

长寿命星载 NAND Flash 自适应坏块管理策略

王文思^{1,2} 林宝军¹

(中国科学院光电研究院 北京 100094)¹ (中国科学院大学 北京 100190)²

摘要 针对长期在轨运行设备的存储可靠性问题,提出了一种自适应坏块管理策略。首先对星载 NAND Flash 存储系统建立马尔可夫可靠性模型,其次根据使用情况预估设备产生的坏块数量,并设置数据存储空间的大小;然后根据设备实际在轨坏块数量动态调整数据存储空间,在确保一定时间内有稳定的存储空间的基础上保持较高的空间利用率;最后对自适应坏块管理策略进行了仿真分析。结果表明,在某一写入速率下,该自适应坏块管理策略的空间利用率不低于 85%。

关键词 星载存储器, NAND Flash, 自适应, 坏块管理, 马尔可夫

中图分类号 TP302.7 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.10.036

Self-adaptive Management Strategy for Bad Blocks Based on Long Lifetime On-board NAND Flash

WANG Wen-si^{1,2} LIN Bao-jun¹

(Academy of Opto-Electronics, Chinese Academy of Sciences, Beijing 100094, China)¹

(University of Chinese Academy of Sciences, Beijing 100190, China)²

Abstract Aiming at the storage reliability problem on long-term on-orbit operation equipment, a self-adaptive management strategy for bad blocks was proposed. Firstly, a Markov reliability model is built for the on-board NAND Flash storage system. Secondly, according to the wear condition, the bad block numbers in the device are estimated, and the size of the data storage space is set. Then, based on the actual bad block numbers on orbit, the data storage space is adjusted dynamically, ensuring that a stable storage space is maintained and a higher utilization is achieved within a certain period of time. Finally, the simulation analysis of the self-adaptive management strategy is carried out. The results show that using the strategy at a certain write speed the space utilization rate of the device is not less than 85%.

Keywords On-board memory, NAND Flash, Self-adaptive, Bad block management, Markov

1 引言

NAND Flash 由于拥有高存储密度、低功耗、低成本、高数据吞吐量等优点而广泛用作大容量存储器件^[1]。为保证存储数据的可靠性,使用时必须考虑坏块的处理方法。文献[2]提出跳过坏块法,即构建映射表的逻辑地址和物理地址一一对应。文献[3,4]提出映射表只存储标志为好块或只存储标识为坏块的物理地址,大大缩小了映射表。文献[5]提出一种替换坏块法,将 NAND Flash 划分为数据区和预留区,数据区用来存放代码和数据,数据区的坏块映射到预留区中空闲的好块,被映射的预留区的好块称为替换块,以代替坏块进行操作。文献[6]提出一种动态坏块管理方法,引入交换块,每次对已用块的写入操作都是通过交换块来完成的,写入完成后,将旧块擦除,将其作为新的交换块,从而实现了损耗平衡,并且实现了数据的先写后擦,能防止意外掉电导致的数据丢失。

2 自适应坏块管理方法

2.1 问题的提出

在某应用场景中,卫星的使用寿命为 10 年, NAND Flash 作为卫星存储系统,要求达到 10 年的可靠性。应用程序对 NAND Flash 的操作为写满存储空间后,对存储空间进行全擦,新的数据覆盖历史数据,并要求在尽可能保证芯片较高利用率的情况下,坏块管理策略能提供一个地址连续、大小一定的存储空间。

针对以上需求,跳坏块法不能提供一个稳定的存储空间,不满足应用程序的要求。应用程序对 NAND Flash 的操作不存在擦写平衡的问题,采用动态坏块管理方法并没有利用到该方法的优点,反而由于每次擦写都要更新映射表而消耗大量的内存时间,且存储映射表的物理块擦写过于频繁更容易损坏。采用替换块法能为用户提供一个稳定的存储空间,但是在 NAND Flash 使用初期坏块数量较少,预留区有大量的

到稿日期:2015-09-26 返修日期:2016-02-06

王文思(1989—),女,硕士,主要研究方向为计算机应用技术、计算机应用系统可靠性设计, E-mail: wangws99@126.com; 林宝军(1963—),男,研究员,主要研究方向为计算机应用技术、空间飞行器总体设计。

有效块不能使用,空间利用率较低。

针对该应用场景提出一种自适应坏块管理方法,该方法是在替换块法的基础上,通过自适应调整数据区的大小实现的。根据坏块的增长数量,将数据区擦除一定的存储空间作为预留区。为方便映射,预留区按“从后往前”的顺序分配空闲的替换块,将每次更新过的映射表分别存储在3个好块中。该方法保证了 NAND Flash 在一定时间内有稳定的存储空间且能达到较高的空间利用率。

2.2 模型的建立

该方法首先要预估设备产生的坏块数量,从而在合适的时间调整数据区大小。目前国内并没有对 NAND Flash 进行建模估算的相关文献,本文利用马尔可夫可靠性模型对车载 NAND Flash 存储系统进行建模研究。首先作下述假设:

- (1)初始有 n 个有效块,每个块只能取好块和坏块两种状态。
- (2)每个块的状态转移率从好块变成坏块的概率为 λ ,而坏块不能转换成好块。
- (3)状态转移可在任一时刻进行,但在相当小的时间区间 Δt 内,不会发生两个以上的好块变为坏块,这个概率是一个高阶无穷小,可以忽略。

$$\begin{bmatrix} P_0'(t) \\ P_1'(t) \\ P_2'(t) \\ \vdots \\ P_{n-1}'(t) \\ P_n'(t) \end{bmatrix} = \begin{bmatrix} -n\lambda & 0 & 0 & \cdots & 0 & 0 \\ n\lambda & -(n-1)\lambda & 0 & 0 & \cdots & 0 \\ 0 & (n-1)\lambda & -(n-2)\lambda & 0 & 0 & \vdots \\ \vdots & 0 & 0 & \ddots & \ddots & 0 \\ 0 & \vdots & \cdots & 2\lambda & -\lambda & 0 \\ 0 & 0 & \cdots & 0 & \lambda & 0 \end{bmatrix} \begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ \vdots \\ P_{n-1}(t) \\ P_n(t) \end{bmatrix}$$

对以上微分方程组两端作拉普拉斯变换,并利用初始条件 $P_0(0)=[1 \ 0 \ 0 \ \cdots \ 0]^T$ 求解这些方程,可以得到各状态概率函数 $P_k(t)$, $P_k(t)$ 表示在 t 时刻有 k 个块坏掉的概率,经计算得:

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-n\lambda t} \quad (1)$$

参数 λ 是擦写次数的函数,由文献[7]的图1可得失效率与擦写次数的函数曲线。擦写次数是写入速率的函数,设某应用场景下的写入速率为 ω ,由该速率可求出特定时刻的擦写次数,再由文献[7]的失效率曲线得到该时刻的失效率,从而由式(1)求得该时刻产生一定坏块的概率。当概率达到1时,所对应的坏块数可认为是该时刻设备产生的坏块的数量。

2.3 数据区的划分

数据区可按坏块达到不同数量级的时刻来划分,设估算出的坏块数有 n 个不同的数量级 $\{N_1, N_2, \dots, N_n\}$, 对应的工作时长为 $\{t_1, t_2, \dots, t_n\}$, 则在每个工作时长下调整一次数据区的大小,保证在 $[0, t_1], [t_1, t_2], \dots, [t_{n-1}, t_n]$ 每个时间区间内有稳定的存储空间 $\{C_1, C_2, \dots, C_n\}$, 且保证较高的空间利用率。划分数据区时,预留区内的替换块数要满足 $t(t=t_1, t_2, \dots, t_n)$ 时刻到达时的坏块数量,为确保系统运行参数在合理范围内发生变化时,整个系统能够稳定运行,每次调整数据区时都设置一定裕度 $\{N_{d1}, N_{d2}, \dots, N_{dn}\}$ 。

设 NAND Flash 的所有块数为 N_c , 出厂固有坏块为 N_b , 每块的容量为 B , 则 $[0, t_1]$ 时间内系统能提供的有效块数为 $N_c - N_b - N_1 - N_{d1}$, 存储空间 $C_1 = (N_c - N_b - N_1 - N_{d1})B$ 。 $[0, t_1]$ 时间内的空间利用率为 $\frac{N_c - N_b - N_1 - N_{d1}}{N_c}$ 。同理可得

其它时间区间内系统提供的存储空间和该时间段内系统的空

NAND Flash 系统状态如图1所示。

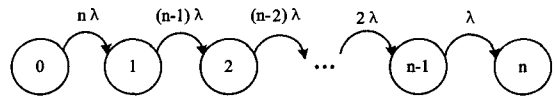


图1 NAND Flash 系统状态

设 k 在有限集合 $\{0, 1, 2, \dots, n\}$ 上取值。 k 状态表示 NAND Flash 产生了 k 个坏块, k 的所有取值为系统的所有状态。设系统在 t 时刻处于 k 状态的概率为 $P_k(t)$, 由状态图得, 在 Δt 时间内状态 k 转到状态 $k+1$ 的概率为 $P_k(t)k\lambda\Delta t$, 转到 $k-1$ 的概率为0, 则在 $t+\Delta t$ 时刻, 系统处于各状态的概率为:

$$P_0(t+\Delta t) = P_0(t)(1-n\lambda\Delta t)$$

$$P_1(t+\Delta t) = P_0(t)n\lambda\Delta t + P_1(t)[1-(n-1)\lambda\Delta t]$$

依次类推

⋮

$$P_k(t+\Delta t) = P_{k-1}(t)(n-k+1)\lambda\Delta t + P_k(t)[1-(n-k)\lambda\Delta t]$$

$\Delta t \rightarrow 0$ 时, 可得系统状态方程为:

间利用率。

在不同工作时长下, 系统提供的存储空间如图2所示。

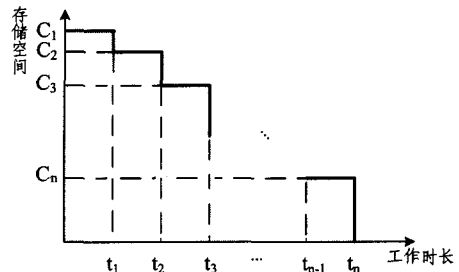


图2 不同工作时长下的存储空间

设备工作一段时间后, 可得到实际产生的坏块数, 坏块数和估算值会有一定的偏差, 数据区的划分要根据坏块的实际值作动态的调整, 分为3种情况来讨论。

(1)估算值与实际值偏差不大, 认为模型估算正确, 当工作时间到达 $t_k (k=1, 2, \dots, n)$ 后, 预留区大小逐渐不能满足产生的坏块数量, 此时需要根据下一个划分数据区大小的时刻 t_{k+1} 产生的坏块数来调整数据区的大小。设 t_{k+1} 时刻比 t_k 时刻增加了 N_z 个坏块, 则将用户的存储空间擦除 $(N_z + N_{dk})$ 个坏块大小的容量, 作为 t_k 时刻以后的预留区。

(2)当坏块数估算过低, 会导致预留区的替换块数不能满足坏块的需求。设 $t_0 (t_0 < t_1)$ 时刻的估算值为 N_0 , 而实际 t_0 已经达到坏块数 N_{max} , N_{max} 为 t_1 时刻划分的预留区的块数 ($N_{max} > N_T$), 则说明模型估算偏低, 需要调整模型估算参数。步骤如下:

1)估算 t_0 时刻的坏块数对应的失效率为 λ_0 。逐渐增大

λ_0 ,按该模型计算,直到得到一个失效率 λ_T ,能使 t_0 时刻的坏块数达到或接近 N_{max} 。

2)考虑 t_0 时刻估算值对应的 λ_0 和实际值对应的 λ_T 之间的函数关系,可由这种关系调整 t_1 时刻的失效率 λ_1 ,对 t_1 时刻的坏块数重新估算。

3)按重新估算的 t_1 时刻的坏块数重新设置数据区大小,使之不会发生还未到达 t_1 时刻预留块已不够坏块映射的情况。

(3)当估算值过高,会降低设备的空间利用率。当到达 t_1 时刻,实际产生的坏块数远小于估算值,则说明模型估算偏高,设备的空间利用率偏低,此时也需要调整模型估算参数,方法同上。

3 实例分析

该应用场景中所用 NAND Flash 大小为 4GB,擦写速率为 21kbps,则 1 年后每块的擦写次数约为 20 次,5 年后擦写次数约为 100 次。根据文献[7]图 1 得擦写次数和失效率的函数关系,从而可知,当擦写次数小于 100 次时,失效率很小,出现很少的坏块,不利于分析。从 5 年后开始分析,设备工作 5 年后擦写次数为 100 时,得到的失效率约为 $\lambda=10^{-5}$ 。ECC 校验能使失效率降低一个数量级,所以计算坏块产生概率必须考虑 ECC 校验,擦写次数为 100 时 $\lambda=10^{-6}$,由式(1)得设备工作 5 年后产生的坏块的概率。同理可估算出设备工作 6 年、8 年、10 年后产生坏块数的概率,如图 3 所示。

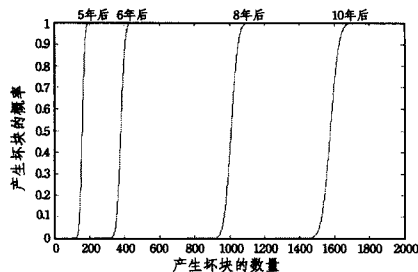


图 3 产生坏块的概率

由图 3 可知,卫星工作 5 年后产生 190 个坏块的概率达到 1,则可认为工作 5 年后约产生 190 个坏块,同理可得工作 6 年后到工作 10 年后产生的坏块的数量。表 1 给出了 NAND Flash 在擦写速率为 21kbps 时,卫星工作 5 年、6 年、8 年、10 年后对应的失效率和预计产生的坏块数量,估算值符合工程实践。

表 1 21kbps 下 NAND Flash 失效情况

	5 年后	6 年后	8 年后	10 年后
失效率	10^{-6}	0.2×10^{-5}	0.4×10^{-5}	0.5×10^{-5}
坏块数	190	420	1100	1690

在相同工作时长下,如果 NAND Flash 的擦写速率不同,则擦写次数不同,从而失效率函数不同。擦写速率越大,越容易产生坏块。这里计算出了 NAND Flash 擦写速率在 100kbps 和 1Mbps 下的失效情况,如表 2 和表 3 所列。

表 2 100kbps 下 NAND Flash 失效情况

	1 年后	2 年后	3 年后	4 年后
失效率	10^{-6}	0.5×10^{-5}	0.8×10^{-5}	10^{-5}
坏块数	35	320	800	1300

表 3 1Mbps 下 NAND Flash 失效情况

	1 年后	2 年后	3 年后	4 年后
失效率	10^{-5}	0.2×10^{-4}	0.4×10^{-4}	0.5×10^{-4}
坏块数	320	1300	3980	6500

据此可得工作时长和坏块数量的关系如图 4 所示,图 4 同时给出了擦写速率为 100kbps 和 1Mbps 时,工作时长和坏块数量的关系,以便以后分析不同写入速率下 NAND Flash 的使用寿命。

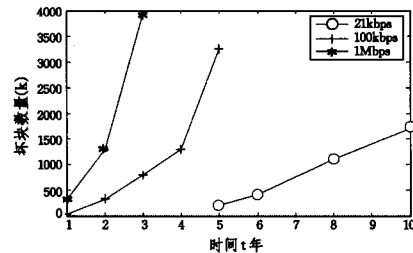


图 4 不同速率下坏块数量随时间的变化

由图 4 可知,NAND Flash 前 5 年内出现的坏块数量较少,前 5 年内的数据区大小按能保证工作 5 年的可靠性来计算,不进行变动。5 年后坏块数量才有显著增长,可根据坏块数量调整数据区的大小。

设置设备工作前 5 年数据区大小:1)使用 NAND Flash 前先检测固有坏块的数量,为 324 块,预留区首先要保证能满足 324 个初始坏块的占用空间;2)由仿真结果估算出 5 年后约产生 190 个坏块,预留区还要提供 190 个使用坏块的占用空间;3)设备工作前 5 年的预留区大小至少满足 514 个坏块的占用空间。设置预留区时要考虑一定裕度,适当增大坏块所需空间,故将预留区大小设为 600 块。本文所用 NAND Flash 共 16384 块,则数据区有 15784 块的存储空间,此时 NAND Flash 的空间利用率可达 96.34%以上。

当 5 年后预留区替换块差不多被映射完后,用户需要新的预留空间来映射使用坏块。由于预留区“从后往前”分配空闲的替换块,因此只需计算出后 5 年增加的使用坏块数,减少数据区相应的存储空间,使之用作预留区来替换设备工作后 5 年产生的坏块的数量。

设置设备工作后 5 年数据区大小:1)由模型估算出 10 年后约产生 1690 个坏块,比前 5 年增长了约 1500 个使用坏块,则减少的数据区空间至少满足 1500 个使用坏块的占用空间;2)将坏块裕度设为 300,即 5 年后用户将存储空间减去 1800 块作为预留区,则 5 年后用户数据区有 13984 块的存储空间,此时 NAND Flash 的空间利用率达到 85.35%。

结束语 本文通过对现有几种坏块管理策略进行分析,针对某特定的应用场景,综合考虑几种策略的优缺点,提出一种自适应坏块管理策略。并通过实例进行仿真,结果表明本文提出的自适应坏块管理策略在为用户提供稳定的存储空间的同时,可保证其空间利用率达到 85%以上,达到了应用程序在该应用场景下的需求。该方法主要针对该特定应用场景,在不同的应用场景下需要综合考虑更多问题,将自适应坏块管理策略应用在不同的应用场景中可作为进一步的研究内容。

参考文献

[1] Furano G, Fabiano M. NAND flash storage technology for mission-critical space applications[J]. IEEE Aerospace and Electronic Systems Magazine, 2013, 9: 30-36

带来的问题。特别是未登录词在当前查错文档中会出现多次,从而减少由于未登录词而带来的误判。

参 考 文 献

- [1] Kukich K. Techniques for automatically correcting words in text [J]. ACM Computing Surveys (CSUR), 1992, 24(4): 377-439
- [2] Zhang Yang-sen, Cao Yuan-da, Yu Shi-wen. A Hybrid Model of Combining Rule-based and Statistics-based Approaches for Automatic Detecting Errors in Chinese Text [J]. Journal of Chinese Information Processing, 2006, 20: 1-7 (in Chinese)
张仰森, 曹元大, 俞士汶. 基于规则与统计相结合的中文文本自动查错模型与算法 [J]. 中文信息学报, 2006, 20: 1-7
- [3] Shi De-sheng, Wang Liang-zhi, Chen Zhi-da, et al. A Statistics-based Approach for Automatic Detecting Errors in Chinese Text [J]. Computer and Communications, 1992, 8: 19-26 (in Chinese)
施得胜, 王良志, 陈志达, 等. 基于统计的中文错字侦测法 [J]. 电脑与通讯, 1992, 8: 19-26
- [4] Zhang Zhao-huang. Automatic Error Detection and Correction of Chinese Text [J]. Communications of COLIPS, 1994, 4(2): 143-149 (in Chinese)
张照煌. 中文错别字自动订正方法初探 [J]. Communications of COLIPS, 1994, 4(2): 143-149
- [5] Zhang L, Zhou M, Huang C, et al. Multifeature-based approach to automatic error detection and correction of Chinese text [C] // Proceedings of the First Workshop on Natural Language Processing and Neural Networks, 2000: 2744-2748
- [6] Ma Jin-shan, Zhang Yu, Liu Ting, et al. Detecting Chinese Text Errors Based on Trigram and Dependency Parsing [J]. Journal of the China Society for Scientific and Technical Information, 2005, 23(6): 723-728 (in Chinese)
马金山, 张宇, 刘挺, 等. 利用三元模型及依存分析查找中文文本错误 [J]. 情报学报, 2005, 23(6): 723-728
- [7] Zhang Yang-sen, Cao Yuan-da, Yu Shi-wen. A Hybrid Model of Combining Rule-based and Statistics-based Approaches for Automatic Detecting Errors in Chinese Text [J]. Journal of Chinese Information Processing, 2006, 20(4): 1-7 (in Chinese)
- [8] Zhang Yang-sen, Ding Bing-qing. Automatic Errors Detecting of Chinese Texts Based on the Bi-neighborship [J]. Journal of Chinese Information Processing, 2001, 15(3): 36-43 (in Chinese)
张仰森, 丁冰青. 基于二元接续关系检查的字词级自动查错方法 [J]. 中文信息学报, 2001, 15(3): 36-43
- [9] Zhang Lei, Sun Mao-song, Zhou Ming, et al. Automatic Chinese Text Error Correction Approach Based on Fast Approximate Chinese Word-Matching Algorithm [C] // Proceeding of the 3rd World Congress on Intelligent Control and Automation, 2000: 2739-2743 (in Chinese)
张磊, 孙茂松, 周明, 等. 基于快速模糊词匹配算法的中文自动校对方法 [C] // 第三届全球智能控制与自动化大会, 2000: 2739-2743
- [10] Liu Liang-liang, Wang Shi, Wang Dong-sheng, et al. Automatic Text Error Detection in Domain Question Answering [J]. Journal of Chinese Information Processing, 2013, 27(3): 77-83 (in Chinese)
刘亮亮, 王石, 王东升, 等. 领域问答系统中的文本错误自动发现方法 [J]. 中文信息学报, 2013, 27(3): 77-83
- [11] 朱德熙. 语法讲义 [M]. 商务印书馆, 1982
- [12] Zhang Hua-ping, Liu Qun. Model of Chinese Words Rough Segmentation Based on N-Shortest-Paths Method [J]. Journal of Chinese Information Processing, 2002, 16(5): 1-7 (in Chinese)
张华平, 刘群. 基于 N-最短路径方法的中文词语粗分模型 [J]. 中文信息学报, 2002, 16(5): 1-7
- [13] 丰泽洋, 曹存根. 语音查询中的辨音方法 [M]. Google Patents, CN Patent App. CN02, 160, 272
- [14] 王石, 王卫民, 符建辉. 一种汉字字形认知相似度计算方法 [M]. Google Patents, 28. CN Patent App. CN 201, 110, 205, 807
- [15] Wang Si-li, Zhang Hua-ping, Wang Bin. Research of Optimization on Double-Array Trie and its Application [J]. Journal of Chinese Information Processing, 2006, 20: 24-30 (in Chinese)
王思力, 张华平, 王斌. 双数组 Trie 树算法优化及其应用研究 [J]. 中文信息学报, 2006, 20: 24-30
-
- (上接第 195 页)
- [2] Liu Rui. Design of Imaging Sonar Data Acquisition and Storage System Based on NAND Flash [D]. Harbin: Harbin Engineering University, 2012 (in Chinese)
柳睿. 基于 Nand Flash 的图像声纳数据采集存储系统设计 [D]. 哈尔滨: 哈尔滨工程大学, 2012
- [3] Shu Wen-li, Wu Yun-feng, Zhao Qi-yi, et al. Bad Block Management Method of NAND FLASH Memory [J]. Chinese Journal of Election Devices, 2011, 5: 580-583 (in Chinese)
舒文丽, 吴云峰, 赵启义, 等. NAND Flash 存储的坏块管理方法 [J]. 电子器件, 2011, 5: 580-583
- [4] Zhang Sheng-yong, Gao Shi-jie, Wu Zhi-yong, et al. Bad Block Handle Method of NAND Flash Memory based on FPGA [J]. Computer Engineering, 2010, 6: 239-240, 243 (in Chinese)
张胜勇, 高世杰, 吴志勇, 等. 基于 FPGA 的 NAND Flash 坏块处理方法 [J]. 计算机工程, 2010, 6: 239-240, 243
- [5] Li You-meng, Li Qing-cheng, Gong Xiao-li. Research and implementation of NAND-FLASH bad block management algorithm for FTL layer [A] // Proceedings of 2010 International Conference on Services Science, Management and Engineering (Volume 2) [C]. Civil Aviation University of China, 2010
- [6] Han Yong-hao, Wang Shao-yun. Design and implementation of a NAND Flash dynamic bad block management algorithm [J]. Informatization Research, 2011, 37(3): 23-26 (in Chinese)
韩勇豪, 王少云. 一种 NAND Flash 动态坏块管理算法的设计与实现 [J]. 信息化研究, 2011, 37(3): 23-26
- [7] Lara D, Eitan Y, Ryan G. Graded Bit-Error-Correcting Codes With Applications to Flash Memory [J]. IEEE Transactions on Information Theory, 2012, 4: 2315-2327
- [8] Hu Ning, Yang Qing, Wang Dong. NandFlash Memory Management Based on Blocks Set [J]. Microelectronics and Computer, 2015, 32(3): 19-22 (in Chinese)
胡宁, 杨琼, 王冬. 基于分组的 NandFlash 块管理方法 [J]. 微电子学与计算机, 2015, 32(3): 19-22
- [9] Zhao Zhi-cao. Reliability Analysis and Optimization Design of Load-sharing Redundant System [D]. Xi'an Northwestern Polytechnical University, 2015 (in Chinese)
赵志草. 共载冗余系统可靠性分析与优化设计 [D]. 西安: 西北工业大学, 2015
- [10] Gregory F. Lawler. 随机过程导论 (第 2 版) [M]. 张景肖, 译. 北京: 机械工业出版社, 2010: 48-61
- [11] 邹逢兴, 张相平, 龙志强, 等. 计算机应用系统的故障诊断与可靠性技术基础 [M]. 中国水利水电出版社, 2011: 342-348