

# 一种 NAND Flash 坏块管理优化方法的实现

乔立岩<sup>1</sup> 李元亨<sup>1</sup> 王 戈<sup>2</sup> 徐红伟<sup>1</sup>

(1. 哈尔滨工业大学 哈尔滨 150080; 2. 第二炮兵驻 699 厂军代室 北京 100039)

**摘要:** 由于高速数据采集系统中数据传输的高速性,系统后端需要保证高速的数据存储速度。在基于 NAND Flash 的存储系统中,提出了一种以页为单元跳过和替换坏块的 NAND Flash 坏块管理方法。与传统的以块为单位处理坏块的方法不同,此方法保证了不会因为坏块的出现而大幅降低写入速度,提高了 Flash 存储资源的利用率,并且降低了缓存资源的占用。此设计应用在一个基于 FPGA 的高速数据采集系统中,测试表明该方法能够正确处理 NAND Flash 的相关坏块问题,在坏块出现时使存储的瞬时速度几乎保持不变,并且具有较高的稳定性。

**关键词:** NAND Flash; 坏块管理方法; 高速数据传输

**中图分类号:** TP274.2    **文献标识码:** A    **国家标准学科分类代码:** 510.4030

## Realization of an optimizing method of bad block management in NAND flash

Qiao Liyan<sup>1</sup> Li Yuanheng<sup>1</sup> Wang Ge<sup>2</sup> Xu Hongwei<sup>1</sup>

(1. Harbin Institute of Technology, Harbin 150080, China;

2. The Military Representative office in 699 factory, the Second Artillery Corps, Beijing 100039, China)

**Abstract:** Due to the high speed of data transmission in data acquisition system, the system need ensure the high speed data storage speed. This paper introduces a new bad block management of NAND Flash in a NAND Flash storage system, which is based on judging and jumping bad blocks by the unit of page. Compared with the traditional management by the unit of block, this design guarantees the speed of programming won't decrease if bad blocks appear, improves the utilization rate of Flash storage resource, and decreases occupying the resource of cache. This design used in a high speed data acquisition system based on FPGA, test result shows that the bad block management makes sure the speed almost remain the same when bad blocks occur, and handles the NAND Flash bad block issues correctly with highly stability.

**Keywords:** NAND Flash; bad block management; high speed data transfer

## 1 引言

数据采集和存储系统<sup>[1]</sup>是现代测控系统中不可或缺的关键一环,对于要求高速数据采集的场合,由于采集的高速性,无法做到采集和分析数据同时进行,需要将数据先存储,待采集完毕后将数据传入计算机再进行分析。所以,高速可靠的存储系统是制约其速度的关键因素,保证存储的高速性对于保证数据的准确性与快速性具有重要意义。

近年来,NAND Flash 发展迅速,它以半导体作为载体,相比于传统的存储设备,可靠性更高,易于实现高速度、低功耗和小型化等指标,成为设计高速存储系统<sup>[2-3]</sup>的首选。NAND Flash 的基本存储单元是由块(Block)构成。每块又由若干页(Page)构成。其中,最小读写单元是页,最小擦除单元是块。在对一页进行编程之前,需要先将该页所在的块进行擦除操作。

在使用 NAND Flash 时,会出现一些坏块(Bad Block),如何对这些坏块进行管理,是其应用中的一个重要问题,管理方法的好坏直接影响到存储系统的可靠性<sup>[4]</sup>。坏块出现的原因,是在工作过程中会出现一些坏页,导致对该块无法进行擦除操作,从而无法再对此块进行编程,而此块中其他的页工作还是正常的。所以,该文提出的一种新的坏块管理方法是基于页的判断和跳过,保证了编程速度不会因为坏块的出现而大幅降低,提高了 Flash 资源的利用率,并且降低了缓存资源的占用。

## 2 应用背景

### 2.1 坏块的产生

由于 NAND Flash 制造工艺的原因,生产厂家不能保证出厂时每个块都是好块,会有一些出厂坏块,这些坏块厂家会标记出来,不影响 Flash 的正常使用。在使

用过程中进行编程操作或擦除操作时,也会有一些坏块的出现。使用过程中出现的坏块会带来一系列问题。它会降低写入速度,甚至会造成输入数据的堵塞无法及时存储,造成写入错误。所以,需要一个实用的坏块管理方法去解决以上问题。

### 2.2 高速数据传输中出现的问题

数据采集系统对于实时性有很高的要求。对于高速数据采集系统,后端必须要保证高速的数据存储。传统的数据传输系统框图如图 1 所示<sup>[5]</sup>。

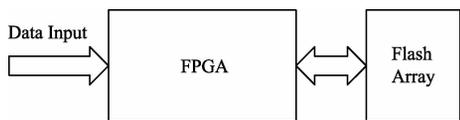


图 1 传统数据传输系统

在这个数传系统中,数据经采集电路进入 FPGA,在 FPGA 的控制下写入 NAND Flash 阵列。此过程中对于坏块的管理有 3 个关键技术<sup>[6]</sup>。首先是坏块识别,通过读取块中第 1 页或第 2 页(如果第 1 页是坏的)扩展区的信息来判断当前块是否是坏块;其次是坏块跳过,对于固有坏块,控制器会识别且跳过此块,对于使用过程中出现的坏块,通过对应关系表把坏块的数据转存到其他好块中;最后是坏块替换,如果一块在编程操作中出现了坏块,控制器就要先将写入的数据读出至缓存,然后写入下一好块中。

对于坏块的识别,传统方法需要访问 Flash 的扩展区来识别该块是否为坏块,此操作对于数据的写入来说可以说是“多余的”;坏块在替换时,需要把已经成功写入的数据重新读出再写入下一好块中,此过程也浪费了大量时间,如果写到最后一页出现坏块,那么写入就需要花费超过平常两倍的时间。

基于以上原因,本文提出了一种基于页跳过和页替换的坏块管理方法,使用 1 片 E<sup>2</sup>PROM 来存储坏块信息。在出现坏块时,只需要重新写入坏掉的那 1 页数据,这样就保证了数据在写入过程中不会因为坏块的出现而速度下降。

### 3 坏块管理方法

在传统的管理方法中<sup>[6-8]</sup>,当编程时出现了坏块,将其标记,并且将写入此块的数据重新读出来至缓存,然后写入下一好块。然而,坏块出现时,写入的最后一页数据写入失败,而之前页中的数据都已被成功写入。所以,使用传统的方法管理,那些成功写入的数据也会舍弃掉。一方面,浪费了 Flash 的存储资源;另一方面,重新写入数据将会降低整体写入速度,占用过多缓存。传统管理方法流程图如图 2 所示。

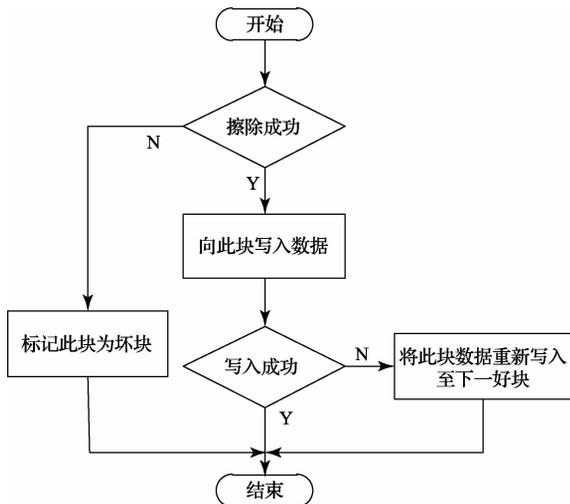


图 2 传统坏块管理方法流程

本文提出的新的管理方法中,向主控制器申请一个 8KB(1 页)大小的 RAM 作为一页数据的缓存,同时,在电路中增加一片 E<sup>2</sup>PROM(8K × 8bit)来存放坏块的信息。E<sup>2</sup>PROM 的地址与 Flash 逻辑块的地址一一对应,这样就保证了 E<sup>2</sup>PROM 中存放的数据与相应块地址的统一。在第一次编程时,主控制器向 E<sup>2</sup>PROM 所有地址写入 0xFF,表明每个块都是好块。在每次编程之前,主控制器会读取 E<sup>2</sup>PROM 中当前块的信息,来判断此块的好坏。在编程和擦除过程中,有些好块会变成坏块。当其在擦除过程中变

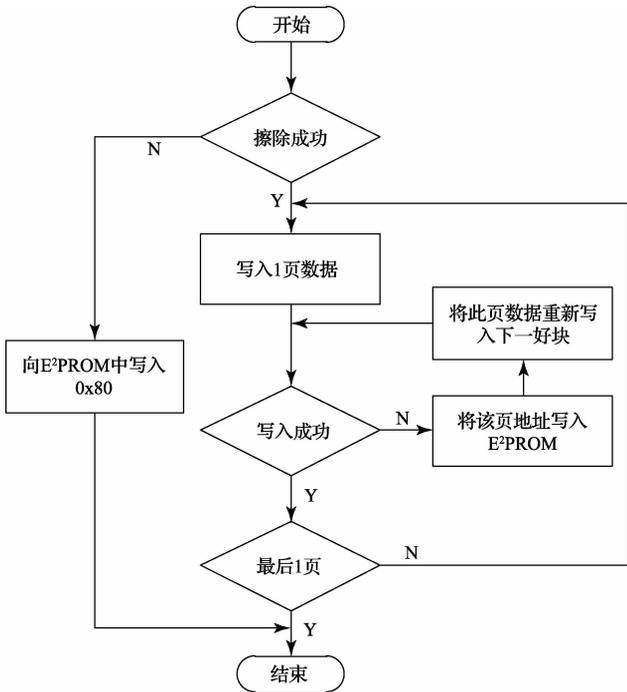


图 3 编程操作中坏块管理流程

成坏块时，主控制器将相应 E<sup>2</sup>PROM 地址中写入 0x80，表示此块是全坏块，里面没有任何数据；当其在编程操作中变成坏块时，主控制器将相应 E<sup>2</sup>PROM 地址中写入当前坏掉的页的页地址(0x00~0x7F)，然后将最后一页的数据重新写入下一好块，保留之前成功写入的数据。本坏块管理方法在编程操作中的工作流程如图 3 所示。

在读取操作中，主控制器首先读取 E<sup>2</sup>PROM 中当前块的信息，如果当前块是好块，则读出全部数据；如果当前块部分坏块，也就是说，此块中有成功写入的数据，主控制器就会读出坏页之前所有页中成功写入的数据；如果是全坏块，就直接跳过读取。其在读取操作中的工作流程如图 4 所示。

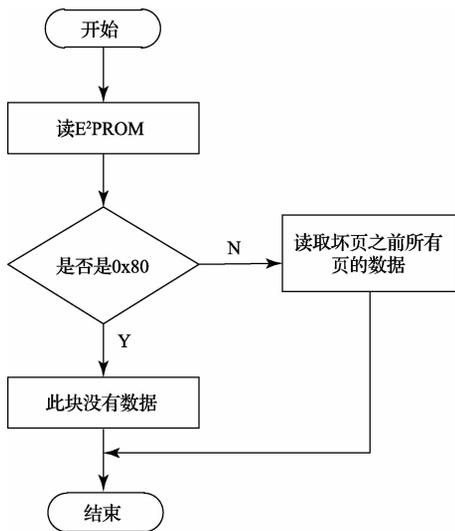


图 4 读取操作中坏块管理流程

#### 4 系统结构

设计应用在一个数据传输系统中，此系统的结构框图如图 5 所示。

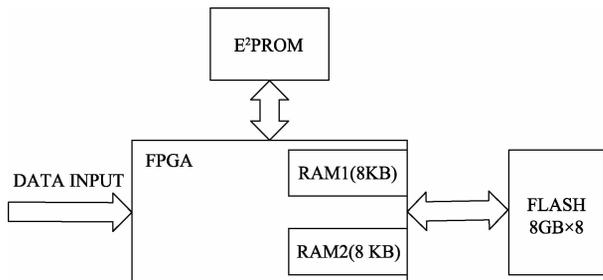


图 5 数据传输系统结构

在本系统中，采用 FPGA<sup>[9]</sup>作为主控制器，向主控制器申请两块 8 KB 大小的 RAM 作为缓存。在编程操作过程中，FPGA 首先读取 E<sup>2</sup>PROM 中的信息来判断当前块的好坏，若编程操作失败，FPGA 就向 E<sup>2</sup>PROM 对应地址中写

入坏掉的页地址，然后将此页的数据从 RAM 中读出。这样，我们只需要重新读写 8 KB 的数据，相较于之前重新读写 1 整块(1MB)数据有了很大的改善，保证了写入速度不会因为坏块的出现而下降。

#### 5 实验结果

系统以 Xilinx 公司的 FPGA 芯片 XC5VSX240T 作为控制核心，采用 8 片 Flash 并联的方式，在前端连续不断的输入 32 GB 测试数据。Flash 工作模式采用异步模式，控制频率 100 MHz。在原有坏块的基础上，写入时每 32 块设置一个坏块，通过比较本方法与传统方法写入时间及写入数据的正确性来判断方法是否可行，测试结果如表 1 和 2 所示。

表 1 编程 1 块所用时间

坏页位置	传统方法	本方法	提升率
无	45 ms	45 ms	无
第 60 页	73 ms	48 ms	52.08%
第 100 页	87 ms	48 ms	81.25%
第 127 页	92 ms	47 ms	95.74%

表 2 平均写入速度测试结果

实验次数	传统方法	本方法	提升率
1	170.10MB/s	175.61MB/s	3.23%
2	169.53MB/s	175.48MB/s	3.51%
3	170.03MB/s	175.80MB/s	3.39%

从表 1 的测试结果可以看出，在不出现坏块时，编程速度与之前相比没有提高，当出现坏块时，本方法能保证写入速度不明显下降，并且坏页位置出现的越靠后，之前的方法速度下降越大，因为要重新写入已写入的数据。而本方法不受此影响，本方法只是重新写入坏掉页的数据，保留之前成功写入的数据，保证了写入速度不明显下降。

对于数据的准确性，写入 4 GB 的测试数据，数据依次递增，上传后通过 MATLAB 进行分析，通过计算相邻数据之间的差值来确定数据没有丢失，也没有多余。实验证明数据的准确性得到保证。

对于数据的写入速度，单片 Flash 写入速度可达 22 MB/s。从表中可以看出，应用本方法使得总体速度与之前相比有一定提升，实质上是本方法保证了在坏块出现时的写入速度不会大幅下降，从而保证了写入平均速度。

通过实验表明，本方法有效实现了 NAND Flash 的坏块管理，使系统性能得到提升。

#### 6 结 论

本文提出了一种基于页跳过和页替换的 NAND Flash 坏块管理方法，并成功应用于在一个数据存储系统中。本方法保证了存储数据的实时性，保证了在出现坏块时写入

(下转第 26 页)