

NAND 存储器均衡算法的实现

董 萍

(三门峡职业技术学院 信息工程系, 河南 三门峡 472000)

摘 要: 文中提出了一种可用于 NAND 存储器的块寿命均衡算法. 此算法使用简易日志系统和垃圾块回收机制, 以降低极低的写入速度为代价, 成功的对 NAND 存储器的块寿命实现了均衡管理, 因而增强了存储器在嵌入式系统中的使用寿命和文件稳定性. 本算法经过基于 FAT 文件系统的实际测试显示了较好的性能, 具有很高的推广与应用价值.

关键词: NAND 存储器; 均衡算法; FAT 文件系统; Flash

中图分类号: TP333.3

文献标识码: A

文章编号: 1000-7180(2013)02-0053-04

Implementation of NAND Memory Equalized Algorithm

DONG Ping

(Information Engineering Department, Sanmenxia Polytechnic, Sanmenxia 472000, China)

Abstract: In this article, a NAND memory equalized algorithm of file system has been improved. This algorithm using a simple log system and garbage block recycling mechanism successfully equalize all block lifetimes and thus overcome the above-mentioned obstacles, at the cost of a little depressed write speed. The actual FAT application has proved the above-mentioned advantages, and it has very high value of promotion and application.

Key words: NAND memory; equalized algorithm; FAT file system; Flash

1 引言

NAND 存储器是 Flash 存储器的一种, 具有高密度、大容量等优点. 但 NAND 存储器的每一个块的擦写寿命都是独立计算并有一个上限值, 对某些块的频繁擦除操作很容易导致该块先于其他块失效. 这就需要有一个经过改进的算法来均衡每个块的擦写次数, 达到使 NAND 存储器的使用寿命更长的目的.

文件系统负责对文件进行管理. 使用文件系统管理硬件的数据存取方式和结构, 可以更好的利用存储空间和效率. 而由于 Windows 操作系统的普及, 微软公司的 FAT 文件系统在计算机设备中, 是一种很典型的文件系统.

一个 FAT 文件系统由 4 部分组成: (1)FAT 保留区; (2)FAT 表区; (3)FAT 根目录区; (4)FAT 文件数据区. FAT 表区用于存放存储器数据索引, 其中包括两个完全相同的 FAT 表: FAT 表和

FAT1 表, 其中后者为前者的备份, 也是本文验证算法的重点^[1-2].

2 NAND Flash 均衡存储方案

2.1 要解决的问题

目前 NAND 存储器文件系统主要存在两点问题:

(1)NAND 出厂时有随机分布的坏块, 文件系统对此需要调整;

(2)文件系统在使用 NAND 的过程中某些块会由于频繁写而先坏掉, 这严重影响了 NAND 存储器整体的使用寿命^[3].

解决的基本方法: (1)对于出厂坏块, 文件系统需要对 NAND 进行扫描并为其登记坏块表; (2)对于读写寿命问题, 使用改进的读写算法重新分配对块的读写, 使 NAND 存储器中的每个块的擦写次数十分接近, 来实现寿命的极大延长.

2.2 具体实现方法

以三星公司的 K9G8G08U0M 芯片为例来说明实现方法。

2.2.1 日志系统的设计

每个 NAND Flash 存储器的块并不是按照文档说明的大小,而是有一个“附加”存储空间,称作“Spare Space”.在 K9G8G08U0M 芯片中,每页的大小是 2KB,会有 64 个字节的 Spare Space^[4],那么这个空间可以用来作为均衡读写算法的一个很重要的实现工具。

我们使用 Spare Space 为每个页建立日志信息,每个 Spare Space 都包含以下信息:

- (1)PageID,此页的编号;
- (2)SN,此页的序号,这个序号可以用来在突然断电时恢复文件;
- (3)PageStatus,此块的状态;
- (4)数据的 ECC 校验信息。

这些区域的每一个 bit 随着块的擦写一同被初始化为全为 1 的状态。

记录信息在页中的格式如图 1 所示。

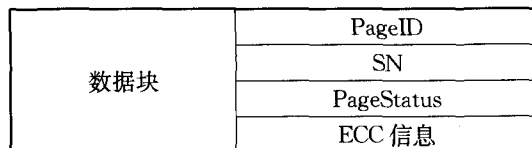


图 1 Spare Space 信息组织图

同时,对每个块也都建立了类似的信息记录,每个块有以下几个信息:

- (1)BlockID,此块的编号;
- (2)BlockStatus,此块是否为坏块;
- (3)BlockInUse,此块是否正在使用;
- (4)BlockRecycle,此块是否等待回收;
- (5)ECC,此块的数据校验信息。

改进的文件系统基于以下方法工作:

(1)如果有数据写入,那么会查找新块,并严格按照每块内页的地址从低到高的顺序将新页写入,然后把旧块中对应的 PageStatus 标记为“已删除”;

(2)如果新块的数目低于一个门限值,会回收那些块中所有 Page 已经标记为“已删除”的旧块,将他们转变为新块;

(3)每一次写页的操作 SN 会加 1 并和数据一起存入新块对应页的 Spare Space 内;

(4)每个文件都按页储存。

当文件系统对文件进行写操作时,我们会遇到两种情况:

(1)删除文件,这时我们仅仅将文件对应的页的 PageStatus 置为“已删除”即可,并不真正去从物理上删除这个页;

(2)改写文件(文件内容增加或减少),我们会找到一个新块,将新的数据写入并将新块的 BlockInUse 标记为 True,表示此块正在被写入。随后,改写文件系统的文件记录将新写入的页加入原来的文件序列,旧页被标记为“已删除”,等待回收。如果这个页是最后一个所在块的可用页,那么就将该块的 BlockRecycle 也标记为“已删除”,等待块被回收。

除此之外,文件系统对硬件的接口也需要进行改写,在系统启动时将改进的文件系统信息从 Flash 的 SpareSpace 中读取上来并严格按照块的顺序组织成文件系统的基本存储形式。

上述软件实现流程如图 2、图 3 所示。

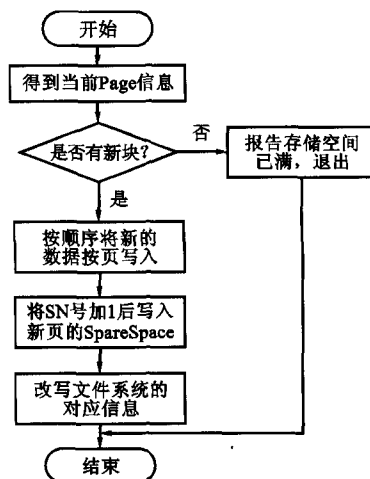


图 2 改进文件系统的写操作

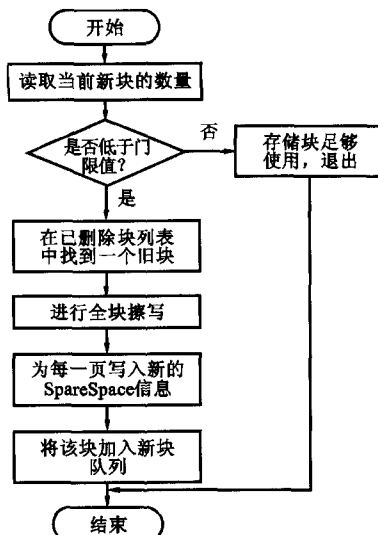


图 3 新块查找算法流程图

2.2.2 文件系统的坏块处理流程

NAND Flash 在使用时也会突然发生错误导致某块无法继续使用,因此需要为此建立坏块机制.本改进文件系统对坏块的处理十分简单,每个 NAND Flash 都有坏块测试方法,按照这个方法测试 ECC 值就可以知道这个块是不是有效的.如果是无效块,只需将此块在内存中的 BlockStatus 登记为 Invalid,随后在文件系统的存储管理表内将对应的存储单元登记为“已损坏”即可.

如果在使用中发生了块失效的情况,一般此时读取块数据是没有问题的,因此只需找一个新块,将失效块的数据搬移进去,随后改写对应的存储管理表即可.如果发生了没有新块的情况,由于本文件系统是按照块内的页顺序分配空间的,也就意味着已经没有任何空间可以为失效块替换了,此时文件系统会返回严重错误并提示文件可能已损坏.

坏块处理流程图如图4所示.

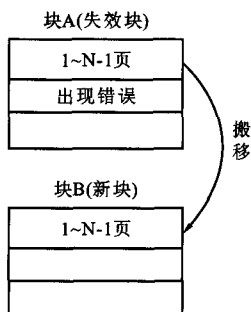


图4 失效块替换算法

2.2.3 意外中断的文件恢复

本文件系统对文件进行写操作的过程中,会将 SN 增 1 并写入新页的 SpareSpace 中,SN 号占用 2 个 bit.从图2可以看到,改写文件时会先写入数据到一个新块内,然后才将原数据页的 PageStatus 置为“已删除”.如果在数据写入完毕但是还未将原块进行 PageStatus 置为“已删除”时发生意外(例如断电等),那么系统恢复后,文件系统内会有 2 个页的 PageID 号完全相同,但是它们的 SN 号不同.由于 SN 号只有 2bit,因此只有 0、1、2、3 这 4 个可能的值.此时我们使用以下算法就能分辨谁是更新的数据并恢复文件内容:

(1)如果两个 SN 不是 0 和 3,那么哪个 SN 号较大,哪个就是更新的数据页;

(2)如果两个 SN 号是 0 和 3,那么将较大的那个 SN 进行 $(SN + 1) \bmod 4$ 的运算后,如果与较小的 SN 相等,那么就说明较小的 SN 代表的页是更新的页.

例如,文件系统扫描时发现两个页的 PageID

相同,但是 SN 一个是 2,另一个是 3,则 SN 为 3 的页存储的是最新的数据;

如果 A 页的 SN 是 3,而 B 页的 SN 是 0,则将较大的 SN 号 3 进行 $(3 + 1) \bmod 4$ 运算,结果是 0,这个结果等于 B 页的 SN 号,因此可知 B 页存储的是最新的数据.

2.2.4 基于 FAT 文件系统的实现

考虑到算法在 FAT 文件系统实现,我们在前文所述的两个写操作的情况下需要做出以下处理:

(1)如果是文件删除操作,将 FAT 表中对应的簇标记为可用;

(2)如果是文件改写操作,将 FAT 表中簇的信息改写,使其指向新块中的页即可.

对于 FAT 系统不同的簇大小,建议使用和页相同的簇大小.例如 NAND Flash 的页为 2KB,那么就尽量使用 2KB 的簇大小定义,这样可以达到最高的读写效率而无需任何转换.对于嵌入式系统来说,这样的设计是可以接受的.

根据参考文献[1]的描述,设计时将 FAT 表和 FAT1 表也纳入了均衡磨损算法的控制范围内,并且在 NAND Flash 首块的固定地址处写入当前 FAT 表的物理地址以便查找,由此实现了整个 FAT 文件系统的均衡磨损.

3 分析与测试

本文改进了 FAT 文件系统,为其增加了硬件操作来提高存储器的使用寿命,因此在使用时可能会造成文件读写的时间延长.

NAND Flash 总容量为 1GB,采用了实际测试的方案对改进 FAT 文件系统检验.检验方法为两部分.

3.1 大批量读写操作的文件系统稳定性.

将上述 NAND 存储器应用于 2 部监控设备中,此设备会记录新的音频数据并擦除掉最旧的数据.长时间使用后对文件系统信息进行读取.

由于每块 NAND 在出厂时就会有较多的坏块,因此无法达到 100% 的块都可用.从表 1 中可以看出,原始的 FAT 文件系统坏块数比改进的文件系统高了大约 21%.

表1 两种文件系统的存储可靠性对比

	Block 总数	可用数	可用率
原文件系统信息	4096	2889	71%
改进文件系统信息	4096	3754	92%

3.2 改进文件系统对读写速度的影响

改进文件系统比原文件系统多了许多处理步骤,会对 NAND 存储器的读写速度产生影响.将上述监测设备采用 USB 连接电脑,并使用权威读写测试软件 ATTO 测试,其典型结果如图 5、图 6 所示.

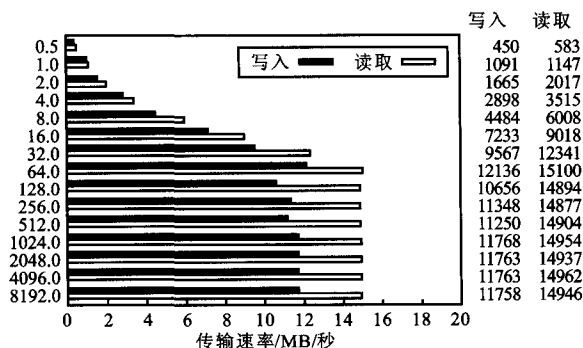


图 5 原文件系统的文件读写速度

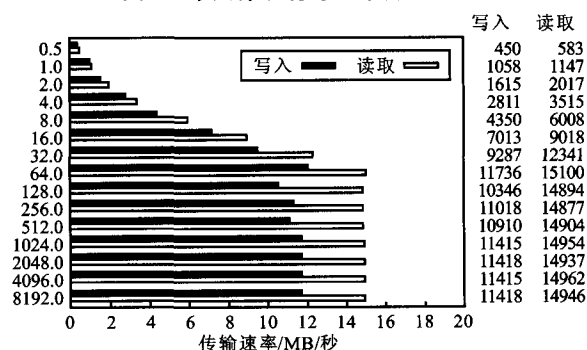


图 6 改进文件系统的读写速度

可以看出,读取速度没有影响,而写入速度大约降低了 3%.因为写入时是直接向新块中写入,并将旧页标记为已删除,因此对写入速度的影响并不是

很大.使用原 FAT 系统的设备使用改进系统后对系统性能也没有显著影响,反而可以有效的提高系统的使用寿命.

4 结束语

分析与测试表明,改进型文件系统对较大容量的 NAND 存储器的使用寿命和文件可靠性均有较大提升,在 FAT 文件系统上的实现证实了它的可靠性,具有较高的推广和使用价值.

参考文献:

- [1] 姚堃,张俊涛.提高 FAT 文件系统在 NAND 存储器上可靠性的研究[J].微电子学与计算机,2010,27(4):75-81.
- [2] 丁双喜,张盛兵,贾宝峰.双通道流水线 Flash 存储系统的设计[J].微电子学与计算机,2008,25(8):224-228.
- [3] Microsoft Corporation. Microsoft extensible firmware initiative FAT32 file system specification[S]. Version 1.03,2000.
- [4] Samsung Corporation. Advance flash memory-K9G8G08U0M[S/OL]. [2000-10-11]. <http://wenku.baidu.com/view/b8d70ce9856a561252d36ff4.html>.

作者简介:

董萍女,(1980-),硕士,讲师.研究方向为嵌入式计算机应用技术.

(上接第 52 页)

- [9] 刘碧海,谢嘉宏.格雷码的关键位位置决定装置及方法 1756085A[P].中国,2006-04-05.
- [10] 李正生,马文彦.一个人工智能问题:用格雷码±1 的规律套解九连环[C]//Anon. 3rd World Congress on Intelligent Control and Automation. China:Hefei: Proceedings of the 3rd World Congress on Intelligent Control and Automation. 2000, 1-5/304-307.

作者简介:

李正生男,(1957-),博士研究生,教授.研究方向为信号与信息处理,自动化测试与控制,电子系统故障诊断.

马文彦女,(1956-),硕士,高级实验师.研究方向为核科学技术.

闫杰男,(1960-),博士,教授,博士生导师.研究方向为飞行器控制与仿真.