根据第二页条带的第一部分数据生成的第二校验数据后,备份第二校验数据;以及响应于备份了第二校验数据,将所述第一校验数据传送到所述第一存储器。

[0036] 根据本申请第一方面的第十五至第十七向非易失存储器写入数据的方法之一,提供了根据本申请第一方面的第十八向非易失存储器写入数据的方法,其中:在将第二页条带的第一部分数据写入来自多个逻辑单元的多个第二物理页后,备份第二校验数据;以及在将第一页条带的第二部分数据写入来自所述多个逻辑单元的多个第三物理页过程中,在所述第一存储器中缓存根据第一页条带的第二部分数据生成的第一页条带的校验数据。

[0037] 根据本申请第一方面的第一至第九向非易失存储器写入数据的方法之一,提供了根据本申请第一方面的第十九向非易失存储器写入数据的方法,还包括:将第三页条带的第一部分数据写入来自第二多个逻辑单元的多个第一物理页;将第四页条带的第一部分数据写入来自所述第二多个逻辑单元的多个第二物理页;将第三页条带的第二部分数据写入来自所述第二多个逻辑单元的多个第三物理页;其中所述第二多个逻辑单元的第一物理页与所述第二多个逻辑单元的第三物理页在非易失存储器的物理空间上不相邻。

[0038] 根据本申请第一方面的第一至第九或第十九向非易失存储器写入数据的方法之一,提供了根据本申请第一方面的第二十向非易失存储器写入数据的方法,还包括:在将第一页条带的第一部分数据写入来自多个逻辑单元的多个第一物理页期间,在第一存储器中缓存根据第一页条带的第一部分数据生成的第一校验数据,将第三页条带的第一部分数据写入来自第二多个逻辑单元的多个第一物理页,以及在第二存储器中缓存根据第三页条带的第一部分数据生成的第四校验数据。

[0039] 根据本申请第一方面的第十九或第二十向非易失存储器写入数据的方法,提供了根据本申请第一方面的第二十一向非易失存储器写入数据的方法,还包括:备份第一存储器中的根据第一页条带的第一部分数据生成的第一校验数据;在所述第一存储器中缓存根据第二页条带的第一部分数据生成的第二校验数据;备份第二存储器中的根据第三页条带的第一部分数据生成的第四校验数据;在所述第二存储器中缓存根据第四页条带的第一部分数据生成的第五校验数据。

[0040] 根据本申请第一方面的第十九或第二十向非易失存储器写入数据的方法,提供了根据本申请第一方面的第二十二向非易失存储器写入数据的方法,还包括:备份第一存储器中的根据第一页条带的第一部分数据生成的第一校验数据;将所述第一校验数据传送到所述第一存储器;在所述第一存储器中缓存根据所述第一校验数据以及第一页条带的第二部分数据生成的所述第一页条带的校验数据;备份第二存储器中的根据第三页条带的第一部分数据生成的第四校验数据;将所述第四校验数据传送到所述第二存储器;以及在所述第二存储器中缓存根据所述第四校验数据以及第三页条带的第二部分数据生成的所述第



三页条带的校验数据。

[0041] 根据本申请的第二方面,提供了根据本申请第二方面的第一介质接口控制器,包括校验数据计算器以及第一存储器,所述校验数据计算器对编程命令对应的数据计算校验数据,校验数据存储在第一存储器;介质接口控制器还耦合到CPU与第二存储器;校验数据计算器也耦合到CPU与第二存储器;响应于页条带的第一部分数据的校验数据计算完成,介质接口控制器或校验数据计算器将第一存储器中的校验数据存储到第二存储器。

[0042] 根据本申请第二方面的第一介质接口控制器,提供了根据本申请第二方面的第二介质接口控制器,其中介质接口控制器或校验数据计算器向CPU发送第一中断信号,并向CPU提供校验数据的在第二存储器中的存储地址。

[0043] 根据本申请第二方面的第一或第二介质接口控制器,提供了根据本申请第二方面的第三介质接口控制器,其中响应于向页条带写入校验数据的写入操作完成,介质接口控制器或校验数据计算器向CPU发送第二中断信号,以指示可释放第二存储器中的校验数据。

[0044] 根据本申请第二方面的第一至第三介质接口控制器之一,提供了根据本申请第二方面的第四介质接口控制器,其特征在于,响应于第一中断信号,CPU生成编程命令,将第二存储器中的校验数据写入页条带,并且指示无需为所生成的编程命令计算校验数据。

[0045] 根据本申请第二方面的第一至第三介质接口控制器之一,提供了根据本申请第二方面的第五介质接口控制器,其中CPU、介质接口控制器或校验数据计算器将第二存储器中的校验数据存储到第一存储器。

[0046] 根据本申请第二方面的第一至第五介质接口控制器之一,提供了根据本申请第二方面的第六介质接口控制器,其中所述校验数据计算器对第一存储器中的数据与编程命令对应的数据计算校验数据。

[0047] 根据本申请第二方面的第一至第六介质接口控制器之一,提供了根据本申请第二方面的第七介质接口控制器,其中所述介质接口控制器将编程命令对应的数据发送给非易失去存储器。

[0048] 根据本申请第二方面的第一至第七介质接口控制器之一,提供了根据本申请第二方面的第八介质接口控制器,还包括第三存储器,用于存储校验数据。

[0049] 根据本申请第二方面的第一至第八介质接口控制器之一,提供了根据本申请第二方面的第九介质接口控制器,其中响应于CPU提供的编程命令,介质接口控制器将第一页条带的第一部分数据写入来自多个逻辑单元的多个第一物理页;响应于CPU提供的编程命令,介质接口控制器将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页;响应于CPU提供的编程命令,介质接口控制器将第一页条带的第二部分数据写入来自所述多个逻辑单元的多个第三物理页;其中第一物理页与第三物理页在非易失存储器的物理空间上不相邻。

[0050] 根据本申请第二方面的第九介质接口控制器,提供了根据本申请第二方面的第十介质接口控制器,其中第一物理页与第三物理页在非易失存储器的物理空间上不相邻包括,第一物理页与第三物理页不属于相同字线、不属于3D存储器的相同层和/或不属于3D存储器的相邻层。

[0051] 根据本申请第二方面的第九或第十介质接口控制器,提供了根据本申请第二方面的第十一介质接口控制器,其中第一物理页与第二物理页在非易失存储器的物理空间上相

邻。

[0052] 根据本申请第二方面的第九至第十一介质接口控制器之一,提供了根据本申请第二方面的第十二介质接口控制器,其中第二物理页与第三物理页在非易失存储器的物理空间上相邻。

[0053] 根据本申请第二方面的第九至第十一介质接口控制器之一,提供了根据本申请第二方面的第十三介质接口控制器,其中响应于CPU提供的编程命令,介质接口控制器将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页之后,将第一页条带的第三部分用户数据写入来自所述多个逻辑单元的多个第三物理页之前,还将第三页条带的第一部分数据写入来自所述多个逻辑单元的多个第四物理页;其中第二物理页与第四物理页在非易失存储器的物理空间上相邻。

[0054] 根据本申请第二方面的第九至第十三介质接口控制器之一,提供了根据本申请第二方面的第十四介质接口控制器,其中所述多个逻辑单元的每个为相同页条带提供的两个或更多物理页彼此在非易失存储器的物理空间上不相邻。

[0055] 根据本申请第二方面的第九至第十四介质接口控制器之一,提供了根据本申请第二方面的第十五介质接口控制器,其中所述多个逻辑单元的每个提供所述多个第一物理页中的一个物理页;所述多个逻辑单元的每个提供所述多个第二物理页中的一个物理页;以及所述多个逻辑单元的每个提供所述多个第三物理页中的一个物理页。

[0056] 根据本申请第二方面的第十五介质接口控制器,提供了根据本申请第二方面的第十六介质接口控制器,其中所述多个逻辑单元的每个为页条带提供两个或更多物理页。

[0057] 根据本申请第二方面的第九至第十六介质接口控制器之一,提供了根据本申请第二方面的第十七介质接口控制器,其中CPU、介质接口控制器或校验数据计算器备份第一存储器中的根据第一页条带的第一部分数据生成的第一校验数据;以及将所述第一存储器用于缓存根据第二页条带的第一部分数据生成的第二校验数据。

[0058] 根据本申请第二方面的第十七介质接口控制器,提供了根据本申请第二方面的第十八介质接口控制器,其中CPU、介质接口控制器或校验数据计算器备份所述第一存储器中的根据第二页条带的第一部分数据生成的第二校验数据;以及将所述第一存储器用于缓存根据第一页条带的第二部分数据生成的第三校验数据。

[0059] 根据本申请第二方面的第十七介质接口控制器,提供了根据本申请第二方面的第十九介质接口控制器,其中介质接口控制器或校验数据计算器备份所述第一存储器中的根据第二页条带的第一部分数据生成的第二校验数据;CPU将第二存储器中的第一校验数据传送到所述第一存储器;以及校验数据计算器在所述第一存储器中缓存根据所述第一校验数据以及第一页条带的第二部分数据生成的所述第一页条带的校验数据。

[0060] 根据本申请第二方面的第一至第十九介质接口控制器之一,提供了根据本申请第二方面的第二十介质接口控制器,还包括第三存储器,用于存储校验数据计算器生成的校验数据;其中在介质接口控制器将第一页条带的第一部分数据写入来自多个逻辑单元的多个第一物理页期间,校验数据计算器在第一存储器中缓存根据第一页条带的第一部分数据生成的第一校验数据,介质接口控制器将第三页条带的第一部分数据写入来自第二多个逻辑单元的多个第一物理页,以及校验数据计算器还在第三存储器中缓存根据第三页条带的第一部分数据生成的第四校验数据。

[0061] 根据本申请的第三方面,提供了根据本申请第三方面的第一向非易失存储器写入数据的装置,包括:第一写入模块,用于将第一页条带的第一部分数据写入来自多个逻辑单元的多个第一物理页;第二写入模块,用于将第二页条带的第一部分数据写入来自所述多个逻辑单元的多个第二物理页;第三写入模块,用于将第一页条带的第二部分数据写入来自所述多个逻辑单元的多个第三物理页;其中第一物理页与第三物理页在非易失存储器的物理空间上不相邻。

### 附图说明

[0062] 当连同附图阅读时,通过参考后面对示出性的实施例的详细描述,将最佳地理解本申请以及优选的使用模式和其进一步的目的和优点,其中附图包括:

[0063] 图1为现有技术中的固态存储设备的示意图;

[0064] 图2为页条带的构造示意图;

[0065] 图3为本申请实施例的固态存储设备的控制部件的示意图;

[0066] 图4为本申请又一实施例的介质接口控制器的示意图;

[0067] 图5为本申请另一实施例的介质接口控制器的示意图;

### 具体实施方式

[0068] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0069] 图3是根据本申请实施例的固态存储设备的控制部件的框图。固态存储设备的控制部件104中包括主机接口310、前端处理模块320、闪存管理模块330以及一个或多个介质接口控制器340。主机接口310用于同主机交换命令与数据。闪存管理模块330提供逻辑地址到物理地址映射、磨损均衡、垃圾回收等功能,并生成IO命令发送给介质接口控制器340。介质接口控制器耦合到NVM芯片105。介质接口控制器有一个或多个,每个接口控制器耦合到各自的NVM芯片105。介质接口控制器接收IO命令,并根据IO命令向NVM芯片发出IO命令(读、编程、擦除、暂停、读取特征(feature)和/或设置特征等命令)。

[0070] 在一个例子中,介质接口控制器提供具有不同优先级的多个队列(例如队列342与队列344,其中队列342是高优先级队列而队列344是低优先级队列)来接收IO命令。闪存管理模块330将读命令填入接口控制器的高优先级队列,使介质接口控制器优先处理读命令。而将编程命令、擦除命令和/或读命令(例如不要求处理延迟的读命令)填入介质接口控制器的低优先级队列。介质接口控制器会优先处理高优先级队列342中的命令,而以低优先级处理队列344中的命令。可以理解地,闪存管理模块也可将其他类型的IO命令填入高优先级队列。

[0071] 根据本申请的实施例,一般情况下,介质接口控制器优先处理高优先级队列中的IO命令,而以低优先级处理低优先级队列中的IO命令。

[0072] 根据本申请的一个实施例,为了降低固态存储设备的IO命令的处理延迟,若在逻辑单元(L1)上有正在执行的编程命令(P1)或擦除命令(E1),而在高优先级队列中有待处理

的读命令 (R1), 其中, 读命令 R1 与编程命令 P1 访问相同的逻辑单元 (L1)。则介质接口控制器向逻辑单元 (L1) 发出暂停命令, 暂停编程命令 (P1) 或擦除命令 (E1) 的处理, 以及处理读命令 (R1)。以及在读命令 (R1) 处理完成后, 向逻辑单元 (L1) 发出恢复命令, 恢复对编程命令 (P1) 或擦除命令 (E1) 的处理。

[0073] 可以理解地, 高优先级队列/低优先级队列中的 I/O 命令与发送给逻辑单元的 I/O 命令可具有不同的形态, 但以相同的标记 (R1、P1、E1 等) 指示处于不同阶段、具有不同形态但含义相同的同一命令。

[0074] 根据本申请的又一个实施例, 介质接口控制器发现在高优先级队列中有待处理的读命令 (R2), 而在低优先级队列中有待处理的编程命令 (P2)。介质接口控制器不是先处理读命令 (R2), 而是先处理低优先级队列中的编程命令 (P2)。在向逻辑单元 (L2) 发出编程命令 (P2) 后, 若读命令 (R2) 与编程命令 (P2) 访问相同的逻辑单元 (L2), 则紧接着向逻辑单元 (L2) 发出暂停命令, 以暂停对编程命令 (P2) 的处理。以及介质接口控制器处理来自高优先级队列的读命令 (R2), 向逻辑单元 (L2) 发出读命令 (R2), 并在读命令 (R2) 处理完成后, 向逻辑单元 (L2) 发出恢复命令, 以恢复对编程命令 (P2) 的处理。若读命令 (R2) 与编程命令 (P2) 访问不同的逻辑单元, 则直接处理读命令 (R2), 而无需暂停编程命令 (P2)。

[0075] 根据本申请的另一实施例, 介质接口控制器发现高优先级队列中有待处理的读命令 (R3), 而在低优先级队列中有待处理的擦除命令 (E3)。介质接口控制器不是先处理读命令 (R3), 而是先处理低优先级队列中的擦除命令 (E3)。在向逻辑单元 (L3) 发出擦除命令 (E3) 后, 若读命令 (R3) 与擦除命令 (E3) 访问相同的逻辑单元 (L3), 则紧接着向逻辑单元 (L3) 发出暂停命令, 以暂停对擦除命令 (E3) 的处理。以及介质接口控制器处理来自高优先级队列的读命令 (R3), 向逻辑单元 (L3) 发出读命令 (R3), 并在读命令 (R3) 处理完成后, 向逻辑单元 (L3) 发出恢复命令, 以恢复对擦除命令 (E3) 的处理。若读命令 (R3) 与擦除命令 (E3) 访问不同的逻辑单元, 则直接处理读命令 (R3), 而无需暂停擦除命令 (E3)。进一步地, 若接下来低优先级队列中又出现了访问其他逻辑单元的编程 (P4)/擦除 (E4) 命令 (不同于逻辑单元 L3 的其他逻辑单元), 高优先级队列中出现了读命令 (R4), 则执行编程 (P4)/擦除 (E4) 命令, 以及暂停编程 (P4)/擦除 (E4) 命令, 接下来执行读命令 (R4) 以及恢复编程 (P4)/擦除 (E4) 命令的处理。

[0076] 可选地, 每个逻辑单元有对应的介质接口控制器或介质接口控制器的执行上下文, 从而由同一介质接口控制器正在处理的 I/O 命令访问相同的逻辑单元。在此情况下, 介质接口控制器无需判断读命令与编程/擦除命令是否访问相同的逻辑单元, 而在执行了编程/擦除后又从高优先级队列收到读命令时, 都向逻辑单元发出暂停命令, 以暂停对编程/擦除命令的处理。接着发出读命令, 并在读命令执行完成后, 恢复编程/擦除命令的执行。

[0077] 图4是根据本申请又一实施例的介质接口控制器440的框图。介质接口控制器包括多个队列 (例如, 队列342与队列344)。各个队列具有不同的优先级, 例如队列342是高优先级队列, 而队列344是低优先级队列。介质接口控制器440包括命令调度模块410、校验数据计算器420。作为举例, 校验数据计算器420对输入数据做异或以得到校验数据, 在校验数据计算器420中还包括XOR缓存 (422/424/426), 计算得到的校验数据被存储在XOR缓存 (422/424/426) 中。

[0078] 可选地, 介质接口控制器440还耦合到DRAM (参看图1, DRAM 110)。

[0079] 命令调度模块410从队列342/344中取出命令,并执行命令以访问NVM芯片。对于编程命令,编程命令对应的数据被发送给NVM芯片的过程中,指定XOR缓存(例如,XOR缓存422),校验数据计算器420对编程命令对应的数据同XOR缓存(例如,XOR缓存422)所缓存的数据执行异或操作,异或操作的结果存储在XOR缓存(例如,XOR缓存422)中。

[0080] 作为举例,在NVM芯片405/415/425的每个上分配一个物理页来构造页条带(S1)。页条带(S1)中的2个物理页用于存储用户数据,而1个物理页用于存储校验数据。

[0081] 命令调度模块410从命令队列中获取编程命令(P5),分配XOR缓存(422),将编程命令(P5)的数据同XOR缓存(422)做异或,计算结果存储在XOR缓存(422)中,以及将编程命令(P5)发送给NVM芯片405。

[0082] 接下来,命令调度模块410从命令队列中获取编程命令(P7),分配XOR缓存(426),将编程命令(P7)的数据同XOR缓存(426)做异或,计算结果存储在XOR缓存(426)中,以及将编程命令(P7)发送给NVM芯片435。

[0083] 其中,编程命令(P5)向页条带(S1)写入数据,而在页条带(S1)被完整写入前,XOR缓存(422)被占用来生成用于页条带(S1)的校验数据。类似地,编程命令(P7)向页条带(S2)写入数据,XOR缓存(426)被占用来生成用于页条带(S2)的校验数据。

[0084] 接下来,在高优先级队列中出现读命令(R5),而低优先级队列中出现编程命令(P6)。虽然读命令(R5)的优先级高于编程命令(P6),为了减少对XOR缓存(422)的占用时间,命令调度模块410优先处理编程命令(P6),编程命令(P6)向页条带(S1)写入数据。将编程命令(P6)对应的数据同XOR缓存(422)做异或,异或结果被存储在XOR缓存(422)中。以及将编程命令(P6)发送给NVM芯片415。

[0085] 此时,由于已经收到了要写入页条带(S1)的全部2页用户数据,将XOR缓存(422)中存储的数据作为页条带(S1)的校验数据并写入NVM芯片425(通过生成编程命令PX5),并释放XOR缓存(422)。

[0086] 在根据本申请的实施例中,将编程命令(P6)对应的数据同XOR缓存(422)做异或,异或结果被存储在XOR缓存(422)中之后,命令调度模块410处理来自高优先级队列342的读命令(R5)。若读命令(R5)所访问的逻辑单元(L5)上有正在处理的编程命令(例如,编程命令P6/P7/PX5)或擦除命令,向逻辑单元(L5)发出暂停命令,暂停逻辑单元(L5)上正在执行的编程命令或擦除命令,向逻辑单元(L5)发出读命令(R5),在读命令(R5)执行完成后,向逻辑单元(L5)发出恢复命令,以恢复被暂停的编程命令或擦除命令。

[0087] 可选地,将编程命令对应的数据同XOR缓存(例如,XOR缓存422)做异或,异或结果被存储在XOR缓存(例如,XOR缓存422),并将编程命令发送给逻辑单元后,将XOR缓存(422)存储的内容(例如,用于页条带(S1)的用户数据或校验数据)写入DRAM(参看图1,DRAM 110),从而可释放XOR缓存(422),并将XOR缓存(422)分配来计算另一页条带(例如,页条带(S3))的校验数据。以及响应于收到向页条带(S1)写入数据的编程命令,从DRAM中获取所存储的XOR缓存(422)的内容,并存储到XOR缓存(例如,422/426),并继续处理向页条带(S1)写入数据的编程命令。

[0088] 图5展示了本申请又一实施例的介质接口控制器。图5的实施例中,控制部件104(参看图1)包括多个介质接口控制器(540/550)。介质接口控制器包括多个队列(例如,队列542/544,552/554)。各个队列具有不同的优先级,例如队列542/552是高优先级队列,而队

列544/554是低优先级队列。介质接口控制器540/550耦合到校验数据计算器520。作为举例,校验数据计算器520对输入数据做异或以得到校验数据,在校验数据计算器520中还还包括XOR缓存(522/524/526)。

[0089] 可选地,介质接口控制器540/550还耦合到DRAM(参看图1,DRAM 110)。

[0090] 其中,介质接口控制器540专用于访问逻辑单元(LUN 505),介质接口控制器550专用于访问逻辑单元(LUN 515)。因而,为访问LUN 505,将相应的命令添加队列542/544,以及为访问LUN 515,将相应的命令添加到队列552/554。

[0091] 可以理解地,介质接口控制器中可存储多组执行上下文,每组执行上下文专用于访问逻辑单元之一。通过对介质接口控制器的执行上下文进行切换,使得在每一时刻,介质接口控制器专用于访问同当前执行上下文对应的(一个)逻辑单元。

[0092] 在图5的实施例中,多个介质接口控制器(540/550)共享校验数据计算器520,也共享DRAM。

[0093] 作为举例,响应于处理编程命令(P10),编程命令(P10)向页条带(S10)写入数据。介质接口控制器540为编程命令(P10)分配XOR缓存(522),将编程命令(P10)对应的数据同XOR缓存(522)做异或,异或结果被存储在XOR缓存(522)中,以及将编程命令(P10)发送给LUN 505。

[0094] 接下来,高优先级队列542上出现待处理的读命令(R10)。由于介质接口控制器540专用于访问逻辑单元(LUN 505),因而暗示了读命令(R10)与编程命令(P10)都访问逻辑单元(LUN 505)。介质接口控制器540发现编程命令(P10)正在LUN 505上执行,为降低读命令(R10)的处理延迟,向LUN 505发出暂停命令,暂停编程命令(P10)的执行,并将读命令(R10)发送给LUN 505。以及响应于读命令(R10)执行完成,向LUN 505发出恢复命令,以恢复编程命令(P10)的执行。

[0095] 接下来,高优先级队列542上出现待处理的读命令(R11),低优先级队列544上出现待处理的编程命令(P11)。以及编程命令(P10)已执行完成。由于介质接口控制器540专用于访问逻辑单元(LUN 505),因而暗示了读命令(R11)与编程命令(P11)都访问逻辑单元(LUN 505),介质访问控制器540无需再检查收到的命令是否访问相同的逻辑单元。虽然读命令(R11)的优先级高于编程命令(P11),但介质访问控制器540优先处理编程命令(P11)。编程命令(P11)用于向页条带(S11)写入数据。介质接口控制器540为编程命令(P11)分配XOR缓存(524),将编程命令(P11)对应的数据同XOR缓存(524)做异或,异或结果被存储在XOR缓存(524)中,以及将编程命令(P11)发送给LUN 505。接下来,介质访问控制器540向LUN 505发出暂停命令,暂停编程命令(P11)的执行,并将读命令(R11)发送给LUN 505。以及响应于读命令(R11)执行完成,向LUN 505发出恢复命令,以恢复编程命令(P11)的执行。

[0096] 接下来,响应于高优先级队列552上出现待处理的读命令(R12),低优先级队列554上出现待处理的编程命令(P12),由于介质接口控制器550专用于访问逻辑单元(LUN 515),因而暗示了读命令(R12)与编程命令(P12)都访问逻辑单元(LUN 515),介质访问控制器550无需再检查收到的命令是否访问相同的逻辑单元。即使读命令(R12)的优先级高于编程命令(P12),介质访问控制器550也优先处理编程命令(P12)。编程命令(P12)用于向页条带(S12)写入数据。介质接口控制器550为编程命令(P12)分配XOR缓存(526),将编程命令(P12)对应的数据同XOR缓存(526)做异或,异或结果被存储在XOR缓存(526)中,以及将编程

命令 (P12) 发送给 LUN 515。接下来,介质访问控制器550向 LUN 515发出暂停命令,暂停编程命令 (P12) 的执行,并将读命令 (R12) 发送给 LUN 515。以及响应于读命令 (R12) 执行完成,向 LUN 515发出恢复命令,以恢复编程命令 (P12) 的执行。

[0097] 接下来,响应于高优先级队列552上出现待处理的读命令 (R13),低优先级队列554上出现待处理的编程命令 (P13),即使读命令 (R13) 的优先级高于编程命令 (P13),介质访问控制器550也优先处理编程命令 (P13)。编程命令 (P13) 用于向页条带 (S14) 写入数据。介质接口控制器550为编程命令 (P13) 分配XOR缓存。由于XOR缓存 (522/524/526) 均被占用,将XOR缓存之一 (例如,XOR缓存526) 所存储的内容传输到DRAM。初始化XOR缓存 (526),将编程命令 (P13) 对应的数据同XOR缓存 (526) 做异或,异或结果被存储在XOR缓存 (526) 中,以及将编程命令 (P13) 发送给 LUN 515。接下来,介质访问控制器550向 LUN 515发出暂停命令,暂停编程命令 (P13) 的执行,并将读命令 (R13) 发送给 LUN 515。以及响应于读命令 (R13) 执行完成,向 LUN 515发出恢复命令,以恢复编程命令 (P13) 的执行。

[0098] 接下来,无论介质接口控制器540/550的哪个收到向页条带 (S12) 写入数据的编程命令,都将XOR缓存 (526) 的用于页条带 (S12) 的存储信息传送到DRAM中,并从DRAM中获取之前所存储的用于页条带 (S12) 的XOR缓存的内容,并存储到XOR缓存 (526) 中。

[0099] 图6展示了本发明又一实施例的介质接口控制器。图6的实施例中,介质接口控制器640包括多个队列 (例如,队列542/544)。各个队列具有不同的优先级,例如队列542是高优先级队列,而队列544是低优先级队列。介质接口控制器640耦合到校验数据计算器620。作为举例,校验数据计算器620对输入数据做异或以得到校验数据,在校验数据计算器620中还包括XOR缓存 (622/624/626)。

[0100] 可选地,介质接口控制器640还耦合到DRAM (参看图1,DRAM 110)。

[0101] 其中,介质接口控制器640专用于访问逻辑单元 (LUN 505),因而为访问 LUN 505,将相应的命令添加队列542/544。

[0102] 可以理解地,介质接口控制器中可包括多组执行上下文,每组执行上下文专用于访问逻辑单元之一。对介质接口控制器的执行上下文进行切换,使得在每一时刻,介质接口控制器专用于访问同当前执行上下文对应的 (一个) 逻辑单元。

[0103] 图6所示的介质接口控制器640还通过总线630耦合到CPU与DRAM,从而在CPU的协助下更高效地处理I/O命令。校验数据计算器620通过总线630同DRAM交换数据,以及通过总线630向CPU指示中断,中断包括指示用于页条带的校验数据计算完成的中断 (R\_CPL),以及用于指示向页条带的写入操作处理完成的中断 (P\_CPL)。可以理解地,中断 (P\_CPL) 也可以由介质接口控制器640产生。向页条带的写入操作处理完成,指页条带的用户数据与校验数据被通过多个编程命令写入NVM芯片105。在一些情况下,一个或多个编程命令执行失败,通过响应中断 (P\_CPL),CPU也将知晓执行失败的编程命令。

[0104] 作为举例,向页条带 (S14) 写入数据,页条带 (S14) 包括3个物理页,用于存储2页用户数据以及1页校验数据。其中已经向页条带 (S14) 写入了1页用户数据。介质接口控制器640收到编程命令 (P14),编程命令 (P14) 用于向页条带 (S14) 写入第2页用户数据。在校验数据计算器620的XOR缓存 (622) 中记录了用于页条带 (S14) 的校验数据。介质接口控制器640为编程命令 (P14) 分配XOR缓存 (622),将DRAM中存储的编程命令 (P14) 对应的数据同XOR缓存 (622) 所存储的数据做异或,异或结果存储在XOR缓存 (622) 中,以及将编程命令 (P14) 发

送给LUN 505。

[0105] 接下来,由于已经完成了页条带(S14)的所有用户数据的异或计算,XOR缓存(622)中存储的是用于页条带(S14)的校验数据(X1)。校验数据计算器620将XOR缓存(622)中的校验数据通过总线传输到DRAM,生成中断(R\_CPL)并发送给CPU(例如,通过总线630)。CPU基于中断(R\_CPL)获知用于页条带(S14)的校验数据已经计算完成,并且获知用于页条带(S14)的校验数据在DRAM中的存储位置。接下来,CPU或者介质接口控制器640可将XOR缓存(622)初始化并分配给其他要写入数据的页条带。

[0106] 在接下来的任意时刻,CPU向命令队列542/544填充编程命令(P15),以将DRAM中的校验数据(X1)写入页条带(S14),并且向介质接口控制器640指示无需为编程命令(P15)计算校验数据。

[0107] 接下来,响应于LUN 505指示编程命令(P15)执行完成,介质接口控制器640向CPU产生中断(P\_CPL),在中断中指示编程命令(P15)的执行结果。若编程命令(P15)执行成功,CPU丢弃DRAM中的校验数据(X1);若编程命令(P15)执行失败,CPU生成另一编程命令(P15),以再次将DRAM中的校验数据(X1)写入NVM芯片。进一步地,若编程命令(P15)执行失败,由于页条带(S14)未被完整写入,需要读出页条带(S14)已被写入的数据,并写入另一页条带。

[0108] 继续参看图6,在另一种实施方式中,响应于收到并执行编程命令(P14),在XOR缓存(622)中生成了用于页条带(S14)的校验数据(X1)。介质接口控制器640通过编程命令(P16)将XOR缓存(622)中的数据(X1)写入LUN 505,以及将XOR缓存(622)中的数据通过总线写入DRAM。响应于校验数据(X1)被写入LUN 505的寄存器以及DRAM,向CPU生成中断(R\_CPL)。响应于中断(R\_CPL),CPU知晓校验数据(X1)已被发送到LUN 505,且在DRAM中存在校验数据(X1)的副本。接下来,CPU或者介质接口控制器640可将XOR缓存(622)初始化并分配给其他要写入数据的页条带。以及当LUN 505向介质接口控制器640指示编程命令(P16)执行完成,介质接口控制器740生成中断(P\_CPL)以向CPU指示对页条带(S14)的操作完成。

[0109] 在图6的实施例中,写入页条带时,可以更早地释放XOR缓存,并将释放的XOR缓存分配给其他编程命令,从而在固态存储设备能够并发执行更多的编程命令,降低了有限的XOR缓存资源对并发执行编程命令的数量的限制,也减少了编程命令等待XOR缓存资源的概率,降低了编程命令的处理延迟,提高了固态存储设备的性能。

[0110] 可选地,即使未接收到中断,CPU也可指示将XOR缓存中的数据通过总线630写入DRAM。或者介质接口控制器640指示将XOR缓存中的数据写入DRAM。

[0111] 图7展示了根据本申请又一实施例的在NVM芯片上的数据组织的示意图。页条带中,用户数据与校验数据的数量的比例,影响固态存储设备的NVM芯片的存储空间利用率。页条带中用户数据越多,存储空间的更多部分被用于存储用户数据,存储空间利用率越高。而受限于固态存储设备的NVM芯片数量,逻辑单元(LUN)的数量可能较少。在图7的实施例中,页条带中的物理页数量大于为页条带提供物理页的逻辑单元的数量,从而一个或多个逻辑单元为页条带提供两个或更多物理页。

[0112] 在图7的实施例中,展示了4个逻辑单元(LUN 0、LUN 1、LUN 2以及LUN3),以及2个页条带(页条带0与页条带1)。每个逻辑单元为一个页条带提供2个物理页(图7中,由形如Sa-b的附图标记指示物理页之一)。页条带0占用了物理页S0-0、物理页S0-1、物理页S0-2、物理页S0-3、物理页S0-4、物理页S0-5、物理页S0-6以及物理页S0-P。其中,物理页S0-P存储



校验数据,而其他物理页存储用户数据。页条带1占用了由形如S1-b的附图标记所指示的8个物理页,其中物理页S1-P存储校验数据,其他物理页存储用户数据。

[0113] 以及图7中,每个逻辑单元中纵向相邻的物理页在NVM芯片的物理位置上相邻。例如,物理页S0-0与其图7中纵向相邻的物理页S1-0中的对应比特来自相同的MLC (Multiple Level Cell, 多级单元) 存储单元。又例如,物理页S0-0与其图7中纵向相邻的物理页属于空间上相邻的字线。作为又一个例子,物理页S0-0与其图7中纵向相邻的物理页属于3D存储器Z轴上相邻的两层。由于来自同一MLC单元或物理位置上相邻的存储单元,图7中每个逻辑单元中纵向相邻的物理页同时发生故障的几率大于纵向不相邻的物理页同时发生故障的概率。因而为使页条带能提供有效的可靠性,需要用同时发生故障概率低的物理页来组成页条带。

[0114] 来自不同逻辑单元的物理页同时发生故障的概率低。当受限于逻辑单元数量需要由同一逻辑单元为单一页条带提供多个两个或更多物理页时,选择在NVM芯片的物理空间上不相邻、不属于同一字线和/或不属于3D存储器的同一层的物理页来构造页条带。

[0115] 作为举例,图7的实施例中,同一行的物理页(例如,物理页S0-0、物理页S0-1、物理页S0-2与物理页S0-3)具有相同的物理地址(块地址与页地址),从而便于获取页条带中的各个物理页。也可以有多种其他方式来选择物理页构造页条带。例如,依照中国专利申请201610814552.5中提供的方式构造块条带,以及由块条带的每个物理块提供一个或多个物理页来构造页条带。

[0116] 以及在向NVM芯片写入数据时,通常需按物理页地址顺序地写入数据。参看图7,在向页条带0的第1部分写入数据后,在逻辑单元0上需要向物理页S1-0写入数据,在逻辑单元1上需要向物理页S1-1写入数据,在逻辑单元2上需要向物理页S1-2写入数据,而在逻辑单元3上需要向物理页S1-3写入数据。而物理页S1-0、物理页S1-1、物理页S1-2以及物理页S1-3属于页条带1,因而接下来需要提供要写入页条带1的数据。

[0117] 返回参看图4、图5或图6,为向页条带0的第1部分写入,使用XOR缓存(例如,XOR缓存622)为页条带计算校验数据。当完成对页条带0的第一部分写入后,要向页条带1的第1部分写入数据。在向页条带1的第1部分写入数据期间,用于页条带0的第1部分的XOR缓存(例如,XOR缓存622)不会被使用。为提升XOR缓存的利用率,或者为了向页条带1的第1部分写入数据而分配XOR缓存,将例如XOR缓存622中已经根据页条带0的第1部分计算出的数据存储在DRAM,清除XOR缓存622,并将其用来为页条带1的第1部分计算校验数据。以及在将页条带1的第1部分写入NVM芯片后,将XOR缓存622的数据存储在DRAM,并将DRAM中存储的根据页条带0的第1部分计算出的数据搬移到XOR缓存,再继续根据页条带0的第2部分(不包括要写入物理页S0-P的数据)计算校验数据,得到的结果作为页条带1的校验数据,并写入物理页S0-P。接下来,再将DRAM中存储的根据页条带1的第1部分计算出的数据搬移到XOR缓存,再继续根据页条带1的第2部分(不包括要写入物理页S1-P的数据)计算校验数据,得到的结果作为页条带2的校验数据,并写入物理页S1-P。

[0118] 以此方式,仅使用一份XOR缓存,交替地为两个页条带计算校验数据。在XOR缓存资源有限的情况下,充分利用了NVM芯片写(编程)操作的带宽。不会因XOR缓存资源有限而影响固态存储设备的吞吐能力。

[0119] 可以理解地,在根据本申请的又一实施例中,页条带可包括其他数量的物理页。为

保证同一逻辑单元为单一页条带提供的两个或多个物理页在NVM芯片的物理空间上不相邻、不属于同一字线和/或不属于3D存储器的同一层,这些两个或多个物理页之间可存在一个、两个或更多的物理页。

[0120] 依然可以理解的,根据本申请的另一实施例中,页条带被分为3部分或其他数量的部分。当页条带被分为3部分,在向页条带写入数据的过程中,写完页条带的第一部分或第二部分后,都将XOR缓存的数据搬移到DRAM,以将XOR缓存分配给其他页条带。

[0121] 图8是根据图7的实施例使用XOR缓存向页条带写入数据的示意图。图8中,从左到右展示了时间的流逝方向。利用单一的XOR缓存(例如,图6的XOR缓存624)向页条带0与页条带1(参看图7)写入数据。

[0122] 参看图8,在时间段810,使用XOR缓存624为页条带0的第1部分计算校验数据。由于页条带0的第1部分有4个物理页的数据(参看图8,S0-0、S0-1、S0-2与S0-3),在为一个物理页的数据计算校验数据后,将这部分数据写入物理页。作为举例,在时间段810,首先为物理页S0-0的数据计算校验数据,经过 $T_{xor}$ 时间后,计算完成,校验数据被存入XOR缓存,并开始将数据写入物理页S0-0(由时间段820指示)。接下来,在时间段810,为页条带0的第1部分的其他物理页计算校验数据,并存入XOR缓存,以及将数据写入物理页。由于向NVM芯片的物理页写入数据所需的时间远大于计算校验数据所需的时间,在图8中,时间段820的长度远大于时间段810的长度。

[0123] 在时间段810之后,XOR缓存624中的数据被存储到DRAM,而向页条带0的第1部分写入数据的操作仍在进行。此时,XOR缓存624已被释放,可用于为其他页条带计算校验数据。图8中,在时间段812,利用XOR缓存624为页条带1的第1部分计算校验数据。以及在为页条带1的第1部分的一个物理页(例如,S1-0)生成校验数据后(从时间段820开始的 $T_{xor}$ 时间),开始向页条带1的第1部分写入数据(由时间段822指示)。

[0124] 在时间段812之后,将XOR缓存624中的数据(用于页条带1的第1部分)存储到DRAM。在时间段814,XOR缓存624再次被分配给页条带0。对DRAM中将为页条带0的第1部分生成的校验数据,以及页条带0的第2部分的用户数据,利用XOR缓存624计算校验数据。并在时间段824将数据写入页条带0的第2部分。

[0125] 在时间段816,将XOR缓存624再次分配给页条带1。对DRAM中将为页条带1的第1部分生成的校验数据,以及页条带1的第2部分的用户数据,利用XOR缓存624计算校验数据。并在时间段826将数据写入页条带1的第2部分。

[0126] 图9展示了根据本申请另一实施例的在NVM芯片上的数据组织的示意图。在图9的实施例中,展示了4个逻辑单元(LUN 0、LUN 1、LUN 2以及LUN3),以及3个页条带(页条带0、页条带1与页条带2)。每个逻辑单元为一个页条带提供2个物理页(图9中,由形如Sa-b的附图标记指示物理页之一)。作为举例,页条带0占用了物理页S0-0、物理页S0-1、物理页S0-2、物理页S0-3、物理页S0-4、物理页S0-5、物理页S0-6以及物理页S0-P。其中,物理页S0-P存储校验数据,而其他物理页存储用户数据。

[0127] 图9所示的实施例中,同一逻辑单元中,为页条带0的第1部分与第二部分提供的物理页之间间隔两个物理页。从而即使物理页S0-0发生故障,并导致物理页S1-0与物理页S2-0损坏,依然可以通过页条带0恢复物理页S0-0的数据,通过页条带1恢复物理页S1-0的数据,以及通过页条带2恢复物理页S2-0的数据。

[0128] 作为举例,使用XOR缓存624(参看图6)为页条带0、页条带1以及页条带2计算校验数据。向页条带0的第1部分写入数据后,将XOR缓存624的数据搬移到外部存储器(DRAM),清空XOR缓存624,并为页条带1的第1部分计算校验数据。向页条带1的第1部分写入数据后,将XOR缓存624的数据搬移到外部存储器(DRAM),清空XOR缓存624,并为页条带2的第1部分计算校验数据。接下来,从DRAM中将为向页条带0的第1部分计算的校验数据搬移到XOR缓存,向页条带0的第2部分(不包括S0-P)写入数据,将XOR缓存624的数据写入S0-P作为页条带0的校验数据。以及从DRAM中将为向页条带1的第1部分计算的校验数据搬移到XOR缓存,向页条带1的第2部分(不包括S1-P)写入数据,将XOR缓存624的数据写入S1-P作为页条带1的校验数据。以及类似地,用XOR缓存624生成页条带2的校验数据。

[0129] 作为另一个例子,使用XOR缓存622(参看图6)为页条带0以及页条带1计算校验数据。以及使用XOR缓存624(参看图6)为页条带2计算校验数据。

[0130] 作为依然另一个例子,页条带被分为N个部分(N为正整数),用1个XOR缓存对页条带的每个部分计算校验数据,以及将XOR缓存的数据搬移到外部存储器。在计算页条带的第m部分时(m为正整数),不从外部存储器获取XOR缓存的内容,而是为页条带的第m部分计算校验数据,并将XOR缓存的数据再次搬移到外部存储器。对于页条带的N个部分(各部分可具有不同的大小),在外部存储器中得到N份缓存数据。以及再用XOR缓存为N份缓存数据计算校验数据,得到的结果作为页条带的校验数据。

[0131] 通过将页条带分为多个部分,分别利用XOR缓存计算校验数据,使得在为页条带生成校验数据的过程中,可将XOR缓存暂时用于为其他页条带计算校验数据,并使得单一页条带的各个部分在NVM芯片的不相邻物理页上存储。

[0132] 图10是根据本申请实施例的向页条带写入数据的流程图。页条带被分为多个部分。优选地,页条带的每个部分对应的多个物理页的每个,来自不同的逻辑单元。而每个逻辑单元为页条带的每个部分提供一个物理页。可以理解的,也使用“页条带”指示要写入页条带的数据。

[0133] 为写入数据,将第一页条带(参看图7,页条带0)的部分用户数据写入构成第一页条带第一部分的来自多个逻辑单元的每个的第一物理页(1010)(也参看图7,物理页S0-0、物理页S0-1、物理页S0-2以及物理页S0-3)。

[0134] 接下来,将第二页条带(参看图7,页条带1)的部分用户数据写入构成第二页条带第一部分的来自多个逻辑单元的每个的第二物理页(1020)(也参看图7,物理页S1-0、物理页S1-1、物理页S1-2以及物理页S1-3)。

[0135] 优选地,第一物理页与第二物理页的地址连续,在物理空间上相邻,或者属于3D存储器的同一层。

[0136] 可选地,还将一个或多个其他页条带的部分用户数据写入构成其他页条带部分的来自多个逻辑单元的每个的一个或多个其他物理页。这些物理页与第二物理页的地址连续,在物理空间上相邻,或者属于3D存储器的同一层,从而使得步骤1030中的第三物理页与第一物理页在物理空间上不相邻、不属于同一字线和/或不属于3D存储器的同一层。

[0137] 接下来,将第一页条带(参看图7,页条带0)的部分用户数据与校验数据写入构成第一页条带第二部分的来自多个逻辑单元的每个的第三物理页(1030)(也参看图7,物理页S0-4、物理页S0-5、物理页S0-6以及物理页S0-P)。

[0138] 可选地,为每个页条带分配XOR资源。例如,为页条带0分配XOR资源622,为页条带1分配XOR资源624(也参看图6)。在步骤1010与步骤1030,使用XOR资源622为页条带0计算校验数据,而在步骤1020,使用XOR资源624为页条带1计算校验数据。

[0139] 作为另一种实施方式,多个页条带使用单一XOR资源(例如,XOR资源626)。例如,步骤1010,校验数据计算器620用XOR资源626为页条带0计算校验数据,并将对页条带0的第1部分用户数据的校验数据计算结果存储到外部存储器。步骤1020,校验数据计算器620用XOR资源626为页条带1计算校验数据,并将对页条带1的第1部分用户数据的校验数据计算结果存储到外部存储器。在步骤1030,校验数据计算器620用XOR资源626为页条带0计算校验数据,并将对页条带0的第2部分用户数据以及外部存储器中存储的页条带0的第1部分用户数据的校验数据计算结果计算的校验数据作为页条带0的校验数据写入物理页。

[0140] 异或操作满足交换律与结合律。在利用异或操作计算校验数据时,多个操作数的顺序没有限制。从而对于页条带的多个部分,利用从中生成的多个校验数据生成完整页条带的校验数据。

[0141] 上面以NVM芯片作为例子而公开了本申请的多种实施例。所属领域技术人员将意识到本申请的实施例也可应用于其他类型的支持擦除暂停和/或擦除恢复命令的存储介质,例如相变存储器、电阻存储器、铁电存储器等。

[0142] 已经为了示出和描述的目的而展现了对本申请的描述,并且不旨在以所公开的形式穷尽或限制本申请。对所属领域技术人员,许多调整 and 变化是显而易见的。

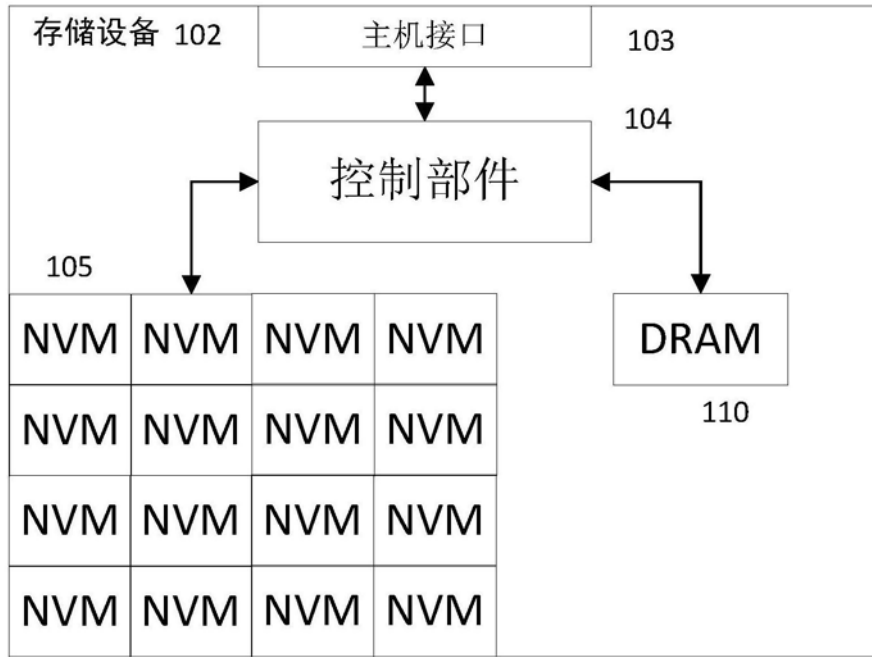


图1

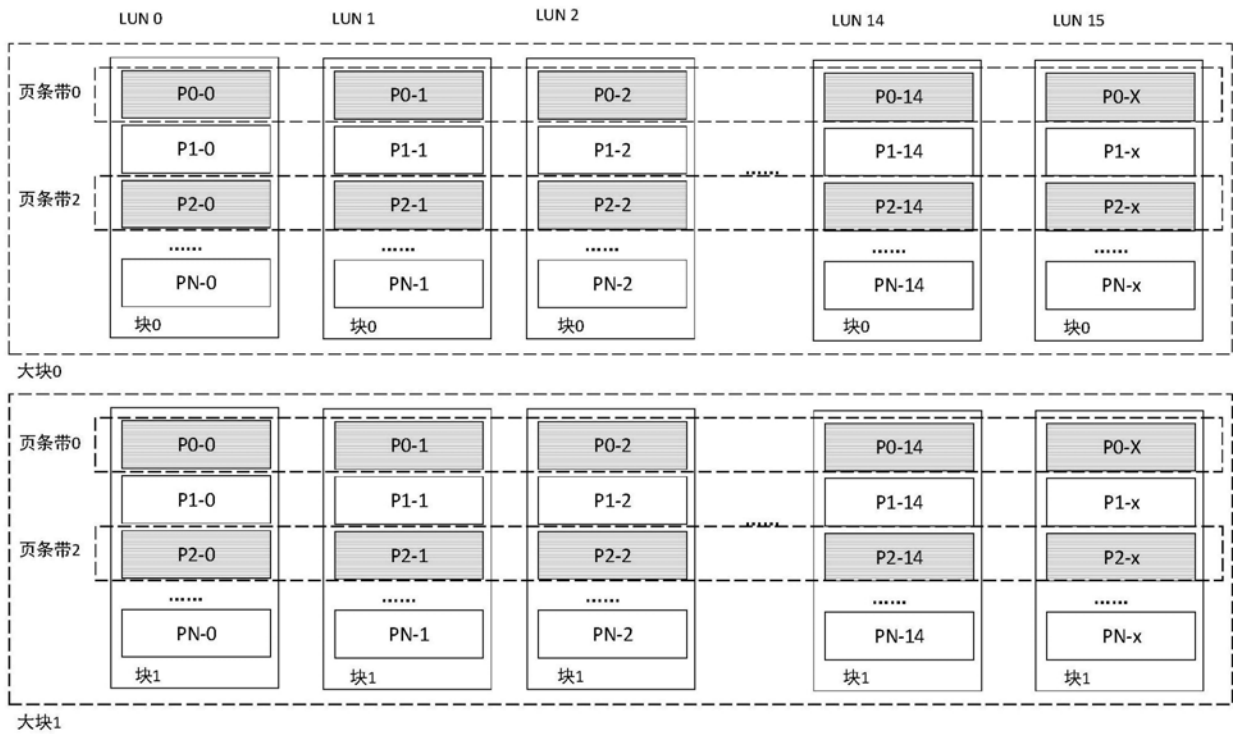


图2

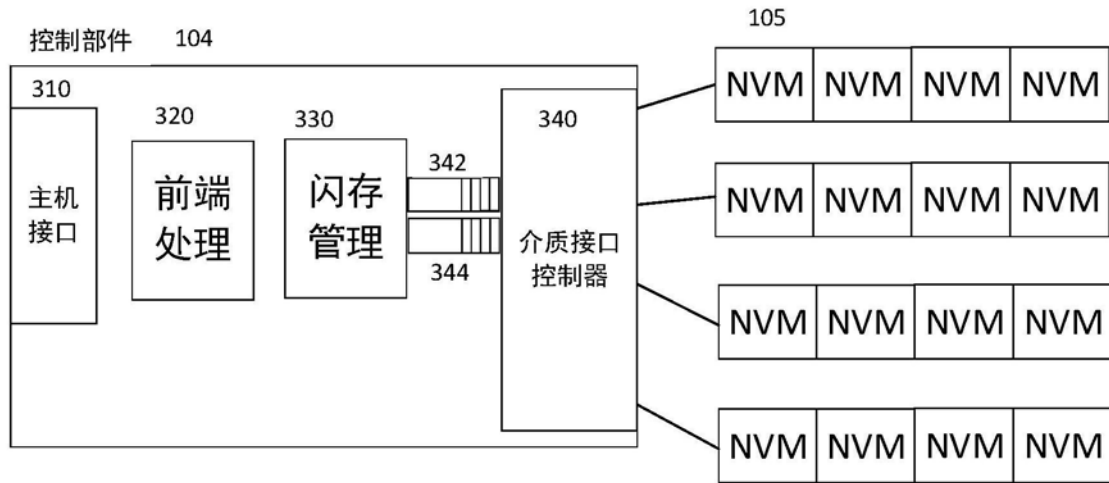


图3

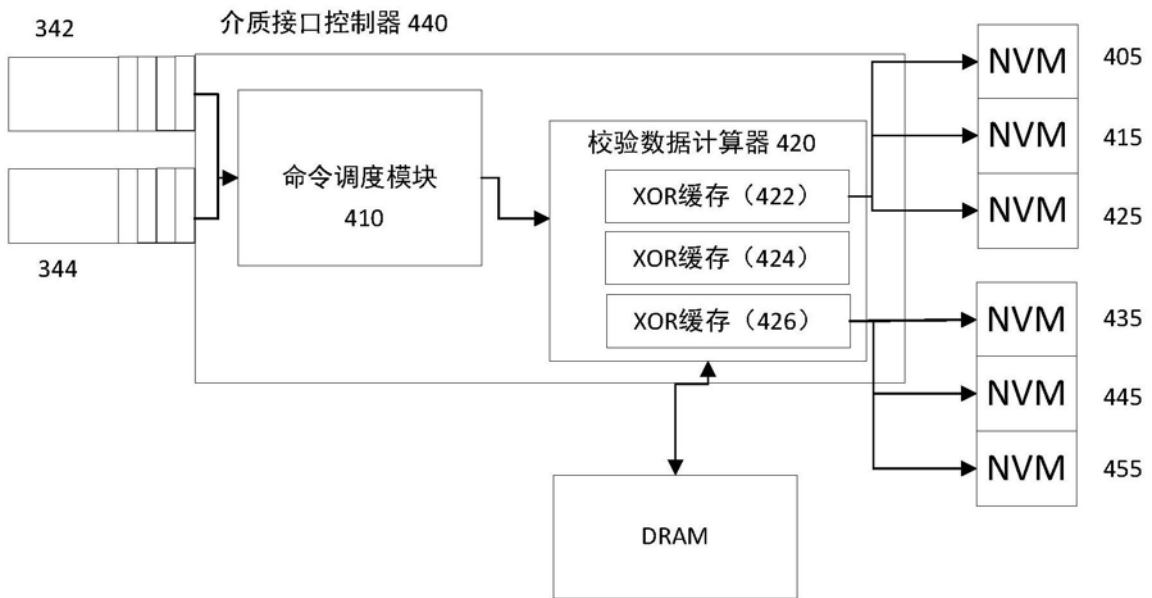


图4

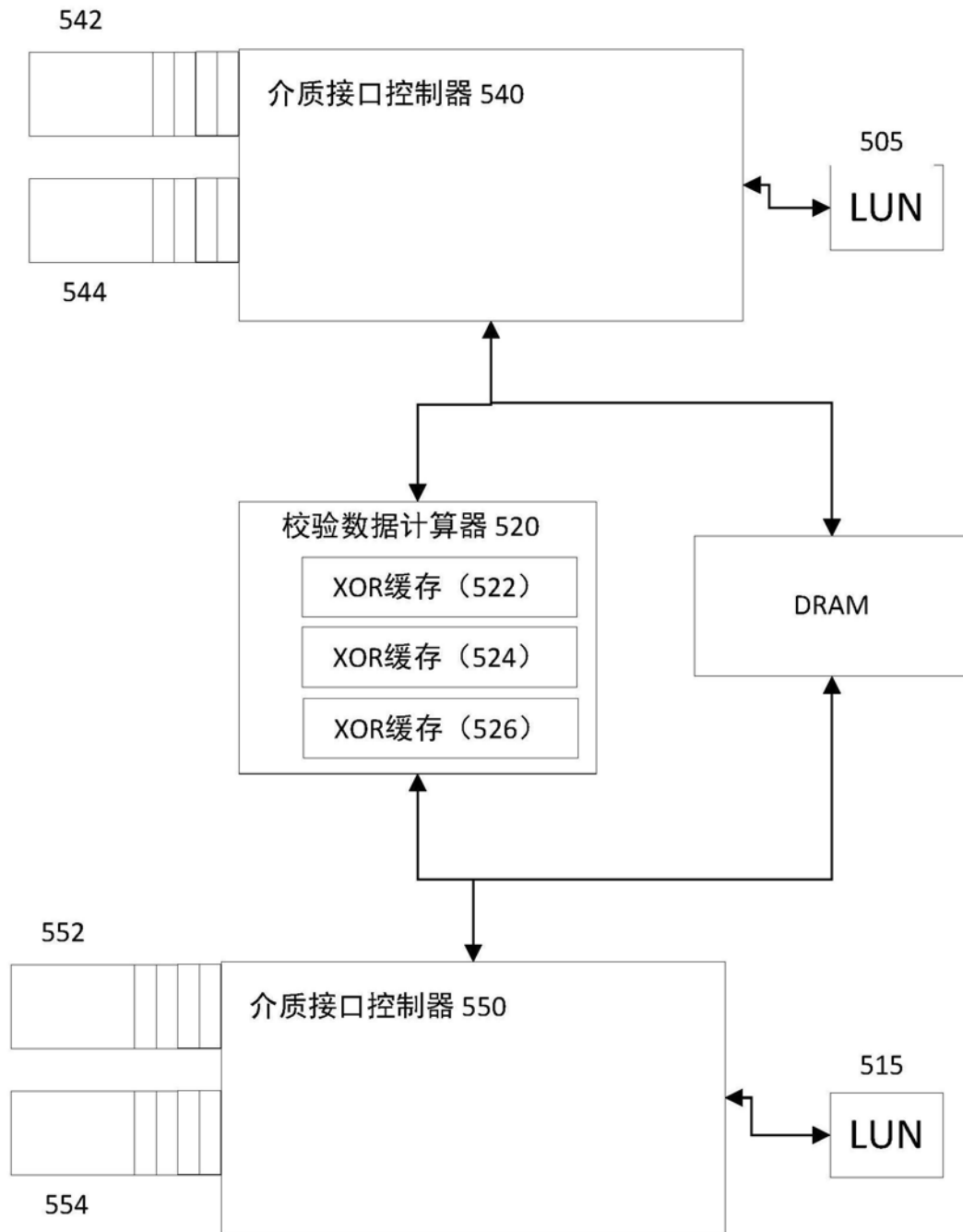


图5

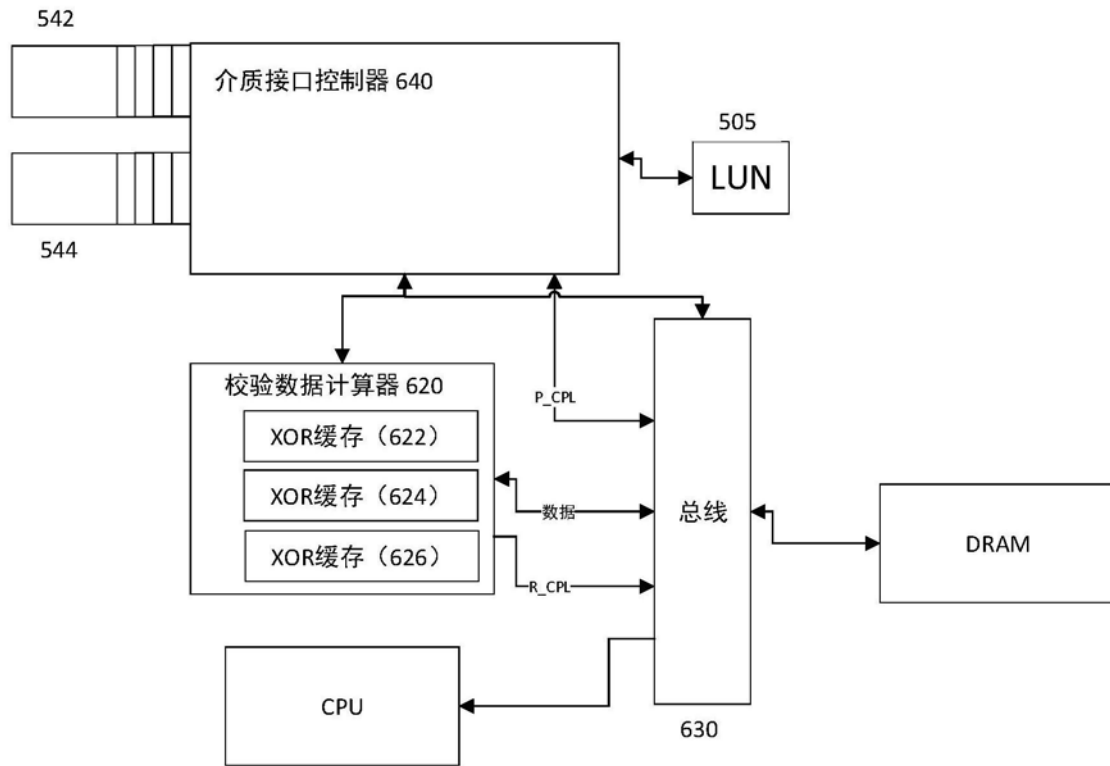


图6

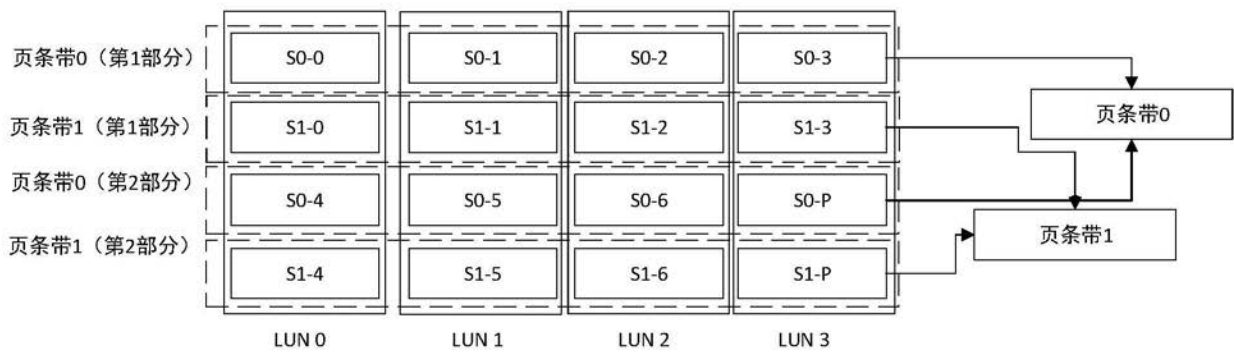


图7



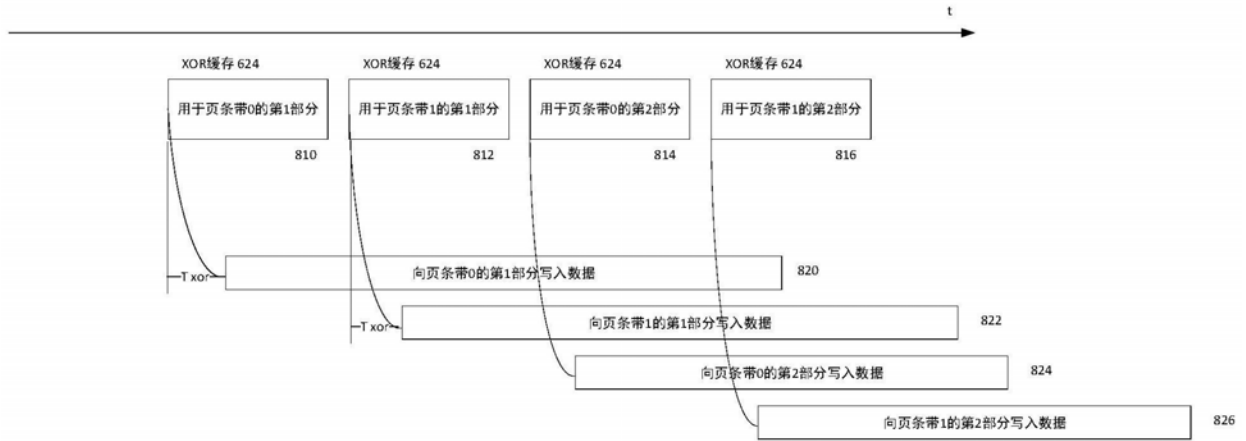


图8

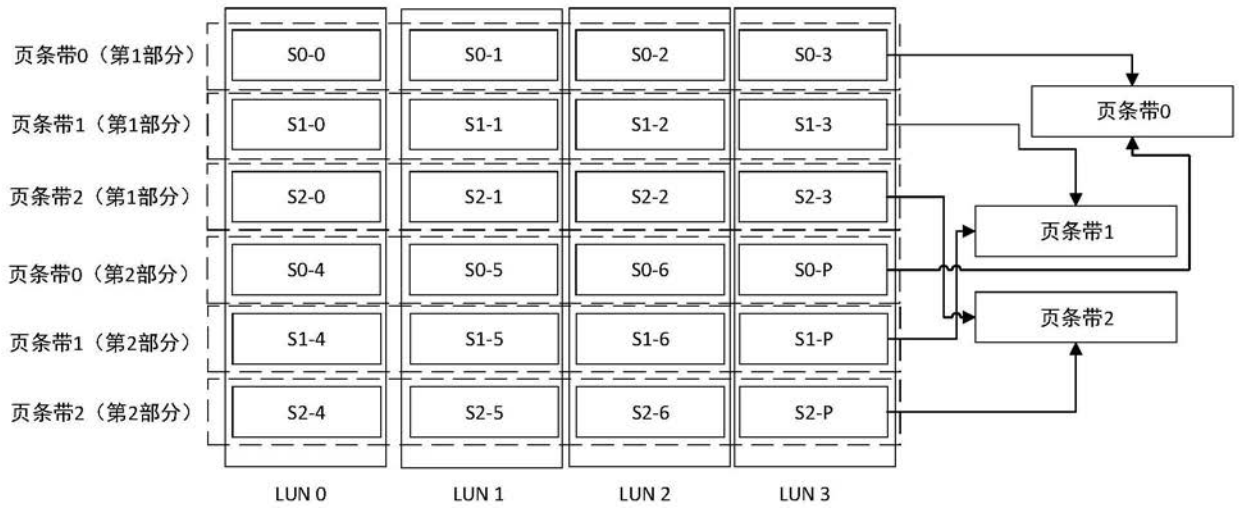


图9

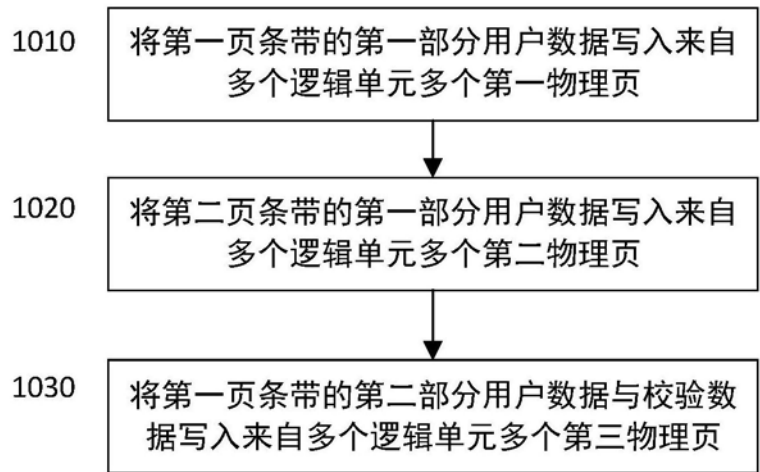


图10