

# 一种适用于 MLC 闪存的 CCI 噪声均衡化算法

张旋<sup>1,2</sup> 周乐<sup>2</sup> 侯爱华<sup>1</sup>

(西安理工大学 西安 710082)<sup>1</sup> (西安电子科技大学 西安 710071)<sup>2</sup>

**摘要** 随着多级单元(Multi-Level Cell, MLC)闪存存储密度的增加,单元间干扰(Cell-to-Cell Interference, CCI)成为影响 NAND 闪存可靠性的主要噪声。在深入研究 MLC 闪存模型和 CCI 噪声模型的基础上,提出了一种 MLC 闪存的 CCI 噪声均衡化算法。该算法通过估计 CCI 干扰强度进而对感知 MLC 阈值电压进行补偿,可以更准确地读取 MLC 单元中存储的信息。仿真结果表明,在 MLC 闪存信道条件下,CCI 噪声均衡化算法可以有效减少相邻状态的阈值电压交叉现象,有助于降低原始比特错误率,增强 MLC 闪存的可靠性。

**关键词** 多级单元,单元间干扰,均衡化

**中图分类号** TP301.6 **文献标识码** A

## CCI Noise Equalization Algorithm for MLC Flash Memory

ZHANG Xuan<sup>1,2</sup> ZHOU Le<sup>2</sup> HOU Ai-hua<sup>1</sup>

(Xi'an University of Technology, Xi'an 710082, China)<sup>1</sup> (Xidian University, Xi'an 710071, China)<sup>2</sup>

**Abstract** With the increase of MLC (Multi-Level Cell) flash memory density, CCI (Cell-to-Cell Interference) is the dominant noise source which affects the reliability of NAND flash memory. On the research of MLC flash memory model and CCI noise model, an equalization algorithm of CCI noise was proposed for MLC flash memory. This method compensates the sensed threshold voltage of MLC flash memory by estimating the CCI interference, so it is more accurate to read the information stored in MLC. The simulation results show that the CCI noise equalization algorithm can reduce the overlap of the adjacent threshold voltage distribution, which help to reduce the raw bit error rate and enhance the reliability of flash memory.

**Keywords** Multi-level cell, Cell-to-cell interference, Equalization

## 1 引用

近年来,以闪速存储器(Flash Memory)为代表的非易失性存储器具有存取速度快、存储容量大、功耗低、体积小、防震抗摔、耐久性等优点,已成为现代存储系统中一种重要的存储介质<sup>[1-2]</sup>。随着制造工艺的不断提升,闪存芯片封装尺寸不断缩小,存储密度正以摩尔定律的速度增长。一方面,闪存的单位比特成本不断降低;另一方面,闪存单元间距越来越小,浮栅中可存储的电子变少,相邻电位窗口间隔变窄,相邻单元之间的寄生耦合电容效应变大,导致单元间干扰(Cell-to-Cell Interference, CCI)噪声的影响越来越大,闪存存储系统的可靠性不断下降<sup>[3-6]</sup>。

随着工业技术的不断革新,多级闪存(Multi-Level Cell, MLC)进一步提升了存储密度,同时由单元间寄生耦合电容所引发的闪存存储信道 CCI 噪声导致闪存系统的可靠性急剧降低,因此借助有效的均衡化算法可以提高 MLC 闪存的可靠性<sup>[7]</sup>。文献<sup>[7]</sup>提出一种后补偿的均衡算法,该算法在存在严重 CCI 干扰的场景下,消除 CCI 噪声的效果不佳。本文提出一种适用于 MLC 闪存的 CCI 噪声均衡化算法,该算法从闪存块中不受 CCI 干扰的最后一页开始向第一页逐页进行 CCI 均衡化处理,均衡化之后得到的 MLC 阈值电压的准

确性更高,因此均衡化算法可以有效减少 MLC 相邻状态的阈值电压交叉现象,有助于降低原始比特错误率(Raw Bit Error Rate, RBER),增强 MLC 的可靠性。

## 2 MLC 型 NAND 闪存

MLC 单元是一个带有浮栅(Floating Gate)的晶体管,每个存储单元有两个栅极:1)在顶部的控制栅(Control Gate);2)下方被隧道氧化绝缘层包围的浮栅(Floating Gate),进入浮栅的电荷会困于其中。单级单元(Single-Level Cell, SLC)通过浮栅中电荷有无来表示逻辑值 0 或 1(见图 1(a)),从而实现 1 bit 数据的存储。而 MLC 闪存浮栅通过存储 4 种不同的阈值电压来实现 2 bit 数据的存储( $s_0 = 11, s_1 = 01, s_2 = 00, s_3 = 10$ )(见图 1(b))<sup>[5]</sup>。



图 1 闪存单元结构示意图

本文受国家自然科学基金(61271004)资助。

张旋(1980—),男,硕士,讲师,主要研究方向为差错控制编码技术、大数据系统中数据的可靠性技术, E-mail: zhangyic2014@163.com; 周乐(1990—),男,硕士生,主要研究方向为差错控制编码技术、大数据系统中数据的可靠性技术, E-mail: 66877905@qq.com; 侯爱华(1969—),女,副教授,主要研究方向为应用数学、空间统计等, E-mail: airrs2015@sina.com。

## 2.1 MLC 阈值电压模型

NAND 闪存的组织结构依次是: 阵列 (Array)、块 (Block)、页 (Page) 和闪存单元。大容量的闪存阵列由多个块组成, 每个块由多个页构成, 页是闪存编程和读取的最小单位。\$N\_b\$ 表示存储单元存储的比特数量, 则每个存储单元有 \$K=2^{N\_b}\$ 个存储状态。用 \$p^{(k)}(x)\$ 表示第 \$k\$ 个存储状态阈值电压的概率密度函数, \$0 \le k \le K-1\$。对于 MLC 闪存, \$N\_b=2\$, \$K=2^2=4\$。在编程之前, MLC 闪存单元的闪存块必须处于擦除状态, 擦除状态阈值电压随机变量 \$X\$ 的概率密度函数如式(1)所示<sup>[5,8-10]</sup>:

$$p_e(x) = p^{(0)}(x) = \frac{1}{\sigma_e \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_e)^2}{2\sigma_e^2}\right) \quad (1)$$

其中, \$\mu\_e\$ 和 \$\sigma\_e\$ 是擦除状态的均值和标准方差。

在闪存单元的编程过程中, 需利用增量步长脉冲程序 (Incremental-Step-Pulse-Program, ISPP) 严格控制电荷的注入量<sup>[11]</sup>。如图 2 所示, 每一次编程操作之后, 闪存单元的阈值电压增加 \$\Delta V\_{pp}\$, 不断逼近校验电压 \$V\_p\$。编程状态的阈值电压趋于 \$[V\_p, V\_p + \Delta V\_{pp}]\$ 上的均匀分布。第 \$k\$ 个 \$p^{(k)}(x)\$ (\$1 \le k \le K-1\$) 编程状态的 MLC 闪存单元阈值电压分布理想的数学模型可近似为均匀分布, 如式(2)所示<sup>[5,8-10]</sup>:

$$p_p(x) = p^{(k)}(x) = \begin{cases} \frac{1}{\Delta V_{pp}}, & \text{if } V_p \le x \le V_p + \Delta V_{pp} \\ 0, & \text{else} \end{cases} \quad (2)$$

其中, \$V\_p\$ 是写的校验电压, \$\Delta V\_{pp}\$ 为编程的步长电压增量。\$V\_p\$ 和 \$V\_p + \Delta V\_{pp}\$ 表示为 \$V\_l^{(k)}\$ 和 \$V\_r^{(k)}\$。

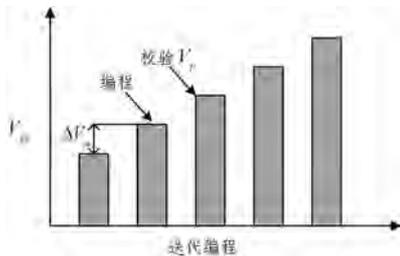


图2 编程/校验编程操作

MLC 闪存信道仿真参数的设置如下: 擦除状态阈值电压分布概率密度函数 \$p\_e(x)\$ 的均值 \$\mu\_e=1.4\$; 标准差 \$\sigma\_e=0.4\$; 迭代编程步长电压增量 \$\Delta V\_{pp}=0.3\text{V}\$; 3 个编程状态的校验电压 \$V\_p\$ 分别为 \$2.8\text{V}\$、\$3.4\text{V}\$ 和 \$4.0\text{V}\$。对 10 个闪存块, 每块 32 页, 每页 17260 个 MLC 进行 100 次蒙特卡洛模拟仿真。在无 CCI 噪声干扰的理想情况下, MLC 闪存经过擦除和编程操作之后, 得到闪存单元阈值电压分布的统计直方图, 如图 3 所示。

$$p^{(0)}(x) = p_e(x) \otimes p_c(x) = \int_t p_e(t) p_c(x-t) dt = \frac{1}{2\sqrt{2\pi}\sigma_e\gamma_y K\Delta V_{pp}} \int_t e^{-\frac{(t-\mu_e)^2}{2\sigma_e^2}} \times \left( \sum_{v=1}^{K-1} \left( \text{erf}\left(\frac{V_r^{(v)} - \frac{x-t}{\gamma_y} - \mu_e}{\sqrt{2}\sigma_e}\right) - \text{erf}\left(\frac{V_l^{(v)} - \frac{x-t}{\gamma_y} - \mu_e}{\sqrt{2}\sigma_e}\right) \right) + 2\gamma_y\Delta V_{pp}\delta(x-t) \right) dt \quad (6)$$

如果被干扰单元被编程为第 \$k\$ 个编程状态, 那么在受到

$$p^{(k)}(x) = p_p^{(k)}(x) \otimes p_c(x) = \int_t p_p^{(k)}(t) p_c(x-t) dt = \int_{V_l^{(k)}}^{V_r^{(k)}} \left( \frac{1}{2K\Delta V_{pp}^2} \times \sum_{v=1}^{k-1} \left( \text{erf}\left(\frac{V_r^{(v)} - \frac{x-t}{\gamma_y} - \mu_e}{\sqrt{2}\sigma_e}\right) - \text{erf}\left(\frac{V_l^{(v)} - \frac{x-t}{\gamma_y} - \mu_e}{\sqrt{2}\sigma_e}\right) \right) + \frac{\delta(x-t)}{K\Delta V_{pp}} \right) dt \quad (7)$$

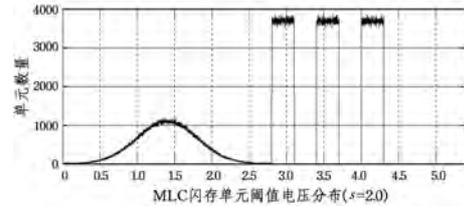


图3 理想情况下的 MLC 阈值电压统计直方图

## 2.2 单元间干扰噪声模型的建立

MLC 单元由于编程操作使其阈值电压变化, 这种电压的变化会在相邻的 MLC 单元之间形成寄生耦合电容 (Parasitic Capacitance-Coupling), 从而使邻近单元的阈值电压增加, 这种现象称为单元间干扰 (Cell-to-Cell Interference, CCI)<sup>[12]</sup>。随着半导体生产工艺的不断发展, CCI 已经成为影响 NAND 闪存可靠性的一个主要噪声源。

MLC 存储单元 CCI 噪声的概率密度函数为 \$p\_c(x)\$。设一个存储单元受到了相邻字线 \$N\$ 个存储单元的干扰, \$x\_n\$ 和 \$y\_n\$ 表示第 \$n\$ 个干扰单元 (Interfering Cell) 在编程前和编程后的阈值电压, \$\gamma\_n\$ 表示第 \$n\$ 个干扰单元和被干扰单元 (Victim Cell) 之间的电容耦合系数, 那么该存储单元受到相邻存储单元的 CCI 噪声干扰的大小如式(3)所示<sup>[5]</sup>:

$$F = \sum_n (\gamma_n \cdot \Delta V_n) = \sum_{n=0}^{N-1} \gamma_n (y_n - x_n) = \sum_{n=0}^{N-1} \gamma_n y_n - \sum_{n=0}^{N-1} \gamma_n x_n \quad (3)$$

由于擦除状态的阈值电压服从高斯分布, 因此所有 \$x\_n\$ 的联合分布也服从高斯随机变量 \$x'\$, 即 \$x' = \sum\_{n=0}^{N-1} \gamma\_n x\_n\$ 服从 \$N(\mu', \sigma')\$, 其中 \$\mu'\$ 和 \$\sigma'\$ 如式(4)所示, 用 \$p\_{x'}(x)\$ 表示高斯变量 \$x'\$ 的概率密度函数。

$$\mu' = \sum_{n=0}^{N-1} \gamma_n \mu_e, \sigma' = \left( \sum_{n=0}^{N-1} \gamma_n \sigma_e^2 \right)^{1/2} \quad (4)$$

当一个存储单元被编程后, 它的阈值电压服从均匀分布, 因此 \$\gamma\_n y\_n\$ 仍为均匀分布。假定所有 \$y\_n\$ 的联合分布服从均匀分布 \$y'\$, \$y' = \sum\_{n=0}^{N-1} \gamma\_n y\_n\$, 其概率密度函数为 \$p\_{y'}(y)\$。在单元间干扰强度为 \$F = y' - x'\$ 的情况下, 可以估计存储单元间干扰的概率密度函数, 如式(5)所示:

$$p_c(x) = \int_t p_{y'}(x+t) p_{x'}(t) dt \quad (5)$$

通过式(6)和式(7)可以计算 CCI 干扰后 MLC 闪存单元阈值电压的概率分布。若受干扰单元保持擦除状态, 在存储单元之间发生 CCI 干扰之后, 它的阈值电压分布如式(6)所示 \$\otimes\$ 表示卷积运算, \$\delta(x)\$ 表示 Dirac delta 函数<sup>[5]</sup>:

CCI 干扰后, 它的阈值电压分布如式(7)所示<sup>[5]</sup>:

其中,  $erf(x)$  记作错误函数,  $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ 。

$s$  表示电容耦合强度因子, 耦合强度因子  $s$  的大小表示 CCI 单元间干扰的强弱。如图 4 所示, 在全位线结构中, 将受干扰闪存单元与垂直方向上方干扰单元之间的电容耦合率记为  $\gamma_y$ , 受干扰单元与两侧斜上方干扰单元之间的电容耦合系数记为  $\gamma_{xy}$ 。垂直方向电容耦合系数  $\gamma_y = 0.08s$ , 对角线方向的电容耦合系数  $\gamma_{xy} = 0.006s$ , 通过蒙特卡罗 (Monte Carlo) 统计模拟方法可得 CCI 噪声干扰后的统计直方图, 如图 5 所示<sup>[5]</sup>。

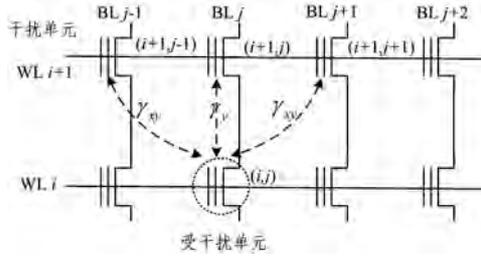


图 4 CCI 干扰示意图

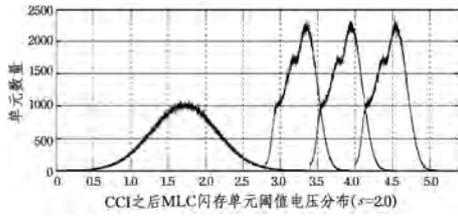


图 5 CCI 噪声干扰后的 MLC 阈值电压统计直方图

由图 5 分析可得, MLC 闪存存储信道在加入 CCI 噪声后, MLC 闪存单元阈值电压发生波动, 相邻的阈值电压分布出现重叠区域, 位于此区域的 MLC 单元在判决时的出错概率较高, 进而导致错误读取存储的数据, 这就是 MLC 闪存发生存储错误的原因。

### 3 MLC 闪存的 CCI 均衡化算法

#### 3.1 MLC 闪存的 CCI 均衡化思路

通过对式(3)的深入分析可知, 受干扰单元阈值电压的变化值  $\tilde{F}$  与到相邻单元编程前后阈值电压的变化量相关。本文 MLC 闪存的 CCI 均衡化思路是通过估计闪存单元编程前后的电压变化量来补偿受干扰单元的阈值电压, 进而削弱 CCI 对其的影响。具体的方法是读取一个受 CCI 干扰的单元的阈值电压, 在读出的阈值电压的基础上减去估计 CCI 影响的变化值  $\tilde{F}$ , 进而可以得到更逼近正确的阈值电压。假定第  $k$  个干扰单元感知得到的阈值电压为  $\tilde{V}_i^{(k)}$ , 其擦除状态阈值电压的期望值为  $\mu_e$  (第  $k$  个干扰单元在编程之前阈值电压值的估计), 则第  $k$  个受干扰单元的阈值电压变化量  $\Delta \tilde{V}_i^{(k)}$  可估计为  $\tilde{V}_i^{(k)} - \mu_e$ 。因此, 可以使用式(8)来估计单元间的干扰强度:

$$\tilde{F} = \sum_k (\gamma^{(k)} \cdot \Delta \tilde{V}_i^{(k)}) = \sum_k (\gamma^{(k)} \cdot (\tilde{V}_i^{(k)} - \mu_e)) \quad (8)$$

#### 3.2 MLC 闪存的 CCI 均衡化算法

由于闪存单元之间存在寄生耦合电容, 正在编程的闪存

页会对其上一页闪存单元的阈值电压产生影响进而改变存储的数据。本文设计的 MLC 闪存 CCI 均衡化算法从闪存块中的最后一页到第一页逐页进行 CCI 均衡化(见算法 1)。其中, 闪存块最后一页不会受到其他页 CCI 的影响。该算法的优点是均衡化式(8)中用到的阈值电压  $\tilde{V}_i^{(k)}$  是已进行 CCI 均衡化后的结果, 即读取的阈值电压  $\tilde{V}_i^{(k)}$  已经减去了 CCI 对其的阈值电压变化估计量  $\Delta \tilde{V}_i^{(k)}$ , 因此使用  $\tilde{V}_i^{(k)}$  更加准确, 进而式(8)估计的阈值电压变化量  $\Delta \tilde{V}_i^{(k)}$  更加符合实际的情况, 均衡化之后得到的 MLC 阈值电压的准确性更高。

#### 算法 1 MLC 闪存信道的 CCI 均衡化算法

参数说明:

delta\_Vth[i]: MLC 编程前后阈值电压变化估计量

cell\_Vth[j][i]: 感知得到的 MLC 阈值电压

$\tilde{F}$ : 估计受干扰单元的电压变化量

$\gamma_{xy}, \gamma_y$ : 对角线和垂直方向的电容耦合率

$\mu_e$ : 擦除状态的闪存单元阈值电压的期望值

for j=(32, ..., 1) do //从闪存块最后一页开始

//Block 最后一页不受 CCI 干扰, 计算 Cell 的电压变化量

if j=32 //从闪存块最后一页

//计算当前页中 Cell 估计的电压变化量(式(8))

for i=(1, ..., 17260) do

delta\_Vth[i]=cell\_Vth[j][i]- $\mu_e$

end for

else

//一页中所有 MLC 单元逐个进行均衡化处理

for i=(1, ..., 17260) do

//估计闪存页中每个 Cell 受到 CCI 干扰的强度

$$\tilde{F} = \gamma_{xy} \times \text{delta\_Vth}[i-1] + \gamma_y \times \text{delta\_Vth}[i] + \gamma_{xy} \times \text{delta\_Vth}[i+1]$$

//均衡化 CCI, 得到更加准确的 cell\_Vth[j][i]

$$\text{cell\_Vth}[j][i] = \text{cell\_Vth}[j][i] - \tilde{F}$$

end for

//计算当前页中 Cell 估计的电压变化量(式(8))

for i=(1, ..., 17260) do

$$\text{delta\_Vth}[i] = \text{cell\_Vth}[j][i] - \mu_e$$

end for

end if

end for

### 4 仿真实验及分析

MLC 闪存仿真参数的设置如表 1 所列, 信道仿真参数设置与 2.1 节中的设置相同。通过对比图 6 和图 7 可知, MLC 闪存单元相邻状态的阈值电压分布交叉现象会随着电容耦合强度因子  $s$  的增大而更加严重。在闪存单元受到 CCI 噪声之后, 通过使用均衡化算法, 可以有效减少相邻状态的阈值电压交叉现象, 原因是使用均衡化算法可以对受干扰闪存单元的阈值电压进行有效的补偿, 进而提高读取 MLC 单元阈值电压的精度。本文均衡化算法是从闪存块中未遭受到 CCI 干扰的最后一页开始向第一页逐页进行 CCI 均衡化, 进而使估计出的阈值电压变化量更加符合实际情况, 均衡化之后得到的 MLC 阈值电压的准确性更高, 因此均衡化算法增强了 MLC 闪存的可靠性。

表1 MLC 闪存仿真参数

闪存结构	参数
Block	32Pages/Block
Page	17260Cells/Page
NAND 闪存类型	MLC(2bits/Cell)

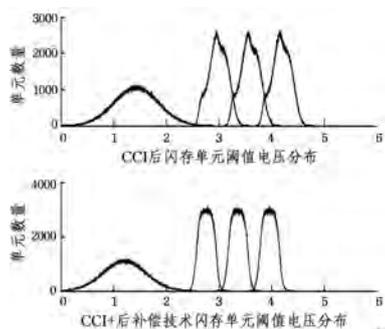


图6  $s=0.8$  时闪存模型阈值电压分布直方图统计

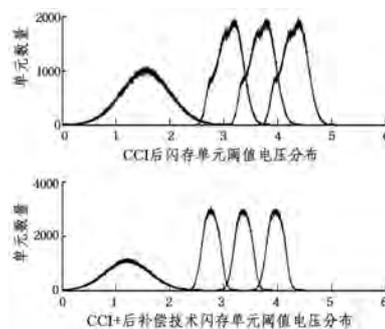


图7  $s=1.4$  时闪存模型阈值电压分布直方图统计

图8给出了 MLC 闪存单元阈值电压在均衡化前后的误比特率 BER 的性能比较,均衡化算法可以有效降低闪存单元中的原始比特错误率。随着耦合强度因子  $s$  的减小,CCI 干扰强度变弱,与 RBER 相比,均衡化算法的性能得到提升。

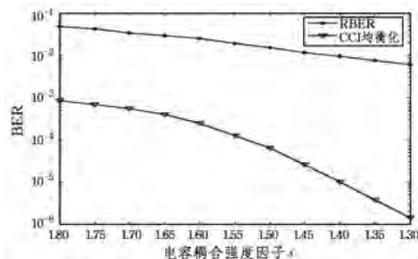


图8 MLC 闪存单元在均衡化前后的 BER 性能

**结束语** 针对 CCI 成为影响 MLC 型 NAND 闪存可靠性的主要因素,提出了一种 MLC 闪存的 CCI 噪声均衡化算法。该算法对感知后的 MLC 闪存单元阈值电压进行有效补

偿,提高了 MLC 闪存单元阈值电压的准确度。CCI 均衡化补偿算法的使用,可以有效减少相邻状态的阈值电压交叉现象,有助于降低原始比特错误率,提高 MLC 闪存的可靠性。

参考文献

- [1] TAKEUCHI K. NAND flash application and solution[J]. IEEE Solid-State Circuits Magazine,2013,5(4):34-40.
- [2] DEAN K. The history of semiconductor memory:from magnetic tape to NAND flash memory [J]. IEEE Solid-State Circuits Magazine,2016,8(2):16-22.
- [3] LIU R,CHUANG M,YANG C,et al. Improving read performance of NAND flash SSDs by exploiting error locality [J]. IEEE Transactions on Computers,2016,64(4):1090-1102.
- [4] SALA F,IMMINK K,DOLECEK L. Error control schemes for modern flash memories solutions for flash deficiencies[J]. IEEE Consumer Electronics Magazine,2015,4(1):66-73.
- [5] DONG G,XIE N,ZHANG T. On the use of soft-decision error-correction codes in NAND flash memories [J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2011, 58(2): 429-439.
- [6] CHAUDHRY A,GUANG Y,CAI K. Detector for MLC NAND flash memory using neighbor-a-priori information [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2016,24(9):2827-2836.
- [7] DONG G,LI S,ZHANG T. Using data postcompensation and predistortion to tolerate cell-to-cell interference in MLC NAND flash memory[J]. IEEE Transactions on Circuits and Systems I: RegularPapers,2010,57(10):2718-2728.
- [8] PARNELL T,DÜNNER C,MITTELHOLZER T, et al. Capacity of the MLC NAND flash channel[J]. IEEE Journal on Selected Areas in Communications,2016,34(9):2354-2365.
- [9] XU Q,GONG P,CHEN T, et al. Modelling and characterization of NAND flash memory channels[J]. Measurement, 2015, 70: 225-231.
- [10] CHAUDHRY A,GUANG Y,CAI K. Read and write voltage signal optimization for multi-level-cell (MLC) NAND flash memory [J]. IEEE Transactions on Communications, 2016, 64(4):1613-1623.
- [11] SUH K,SUH B,LIM Y, et al. A 3.3V 32Mb NAND flash memory with incremental step pulseprogramming scheme[J]. IEEE Journal of Solid-State Circuits,1995,30(11):1149-1156.
- [12] PARK K,KANG M,KIM D, et al. A zeroing cell-to-cell interference page architecture with temporary LSB storing and parallel MSB program scheme for MLC NAND flash memories[J]. IEEE Journal of Solid-State Circuits,2008,43(4):919-928.

(上接第 540 页)

- [3] HOLTZMAN G. The SPIN Model Checker,Primer and Reference Manual[M]. Addison-Wesley Professional,2001.
- [4] CHAN W, ANDERSON R J, BEAME P, et al. Model checking large software specifications[J]. IEEE Transactions on Software Engineering, 1998,24(7):498-520.
- [5] HOLZMANN G J. The model checker SPIN[J]. IEEE Transactions on Software Engineering,1997,23(5):279-295.
- [6] HOLZMANN G J,SMITH M H. A practical method for verifying event-driven software[C]// Proceedings of the 21st Inter-

national Conference on Software Engineering. ACM,1999:597-607.

- [7] 王大伟,张大方,缪力. 一种自动化模型检测 ANSI-C 程序的实用方法[J]. 计算机工程与科学,2010,32(4):79-82.
- [8] 郭伟,缪力,张大方,等. 基于 Spin 的 UML 状态图模型检查的设计与实现[J]. 计算机工程与应用,2008,44(10):43-47.
- [9] 高晓星,李晓霞,薛冰. 基于 UML 和 SPIN 的软件安全模型验证[J]. 长沙大学学报,2013,27(5):69-71.
- [10] 舒良春,饶俊,肖美华,等. 基于 Promela 的 UML 建模方法及其应用[J]. 计算机与现代化,2010,2010(2):101-104.