

NAND 型闪存不溢出存储设计

陈 荔¹, 张 菁²

(1. 长安大学 教育技术与网络中心, 陕西 西安 710064; 2. 长安大学 电子与控制学院, 陕西 西安 710064)

摘 要:针对在无人值守的工作环境下, 数据存储系统由于数据量太大而发生溢出的现象, 以 K9K8G08U0A 闪存为例, 设计读、写、擦除基本的操作, 引入非离散数据与离散数据的概念来标识需要存储的数据, 利用数据回卷机制来存储数据, 把存储空间分成坏块记录区、非离散数据存储区和离散数据存储区 3 部分, 同时设计数据写入方式, 确保每次写入的数据在离散履历区和非离散履历区均存在。设计结果表明: 在达到离散存储区最大值时, 将存储引向离散存储区的开始位置, 使离散存储区变成首尾相连的闭合环, 从而实现数据存储不溢出; 针对数据回卷造成存储时间较早的数据部分或全部被覆盖, 造成数据的不完整, 采用数据反向分析的方法, 通过比较相邻数据存储空间是否有交集来分解出正确的数据; 使用数据回卷的存储机制, 使存储空间变成一个“永远存不满的空间”, 确实保证了数据存储不溢出, 保存了最新最近的数据。

关键词: NAND 型闪存; 离散数据; 事务回滚; 反向分析; 存储机制

中图分类号: TP311

文献标志码: A

NAND flash-based storage design with no overflow

CHEN Li¹, ZHANG Jing²

(1. Center of Educational Technology and Network, Chang'an University, Xi'an 710064, Shaanxi, China;
2. School of Electronic and Control Engineering, Chang'an University, Xi'an 710064, Shaanxi, China)

Abstract: According to the data storage system occurs as a result of data is too large and overflow phenomenon in the unattended working environment, based on the K9K8G08U0A flash as an example, reading, the writing, erasing were designed. The concept of non-discrete data and discrete data were introduced to identify the need to store data, the rollback mechanism was used to store data, the storage space was divided into the bad block record area, non-discrete storage area and discrete data storage area. It is ensured that every write data in discrete resume and non-discrete resume area existed with the design data write mode. The results show that when the discrete storage area upon reaching the maximum value, it will be stored into discrete storage area become to the start position, the discrete storage area into a closed ring of end to end, so as to realize the data storage do not overflow. According to the data from an earlier time data storage back part or all of the cover, which cause the incompleteness of data, the data of reverse analysis method was used, through the comparison of the neighboring data storage space is the intersection of decomposition of the correct data. The data rollback storage mechanism was used to make the memory space into a “forever dissatisfied space”, the data storage does not overflow is ensured and the latest data is preserved. 3 tabs, 5 figs, 9 refs.

Key words: NAND flash memory; discrete data; transaction rollback; reverse analysis; storage mechanism

0 引言

嵌入式自动化系统已经广泛地应用在各行各业。NAND Flash 闪存存储器(简称闪存)以其掉电非易失、功耗低、寿命长、升级容易等独有的特点,迅速成为数据存储的最佳选择。目前,基于 NAND 闪存的文件系统大致有 2 种。一种是集中索引的文件系统;另一种是专门针对 NAND 闪存而设计的文件系统^[1]。集中索引的文件系统就是通过将 NAND 闪存模拟成可随意读写的块设备对闪存进行管理,这种文件系统是比较传统的文件系统,比如 FAT32 和 Ext2^[2]。传统的文件存储系统因具有符合多数人的使用习惯、存储利于理解、操作方便等优点得以广泛的使用^[3]。但是在一些需要长时间无人值守或是短时间需要记录可能超过闪存容量的数据记录的情况下,这种文件系统明显存在缺点,常常丢失最新最近的数据。一般的解决方式就是不断的增加闪存的数量,但是这种解决方式系统造价就会比较高,并且系统的体积也会变大^[4-5]。如果通过软件设计解决这个问题,将大大提高系统的适用范围。本文在这种前提下,进行存储模式的研究,引入数据自动删除功能。闪存是按照块进行存储的,删除操作是以块为单元。一个闪存根据容量大小分成若干个数据块。数据存入数据块前,先对块进行删除操作,然后写入数据。存储数据到最大块后,重新开始对最小块进行删除,写入最新数据^[6-7]。通常将这种存储方式称之为数据回卷。软件设计中融入了数据回卷的机制,解决了存储系统在任何情况下始终能保留最新最近的数据记录的问题。

1 闪存体简介

1.1 闪存体

本次设计以三星 K9K8G08U0A 闪存为例,K9K8G08U0A 是大容量、高可靠性的 NAND Flash 存储器。其存储容量为 1 G 字节。

图 1 为 K9K8G08U0A 的阵列组织,有 1 G 存储空间。芯片有 8 192 个块(Blocks),一个块分为 64 页(Pages),1 个页包含 (2K+64) 字节(Bytes)。I/O0~I/O7 寻址字节序列。

1.2 基本命令设计

对于存储文件或者日志,基本操作有写入、擦除

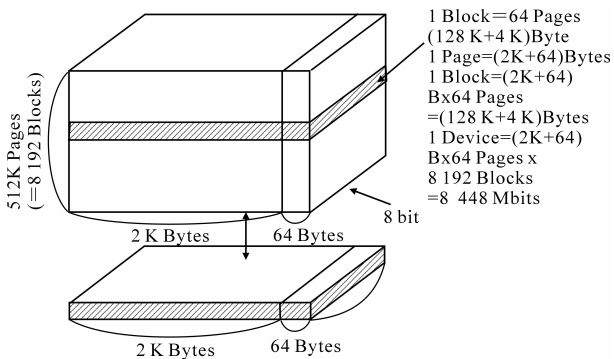


图 1 K9K8G08U0A 阵列组织

Fig. 1 Array organizations of K9K8G08U0A

和读出动作,见表 1。根据芯片的阵列结构与寻址字节及表 1 所示的芯片提供的访问命令字,设计出对芯片的读、写及擦除的基本操作。芯片读、写操作以页为单位,因为对 NAND Flash 写数据时,位数数据只能由 1 变为 0,所以芯片擦除操作需要以块为单位,对 Flash 写操作前必须把要写单元所在块整体擦除^[3]。读数据命令需要携带芯片标识,需要读出数据的长度,读取数据位于芯片的起始位置(块号和页号),及读数据关键字。写数据命令需要携带知芯片标识,写入数据的长度,数据存储到芯片的位置(块号和页号),写入数据的内容及写数据关键字。擦除数据命令也需要携带芯片标识,要擦除的块的位置(块号)及擦除关键字。根据以上要求设计出如下命令字格式。

表 1 K9K8G08U0A 命令字

Tab. 1 K9K8G08U0A command summary

功能	第 1 周期	第 2 周期
读	00H	30H
写	80H	10H
擦除	60H	D0H
芯片标识	F1H	

读数据:F1H+Int64ConvertChar2(读取数据长度)+00H+00H+00H+Int64ConvertChar3(Block * 64 + Page)+30H

写数据:F1H+Int64ConvertChar2(写入数据的长度)+80H+00H+00H+Int64ConvertChar3(Block * 64 + Page)+Data(写入数据内容)+10H

擦除数据:F1H+00H+04H+60H+D0H+00H+00H+ Int64ConvertChar3(擦除块编号)

上述命令字中 Int64ConvertChar2 功能是将数字转换为 2 字节的字符串,Int64ConvertChar3 功能是将数字转换为 3 字节的字符串,并且调整低字节在前、高字节在后的顺序。本命令设计只对整页进行操作,所以寻找地址的 5 个字节中低 2 字节用 00H+00H 表示,高 3 字节用 Int64ConvertChar3 (Block * 64 + Page)表示。有这 3 个基本的操作命令,芯片就可以自由的使用^[4]。

2 存储方式设计

2.1 设计需求

数据的储存设计好坏直接影响到数据的存取效率、系统实现效率和系统的运行效率。为了更好地方便快捷地存储数据,引入非离散履历与离散履历的概念,对于新存入的数据文件,把数据文件的存储时间、类型、大小等代表文件特征的部分称为非离散履历,数据文件内容称为离散履历。由于 NAND Flash 同其他固体存储器一样会产生坏块,如果将数据存储坏块区域,将导致无法正常读出,而系统的正常工作状态是不允许把数据存储坏块区,所以需要根据器件出厂时自带的坏块信息表标记坏块区域,以保证数据不被写到坏块区域。鉴于以上需求,将存储空间划分为坏块存储区、非离散存储区和离散存储区。非离散存储区存储数据的特征信息,类似于数据的索引,根据非离散履历可以按图索骥找到离散存储空间中的数据内容。坏块区存储该闪存体的坏块记录,其存储空间划分如图 2 所示。

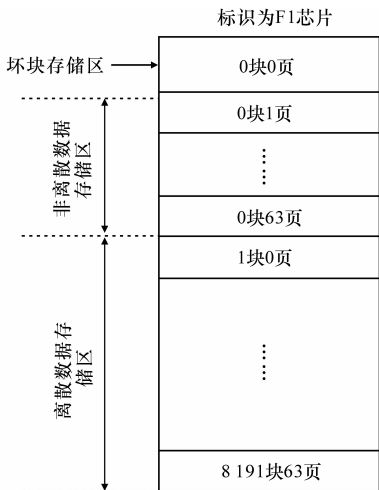


图 2 存储空间划分
Fig.2 Storage space division

2.2 读取设计原理

NAND Flash 坏块区域一般不超过器件总容量的 3%,出厂芯片会说明坏块,把坏块的编号存储在 F1 芯片的第 0 块的第 0 页,以逗号分隔坏块号。如

坏块记录为是 8,18,8 100 等,在对芯片进行写操作时,判断将要写入的块是否存在于坏块列表中,以防止数据存储坏块区域而无法恢复。对于坏块区,需要在使用前对芯片进行全盘测试,准确测试出所使用芯片的坏块数量和位置。测试坏块区需要对测试的块进行擦除、写和读的操作。准备一页的数据,首先对需要测试的块进行擦除操作,然后将准备好的数据写入测试块中,再将写入数据读回来,比较写入数据和读出的数据是否完全一致,如果发现不一致,说明所测试的块是坏块,反之就是正常的。

非离散数据相当于文件系统的索引区,每一组数据形成一条记录。每一条记录包含着数据文件的特征,数据特征有文件创建时间、文件类型、文件存储的位置等信息。文件创建时间取在文件存储时的系统时间,文件类型根据文件的属性进行区分,例如有数据文件、图像文件、声音文件等,文件存储位置包括起始块号、起始页号、终止块号、终止页号。非离散履历以一条长度为 21 字节的记录存储在 F1 芯片的第 0 块的第 1 页,记录头尾各为一个字节的标志,利用记录的长度以及头尾标识可以将非离散履历进行正确分离。非离散履历中记录序列号是一组唯一不重复并且递增的一组整数,该整数可以唯一标识非离散履历,见下页表 2。

离散履历是数据文件内容,存储在 F1H 芯片的第 1 块到 8 191 块区间,为了对数据信息进行多重备份,在离散数据的前面添加一些数据文件简单的信息,称为离散履历数据头,将非离散数据部分信息贴在离散数据的头部,见下页表 3。

离散履历数据头中序列号等同于非离散履历中的履历记录序列号,数据长度表示数据文件的真实长度,离散履历的存储方式是离散数据头信息+数据。离散数据前面携带简单的数据信息,如果非离散数据出现意外的情况无法读取,也可以通过这些信息来归类数据的属性,正常情况这些信息被忽略。

3 基于数据回卷机制的数据读取设计

3.1 数据回卷机制

存储空间正确划分后,系统就可以正常的存储读取数据。但是,如果是长时间无人值守的环境,系统运行需要存储的信息就可能超过存储芯片的容量,从而导致最新信息丢失。为解决这一问题,引入数据回卷的机制。数据回卷取义于数据库事务日志的回滚,数据库在执行一个指定事务出现异常后,整个事务回到事务执行前。数据回卷指系统进行正常数据存储,如果存储到 8 191 块的 63 页后,系统将存储重新指向第 1 块的第 0 页继续进行数据存储。

表 2 非离散履历定义
Tab. 2 Definition of non-discrete resume

序号	名称(助记符)	长度/ 字节	备注
1	履历条目标识(PE)	1	固定值 0xB8
2	数据类型(TYP)	1	0xFF 表示本条目为空, 0x11 表示数据履历条目, 0x44 表示图像履历条目, 0x88 表示音频履历条目
3	履历记录起始时间 (START_SER)	6	离散履历开始记录的时间, 日期按照年月日时分秒格式 组织:YYMMDDHHmmSS
4	履历记录序列号 (SERNUM)	2	履历记录序列号按照年月日 时分秒格式+16 bits,流水 号 组 织: YYMMDDH- HmmSS+AA+BBAA+BB 为 16 bits 流水号,每次存储 过程后自动加 1
5	履历记录起始块号 (START_SBK)	2	离散履历开始记录的块号, 有效值为 1~8 191 块
6	履历记录起始页号 (START_SPG)	2	离散履历开始记录的页号, 有效值为 0~63 页
7	履历记录结束块号 (STOP_SBK)	2	离散履历结束记录的块号, 有效值为 1~8 191 块
8	履历记录结束页号 (STOP_SPG)	2	离散履历结束记录的页号, 有效值为 0~63 页
9	履历结束标志(EOP)	1	0xFA

表 3 离散履历数据头定义
Tab. 3 Definition of discrete resume data header

序号	名称(助记符)	长度/ 字节	备注
1	履历条目标识(PE)	1	固定值 0xB8
2	履历序列号(SER)	2	序列号
3	数据存储的起始块号 (SBK)	2	有效值为 1~8 191
4	数据存储的起始页号 (SPG)	1	有效值为 0~63
5	数据长度(DLEN)	4	本履历条目录的有效数 据长度(字节,不算数据信 息部分的长度)。
6	履历结束标记(EOP)	1	固定值 0xF5

整个芯片存储空间成为一个首尾相连的闭合环,变成一个“永远存不满的空间”。

数据回卷机制如图 3 所示,存储顺序从小 Block 向大 Block 依次存储,需要数据回卷时,系统总是把时间最久的数据覆盖,从而保证留下的数据最新。首尾相连的存储空间实现了存储不溢出。

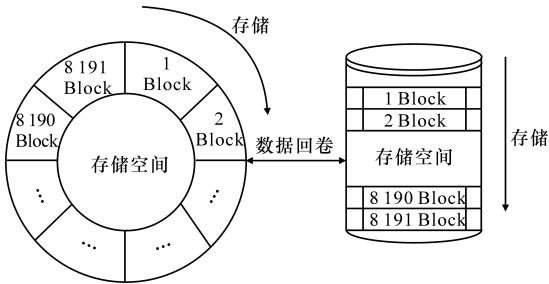


图 3 数据回卷机制
Fig. 3 Data rollback mechanism

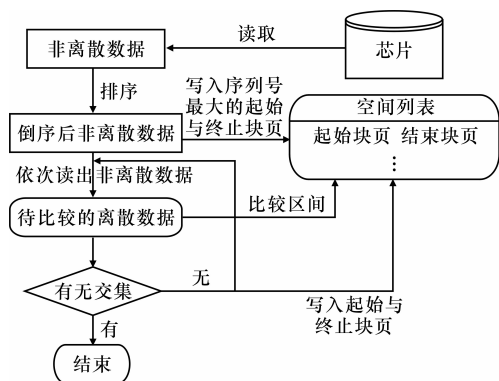
3.2 写入数据设计

对整个芯片的空间进行功能划分,写入数据发生溢出引入回卷机制后,就需要将数据正确的写入芯片。采集前端获得数据后,读出非离散履历记录号中最大的记录,分离出其中的终止块终止页。将页数加 1 作为写入新数据的起始页,如果加 1 后大于 63,则需要对块加 1,页置 0。得到新的块号后,需要将块号带入坏块区进行比较,如果块号存在于坏块区,说明该块不能使用,需要将块号继续加 1,同时再次检查是否是坏块,直到找到正常的块(每次存储使用新块都需要进行坏块检查)。得到数据存储的起始块号页号后,再计算出新的履历记录号(通常为最大的履历号加 1),同时整理出每一页数据前面的数据头,将数据头和数据内容组成的数据包利用写命令字逐页写入芯片。数据正确写入后,数据记录号、起始块号页号、终止块号页号和获取系统当前的时间作为存储时间,把这些数据整理成一条非离散履历,写入非离散履历空间,完成了一次数据的写入。在写入数据时发现块号大于 8 191,页号大于 63,说明芯片已经全部写满,剩余的数据需要重新写入 1 块 0 页及之后的空间中。数据写入新块前,需要对新选的块进行删除操作。每次数据到来,先将数据写入离散数据区,然后根据得到的数据在非离散履历区写一条记录,完成一次数据的正确写入^[5]。

3.3 读出数据分析

存储空间的首尾相连实现了存储不溢出,但是也带来了一个问题,数据回卷后,一些数据被完全覆盖或部分覆盖,如果在读出数据方面不进行正确的分析,可能出现数据混乱。可以采用倒推分析的方式来逐一一对非离散数据进行分析,如果发现非离散数据间起始块页与结束块页出现交集,说明数据出现过覆盖。首先把所有的非离散数据读出,通过序列号进行由大到小的排序,同时建立一个空间列表;根据存储规则,序号最大的数据必定是完整的,将其起始块页与终止块页划定的区间写入空间列表,如果出现起始块页大于终止块页,说明本次数据存储

出现回卷,把本次数据区间化成 2 个区间范围写入空间列表,一个区间是由起始块页到 8 191 块 63 页,另一个区间是 1 块 0 页到终止块页;分析序列号次小的非离散数据,用该数据的起始块页和终止块页与空间列表中的数据进行比较,同样,如果该数据的起始块页大于终止块页,也是要分成 2 个区间分别与空间列表数据比较,2 个区间都引入,没有交集才能说明本数据完整,如果发现交集,说明该数据已经被部分覆盖,该数据可以丢弃,比该序列号大的记录是正常记录,比该序列号小的记录已经被完全覆盖^[8]。通过数据倒推,从非离散数据中分析出正常的记录,根据非离散数据记录的起始块页与终止块页将离散数据的内容读出,将数据前面的 11 (离散履历数据头信息) 直接去掉,剩余的部分保留成记录文件,如图 4 所示^[6]。



注:出现交集后,比该序列号大的数据为正常数据,比该序列号小(包括该序列号)的数据信息丢失。

图 4 数据反向分析

Fig. 4 Reverse analysis of data

3.4 芯片损坏数据读出设计

通过以上的设计,完成了数据内容的不溢出存储,并且通过查询非离散履历就可以下载到需要的离散履历。但是任何芯片在使用过程中都可能损坏其中的数据块,如果损坏的是非 0 块,通过记录这个坏块到坏块区,芯片还可以继续使用。如果损坏的是 0 块,那这个芯片就不能使用了(芯片出产前规定 0 块不能是坏块)。不能使用的芯片如何将它已经存储的内容读出呢? 由于 0 块损坏,非离散履历已经无法读出,这种情况下就需要使用离散履历数据前面加贴的数据头进行分析分解芯片数据。首先将对芯片从 1 块 0 页的数据读出,设置一个变量用于临时保存离散履历的履历记录号,同时创建以该记录号为文件名的空文件,将读出的数据部分写入文件。继续读取下一页的数据,读出履历记录号,与变量中保存的记录号比较,如果一致,说明该数据也可以写入文件,如果不一致,说明该记录号包含的数据

已经读取完毕,关闭文件同时创建以新记录号为文件名的空文件,同时更新临时变量记录的离散履历号,并将读出的数据写入文件,依此类推,逐页逐块就可以将芯片中的数据读取整理出来^[7],数据读出流程如图 5 所示。需要注意的是,如果这块芯片的数据发生过回卷,回卷的结果是将一个整体数据分别存储在芯片的尾部和头部,并且尾部存储文件的头,头部存储文件的尾。为了保证数据的正确性,在读完最后一页数据后,应该重新读取芯片 1 块 0 页的数据,读出新记录号与保存在临时变量中的记录号相比较,如果不一致,说明整个芯片数据读取完成,关闭文件。如果一致,说明数据的内容由于回卷被分别保存了,需要进行数据读取并且写入文件,直至发现记录号不一致,关闭文件,整个芯片数据读取完毕^[9]。

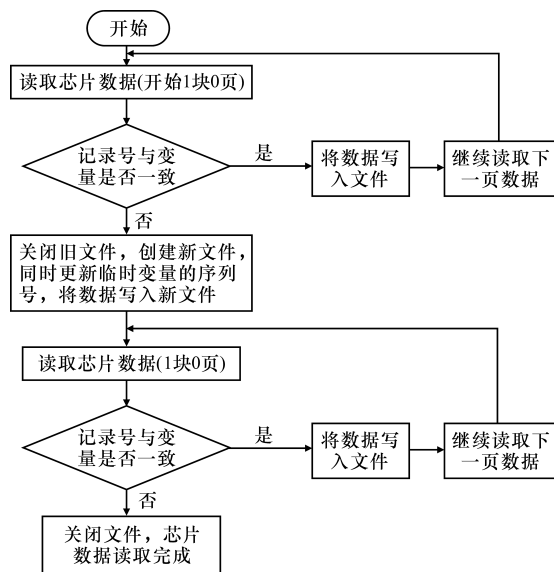


图 5 损坏芯片数据读出流程

Fig. 5 Read-out schematic of the damage to the chip data

4 结 语

(1)先对存储空间进行有效划分,分成坏块存储区、非离散数据存储区和离散数据存储区 3 部分,对需要存储的数据以离散数据和非离散数据的概念来标识并分别存储,利用数据回卷的机制使离散存储区成为一个闭合环,存储到最大位置后,开始自动从头开始存储数据;设计数据写入方式,确保每次写入的数据在离散履历区和非离散履历区均存在。

(2)由于数据的回卷导致一些数据被部分或全部覆盖,利用数据倒推方法分析数据,通过分析相邻间数据存储空间的交集来确定文件的完整性。

(3)如果芯片非离散数据区损坏后,利用离散数据区的数据头信息将存储在芯片中的数据整理读

出。设计引入数据回卷的存储机制,使存储空间变成一个“永远存不满的空间”,确实保证了数据存储不溢出;保存了最新最近的数据。通过以上的详细设计,设计出了一个在特殊情况下也可以正常使用的数据记录系统模式。该记录系统已经在个各种教练车上得到了应用,用于记录学员操作仪器的每一个动作。

参考文献:

References:

- [1] 汤显,孟小峰. FClock: 一种面向 SSD 的自适应缓冲管理算法[J]. 计算机学报, 2010, 33(8): 1460-1471.
TANG Xian, MENG Xiao-feng. FClock: an adaptive buffer replacement algorithm for SSD[J]. Chinese Journal of Computers, 2010, 33(8): 1460-1471. (in Chinese)
- [2] 周大,梁智超,孟小峰. HF-Tree: 一种闪存数据库的高更新性能索引结构[J]. 计算机研究与发展, 2010, 47(5): 832-840.
ZHOU Da, LIANG Zhi-chao, MENG Xiao-feng. HF-Tree: an update-efficient index for flash memory[J]. Journal of Computer Research and Deveopment, 2010, 47(5): 832-840. (in Chinese)
- [3] 朱宁太,姚智慧,张淑萍. 基于可信存储的固态硬盘安全机制研究[J]. 计算机工程与设计, 2010, 31(6): 1212-1215.
ZHU Ning-tai, YAO Zhi-hui, ZHANG Shu-ping. Authentication scheme research on solid-state disk based on trusted storage[J]. Computer Engineering and Design, 2010, 31(6): 1212-1215. (in Chinese)
- [4] 高文青,沈维聪. NAND 存储器在嵌入式系统中的应用[J]. 电测与仪表, 2008, 45(2): 45-48, 57.

(上接第 54 页)

- [2] 景思督,张鸣远. 流体力学[M]. 西安:西安交通大学出版社,2001.
JING Si-du, ZHANG Ming-yuan. Fluid mechanics [M]. Xi'an: Xi'an Jiaotong University Press, 2001. (in Chinese)
- [3] 王惠宾. 矿井通风网络理论与算法[M]. 北京:中国矿业大学出版社,1996.
WANG Hui-bin. Mine ventilation network theory and algorithms[M]. Beijing: China University of Mining and Technology Press, 1996. (in Chinese)
- [4] 王永东,夏永旭. 公路隧道纵向通风数值模拟[J]. 中国公路学报, 2002, 15(1): 82-85.
WANG Yong-dong, XIA Yong-xu. The road tunnel longitudinal ventilation numerical simulation[J]. China Journal of Highway and Transport, 2002, 15(1): 82-

GAO Wen-qing, SHEN Wei-cong. NAND flash application in embedded system[J]. Electrical Measurement & Instrumentation, 2008, 45(2): 45-48, 57. (in Chinese)

- [5] 罗晓,刘昊. 一种基于 FAT 文件系统的 NAND Flash 坏块处理方法[J]. 电子器件, 2008, 31(2): 716-719.
LUO Xiao, LIU Hao. Method of NAND flash memory bad block management based on FAT file system[J]. Chinese Journal of Electron Devices, 2008, 31(2): 716-719. (in Chinese)
- [6] 王宏强,王太勇,邓辉,等. 面向闪存类存储设备嵌入式文件系统研究与实现[J]. 计算机应用研究, 2008, 25(3): 814-819.
WANG Hong-qiang, WANG Tai-yong, DENG Hui, et al. Research on embedded file system faced flash memory[J]. Application Research of Computers, 2008, 25(3): 814-819. (in Chinese)
- [7] 杨春林,雷航. 基于 NAND Flash 的嵌入式文件系统的改进与优化[J]. 计算机应用, 2007, 27(12): 3102-3104.
YANG Chun-lin, LEI Hang. Improvement and optimization of embedded file system based on NAND Flash[J]. Computer Applications, 2007, 27(12): 3102-3104. (in Chinese)
- [8] Robert T. Trusted computing for disk drives and other peripherals[J]. IEEE Security and Privacy, 2006(9): 26-33.
- [9] 顾宝根,顾喜梅. 日志结构的嵌入式文件系统研究[J]. 计算机工程与设计, 2004, 25(6): 915-917.
GU Bao-gen, GU Xi-mei. Research for mechanism of journal file system used in embedded environment[J]. Computer Engineering and Design, 2004, 25(6): 915-917. (in Chinese)
- [5] 仇玉良. 公路隧道复杂通风网络分析技术研究[D]. 西安:长安大学, 2005.
QIU Yu-liang. The complicated ventilation network technology research of highway tunnel[D]. Xi'an: Chang'an University, 2005. (in Chinese)
- [6] 王立新. 长大公路隧道网络通风研究[J]. 现代隧道技术, 2008, 45(4): 76-82.
WANG Li-xin. Study on long highway tunnel network ventilation system[J]. Modern Tunneling Technology, 2008, 45(4): 76-82. (in Chinese)
- [7] 邵泽凡. 节点法网络通风计算程序研究[D]. 西安:长安大学, 2010.
SHAO Pan-fan. Study on the calculation program of network ventilation in node method[D]. Xi'an: Chang'an University, 2010. (in Chinese)