

(12) 发明专利申请

(10) 申请公布号 CN 102110028 A

(43) 申请公布日 2011. 06. 29

(21) 申请号 200910189473. X

(22) 申请日 2009. 12. 25

(71) 申请人 康佳集团股份有限公司

地址 518053 广东省深圳市南山区华侨城

(72) 发明人 王丛华

(74) 专利代理机构 深圳中一专利商标事务所

44237

代理人 贾振勇

(51) Int. Cl.

G06F 11/10 (2006. 01)

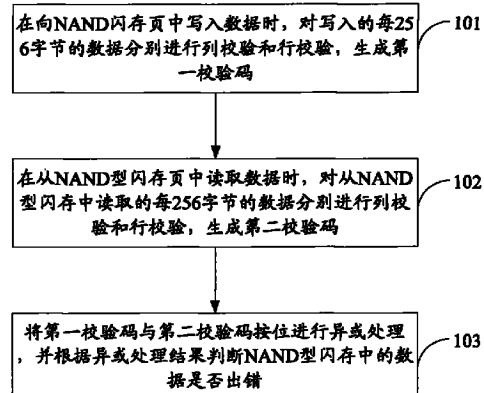
权利要求书 3 页 说明书 6 页 附图 2 页

(54) 发明名称

一种 NAND 闪存及其数据的校验方法和装置

(57) 摘要

本发明适用于存储领域，提供了一种 NAND 闪存及其数据的校验方法和装置，该方法包括：在向 NAND 闪存页中写入数据时，对写入的每 256 字节的数据分别进行列校验和行校验，生成第一校验码；在从 NAND 闪存页中读取数据时，对从 NAND 闪存中读取的每 256 字节的数据分别进行列校验和行校验，生成第二校验码；将第一校验码与第二校验码按位进行异或处理，并根据异或处理结果判断 NAND 闪存中的数据是否出错。本发明实施例由于通过软件形式实现 NAND 闪存中数据的校验方法，因此，不仅可以校验 NAND 闪存中的数据，而且可适用于任何平台。



1. 一种 NAND 闪存中数据的校验方法,其特征在于,所述方法包括下述步骤:

在向 NAND 闪存页中写入数据时,对写入的每 256 字节的数据分别进行列校验和行校验,生成第一校验码,所述第一校验码包括第一列校验码和第一行校验码;

在从 NAND 闪存页中读取数据时,对从 NAND 闪存中读取的每 256 字节的数据分别进行列校验和行校验,生成第二校验码,所述第二校验码包括第二列校验码和第二行校验码;

将第一校验码与第二校验码按位进行异或处理,并根据异或处理结果判断 NAND 闪存中的数据是否出错。

2. 如权利要求 1 所述的方法,其特征在于,对写入的每 256 字节的数据进行列校验的步骤具体为:

将写入的每 256 字节的数据排列成 256 行和 8 列的矩阵,矩阵中每个元素代表 256 字节中的一个比特位;

采用预设的列校验规则对矩阵中的比特位进行列校验,生成对应的列极性码;

将生成的所有列极性码组合成第一列校验码。

3. 如权利要求 2 所述的方法,其特征在于,所述预设的列校验规则为:

对矩阵中异或处理 1 列的比特位再跳过 1 列的比特位,循环进行,直到矩阵中所有列均处理完毕,生成对应的列极性码;

对矩阵中跳过 1 列的比特位后再异或处理 1 列的比特位,循环进行,直到矩阵中所有列均处理完毕,生成对应的列极性码;

对矩阵中异或处理 2 列的比特位再跳过 2 列的比特位,循环进行,直到矩阵中所有列均处理完毕,生成对应的列极性码;

对矩阵中跳过 2 列的比特位再异或处理 2 列的比特位,循环进行,直到矩阵中所有列均处理完毕,生成对应的列极性码;

对矩阵中异或处理 4 列的比特位再跳过 4 列的比特位,循环进行,直到矩阵中所有列均处理完毕,生成对应的列极性码;

对矩阵中跳过 4 列的比特位再异或处理 4 列的比特位,循环进行,直到矩阵中所有列均处理完毕,生成对应的列极性码。

4. 如权利要求 1 所述的方法,其特征在于,对写入的每 256 字节的数据进行行校验的步骤具体为:

将写入的每 256 字节的数据排列成 256 行和 8 列的矩阵,矩阵中每个元素代表 256 字节中的一个比特位;

采用预设的行校验规则对矩阵中的比特位进行行校验,生成对应的行极性码;

将生成的所有行极性码组合成第一行校验码。

5. 如权利要求 4 所述的方法,其特征在于,所述预设的行校验规则为:

对矩阵中异或处理 1 行的比特位再跳过 1 行的比特位,循环进行,直到矩阵中所有行均处理完毕,生成对应的行极性码;

对矩阵中跳过 1 行的比特位后再异或处理 1 行的比特位,循环进行,直到矩阵中所有行均处理完毕,生成对应的行极性码;

对矩阵中异或处理 2 行的比特位再跳过 2 行的比特位,循环进行,直到矩阵中所有行均处理完毕,生成对应的行极性码;

对矩阵中跳过 2 行的比特位再异或或处理 2 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的列极性码；

对矩阵中异或或处理 4 行的比特位再跳过 4 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

对矩阵中跳过 4 行的比特位再异或或处理 4 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

对矩阵中异或或处理 8 行的比特位再跳过 8 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

对矩阵中跳过 8 行的比特位后再异或或处理 8 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

对矩阵中异或或处理 16 行的比特位再跳过 16 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

对矩阵中跳过 16 行的比特位再异或或处理 16 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的列极性码；

对矩阵中异或或处理 32 行的比特位再跳过 32 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

对矩阵中跳过 32 行的比特位再异或或处理 32 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

对矩阵中异或或处理 64 行的比特位再跳过 64 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

对矩阵中跳过 64 行的比特位再异或或处理 64 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的列极性码；

对矩阵中异或或处理 128 行的比特位再跳过 128 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

对矩阵中跳过 128 行的比特位再异或或处理 128 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码。

6. 如权利要求 1-5 任一项所述的方法，其特征在于，所述将第一校验码与第二校验码按位进行异或或处理，并根据异或或处理结果判断 NAND 闪存中的数据是否出错的步骤具体为：

当第一校验码与第二校验码按位进行异或或处理后，异或处理结果为 0，判定 NAND 闪存不存在错误，或者出现了无法检测的错误；

当第一校验码与第二校验码按位进行异或或处理后，异或处理结果中存在 11 个比特位为 1，判定存在一个比特错误，且可纠正；

当第一校验码与第二校验码按位进行异或或处理后，异或处理结果中只存在 1 个比特位为 1，判定 OOB 区出错；

当第一校验码与第二校验码按位进行异或或处理后，异或处理结果为其他情况均判定出现了无法纠正的错误。

7. 一种 NAND 闪存中数据的校验装置，其特征在于，所述装置包括：

第一校验码生成单元，用于在向 NAND 闪存页中写入数据时，对写入的每 256 字节的数据分别进行列校验和行校验，生成第一校验码，所述第一校验码包括第一列校验码和第一

行校验码；

第二校验码生成单元，用于在从NAND闪存页中读取数据时，对从NAND闪存中读取的每256字节的数据分别进行列校验和行校验，生成第二校验码，所述第二校验码包括第二列校验码和第二行校验码；

数据校验单元，用于将第一校验码与第二校验码按位进行异或处理，并根据异或处理结果判断NAND闪存中的数据是否出错。

8. 如权利要求7所述的装置，其特征在于，所述第一校验码生成单元包括：

第一列校验模块，用于将写入的每256字节的数据排列成256行和8列的矩阵，并采用预设的列校验规则对矩阵中的比特位进行列校验，生成对应的列极性码，将生成的所有列极性码组合成第一列校验码；

第一行校验模块，用于将写入的每256字节的数据排列成256行和8列的矩阵，并采用预设的行校验规则对矩阵中的比特位进行行校验，生成对应的行极性码，将生成的所有行极性码组合成第一行校验码。

9. 一种NAND闪存，其特征在于，所述闪存包括所述权利要求7或8所述的NAND闪存中数据的校验装置。

## 一种 NAND 闪存及其数据的校验方法和装置

### 技术领域

[0001] 本发明属于存储器领域,尤其涉及一种 NAND 闪存及其数据的校验方法和装置。

### 背景技术

[0002] 由于 NAND 闪存的工艺不能保证 NAND 闪存内的存储排列 (MemoryArray) 在其生命周期中保持性能的可靠性,因此,在 NAN 型闪存的生产和使用过程中会产生坏块。当操作时序和电路稳定性较差时,如果 NAND 闪存存在坏块,则可能将造成整个闪存块或者闪存页不能读取或者全部出错,则 NAND 如果操作时序和电路稳定性较好,一般存在坏块的闪存页(如 512 字节)中只有一个或者几个位的数据将出错。在 NAND 闪存处理中,一般采用错误检查和纠正算法 (Error Checking and Correction, ECC) 来检查甚至纠正 NAND 闪存中存在的错误。

[0003] 现有技术提供了 ECC 的硬件实现电路,该 ECC 的硬件实现电路可以纠正 NAND 闪存中的单比特错误和检测双比特错误,而且计算速度快,但对 1 比特以上的错误无法纠正,对 2 比特以上的错误不保证能检测。由于现有技术是通过硬件电路实现 NAND 闪存的 ECC 的,其在制作和生产烧片文件时,只能在开发板上完成,因此,效率低,且由于无法移植到其他平台上完成,从而使得 ECC 校验的使用受到限制。

### 发明内容

[0004] 本发明实施例的目的在于提供一种 NAND 闪存中数据的校验方法,旨在解决现有的硬件实现校验方法无法移植到其他平台的问题。

[0005] 本发明实施例是这样实现的,一种 NAND 闪存中数据的校验方法,所述方法包括下述步骤:

[0006] 在向 NAND 闪存页中写入数据时,对写入的每 256 字节的数据分别进行列校验和行校验,生成第一校验码,所述第一校验码包括第一列校验码和第一行校验码;

[0007] 在从 NAND 闪存页中读取数据时,对从 NAND 闪存中读取的每 256 字节的数据分别进行列校验和行校验,生成第二校验码,所述第二校验码包括第二列校验码和第二行校验码;

[0008] 将第一校验码与第二校验码按位进行异或处理,并根据异或处理结果判断 NAND 闪存中的数据是否出错。

[0009] 本发明实施例的另一目的在于提供一种 NAND 闪存中数据的校验装置,所述装置包括:

[0010] 第一校验码生成单元,用于在向 NAND 闪存页中写入数据时,对写入的每 256 字节的数据分别进行列校验和行校验,生成第一校验码,所述第一校验码包括第一列校验码和第一行校验码;

[0011] 第二校验码生成单元,用于在从 NAND 闪存页中读取数据时,对从 NAND 闪存中读取的每 256 字节的数据分别进行列校验和行校验,生成第二校验码,所述第二校验码包括第

二列校验码和第二行校验码；

[0012] 数据校验单元，用于将第一校验码与第二校验码按位进行异或处理，并根据异或处理结果判断 NAND 闪存中的数据是否出错。

[0013] 本发明实施例的另一目的在于提供一种 NAND 闪存，所述闪存包括所述 NAND 闪存中数据的校验装置。

[0014] 在本发明实施例中，通过根据上述方法流程执行的软件来实现 NAND 闪存中的数据进行校验，在制作 NAND 闪存的映像文件时，通过为数据增加该根据上述方法流程执行的数据校验软件，即可实现对 NAND 闪存的数据的校验，该校验方法由于通过软件实现，从而与平台无关，使其可适用于任何平台。

## 附图说明

[0015] 图 1 是本发明实施例提供的 NAND 闪存中数据的校验方法的实现流程图；

[0016] 图 2 是本发明实施例提供的列校验规则的示例图；

[0017] 图 3 是本发明实施例提供的行校验规则的示例图；

[0018] 图 4 是本发明实施例提供的 NAND 闪存中数据的校验装置的结构框图。

## 具体实施方式

[0019] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0020] 在本发明实施例中，在向 NAND 闪存页中写入数据时，根据写入的每 256 字节的数据生成第一校验码，该第一校验码包括第一行校验码与第二列校验码，在从 NAND 闪存页中读取数据时，根据从 NAND 闪存中读取的每 256 字节的数据生成第二校验码，该第二校验码包括第二行校验码与第二列校验码，将第一校验码与第二校验码按位进行异或处理，并根据异或处理结果判断 NAND 闪存中的数据是否出错。

[0021] 图 1 示出了本发明实施例提供的 NAND 闪存中数据的校验方法的实现流程，详述如下：

[0022] 在步骤 S101 中，在向 NAND 闪存页中写入数据时，对写入的每 256 字节的数据分别进行列校验和行校验，生成第一校验码，该第一校验码包括第一列校验码和第一行校验码。

[0023] 其中对写入的每 256 字节的数据进行列校验的步骤具体如下：

[0024] 请参阅图 2，为本发明实施例提供的列校验规则的示例图，详述如下：

[0025] A、将写入的每 256 字节的数据排列成 256 行和 8 列的矩阵，矩阵中每个元素代表 256 字节中的一个比特位。

[0026] B、采用预设的列校验规则对矩阵中的比特位进行列校验，生成第一列校验码，其中预设的列校验规则具体为：

[0027] B1、对上述矩阵中异或处理 1 列的比特位再跳过 1 列的比特位，循环进行，直到矩阵中所有列均处理完毕，生成对应的列极性码；如可采用下述列极性生成公式表示：

[0028]  $CP0 = Bit0 \wedge Bit2 \wedge Bit4 \wedge Bit6$ ；其中 CP0 为生成的列极性码。Bit0、Bit2、Bit4、Bit6 分别为上述矩阵中的第 0 列、第 2 列、第 4 列、第 6 列中的比特位。符号“ $\wedge$ ”表示进行异或

处理。因此公式  $CP0 = Bit0 \wedge Bit2 \wedge Bit4 \wedge Bit6$  是指将该矩阵中第 0 列、第 2 列、第 4 列和第 6 列中的所有比特位进行异或处理，得到对应的列极性码。即  $CP0$  实际是  $256*4 = 1024$  个比特位进行异或处理后生成的列极性码。

[0029] B2、对上述矩阵中跳过 1 列的比特位后再异或处理 1 列的比特位，循环进行，直到矩阵中所有列均处理完毕，生成对应的列极性码；如可采用下述列极性生成公式表示：

[0030]  $CP1 = Bit1 \wedge Bit3 \wedge Bit5 \wedge Bit7$ ；其中  $CP1$  为生成的列极性码。 $Bit1, Bit3, Bit5, Bit7$  分别为上述矩阵中的第 1 列、第 3 列、第 5 列、第 7 列中的比特位。符号“ $\wedge$ ”表示进行异或处理。因此公式  $CP1 = Bit1 \wedge Bit3 \wedge Bit5 \wedge Bit7$  是指将该矩阵中第 1 列、第 3 列、第 5 列和第 7 列中的所有比特位进行异或处理，得到对应的列极性码。即  $CP0$  实际是  $256*4 = 1024$  个比特位进行异或处理后生成的列极性码。

[0031] B3、对上述矩阵中异或处理 2 列的比特位再跳过 2 列的比特位，循环进行，直到矩阵中所有列均处理完毕，生成对应的列极性码；如可采用下述列极性生成公式表示：

[0032]  $CP2 = Bit0 \wedge Bit1 \wedge Bit4 \wedge Bit5$ ；其中  $CP2$  为生成的列极性码。

[0033] B4、对上述矩阵中跳过 2 列的比特位再异或处理 2 列的比特位，循环进行，直到矩阵中所有列均处理完毕，生成对应的列极性码；如可采用下述列极性生成公式表示：

[0034]  $CP3 = Bit2 \wedge Bit3 \wedge Bit6 \wedge Bit7$ ；其中  $CP3$  为生成的列极性码。

[0035] B5、对上述矩阵中异或处理 4 列的比特位再跳过 4 列的比特位，循环进行，直到矩阵中所有列均处理完毕，生成对应的列极性码；如可采用下述列极性生成公式表示：

[0036]  $CP4 = Bit0 \wedge Bit1 \wedge Bit2 \wedge Bit3$ ；其中  $CP4$  为生成的列极性码。

[0037] B6、对上述矩阵中跳过 4 列的比特位再异或处理 4 列的比特位，循环进行，直到矩阵中所有列均处理完毕，生成对应的列极性码；如可采用下述列极性生成公式表示：

[0038]  $CP5 = Bit4 \wedge Bit5 \wedge Bit6 \wedge Bit7$ ；其中  $CP5$  为生成的列极性码。

[0039] C、将得到的所有列极性码组合成第一列校验码。其中组合的方式就是将得到每个比特的列极性码直接组合成一个包含多个比特的第一列校验码。如通过步骤 B 即可得到 6 个比特的列极性码，将得到的 6 个比特的列极性码直接组合，得到一个包含 6 比特的第一列校验码。

[0040] 其中对写入的每 256 字节的数据进行行校验的步骤具体如下：

[0041] 请参阅图 3，为本发明实施例提供的列校验规则的示例图，详述如下：

[0042] D、将写入的每 256 字节的数据排列成 256 行和 8 列的矩阵，矩阵中每个元素代表 256 字节中的一个比特位。

[0043] E、采用预设的行校验规则对矩阵中的比特位进行行校验，生成对应的行极性码，在本发明实施例中，采用 RP0-RP15 表示对应的行极性码。其中预设的行校验规则具体如下：

[0044] E1、对上述矩阵中异或处理 1 行的比特位再跳过 1 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

[0045] E2、对上述矩阵中跳过 1 行的比特位后再异或处理 1 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

[0046] E3、对上述矩阵中异或处理 2 行的比特位再跳过 2 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；

- [0047] E4、对上述矩阵中跳过 2 行的比特位再异或处理 2 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的列极性码；
- [0048] E5、对上述矩阵中异或处理 4 行的比特位再跳过 4 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；
- [0049] E6、对上述矩阵中跳过 4 行的比特位再异或处理 4 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；
- [0050] E7、对上述矩阵中异或处理 8 行的比特位再跳过 8 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；
- [0051] E8、对上述矩阵中跳过 8 行的比特位后再异或处理 8 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；
- [0052] E9、对上述矩阵中异或处理 16 行的比特位再跳过 16 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；
- [0053] E10、对上述矩阵中跳过 16 行的比特位再异或处理 16 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的列极性码；
- [0054] E11、对上述矩阵中异或处理 32 行的比特位再跳过 32 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；
- [0055] E12、对上述矩阵中跳过 32 行的比特位再异或处理 32 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；
- [0056] E13、对上述矩阵中异或处理 64 行的比特位再跳过 64 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；
- [0057] E14、对上述矩阵中跳过 64 行的比特位再异或处理 64 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的列极性码；
- [0058] E15、对上述矩阵中异或处理 128 行的比特位再跳过 128 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码；
- [0059] E16、对上述矩阵中跳过 128 行的比特位再异或处理 128 行的比特位，循环进行，直到矩阵中所有行均处理完毕，生成对应的行极性码。
- [0060] F、将得到的所有行极性码组合成第一行校验码。其中组合的方式就是将得到每个比特的行极性码直接组合成一个包含多个比特的第一行校验码。如通过步骤 E 即可得到 16 个比特的列极性码，将得到的 16 个比特的列极性码直接组合，得到一个包含 16 比特的第一行校验码。
- [0061] 综上所述，当对写入至 NAND 闪存页的数据进行行校验码列校验后，即可生成 6 个比特的第一列校验码和 16 个比特的第一行校验码，将生成的列校验码和行结果直接组合，即可得到第一校验码。为了保存将生成的第一校验码，每个 NAND 闪存页中一般包含 512 字节的数据区和 16 字节的带外数据区 (out-of-band, OOB)，在本发明实施例中，将生成的第一校验码存储至 NAND 闪存页的 OOB 区。由于 NAND 闪存页中一般包含 512 字节的数据区，因此，根据上述方法，即可根据写入至 NAND 闪存页的前 256 字节的数据生成 3 字节的第一校验码，写入至 NAND 闪存页的后 256 字节的数据生成 3 字节的第一校验码，因此，根据写入至 NAND 闪存页的所有数据即可得到两个 3 字节的第一校验码，将该两个 3 字节的第一校验码存储至 NAND 闪存页的 OOB 区。如可以将 NAND 闪存页的 OOB 区的第 0、1、2 和第 3、6、7 字节。

节用于存储第一校验码。

[0062] 在步骤 S102 中,在从 NAND 闪存页中读取数据时,对从 NAND 闪存中读取的每 256 字节的数据分别进行列校验和行校验,生成第二校验码,该第二校验码包括第二列校验码和第二行校验码。

[0063] 其中对从 NAND 闪存中读取的每 256 字节的数据分别进行列校验码行校验的具体过程与对写入的每 256 字节的数据分别进行列校验码行校验相同,在此不再赘述。

[0064] 在步骤 S103 中,将第一校验码与第二校验码按位进行异或处理,并根据异或处理结果判断 NAND 闪存中的数据是否出错。其具体步骤如下:

[0065] 当第一校验码与第二校验码按位进行异或处理后,异或处理结果为 0,则判定 NAND 闪存不存在错误,或者出现了无法检测的错误;当第一校验码与第二校验码按位进行异或处理后,异或处理结果中存在 11 个比特位为 1,判定存在一个比特错误,且可纠正;当第一校验码与第二校验码按位进行异或处理后,异或处理结果中只存在 1 个比特位为 1,判定 OOB 区出错;其他情况均判定出现了无法纠正的错误。

[0066] 在本发明实施例中,通过根据上述方法流程执行的软件来实现 NAND 闪存中的数据进行校验,在制作 NAND 闪存的映像文件时,通过为数据增加该根据上述方法流程执行的数据校验软件,即可实现对 NAND 闪存的数据的校验,该校验方法由于通过软件实现,从而与平台无关,使其可适用于任何平台。

[0067] 图 4 示出了本发明实施例提供的 NAND 闪存中数据的校验装置的结构,为了便于说明,仅示出了与本发明实施例相关的部分。该装置可以是内置于 NAND 闪存中的软件单元、硬件单元或者软硬件相结合的单元,或者作为独立的挂件集成到 NAND 闪存中。其中:

[0068] 第一校验码生成单元 1 在向 NAND 闪存页中写入数据时,对写入的每 256 字节的数据分别进行列校验码行校验,生成第一校验码,该第一校验码包括第一列校验码和第一行校验码。该第一校验码生成单元 1 包括第一列校验模块 11 和第一行校验模块 12。其中:

[0069] 第一列校验模块 11 将写入的每 256 字节的数据排列成 256 行和 8 列的矩阵,并采用预设的列校验规则对矩阵中的比特位进行列校验,生成对应的列极性码,将生成的列极性码组合成第一列校验码。其中矩阵中每个元素代表 256 字节中的一个比特位。预设的列校验规则如上所述,在此不再赘述。

[0070] 第一行校验模块 12 将写入的每 256 字节的数据排列成 256 行和 8 列的矩阵,采用预设的行校验规则对矩阵中的比特位进行行校验,生成对应的行极性码,将生成的所有行极性码组合成第一行校验码。其中矩阵中每个元素代表 256 字节中的一个比特位。预设的行校验规则如上所述,在此不再赘述。

[0071] 第二校验码生成单元 2 在从 NAND 闪存页中读取数据时,对从 NAND 闪存中读取的每 256 字节的数据分别进行列校验码行校验,生成第二校验码,该第二校验码包括第二列校验码和第二行校验码。该第二校验码生成单元 2 包括第二列校验模块 21 和第二行校验模块 22。其中:

[0072] 第二列校验模块 21 将读取的每 256 字节的数据排列成 256 行和 8 列的矩阵,并采用预设的列校验规则对矩阵中的比特位进行列校验,生成第二列校验码。其中矩阵中每个元素代表 256 字节中的一个比特位。预设的列校验规则如上所述,在此不再赘述。

[0073] 第二行校验模块 22 将读取的每 256 字节的数据排列成 256 行和 8 列的矩阵,采用

预设的行校验规则对矩阵中的比特位进行行校验，生成第二行校验码。其中矩阵中每个元素代表 256 字节中的一个比特位。预设的行校验规则如上所述，在此不再赘述。

[0074] 数据校验单元 3 将第一校验码生成单元生成的第一校验码与第二校验码生成的第二校验码按位进行异或处理，并根据异或处理结果判断 NAND 闪存中的数据是否出错。其具体步骤如上所述，在此不再赘述。

[0075] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，是可以通过计算机程序来指令相关的硬件来完成，所述的程序可存储于一计算机可读取存储介质中，该程序在执行时，可包括如上述各方法的实施例的流程。其中，所述的存储介质可为磁碟、光盘、只读存储记忆体 (Read-Only Memory, ROM) 或随机存储记忆体 (Random Access Memory, RAM) 等。

[0076] 在本发明实施例中，通过根据上述方法流程执行的软件来实现 NAND 闪存中的数据进行校验，在制作 NAND 闪存的映像文件时，通过为数据增加该根据上述方法流程执行的数据校验软件，即可实现对 NAND 闪存的数据的校验，该校验方法由于通过软件实现，从而与平台无关，使其可适用于任何平台。

[0077] 以上所述仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等，均应包含在本发明的保护范围之内。

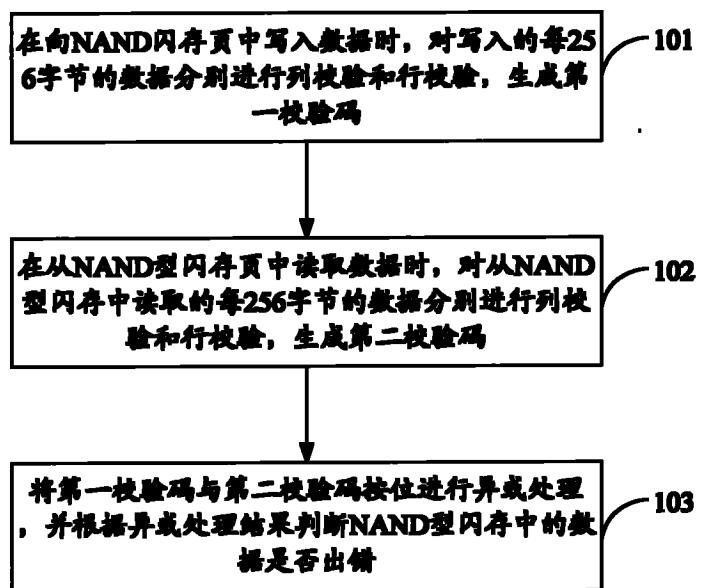


图 1

Byte0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
...	...	...	...	...	...	...	...	...
Byte252	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte253	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte254	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	CP1	CP0	CP1	CP0	CP1	CP0	CP1	CP0
	CP3	CP2		CP3	CP2			
	CP5				CP4	CP2		

图 2

Byte0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	RP0	RP2	RP4	RP14		
Byte1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	RP1					
Byte2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	RP0	RP3				
Byte3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	RP1					
...	...	...	...	...	...	...	...	...	...	...	...	...		
Byte252	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	RP0	RP2	RP5	RP15		
Byte253	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	RP1					
Byte254	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	RP0	RP3				
Byte255	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	RP1					

图 3

