(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2018 年 6 月 28 日 (28.06.2018)

WIPO | PCT

(10) 国际公布号
WO 2018/112980 A1

(51) 国际专利分类号:
H03M 13/11 (2006.01)

(21) 国际申请号: PCT/CN2016/111930

(22) 国际申请日: 2016 年 12 月 24 日 (24.12.2016)

(25) 申请语言: 中文

(26) 公布语言: 中文

(71) 申请人: 华 为 技 术 有 限 公 司 (HUAWEI TECHNOLOGIES CO., LTD.) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(72) 发明人: 曾雁星(ZENG, Yanxing); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 沈建强(SHEN, Jianqiang); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 王 工 艺 (WANG, Gongyi); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 张 进 毅 (ZHANG, Jinyi); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 吕 温 (LV, Wen); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(81) 指定国(除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国(除另有指明，要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,

(54) **Title:** STORAGE CONTROLLER, DATA PROCESSING CHIP, AND DATA PROCESSING METHOD

(54) **发明名称**: 存储控制器、数据处理芯片及数据处理方法

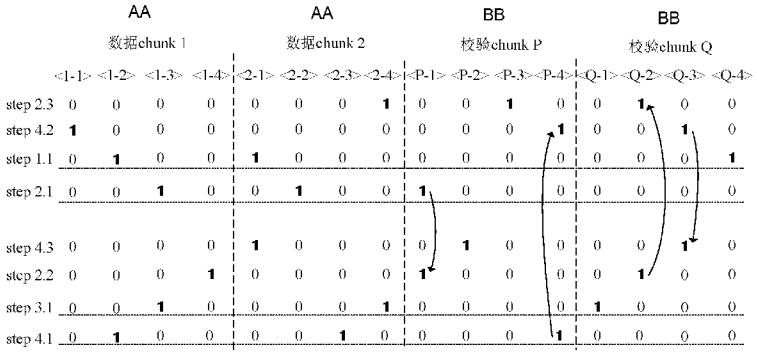

图 5

AA Data
BB Check

(57) **Abstract:** A storage controller. During operation of the storage controller, encoding of *K* data chunks acquired from a client end to be encoded is performed according to a check matrix to generate two check chunks, such that if any of the check chunks is damaged, the damaged chunk can be restored by means of the check matrix and the undamaged check chunk. The storage controller of the present invention improves the efficiency of restoring a damaged chunk.

(57) **摘要**: 一种存储控制器，该存储控制器运行时，根据校验矩阵对从客户端获取的待编码的K个数据大块chunk进行编码，以生成2个校验chunk，以便后续如果有任一chunk损坏的情况下，可以通过该校验矩阵和未损坏的chunk恢复损坏的chunk。该存储控制器提升了恢复损坏的chunk的效率。

WO 2018/112980 A1

[见续页]

CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

存储控制器、数据处理芯片及数据处理方法

技术领域

本申请涉及存储技术领域，尤其涉及一种存储控制器，数据处理芯片以及数据处理方法。

背景技术

大规模存储场景中的存储系统包括多个存储介质和存储控制器，存储介质可以由硬盘(英文：hard disk drive，缩写：HDD)或固态硬盘(英文：solid state drive，缩写：SSD)或两者的组合构成。客户端通过通信网络，将待写入数据发送至存储控制器，存储控制器对待写入的数据进行处理并存入存储介质中。现有的存储系统一般采用了独立磁盘构成的具有冗余能力的阵列(英文：redundant arrays of independent disks，缩写：RAID)技术，而RAID技术的核心就是纠删码(英文：erasure code)的编码和解码技术。

现有的纠删码的编码和解码技术效率较低。

发明内容

本申请提供了一种存储控制器，以提升纠删码的编码和解码效率。

本申请的第一方面提供了一种存储控制器，包括了处理器、存储器和通信接口。该存储控制器运行时，该处理器通过该通信接口获取待编码的K个数据大块chunk，并将所述K个数据chunk缓存入所述存储器，每个数据chunk包括R个数据编码块，R+1为素数且R+1>K。

该处理器通过该通信接口持续接收客户端发来的待写入数据并缓存入该存储器。该存储器中缓存了预设数量大小的待写入数据后，该处理器将所述预设数量的待写入数据分成K个待编码的数据大块，每个数据chunk被分为R个数据编码块。

随后，该处理器，还用于执行所述存储器中的代码执行以下操作：读取所述存储器中存储的所述K个数据chunk，根据校验矩阵和所述K个数据chunk

生成第一校验chunk和第二校验chunk，每个校验chunk包括R个校验编码块。

其中，所述校验矩阵有2*R行，所述校验矩阵中第(k-1)*R+1列至第k*R列为所述K个数据chunk中第k个数据chunk的chunk列集合，K≥k≥1，所述校验矩阵中第K*R+1列至第(K+1)*R列为对应所述第一校验chunk的chunk列集合，所述校验矩阵中第(K+1)*R+1列至第(K+2)*R列为所述第二校验chunk的chunk列集合。

所述校验矩阵为标准校验矩阵H或由标准校验矩阵H执行N次调换操作后得到，N≥1，所述调换操作指将任意两个chunk列集合调换；所述标准校验矩阵H中除以下坐标为1外，其余坐标均为0，2*R≥i≥1，(K+2)*R≥j≥1，

如果i<j，则

$H[i+1][j*R+(R-j+i)\bmod R+1]$

$H[R+i+1][(j+1)*R-(R-j+i)\bmod R]$

如果i>j，则

$H[i+1][j*R+(R-1-j+i)\bmod R+1]$

$H[R+i+1][(j+1)*R-(R-1-j+i)\bmod R]$。

所述校验矩阵在所述存储器中的存储形态可以有两种。第一种存储形态为一个行数为2*R，列数为(K+2)*R的矩阵。由于校验矩阵的每一行代表了一个异或方程，因此该校验矩阵代表了2*R个异或方程，因此，所述校验矩阵的第二种存储形态为2*R个异或方程，通过这2*R个异或方程也可以获得一个行数为2*R，列数为(K+2)*R的矩阵。

结合第一方面，在第一方面的第一种实现方式中，所述校验矩阵中第(k-1)*R+1 列至第 k*R 列分别对应所述 K 个数据 chunk 中第 k 个数据 chunk 的 R 个数据编码块，所述校验矩阵中第 K*R+1 列至第(K+1)*R 列分别对应所述第一校验 chunk 的 R 个校验编码块，所述校验矩阵中第(K+1)*R+1 列至第(K+2)*R 列分别对应所述第二校验 chunk 的 R 个校验编码块。也即，所述校验矩阵中的每一列，对应一个数据编码块或者一个校验编码块。

所述校验矩阵的第 D 行有 3 个坐标为 1，所述第 D 行为所述校验矩阵的任一行。这 3 个为 1 的坐标对应了 3 个编码块。对所述校验矩阵的第 D 行中为 1 的坐标对应的 3 个编码块中的任意 2 个编码块进行一次异或运算可以得

到所述校验矩阵的第 D 行中为 1 的坐标对应的 3 个编码块中未参与本次异或运算的编码块。也即如果所述校验矩阵的某一行的 3 个为 1 的坐标分别对应了编码块 1、编码块 2 和编码块 3，则编码块 1、编码块 2 和编码块 3 中任意两个编码块之间进行异或运算可以得到另一个编码块。

结合第一方面的第一种实现方式，在第一方面的第二种实现方式中，所述处理器在根据所述校验矩阵生成第一校验 chunk 和第二校验 chunk 的过程中，从起始编码行开始编码。该起始编码行为第一校验 chunk 和第二校验 chunk 对应的 2*R 列中仅有 1 个坐标为 1 的行。

由于校验矩阵的每一行对应一个异或方程，而编码开始时，已知的仅有 K*R 个数据编码块，因此编码只能从起始编码行开始。所述校验矩阵一共有 4 个起始编码行。

完成第一起始编码行对应的异或方程后（第一起始编码行为 4 个起始编码行之任一），获取了校验编码块 1。如果校验编码块 1 在所述校验矩阵对应的列中只有 1 个坐标为 1，则第一起始编码行起始的编码过程完成。如果校验编码块 1 在所述校验矩阵对应的列中有 2 个坐标为 1，则跳转到这 2 个坐标所在的行中还未被用于生成校验编码块的行。执行跳转到的行对应的异或方程，获取了校验编码块 2。对校验编码块 2 执行校验编码块 1 相同的处理。如果校验编码块 2 在所述校验矩阵对应的列中只有 1 个坐标为 1，则第一起始编码行起始的编码过程完成。如果校验编码块 2 在所述校验矩阵对应的列中有 2 个坐标为 1，则跳转到这 2 个坐标所在的行中还未被用于生成校验编码块的行。执行跳转到的行对应的异或方程，获取了校验编码块 3，对校验编码块 2 执行校验编码块 1 相同的处理，依次类推，直至第一起始编码行起始的编码过程完成。对所述校验矩阵的 4 个起始编码行，均执行上述第一起始编码行的编码过程，4 个起始编码行起始的编码过程均完成后，则获取了 2*R 个校验编码块。

结合第一方面或第一方面的第一种或第二种实现方式，在第一方面的第三种实现方式中，所述处理器根据所述校验矩阵，获取了所述第一校验 chunk 和所述第二校验 chunk 后，通过所述通信接口将所述 K 个数据 chunk、所述第一校验 chunk 和所述第二校验 chunk 分别存入所述存储控制器所在的存储系统的 K+2 个存储介质中。一般不同的 chunk 存入不同的存储介质中。

获取2*R个校验编码块后，每R个校验编码块形成一个校验chunk，这2个校验chunk和K个数据chunk组成了一个chunk group，一个chunk group中的每个chunk被存入不同的存储介质中，以便后续有存储介质损坏的时候，可以通过该chunk group中未损坏的chunk来恢复损坏的存储介质上存储的chunk。

结合第一方面的第三种实现方式，在第一方面的第四种实现方式中，在所述K个数据chunk、所述第一校验chunk和所述第二校验chunk被存入所述存储控制器所在的存储系统的K+2个存储介质中后，如果所述K+2个存储介质中有存储介质损坏，则所述处理器根据所述校验矩阵和所述K+2存储介质中未损坏的存储介质上存储的数据chunk和所述第一校验chunk和所述第二校验chunk中的至少一个，恢复所述损坏的存储介质。

如果损坏的存储介质上存储的是数据chunk，则根据其余K+1个存储介质上存储的K-1个数据chunk和2个校验chunk恢复损坏的数据chunk。如果损坏的存储介质上存储的是校验chunk，则根据其余K+1个存储介质上存储的K个数据chunk和1个校验chunk，恢复损坏的校验chunk。

恢复过程中，虽然要使用到未损坏的每个数据chunk和未损坏的每个校验chunk，但并不需要使用到未损坏的每个数据chunk的每个数据编码块和未损坏的每个校验chunk的每个校验编码块，具体采用哪些数据编码块和哪些校验编码块来恢复损坏的chunk，需要根据损坏的chunk在所述校验矩阵中对应的列参与了哪几行对应的异或方程决定。

由于解码过程与编码过程强耦合，通过在编码过程中的改进，本申请第一方面提供的存储控制器降低了恢复开销，提升了后续恢复损失的chunk时的效率。

本申请的第二方面提供了一种数据处理芯片，包括电路和读写接口；所述电路用于，通过所述读写接口获取待编码的K个数据大块chunk，每个数据chunk包括R个数据编码块，R+1为素数且R+1>K；所述电路还用于，根据校验矩阵和所述K数据chunk生成第一校验chunk和第二校验chunk，每个校验chunk包括R个校验编码块；其中，所述校验矩阵有2*R行，所述校验矩阵中第(k-1)*R+1列至第k*R列为所述K个数据chunk中第k个数据chunk的chunk列

集合，K≥k≥1，所述校验矩阵中第K*R+1列至第(K+1)*R列为应所述第一校验chunk的chunk列集合，所述校验矩阵中第(K+1)*R+1列至第(K+2)*R列为所述第二校验chunk的chunk列集合；所述校验矩阵为标准校验矩阵H或由标准校验矩阵H执行N次调换操作后得到，N≥1，所述调换操作指示将任意两个chunk列集合调换；所述标准校验矩阵H中除以下坐标为1外，其余坐标均为0，2*R≥i≥1，(K+2)*R≥j≥1，

如果i<j，则

$H[i+1][j*R+(R-j+i)\bmod R+1]$

$H[R+i+1][(j+1)*R-(R-j+i)\bmod R]$

如果i>j，则

$H[i+1][j*R+(R-1-j+i)\bmod R+1]$

$H[R+i+1][(j+1)*R-(R-1-j+i)\bmod R]$。

所述第二方面的具体实现方式及取得的技术效果与所述第一方面的各实现方式类似，在此不再赘述。

本申请的第三方面提供了一种数据处理方法，第一方面中提供的存储控制器和第二方面中提供的数据处理芯片工作时，执行该数据执行方法，包括：

获取待编码的K个数据大块chunk并缓存所述K个数据chunk，每个数据chunk包括R个数据编码块，R+1为素数且R+1>K；

根据校验矩阵和所述K个数据chunk生成第一校验chunk和第二校验chunk，每个校验chunk包括R个校验编码块；

其中，所述校验矩阵有2*R行，所述校验矩阵中第(k-1)*R+1列至第k*R列为所述K个数据chunk中第k个数据chunk的chunk列集合，K≥k≥1，所述校验矩阵中第K*R+1列至第(K+1)*R列为对应所述第一校验chunk的chunk列集合，所述校验矩阵中第(K+1)*R+1列至第(K+2)*R列为所述第二校验chunk的chunk列集合；

所述校验矩阵为标准校验矩阵H或由标准校验矩阵H执行N次调换操作后得到，N≥1，所述调换操作指将任意两个chunk列集合调换；所述标准校验矩阵H中除以下坐标为1外，其余坐标均为0，2*R≥i≥1，(K+2)*R≥j≥1，

如果i<j，则

$H[i+1][j*R+(R-j+i) \bmod R+1]$

$H[R+i+1][(j+1)*R-(R-j+i) \bmod R]$

如果 i>j，则

$H[i+1][j*R+(R-1-j+i) \bmod R+1]$

$H[R+i+1][(j+1)*R-(R-1-j+i) \bmod R]$。

结合第三方面，在第三方面的第一种实现方式中，所述校验矩阵中第(k-1)*R+1 列至第 k*R 列分别对应所述 K 个数据 chunk 中第 k 个数据 chunk 的 R 个数据编码块，所述校验矩阵中第 K*R+1 列至第(K+1)*R 列分别对应所述第一校验 chunk 的 R 个校验编码块，所述校验矩阵中第(K+1)*R+1 列至第(K+2)*R 列分别对应所述第二校验 chunk 的 R 个校验编码块；

所述校验矩阵的第 D 行有 3 个坐标为 1，所述第 D 行为所述校验矩阵的任一行，对所述校验矩阵的第 D 行中为 1 的坐标对应的 3 个编码块中的任意 2 个编码块进行一次异或运算可以得到所述校验矩阵的第 D 行中为 1 的坐标对应的 3 个编码块中未参与本次异或运算的编码块。

结合第三方面的第一种实现方式，在第三方面的第二种实现方式中，在根据所述校验矩阵生成第一校验 chunk 和第二校验 chunk 的过程中，从起始编码行开始编码。该起始编码行为第一校验 chunk 和第二校验 chunk 对应的 2*R 列中仅有 1 个坐标为 1 的行。

完成第一起始编码行对应的异或方程后（第一起始编码行为 4 个起始编码行之任一），获取了校验编码块 1。如果校验编码块 1 在所述校验矩阵对应的列中只有 1 个坐标为 1，则第一起始编码行起始的编码过程完成。如果校验编码块 1 在所述校验矩阵对应的列中有 2 个坐标为 1，则跳转到这 2 个坐标所在的行中还未被用于生成校验编码块的行。执行跳转到的行对应的异或方程，获取了校验编码块 2。对校验编码块 2 执行校验编码块 1 相同的处理。如果校验编码块 2 在所述校验矩阵对应的列中只有 1 个坐标为 1，则第一起始编码行起始的编码过程完成。如果校验编码块 2 在所述校验矩阵对应的列中有 2 个坐标为 1，则跳转到这 2 个坐标所在的行中还未被用于生成校验编码块的行。执行跳转到的行对应的异或方程，获取了校验编码块 3，对校验编码块 2 执行校验编码块 1 相同的处理，依次类推，直至第一起始编码行起始的编码过程完成。对所述校验矩阵的 4 个起始编码行，均执行上述第一起

始编码行的编码过程，4 个起始编码行起始的编码过程均完成后，则获取了 2*R 个校验编码块。

结合第三方面或第三方面的第一种或第二种实现方式，在第三方面的第三种实现方式中，该方法还包括：将所述 K 个数据 chunk、所述第一校验 chunk 和所述第二校验 chunk 分别存入执行该数据处理方法的存储控制器所在的存储系统的 K+2 个存储介质中。

结合第三方面的第三种实现方式，在第三方面的第四种实现方式中，该方法还包括：当所述存储控制器所在的存储系统的 K+2 个存储介质中有存储介质损坏时，根据所述校验矩阵和所述存储控制器所在的存储系统的 K+2 存储介质中未损坏的存储介质上存储的数据 chunk 和所述第一校验 chunk 和所述第二校验 chunk 中的至少一个，恢复所述损坏的存储介质。

本申请第三方面提供的数据处理方法通过在编码过程中的改进，降低了恢复开销，提升了后续恢复损失的 chunk 时的效率。

本申请第四方面提供了一种存储介质，该存储介质中存储了程序，该程序被计算设备运行时，该计算设备执行前述第三方面或第三方面的任一实现方式提供的数据处理方法。该存储介质包括但不限于只读存储器，随机访问存储器，快闪存储器、HDD 或 SSD。

本申请第五方面提供了一种计算机程序产品，该计算机程序产品包括程序指令，当该计算机程序产品被计算机执行时，该计算机执行前述第三方面或第三方面的任一实现方式提供的数据处理方法。该计算机程序产品可以为一个软件安装包，在需要使用前述第三方面或第三方面的任一实现方式提供的数据处理方法的情况下，可以下载该计算机程序产品并在计算设备上执行该计算机程序产品。

附图说明

为了更清楚地说明本申请实施例的技术方法，下面将对实施例中所需要使用的附图作以简单地介绍。

图 1-1 为本申请实施例提供的存储系统的组织结构示意图；

图 1-2 为本申请实施例提供的另一存储系统的组织结构示意图；

图 2 为本申请实施例提供的又一存储系统的组织结构示意图；

图 3 为本申请实施例提供的校验矩阵的结构示意图；

图 4-1 为本申请实施例提供的另一校验矩阵的结构示意图；

图 4-2 为本申请实施例提供的又一校验矩阵的结构示意图；

图 5 为本申请实施例提供的又一校验矩阵的结构示意图；

图 6 为本申请实施例提供的存储控制器的组织结构示意图；

图 7 为本申请实施例提供的另一存储控制器的组织结构示意图；

图 8 为本申请实施例提供的又一存储控制器的组织结构示意图；

图 9 为本申请实施例提供的又一存储控制器的组织结构示意图；

图 10 为本申请实施例提供的数据处理芯片的组织结构示意图。

具体实施方式

下面结合本申请实施例中的附图，对本申请实施例中的技术方法进行描述。

本申请中各个“第一”、“第二”、“第 n”之间不具有逻辑或时序上的依赖关系。

贯穿本说明书，两个编码块之间的异或运算(英文：exclusive OR，缩写：XOR)，指代两个编码块的每一 bit 数据依次进行异或运算。例如编码块 1 的第 1 bit 与编码块 2 的第 1bit 进行异或运算，得到编码块 3 的第 1bit，依次类推，直至编码块的最后一个 bit 与编码块 2 的最后一个 bit 进行异或运算，得到编码块 3 的最后一个 bit。则，编码块 3 由编码块 1 和编码块 2 之间进行异或运算得到。

贯穿本说明书，恢复开销为衡量存储 chunk group 的数据的 K+2 个存储介质中任一存储介质损坏的情况下，恢复损坏的存储介质所需的对存储介质的访问开销的参数。恢复开销等于恢复损坏的存储介质时从未损坏的存储介质读取的编码块的大小与该 chunk group 中全部数据编码块的大小之比。因此，恢复开销越小，说明了在有存储介质损坏的情况下，所需的恢复时间越短。chunk group 的定义，将在下文详细说明。

贯穿本说明书，mod 为求余函数，也即 A mod B 指示 A 除以 B 得到的

余数，A 和 B 均为整数，例如 4 mod 3 = 1。

本申请实施例所应用的架构

如图 1-1 和图 1-2 介绍了两种不同架构的存储系统。图 1-1 中的存储系统也称为存储阵列，存储控制器和存储介质均设置于存储阵列内部。图 1-2 为分布式的存储系统，该存储系统包括多个存储节点，每个存储节点实际可以为服务器。该存储系统的至少一个存储节点包括存储控制器，每个存储节点均包括存储介质，各个存储节点通过通信网络建立通信连接。

图 1-1 的存储阵列中的存储控制器仅对客户端发往该存储阵列的待写入数据进行处理。图 1-2 中的每个存储控制器均可以接收客户端发来的待写入数据并对其进行纠删码的编码和解码处理。一个存储控制器编码后获取的数据不仅可以被存入该存储控制器所在的存储节点的存储介质，还可以通过通信网络发往其他存储节点的存储介质，以实现分布式的存储。由于分布式的存储系统中，可能存在多个存储控制器并行工作，因此这多个存储控制器中的每个存储控制器负责存储系统中的一个存储节点组，每个存储节点组中包括至少一个存储节点。一个存储节点组中的存储控制器负责接收客户端发来的待写入数据，对其进行编码后存入该存储节点组的存储节点中。下文中的存储控制器，可以指代图 1-1 或图 1-2 中的任一存储控制器，该存储控制器用于纠删码的编码和解码。

如图 2 所示，在存储系统运行过程中，存储控制器持续接收客户端发来的待写入数据，接收到预设数量大小的待写入数据后，存储控制器将所述预设数量的待写入数据分成 K 个待编码的数据大块（英文：chunk），每个数据 chunk 被分为 R 个数据编码块，并根据这 K*R 个数据编码块和纠删码的编码方法，生成 2 个校验 chunk，每个校验 chunk 包括 R 个校验编码块。每 K 个数据 chunk 和通过这 K 个数据 chunk 生成的校验 chunk 组成一个 chunk 组(英文:group)。其中，每个 chunk 的大小可以根据需要进行设置，例如为 512 Byte、4k Byte、8k Byte、32k Byte 等。

生成一个 chunk group 后，存储控制器将该 chunk group 中的每个 chunk 存入一个 SSD 中，存储系统采用的存储介质为 HDD 或其他种类的设备的情况与之类似。存储控制器将一个 chunk group 中每个 chunk 存入对应的 SSD

中之后，继续将客户端发来的待写入数据形成另一个 chunk group 并采用类似的方式存储。

每个 chunk 在 SSD 中被分为 R 个编码块进行存储。图 2 中，将数据 chunk 对应的编码块称为数据编码块，将校验 chunk 对应的编码块称为校验编码块。虽然每个 chunk 的全部 R 个编码块都存储于同一个 SSD，但这 R 个编码块的存储地址(可以是物理存储地址或者逻辑存储地址)可以不连续。一般一个 chunk group 中的每个编码块的大小相同。R 和 K 需要符合以下条件：R+1 为素数且 R+1>K，R、K 均为正整数。R 一般为存储系统的配置参数，K 一般为用户设置的参数，为了应对用户的不同需求，存储控制器具有应对不同 K 和 R 的配置下的纠删码的编码和解码方法。

如果任何一个 SSD 损坏了，那么需要用到损坏的 SSD 上的 chunk 所属的 chunk group 上的其余 chunk 来恢复损坏的 SSD 上的 chunk，恢复过程需要使用纠删码的解码方法。

每个校验编码块，是由 2 个来自 2 个不同 chunk 的编码块进行异或运算得到的，这两个 chunk 可以为一个数据 chunk 和一个校验 chunk，或者这两个 chunk 均为数据 chunk。存储控制器在生成校验 chunk 的过程中通过预设于存储控制器中的校验矩阵来确定这 2*R 个校验编码块中的每个校验编码块分别由哪 2 个编码块运算得到。

由于异或运算的特性，生成一个校验编码块的 2 个编码块和该校验编码块之中，任意 2 个编码块进行异或运算都可以得到剩余的 1 个编码块。因此，当任一 SSD 损坏时，存储控制器通过该校验矩阵也可以得知该损坏的 SSD 上存储的 chunk 的每个编码块可以通过哪 2 个编码块运算得出。

该校验矩阵的行数为 2*R 且列数为(K+2)*R。校验矩阵的每一列对应一个编码块，每一行对应一个异或方程。如图 3 所示，X-Y 指代数据 chunk X 的第 Y 个编码块，后文称之为数据编码块 X-Y，K≥X≥1，R≥Y≥1。而两个校验 chunk 分别称之为校验 chunk P 和校验 chunk Q，因此 P-Y 指代校验 chunk P 的第 Y 个编码块，后文称之为校验编码块 P-Y，而 Q-Y 指代校验 chunk Q 的第 Y 个编码块，后文称之为校验编码块 Q-Y，R≥Y≥1。

校验矩阵中每一个 chunk 对应的 R 列，合称为一个 chunk 列集合，因此一个行数为 2*R 且列数为(K+2)*R 的校验矩阵中，一共有 K+2 个 chunk 列集

合。该校验矩阵的第 1 至第 R 列属于数据 chunk 1 对应的 chunk 列集合，该校验矩阵的第 R+1 至第 2R 列属于数据 chunk 2 对应的 chunk 列集合，依次类推，该校验矩阵的第 K*R+1 至第(K+1)*R 列属于校验 chunk P 对应的 chunk 列集合，该校验矩阵的第(K+1)*R 至第(K+2)*R 列属于校验 chunk Q 对应的 chunk 列集合。

校验矩阵的每一行有 3 个坐标为 1，指示这 3 个坐标对应的 3 个编码块中任意 2 个之间进行异或运算可以得到另一个编码块。如图 3 中，校验矩阵第 1 行指示：数据编码块 2-R、数据编码块 K-3、校验编码块 Q-R 三者之中，任意两者之间进行异或运算可以获得没参与异或运算的编码块。需要说明的是，图 3 仅为示例性的展示校验矩阵的结构。

本申请实施例所应用的校验矩阵

本申请实施例所应用的校验矩阵，可以为标准校验矩阵 H，或由标准校验矩阵 H 执行 N 次调换操作后得到，N≥1。一次调换操作指，将一个行数为 2*R 且列数为(K+2)*R 的矩阵中任意两个 chunk 列集合调换。由于标准校验矩阵 H 实际提供了 2*R 个异或方程，每个异或方程用于得出 1 个校验编码块，因此对标准校验矩阵 H 执行 N 次调换操作后得到的矩阵，仍然可以得出 2*R 个校验编码块。

该标准矩阵 H 符合以下条件，下式中 2*R≥i≥1，(K+2)*R≥j≥1：

如果 i<j，则

$$H[i+1][j*R+(R-j+i)\bmod R+1]=1 \qquad \text{式 1}$$

$$H[R+i+1][(j+1)*R-(R-j+i)\bmod R]=1 \qquad \text{式 2}$$

如果 i>j，则

$$H[i+1][j*R+(R-1-j+i)\bmod R+1]=1 \qquad \text{式 3}$$

$$H[R+i+1][(j+1)*R-(R-1-j+i)\bmod R]=1 \qquad \text{式 4}$$

除式 1 至式 4 指示的坐标外，标准校验矩阵 H 中其余坐标均为 0。

如图 4-1 为 K=2，R=4 时的标准矩阵 H。图 4-2 为将标准矩阵 H 的数据 chunk 1 对应的 chunk 列集合和数据 chunk 2 对应的 chunk 列集合调换后得到的校验矩阵。可以看到，图 4-1 提供的标准矩阵 H 的第 1 列至第 4 列的内容被调换到了图 4-2 提供的校验矩阵的第 5 列至第 8 列，而图 4-1 提供的标准

矩阵 H 的第 5 列至第 8 列的内容被调换到了图 4-2 提供的校验矩阵的第 1 列至第 4 列。图 4-2 提供的校验矩阵中，第 1 列至第 4 列依然对应数据 chunk 1 的 4 个数据编码块，第 5 列至第 8 列依然对应数据 chunk 5 的 4 个数据编码块。

以下，介绍编码过程。

首先，从校验矩阵中的起始编码行进行编码。

起始编码行为校验矩阵中 2 个校验 chunk 对应的 2*R 列中仅有 1 个坐标为 1 的行。由于校验矩阵的每一行对应一个异或方程，而编码开始时，已知的仅有 K*R 个数据编码块，因此编码只能从起始编码行开始。

每个校验矩阵中，有 4 个起始编码行。这 4 个起始编码行对应的编码过程互不干扰，实际使用中可以并行执行。

针对起始编码行 1 的编码过程如下：

step 1：根据起始编码行 1 进行异或运算得出校验编码块 1。

step 2：如果校验编码块 1 仅参与了一个异或方程，即校验编码块 1 所在的列仅有一个坐标为 1，则起始编码行 1 起始的编码结束。

step 3：如果校验编码块 1 参与了两个异或方程，即校验编码块 1 所在的列有两个坐标为 1。则根据校验编码块 1 参与的两个异或方程中，未在 step 2 中使用的异或方程所在的行进行编码，得到校验编码块 2。

对校验编码块 2 继续执行 step 2 或 step 3，也即如果校验编码块 2 仅参与了一个异或方程，即校验编码块 2 所在的列仅有一个坐标为 1，则起始编码行 1 起始的编码结束。校验编码块 2 如果参与了两个异或方程，即校验编码块 2 所在的列有两个坐标为 1，则根据校验编码块 2 参与的两个异或方程中，未在中使用的异或方程所在的行进行编码，得到校验编码块 3。

对校验编码块 3 继续执行 step 2 或 step 3。依此类推，直至某次编码后，得到的校验编码块仅参与了 1 个异或方程，则该起始编码行 1 起始的编码结束。

其中，起始编码行 1 为一个校验矩阵中的 4 个起始编码行中之任一。

对校验矩阵的 4 个起始编码行，均执行上述起始编码行 1 的编码过程即可获取 2*R 个校验编码块，则存储控制器获取了全部的数据 chunk 和校验

chunk。

以 K=2、R=4，且采用标准校验矩阵的情况为例，如图 5，第 3、4、7、8 行中，校验 chunk P 和校验 chunk Q 对应的 2*R 列内仅有 1 个坐标为 1，因此第 3、4、7、8 行为该校验矩阵的起始编码行。

根据第 3 行进行编码：

step 1.1 数据编码块 1-2 XOR 数据编码块 2-1 = 校验编码块 Q-4。

由标准校验矩阵的第 16 列可知，校验编码块 Q-4 仅参与了一个异或方程，因此由第 3 行起始的编码结束。

根据第 4 行进行编码：

step 2.1 数据编码块 1-3 XOR 数据编码块 2-2 =校验编码块 P-1。

由标准校验矩阵的第 9 列可知，校验编码块 P-1 参与了两个异或方程，分别对应标准校验矩阵的第 4 行和第 6 行，因此，接下来根据第 6 行进行编码：

step 2.2 数据编码块 1-4 XOR 校验编码块 P-1 =校验编码块 Q-2。

由标准校验矩阵的第 14 列可知，校验编码块 Q-2 参与了两个异或方程，分别对应标准校验矩阵的第 6 行和第 1 行，因此，接下来根据第 1 行进行编码：

step 2.3 数据编码块 2-4 XOR 校验编码块 Q-2 =校验编码块 P-3。

由标准校验矩阵的第 11 列可知，校验编码块 P-3 仅参与了一个异或方程，因此由第 4 行起始的编码结束。

根据第 7 行进行编码：

step 3.1 数据编码块 1-3 XOR 数据编码块 2-4 =校验编码块 Q-1。

由标准校验矩阵的第 13 列可知，校验编码块 Q-1 仅参与了一个异或方程，因此由第 7 行起始的编码结束。

根据第 8 行进行编码：

step 4.1 数据编码块 1-2 XOR 数据编码块 2-3 =校验编码块 P-4。

由标准校验矩阵的第 12 列可知，校验编码块 P-4 参与了两个异或方程，分别对应标准校验矩阵的第 2 行和第 8 行，因此，接下来根据第 2 行进行编码：

step 4.2 数据编码块 1-1 XOR 校验编码块 P-4=校验编码块 Q-3。

由标准校验矩阵的第 15 列可知，校验编码块 Q-3 参与了两个异或方程，分别对应标准校验矩阵的第 2 行和第 5 行，因此，接下来根据第 5 行进行编码：

step 4.3 数据编码块 2-1 XOR 校验编码块 Q-3 =校验编码块 P-2。

由标准校验矩阵的第 10 列可知，校验编码块 P-2 仅参与了一个异或方程，因此由第 8 行起始的编码结束。

至此，4 个起始编码行的编码均结束了，校验编码块 P-1 至校验编码块 Q-4 已经全部被编码得出，因此存储控制器生成了数据 chunk 1 和数据 chunk 2 对应的 chunk group。

以下，介绍解码过程。

通过上述编码过程，存储控制器获取了待存储的数据 chunk 对应的 chunk group 并存入 K+2 个 SSD 后，如果这 K+2 个 SSD 中有 SSD 损坏了，则需要用到纠删码的解码方法恢复损坏的 SSD 上存储的数据 chunk 或校验 chunk。具体的，可以由该存储控制器检测出该损坏的 SSD，或者该存储控制器被通知这 K+2 个 SSD 中有 SSD 损坏。

如果 K+2 个 SSD 中仅有一个 SSD 损坏，也即 chunk group 中只有一个 chunk 需要恢复。对于该损坏的 chunk 的每个编码块的恢复过程如下，编码块 1 为损坏的 chunk 中的 R 个编码块中之任一：

则根据该校验矩阵，获取编码块 1 参与的异或方程，并根据编码块 1 参与的异或方程获取用于恢复编码块 1 的其余两个编码块；

将其余两个编码块做异或运算，得到编码块 1。

由于任一 chunk 的 R 个编码块中，有 R-2 个编码块参与了两个异或方程。每个参与了两个异或方程的编码块在恢复的过程中，可以使用其参与的两个异或方程中的任何一个。因此，实际的解码方法可以有 2^{R-2} 种。

这 2^{R-2} 种解码方法虽然都可以完成损坏的 chunk 的恢复，但由于编码块 1 的恢复过程中，需要将用于编码块 1 恢复的两个编码块从 SSD 中读到存储控制器中，再由存储控制器完成恢复过程。不同的解码方法可能导致恢复全部 R 个编码块的过程中，需要从 SSD 中读出的编码块的数量不同，因此对于一个确定的校验矩阵，对于任一 chunk 损坏的情况下，可以采用一种需要

从 SSD 中读出编码块的数量最少的解码方法。

仍以 K=2、R=4，且采用标准校验矩阵的情况下为例，如图 4-1。

如果数据 chunk 1 所在的 SSD 损坏，则需要恢复数据编码块 1-1、数据编码块 1-2、数据编码块 1-3 和数据编码块 1-4。

数据编码块 1-1 仅参与了校验矩阵第 2 行对应的异或方程：

数据编码块 1-1 =校验编码块 P-4 XOR 校验编码块 Q-3。

数据编码块 1-2 参与了校验矩阵第 3 行和校验矩阵第 8 行对应的异或方程：

数据编码块 1-2 =数据编码块 2-1 XOR 校验编码块 Q-4；

数据编码块 1-2 =数据编码块 2-3 XOR 校验编码块 P-4。

数据编码块 1-3 参与了校验矩阵第 4 行和校验矩阵第 7 行对应的异或方程：

数据编码块 1-3 =数据编码块 2-2 XOR 校验编码块 P-1；

数据编码块 1-3 =数据编码块 2-4 XOR 校验编码块 Q-1。

数据编码块 1-4 仅参与了校验矩阵第 6 行对应的异或方程：

数据编码块 1-4 =校验编码块 P-1 XOR 校验编码块 Q-2。

由于数据编码块 1-2 和数据编码块 1-3 均参与了两个异或方程，因此一共有 2^2=4 种解码方法。

解码方法 1，数据编码块 1-2 的恢复采用了校验矩阵第 3 行对应的异或方程，且数据编码块 1-3 的恢复采用了校验矩阵第 4 行对应的异或方程，则：

恢复开销=(校验编码块 P-4、校验编码块 Q-3、数据编码块 2-1、校验编码块 Q-4、数据编码块 2-2、校验编码块 P-1、校验编码块 Q-2)/8 个数据编码块=0.875。

解码方法 2：数据编码块 1-2 的恢复采用了校验矩阵第 8 行对应的异或方程，且数据编码块 1-3 的恢复采用了校验矩阵第 4 行对应的异或方程，则：

恢复开销=(校验编码块 P-4、校验编码块 Q-3、数据编码块 2-3、数据编码块 2-2、校验编码块 P-1、校验编码块 Q-2)/8 个数据编码块=0.75。

解码方法 3：数据编码块 1-2 的恢复采用了校验矩阵第 3 行对应的异或方程，且数据编码块 1-3 的恢复采用了校验矩阵第 7 行对应的异或方程，则：

恢复开销=(校验编码块 P-4、校验编码块 Q-3、数据编码块 2-1、校验编

码块 Q-4、数据编码块 2-4、校验编码块 Q-1、校验编码块 P-1、校验编码块 Q-2)/8 个数据编码块=1。

解码方法 4：数据编码块 1-2 的恢复采用了校验矩阵第 8 行对应的异或方程，且数据编码块 1-3 的恢复采用了校验矩阵第 7 行对应的异或方程，则：

恢复开销=(校验编码块 P-4、校验编码块 Q-3、数据编码块 2-3、校验编码块 Q-1、数据编码块 2-4、校验编码块 P-1、校验编码块 Q-2)/8 个数据编码块=0.875。

可以看出，解码方法 2 的恢复开销最小。因此，针对数据 chunk 2 损坏的情况下，存储控制器可以优选的采用解码方法 2 来完成数据 chunk 2 的恢复，以提升恢复效率。

针对每个 chunk 损坏的情况下，都有至少一种需要读出编码块的数量最少的解码方法。因此存储控制器内可以存储有多种恢复开销最小的解码方法，不同恢复开销最小的解码方法对应不同 K 和 R 取值下的校验矩阵。

以上介绍了 K+2 个 SSD 中仅有一个 SSD 损坏的场景，如果 K+2 个 SSD 中有两个 SSD 损坏，那么解码方法与前述编码方法类似，也即将损坏的 2 个 SSD 上存储的 chunk 视为校验 chunk，未损坏的 K 个 SSD 上存储的 chunk 的视为数据 chunk。根据所述校验矩阵和未损坏的 K 个 SSD 上存储的 chunk 进行解码，以获取损坏的 2 个 SSD 上存储的校验 chunk。

纠删码的编码方法和解码方法之间强耦合，例如如果编码的时候，数据编码块 1-1 =校验编码块 P-4 XOR 校验编码块 Q-3，那么如果校验编码块 P-4 损坏时，需要采用数据编码块 1-1 XOR 校验编码块 Q-3 恢复校验编码块 P-4，而如果校验编码块 Q-3 损坏时，需要采用数据编码块 1-1 XOR 校验编码块 P-4 恢复校验编码块 Q-3。因此本申请提供的编码方法有效降低了恢复开销，提升了后续恢复损失的 chunk 时的效率。

本申请实施例所应用的存储控制器

如图 6 提供了一种存储控制器 200，存储控制器 200 可以运用于图 1-1 或图 1-2 所示的存储系统中。存储控制器 200 包括总线 202、处理器 204、存储器 208 和通信接口 206。处理器 204、存储器 208 和通信接口 206 之间通过总线 202 通信。

其中，处理器 204 可以为中央处理器(英文：central processing unit，缩写：CPU)。存储器 208 可以包括易失性存储器(英文：volatile memory)，例如随机存取存储器(英文：random access memory，缩写：RAM)。存储器 208 还可以包括非易失性存储器(英文：non-volatile memory)，例如只读存储器(英文：read-only memory，缩写：ROM)，快闪存储器，HDD 或 SSD。

通信接口 206 包括网络接口和存储介质读写接口，分别用于获取客户端发来的待写入数据和将编码后获得的 chunk group 写入存储介质中。

如图 7，当存储控制器 200 在执行编码过程中，存储器 208 中存储有编码程序以及 K 个数据 chunk。

存储控制器 200 运行时，处理器 204 从存储器 208 中读取编码程序和 K 个数据 chunk，以执行前述编码过程生成 chunk group，并通过通信接口 206 将该 chunk group 中的各个 chunk 存入不同存储介质中。

如图 8，当存储控制器 200 在执行解码过程中，存储器 208 中存储有解码程序以及恢复过程中所需的编码块。

当存储控制器 200 所在的存储系统的存储介质损坏时，处理器 204 从存储器 208 中读取解码程序和恢复损坏的存储介质所需的编码块，以执行前述解码过程，恢复损坏的存储介质上存储的 chunk。

编码程序和解码程序可以合并为一个程序。

校验矩阵在存储器 208 中的存储方式有多种，可以直接以矩阵的形式存储，也可以以 2*R 个异或方程的形式存储。并且将这 2*R 个异或方程与编码程序和解码程序融合。

以矩阵的形式存储的情况下，在编码过程中，处理器 204 执行编码程序，访问校验矩阵确定了起始编码行后，每执行完校验矩阵的一行对应的异或运算，处理器 204 再次访问校验矩阵以执行校验矩阵另一行对应的异或运算，直至校验矩阵每一行对应的异或运算均执行完毕。解码过程类似与编码过程类似。

对于每一个校验矩阵，都可以有确定的编码过程和解码过程，因此存储器208中也可以不存储校验矩阵，而是直接在编码程序和解码程序中存储2*R个异或方程。例如图 5 对应的编码方法中，编码程序直接指示执行 step 1.1，step 2.1-2.3，step 3.1 和 step 4.1-4.3，无须存储校验矩阵并且在编码过程中逐

行访问校验矩阵以确定每一个 step 中需要对哪两个编码块作异或运算。类似的，在解码程序中也可以直接存储恢复开销最小的解码方法，例如 K=2、R=4，且采用标准校验矩阵的情况下，如果数据 chunk 1 损坏，为了恢复数据编码块 1-1 至数据编码块 1-4，解码程序直接指示执行以下异或运算：数据编码块 1-1 =校验编码块 P-4 XOR 校验编码块 Q-3；数据编码块 1-2 =数据编码块 2-3 XOR 校验编码块 P-4；数据编码块 1-3 =数据编码块 2-2 XOR 校验编码块 P-1；数据编码块 1-4 =校验编码块 P-1 XOR 校验编码块 Q-2。

以上提供的存储控制器降低了恢复开销，提升了后续恢复损失的 chunk 时的效率。

如图 9，提供了另一种存储控制器 400，存储控制器 400 可以运用于图 1-1 或图 1-2 所示的存储系统中。存储控制器 400 包括总线 402、处理器 404、存储器 408、数据处理芯片 410 和通信接口 406。处理器 404、存储器 408 和通信接口 406 之间通过总线 402 通信。

其中，处理器 404 可以为 CPU。存储器 408 可以包括易失性存储器。存储器 408 还可以包括非易失性存储器。

通信接口 406 包括网络接口和存储介质读写接口，分别用于获取客户端发来的待写入数据和将编码后获得的 chunk group 存入存储介质。

数据处理芯片 410 可以通过电路实现，所述电路可以 Wie 专用集成电路(英文：application-specific integrated circuit，缩写：ASIC)或可编程逻辑器件(英文：programmable logic device，缩写：PLD)。上述 PLD 可以是复杂可编程逻辑器件(英文：complex programmable logic device，缩写：CPLD)，现场可编程门阵列(英文：field programmable gate array，缩写：FPGA)，通用阵列逻辑(英文：generic array logic，缩写：GAL)或其任意组合。

如图 10 所示，数据处理芯片 410 具体可以包括选址单元 4102、运算单元 4104、存储单元 4106 和读写接口 4108。选址单元 4102、运算单元 4104、存储单元 4106 实际可以集成为一个电路。

读写接口 4108 与总线 402 相连，用于在数据处理芯片 410 执行编码的场景下，通过总线 402 获取存储器 408 中存储的数据编码块并存入存储单元 4106，并将编码后获取的校验编码块通过总线 402 发往存储器 208，以便存

储控制器 200 将 chunk group 存入存储介质。读写接口 4108 还用于在数据处理芯片 410 执行解码的场景下，通过总线 402 获取恢复过程中所需的编码块并存入存储单元 4106，并将恢复出的编码块通过总线 402 发往存储器 208。

选址单元4102的功能与校验矩阵类似，选址单元4102指示运算单元4104进行一次异或运算的过程中应当将存储单元 4106 中哪两个编码块进行异或运算，以便运算单元 4104 从存储单元 4106 中获取对应的编码块以完成异或运算。

运算单元 4104 从存储单元 4106 中获取一次异或运算的过程中需要进行异或运算的两个编码块，执行完一次异或运算后将得到的编码块存入存储单元 4106 中，接着执行下一次异或运算。

由于纠删码的编码方法和解码方法之间强耦合，因此本申请提供的存储控制器有效降低了恢复开销，提升了后续恢复损失的 chunk 时的效率。

在上述实施例中，对各个实施例的描述都各有侧重，某个实施例中没有详述的部分，可以参见其他实施例的相关描述。

结合本申请公开内容所描述的方法可以由处理器执行软件指令的方式来实现。软件指令可以由相应的软件模块组成，软件模块可以被存放于 RAM、快闪存储器、ROM、可擦除可编程只读存储器（英文：erasable programmable read only memory，缩写：EPROM）、电可擦可编程只读存储器（英文：electrically erasable programmable read only memory，缩写：EEPROM）、HDD、SSD、光盘或者本领域熟知的任何其它形式的存储介质中。

本领域技术人员应该可以意识到，在上述一个或多个示例中，本申请所描述的功能可以用硬件或软件来实现。当使用软件实现时，可以将这些功能存储在计算机可读介质中或者作为计算机可读介质上的一个或多个指令或代码进行传输。存储介质可以是通用或专用计算机能够存取的任何可用介质。

以上该的具体实施方式，对本申请的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上该仅为本申请的具体实施方式而已，并不用于限定本申请的保护范围，凡在本申请的技术方案的基础之上，所做的任何修改、改进等，均应包括在本申请的保护范围之内。

权 利 要 求 书

1、一种存储控制器，其特征在于，包括处理器、存储器和通信接口；

所述处理器，用于通过所述通信接口获取待编码的K个数据大块chunk，并将所述K个数据chunk缓存入所述存储器，每个数据chunk包括R个数据编码块，R+1为素数且R+1>K；

所述处理器，还用于执行所述存储器中的代码执行以下操作：

读取所述存储器中存储的所述K个数据chunk，根据校验矩阵和所述K个数据chunk生成第一校验chunk和第二校验chunk，每个校验chunk包括R个校验编码块；

其中，所述校验矩阵有2*R行，所述校验矩阵中第(k-1)*R+1列至第k*R列为所述K个数据chunk中第k个数据chunk的chunk列集合，K≥k≥1，所述校验矩阵中第K*R+1列至第(K+1)*R列为对应所述第一校验chunk的chunk列集合，所述校验矩阵中第(K+1)*R+1列至第(K+2)*R列为所述第二校验chunk的chunk列集合；

所述校验矩阵为标准校验矩阵H或由标准校验矩阵H执行N次调换操作后得到，N≥1，所述调换操作指将任意两个chunk列集合调换；所述标准校验矩阵H中除以下坐标为1外，其余坐标均为0，2*R≥i≥1，(K+2)*R≥j≥1，

如果i<j，则

$H[i+1][j*R+(R-j+i) \bmod R+1]$

$H[R+i+1][(j+1)*R-(R-j+i) \bmod R]$

如果i>j，则

$H[i+1][j*R+(R-1-j+i) \bmod R+1]$

$H[R+i+1][(j+1)*R-(R-1-j+i) \bmod R]$。

2、如权利要求 1 所述的存储控制器，其特征在于，所述校验矩阵中第(k-1)*R+1 列至第 k*R 列分别对应所述 K 个数据 chunk 中第 k 个数据 chunk 的 R 个数据编码块，所述校验矩阵中第 K*R+1 列至第(K+1)*R 列分别对应所述第一校验 chunk 的 R 个校验编码块，所述校验矩阵中第(K+1)*R+1 列至第(K+2)*R 列分别对应所述第二校验 chunk 的 R 个校验编码块；

所述校验矩阵的第 D 行有 3 个坐标为 1，所述第 D 行为所述校验矩阵的

任一行，对所述校验矩阵的第 D 行中为 1 的坐标对应的 3 个编码块中的任意 2 个编码块进行一次异或运算可以得到所述校验矩阵的第 D 行中为 1 的坐标对应的 3 个编码块中未参与本次异或运算的编码块。

3、如权利要求 1 或 2 所述的存储控制器，其特征在于，所述处理器还用于，通过所述通信接口将所述 K 个数据 chunk、所述第一校验 chunk 和所述第二校验 chunk 分别存入所述存储控制器所在的存储系统的 K+2 个存储介质中。

4、如权利要求 3 所述的存储控制器，其特征在于，所述处理器还用于，当所述存 K+2 个存储介质中有存储介质损坏时，根据所述校验矩阵和所述 K+2 存储介质中未损坏的存储介质上存储的数据 chunk 和所述第一校验 chunk 和所述第二校验 chunk 中的至少一个，恢复所述损坏的存储介质。

5、一种数据处理芯片，其特征在于，包括电路和读写接口；

所述电路用于，通过所述读写接口获取待编码的K个数据大块chunk，每个数据chunk包括R个数据编码块，R+1为素数且R+1>K；

所述电路还用于，根据校验矩阵和所述K数据chunk生成第一校验chunk和第二校验chunk，每个校验chunk包括R个校验编码块；

其中，所述校验矩阵有2*R行，所述校验矩阵中第(k-1)*R+1列至第k*R列为所述K个数据chunk中第k个数据chunk的chunk列集合，K≥k≥1，所述校验矩阵中第K*R+1列至第(K+1)*R列为应所述第一校验chunk的chunk列集合，所述校验矩阵中第(K+1)*R+1列至第(K+2)*R列为所述第二校验chunk的chunk列集合；

所述校验矩阵为标准校验矩阵H或由标准校验矩阵H执行N次调换操作后得到，N≥1，所述调换操作指示将任意两个chunk列集合调换；

所述标准校验矩阵H中除以下坐标为1外，其余坐标均为0，2*R≥i≥1，(K+2)*R≥j≥1，

如果i<j，则

$$H[i+1][j*R+(R-j+i)\bmod R+1]$$

$$H[R+i+1][(j+1)*R-(R-j+i)\bmod R]$$

如果 i>j，则

$H[i+1][j*R+(R-1-j+i)\bmod R+1]$

$H[R+i+1][(j+1)*R-(R-1-j+i)\bmod R]$。

6、如权利要求 5 所述的数据处理芯片，其特征在于，所述校验矩阵中第(k-1)*R+1 列至第 k*R 列分别对应所述 K 个数据 chunk 中第 k 个数据 chunk 的 R 个数据编码块，所述校验矩阵中第 K*R+1 列至第(K+1)*R 列分别对应所述第一校验 chunk 的 R 个校验编码块，所述校验矩阵中第(K+1)*R+1 列至第(K+2)*R 列分别对应所述第二校验 chunk 的 R 个校验编码块；

所述校验矩阵的第 D 行有 3 个坐标为 1，所述第 D 行为所述校验矩阵的任一行，所述电路对所述校验矩阵的第 D 行中为 1 的坐标对应的 3 个编码块中的任意2个编码块进行一次异或运算可以得到所述校验矩阵的第D行中为 1 的坐标对应的 3 个编码块中未参与本次异或运算的编码块。

7、如权利要求 5 或 6 所述的数据处理芯片，其特征在于，所述数据处理芯片运用于存储控制器中；

所述电路，还用于通过所述读写接口将所述 K 个数据 chunk、所述第一校验 chunk 和所述第二校验 chunk 存入所述存储控制器的存储器中，以便所述存储控制器将所述 K 个数据 chunk、所述第一校验 chunk 和所述第二校验 chunk 分别存入所述存储控制器所在的存储系统的 K+2 个存储介质中。

8、如权利要求 7 所述的数据处理芯片，其特征在于，所述电路还用于，当所述 K+2 个存储介质中有存储介质损坏时，根据所述校验矩阵和所述 K+2 存储介质中未损坏的存储介质上存储的数据 chunk 和所述第一校验 chunk 和所述第二校验 chunk 中的至少一个，恢复所述损坏的存储介质。

9、如权利要求 5 至 8 任一所述的数据处理芯片，其特征在于，所述数据处理芯片包括现场可编程门阵列 FPGA。

10、一种数据处理方法，其特征在于，所述方法适用于存储控制器；所述方法包括：

获取待编码的K个数据大块chunk并缓存所述K个数据chunk，每个数据chunk包括R个数据编码块，R+1为素数且R+1>K；

根据校验矩阵和所述K个数据chunk生成第一校验chunk和第二校验chunk，每个校验chunk包括R个校验编码块；

其中，所述校验矩阵有2*R行，所述校验矩阵中第(k-1)*R+1列至第k*R

列为所述K个数据chunk中第k个数据chunk的chunk列集合，K≥k≥1，所述校验矩阵中第K*R+1列至第(K+1)*R列为对应所述第一校验chunk的chunk列集合，所述校验矩阵中第(K+1)*R+1列至第(K+2)*R列为所述第二校验chunk的chunk列集合；

所述校验矩阵为标准校验矩阵H或由标准校验矩阵H执行N次调换操作后得到，N≥1，所述调换操作指将任意两个chunk列集合调换；所述标准校验矩阵H中除以下坐标为1外，其余坐标均为0，2*R≥i≥1，(K+2)*R≥j≥1，

如果i<j，则

$H[i+1][j*R+(R-j+i) \bmod R+1]$

$H[R+i+1][(j+1)*R-(R-j+i) \bmod R]$

如果i>j，则

$H[i+1][j*R+(R-1-j+i) \bmod R+1]$

$H[R+i+1][(j+1)*R-(R-1-j+i) \bmod R]$。

11、如权利要求 10 所述的数据处理方法，其特征在于，所述校验矩阵中第(k-1)*R+1 列至第 k*R 列分别对应所述 K 个数据 chunk 中第 k 个数据 chunk 的 R 个数据编码块，所述校验矩阵中第 K*R+1 列至第(K+1)*R 列分别对应所述第一校验 chunk 的 R 个校验编码块，所述校验矩阵中第(K+1)*R+1 列至第(K+2)*R 列分别对应所述第二校验 chunk 的 R 个校验编码块；

所述校验矩阵的第 D 行有 3 个坐标为 1，所述第 D 行为所述校验矩阵的任一行，对所述校验矩阵的第 D 行中为 1 的坐标对应的 3 个编码块中的任意 2 个编码块进行一次异或运算可以得到所述校验矩阵的第 D 行中为 1 的坐标对应的 3 个编码块中未参与本次异或运算的编码块。

12、如权利要求 10 或 11 所述的数据处理方法，其特征在于，还包括：

将所述 K 个数据 chunk、所述第一校验 chunk 和所述第二校验 chunk 分别存入所述存储控制器所在的存储系统的 K+2 个存储介质中。

13、如权利要求 12 所述的数据处理方法，其特征在于，还包括：

当所述 K+2 个存储介质中有存储介质损坏时，根据所述校验矩阵和所述 K+2 存储介质中未损坏的存储介质上存储的数据 chunk 和所述第一校验 chunk 和所述第二校验 chunk 中的至少一个，恢复所述损坏的存储介质。

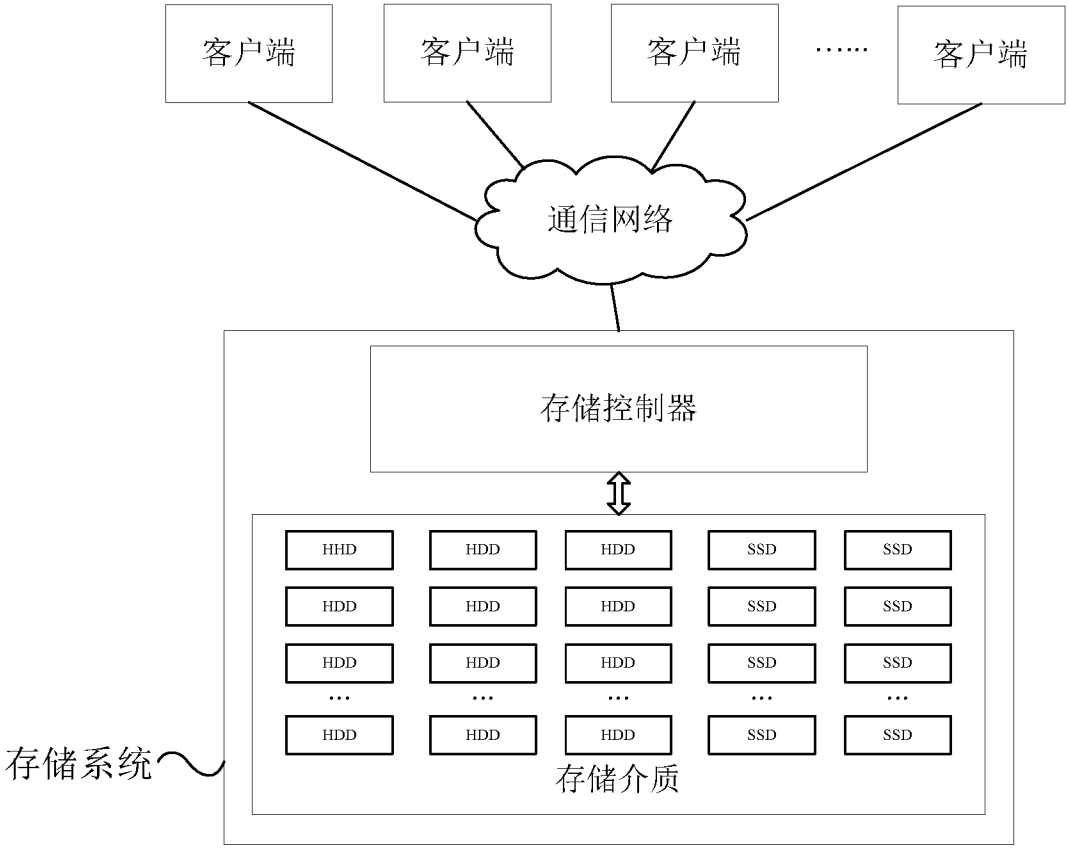
客户端
客户端
客户端
……
客户端
通信网络
存储控制器
HHD
HDD
SSD
存储介质
存储系统

图 1-1

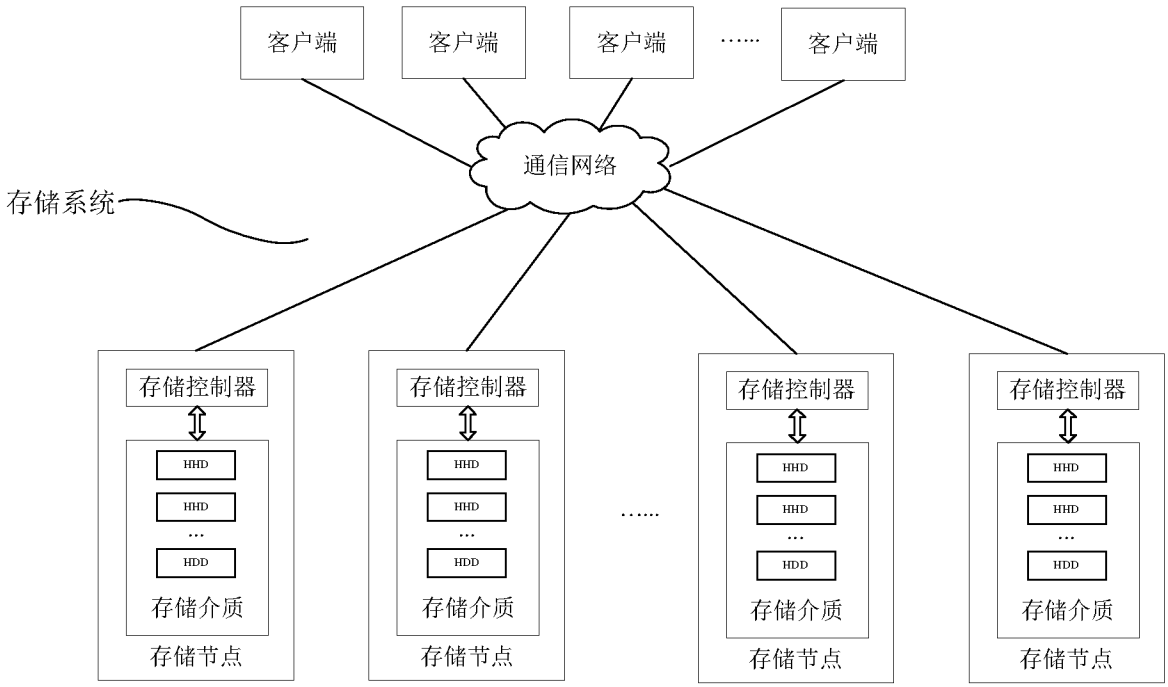
客户端
客户端
客户端
……
客户端
通信网络
存储系统
存储控制器
HHD
HDD
存储介质
存储节点
……

图 1-2

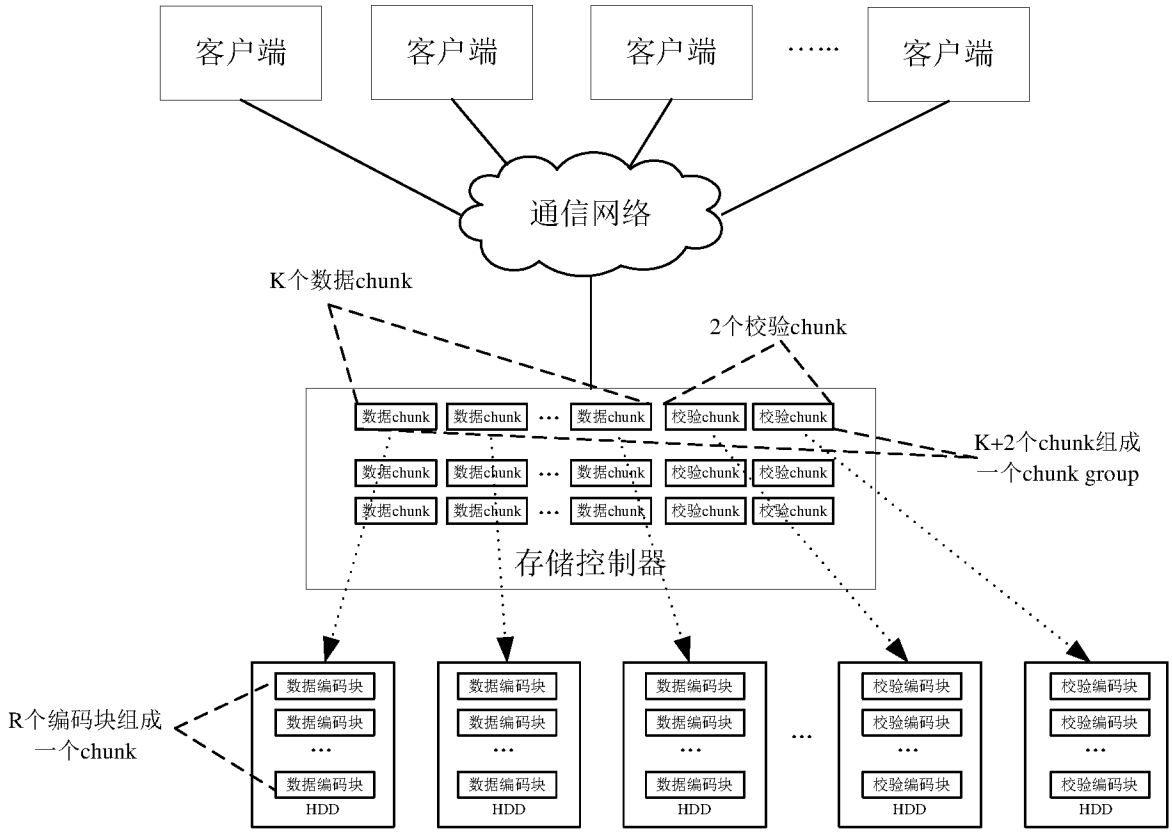
客户端
客户端
客户端
……
客户端
通信网络
K个数据chunk
2个校验chunk
数据chunk
校验chunk
K+2个chunk组成
一个chunk group
存储控制器
R个编码块组成
一个chunk
数据编码块
校验编码块
HDD

图 2

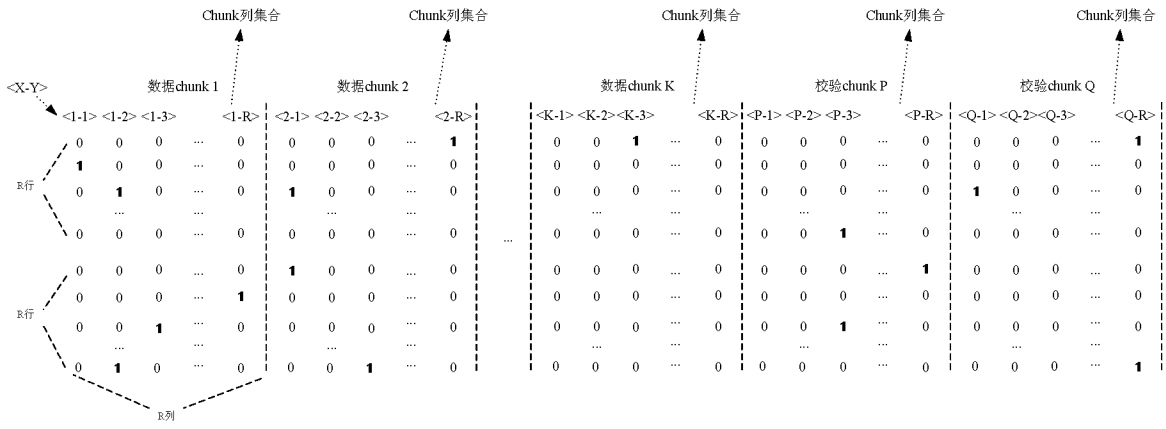
Chunk列集合
<X-Y>
数据chunk 1
数据chunk 2
数据chunk K
校验chunk P
校验chunk Q
R行
R列

图 3

数据chunk 1				数据chunk 2				校验chunk P				校验chunk Q			
<1-1>	<1-2>	<1-3>	<1-4>	<2-1>	<2-2>	<2-3>	<2-4>	<P-1>	<P-2>	<P-3>	<P-4>	<Q-1>	<Q-2>	<Q-3>	<Q-4>
0	0	0	0	0	0	0	**1**	0	0	**1**	0	0	**1**	0	0
1	0	0	0	0	0	0	0	0	0	0	**1**	0	0	**1**	0
0	**1**	0	0	**1**	0	0	0	0	0	0	0	0	0	0	**1**
0	0	**1**	0	0	**1**	0	0	**1**	0	0	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	**1**	0
0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	**1**	0	0
0	0	**1**	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0
0	**1**	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0

图 4-1

数据chunk 1				数据chunk 2				校验chunk P				校验chunk Q			
<1-1>	<1-2>	<1-3>	<1-4>	<2-1>	<2-2>	<2-3>	<2-4>	<P-1>	<P-2>	<P-3>	<P-4>	<Q-1>	<Q-2>	<Q-3>	<Q-4>
0	0	0	**1**	0	0	0	0	0	0	**1**	0	0	**1**	0	0
0	0	0	0	**1**	0	0	0	0	0	0	**1**	0	0	**1**	0
1	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**
0	**1**	0	0	0	0	**1**	0	**1**	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0
0	0	0	0	0	0	0	**1**	**1**	0	0	0	0	**1**	0	0
0	0	0	**1**	0	0	**1**	0	0	0	0	0	**1**	0	0	0
0	0	**1**	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0

图 4-2

	数据chunk 1				数据chunk 2				校验chunk P				校验chunk Q			
	<1-1>	<1-2>	<1-3>	<1-4>	<2-1>	<2-2>	<2-3>	<2-4>	<P-1>	<P-2>	<P-3>	<P-4>	<Q-1>	<Q-2>	<Q-3>	<Q-4>
step 2.3	0	0	0	0	0	0	0	**1**	0	0	**1**	0	0	**1**	0	0
step 4.2	**1**	0	0	0	0	0	0	0	0	0	0	**1**	0	0	**1**	0
step 1.1	0	**1**	0	0	**1**	0	0	0	0	0	0	0	0	0	0	**1**
step 2.1	0	0	**1**	0	0	**1**	0	0	**1**	0	0	0	0	0	0	0
step 4.3	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	**1**	0
step 2.2	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	**1**	0	0
step 3.1	0	0	**1**	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0
step 4.1	0	**1**	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0

图 5

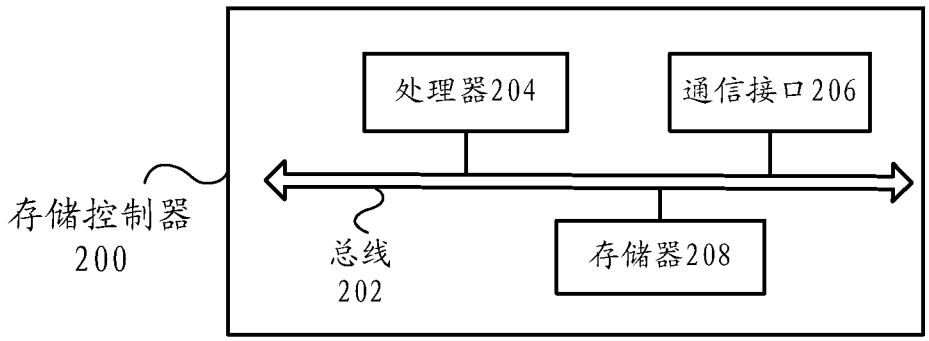

图 6

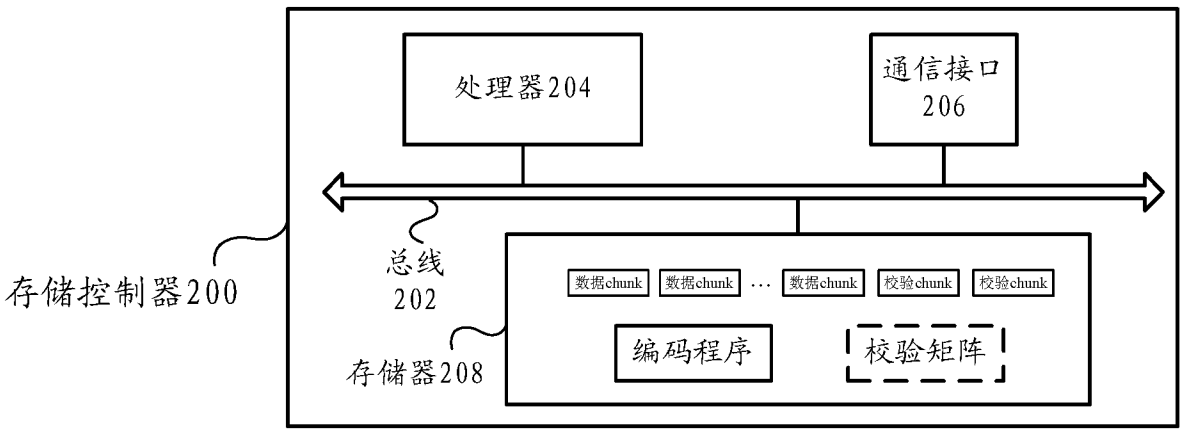

图 7

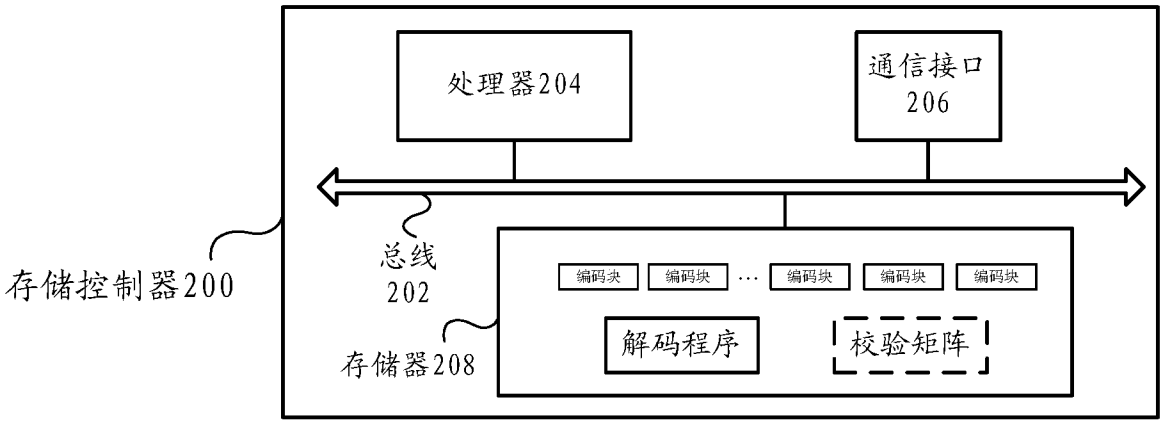
处理器204
通信接口206
总线202
存储控制器200
编码块
编码块
编码块
编码块
编码块
存储器208
解码程序
校验矩阵

图 8

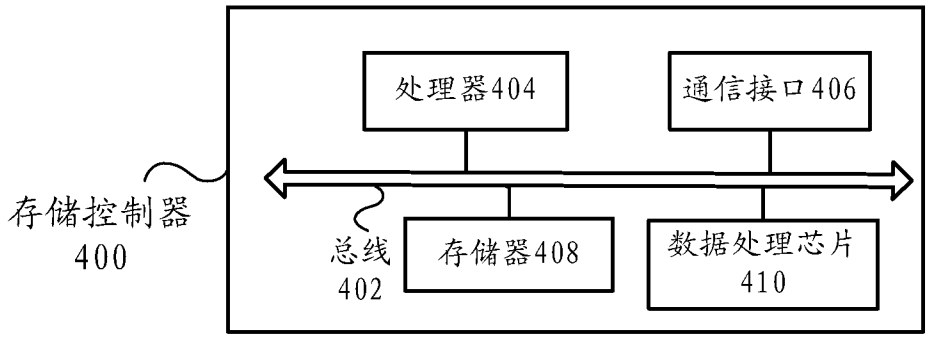
处理器404
通信接口406
存储控制器400
总线402
存储器408
数据处理芯片410

图 9

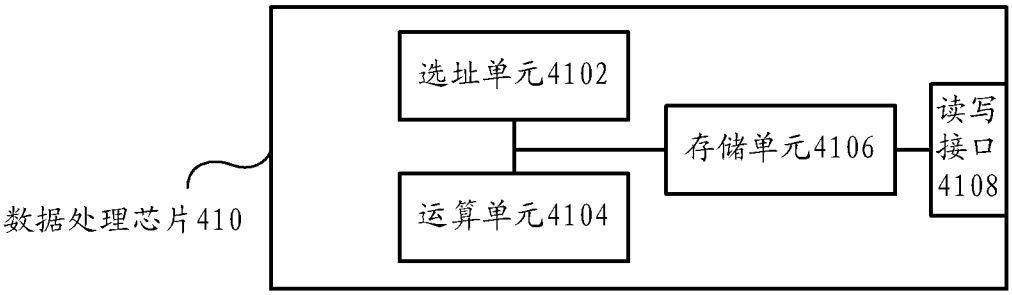
选址单元4102
存储单元4106
读写接口4108
运算单元4104
数据处理芯片410

图 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2016/111930

A. CLASSIFICATION OF SUBJECT MATTER

H03M 13/11 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H03M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, CNKI, WPI, EPODOC: chunk, processor, memory, control, data, cache, block, interface, coding, verify, checkout, matrix, array, exchange

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 104484126 A (HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY) 01 April 2015 (01.04.2015) description, paragraphs [0003]-[0024]	1-13
A	CN 102739346 A (MSTAR SOFTWARE R&D (SHENZHEN) LTD. et al.) 17 October 2012 (17.10.2012) the whole document	1-13
A	CN 104202057 A (ZTE CORPORATION) 10 December 2014 (10.12.2014) the whole document	1-13
A	US 2010251068 A1 (LIN, LI-LIEN et al.) 30 September 2010 (30.09.2010) the whole document	1-13

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&"document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
01 September 2017	28 September 2017
Name and mailing address of the ISA State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No. (86-10) 62019451	Authorized officer MENG, Zishan Telephone No. (86-10) 82246936

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2016/111930

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 104484126 A	01 April 2015	None	
CN 102739346 A	17 October 2012	None	
CN 104202057 A	10 December 2014	EP 3107214 A1	21 December 2016
		WO 2015120719 A1	20 August 2015
		US 2017033804 A1	02 February 2017
US 2010251068 A1	30 September 2010	TW 201035991 A	01 October 2010
		TW 201035988 A	01 October 2010
		CN 101950586 A	19 January 2011
		US 2010251076 A1	30 September 2010
		CN 101847447 A	29 September 2010

国际检索报告

国际申请号 PCT/CN2016/111930

A. 主题的分类

H03M 13/11(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

H03M

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用））

CNPAT,CNKI,WPI,EPODOC: 处理器，存储器，控制，缓存，数据，块，接口，编码，校验，矩阵，调换，列，chunk, processor, control, data, block, coding, verify, checkout, matrix

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
A	CN 104484126 A (华中科技大学) 2015年 4月 1日 (2015-04-01) 说明书第[0003]-[0024]段	1-13
A	CN 102739346 A (晨星软件研发深圳有限公司 等) 2012年 10月 17日 (2012-10-17) 全文	1-13
A	CN 104202057 A (中兴通讯股份有限公司) 2014年 12月 10日 (2014-12-10) 全文	1-13
A	US 2010251068 A1 (LIN, LI-LIEN 等) 2010年 9月 30日 (2010-09-30) 全文	1-13

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2017年 9月 1日	2017年 9月 28日
ISA/CN的名称和邮寄地址 中华人民共和国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451	受权官员 孟子山 电话号码 (86-10)82246936

表 PCT/ISA/210 (第2页) (2009年7月)

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2016/111930

检索报告引用的专利文件			公布日（年/月/日）	同族专利			公布日（年/月/日）
CN	104484126	A	2015年 4月 1日	无			
CN	102739346	A	2012年 10月 17日	无			
CN	104202057	A	2014年 12月 10日	EP	3107214	A1	2016年 12月 21日
				WO	2015120719	A1	2015年 8月 20日
				US	2017033804	A1	2017年 2月 2日
US	2010251068	A1	2010年 9月 30日	TW	201035991	A	2010年 10月 1日
				TW	201035988	A	2010年 10月 1日
				CN	101950586	A	2011年 1月 19日
				US	2010251076	A1	2010年 9月 30日
				CN	101847447	A	2010年 9月 29日