

引用格式: 乌勒, 刘武, 洪亮. 基于 SONOS 型嵌入式 flash 存储器的扰动失效测试算法研究[J]. 微电子学与计算机, 2021, 38(5): 7-13. [WU L, LIU W, HONG L. Investigation on disturbance characteristics and test algorithm of SONOS embedded flash memory[J]. Microelectronics & Computer, 2021, 38(5): 7-13.]

基于 SONOS 型嵌入式 flash 存储器的扰动失效测试算法研究

乌 勒^{1,2}, 刘 武¹, 洪 亮³

(1 上海交通大学 微米/纳米加工技术国家级重点实验室, 上海 200240;

2 上海交通大学 电子信息与电气工程学院微纳电子学系, 上海 200240;

3 上海华力微电子有限公司, 上海 201314)

摘 要: 目前 flash 扰动失效测试算法无法检测 SONOS 型 flash 存储器的全部扰动失效, 特别是对读扰动失效检测的失效覆盖率低, 且测试效率不高. 针对这一问题, 本文对 March-FT 测试算法进行扩展和优化, 提出了一种用于检测 SONOS 型 flash 存储器的扰动失效测试算法——March-SONOS. 结合算法评估系统验证, March-SONOS 测试算法可检测 SONOS 型 flash 存储器的全部扰动失效, 且失效覆盖率达到 100%, 同时算法测试效率提高 55.97%.

关键词: SONOS; 扰动失效模型; March-SONOS 算法; 算法评估系统

中图分类号: TN407

文献标识码: A

文章编号: 1000—7180(2021)05—0007—07

Investigation on disturbance characteristics and test algorithm of SONOS embedded flash memory

WU Le^{1,2}, LIU Wu¹, HONG Liang³

(1 National Key Laboratory of Science and Technology on Micro/Nano Fabrication,
Shanghai Jiao Tong University, Shanghai 200240, China;

2 Department of Micro/Nano Electronics, School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China;

3 Shanghai Huali Micro Electronics Co., LTD., Shanghai 201314, China)

Abstract: The current Flash disturbance failure test algorithm cannot detect all the disturbance failures of SONOS-type Flash memory, especially the failure coverage of read disturbance failure test is low, and the test efficiency is poor. To solve this problem, this paper extends and optimises the March-FT test algorithm, and put forward the March-SONOS test algorithm, which is used to detect the disturbance failure test algorithm of SONOS-type Flash memory. Combined with the verification of the algorithm evaluation system, the March-SONOS test algorithm can detect all disturbance failures of SONOS-style Flash memory, and the failure coverage rate reaches 100%. Meanwhile, the algorithm test and test efficiency is improved by 55.97%.

Key words: SONOS; disturbance failure model; March-SONOS algorithm; algorithm evaluation system

1 引言

Flash 存储器作为非易失性存储器(NVM)的

一种, 具有反复擦写, 掉电数据不会丢失的特性, 被广泛运用于便携式电子设备中. 嵌入式 flash 顾名思义为嵌入整合至大规模逻辑电路中的闪存组件,

起着保存程序代码和中间数据的重要作用。随着嵌入式存储技术的发展,嵌入式 flash 逐渐占据嵌入式存储市场并替代 Mask-ROM、EPROM,成为嵌入式存储领域的主流^[1]。伴随嵌入式 flash 的广泛使用,存储器的可靠性问题也与之俱来,根据 IEEE 1005 标准定义^[2],flash 存储器的可靠性评价包含三个方面,即数据保留特性、抗擦写能力以及扰动特性。文献[3]中建立了 SONOS 型 flash 存储器的扰动失效模型并将其分为编程扰动失效和读扰动失效。

本文基于扰动失效模型的建立,展开 SONOS 型 flash 存储器测试算法的研究。2000 年, March 算法的测试思想第一次运用在 flash 存储器的扰动失效测试中,从此,发展演变出众多 flash 存储器扰动失效测试算法,其中 March-FT 测试算法被广泛使用^[4]。随着存储器的不断发展,测试算法中数据传输的位宽受限,导致无法满足新型 Flash 存储器的测试需求。本文基于 March-FT 测试算法进行改进和优化,提出了一种新型的 SONOS 型 flash 存储器测试算法——March-SONOS。同时为了评估测试算法的测试性能与效率,针对传统的 flash 算法评估系统 RAMSES-FT^[5]进行优化和拓展,设计出一款适用于 SONOS 型 flash 存储器的算法评估系统。从失效覆盖率 (Fault Coverage) 和时间复杂度 (Test Time Complexities) 两个角度^[4]对 March-SONOS 测试算法进行评估分析。实验结果表明 March-SONOS 算法可以检测 SONOS 型 flash 存储器的全部扰动失效,且相比传统的 March-FT 测试算法效率提高。

2 SONOS 单元结构的扰动失效模型

2.1 SONOS 存储单元扰动失效的原因

SONOS 存储器作为电荷俘获型 flash 存储器一种,以更低的操作电压、制造工艺简单、更好的扩展能力以及与 CMOS 工艺相兼容等优点,逐渐被广泛应用于嵌入式存储领域^[3],最为常见的是 NOR 型结构的 SONOS 嵌入式 flash。SONOS 单元结构如图 1 所示,从上向下分别为控制栅,阻挡氧化层,电荷俘获层、隧穿氧化层以及硅衬底。SONOS 存储单元的编程、擦除机制是通过在控制栅和衬底间施加电压,载流子受到纵向电场的影响而产生不同方向的 FN(Fowler-Nordheim)隧穿效应,从而实现存储单元的编程和擦除操作。SONOS 存储单元的读操作是通过对所选中的存储单元,在控制栅上施加读

取电压,源极和漏极施加固定电压,通过判断漏电流的大小区分存储单元的存储状态。

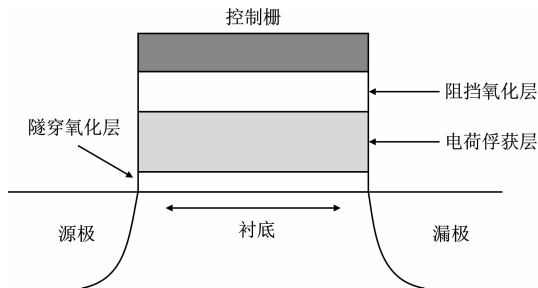


图 1 SONOS 的单元结构

从图 1 可以发现,隧穿氧化层的厚度远远小于阻挡氧化层,这是为了让载流子在电场的作用下,更容易发生隧穿效应。然而,在嵌入式存储芯片的制造过程中,若无法保证工艺质量,会造成氧化层薄膜的厚度不均以及参杂浓度不均等问题。同时,随着集成电路工艺节点的不断提高,存储单元的器件尺寸和单元间的距离逐渐较小,也会导致氧化层薄膜减小。存储单元若存在以上两种情况,就会在存储器操作的过程中导致某些存储单元发生热电子击穿或不必要的隧穿效应,从而产生扰动失效。此外,还存在一些其他的因素使得 SONOS 存储器产生扰动效应,比如编程/擦写周期次数、编程/擦写时间以及栅极和漏极施加电压大小等因素,都会在不同程度上对存储单元产生扰动效应^[3]。

2.2 SONOS 型 flash 存储器的扰动失效模型

本文基于 2T-SONOS 存储阵列结构研究存储器的扰动失效机制,并根据现有的文献研究建立 SONOS 型 flash 存储器扰动失效模型。在图 2 中每个存储单元分别包含选择管 (Select Gate, SG) 和 SONOS 结构器件,也称控制管 (Control Gate, CG)。如图 2 中单元 P,选择管只有在选中该存储单元进行操作时才会导通,其余时刻均为关断状态,此时选择管也可以帮助 SONOS 存储器减小关断状态漏电流,从而达到降低功耗的作用。SONOS 型 flash 编程和擦除的最小操作单位为行 (page),当存储单元进行擦除操作后,单元的逻辑状态应为“0”;当存储单元进行编程“1”操作后,单元的逻辑状态应为“1”;当存储单元进行编程“0”操作后,单元的逻辑状态应为“0”。“0”状态的存储单元可以编程为“1”,但无法将“1”状态的存储单元编程为“0”。本文根据 SONOS 器件特有的结构和工作原理,基于现有的文献研究将 SONOS 型 flash 存储器的扰动特性分为两种,分别为编程扰动 (program disturb) 和读扰动

(read disturb)^[3].

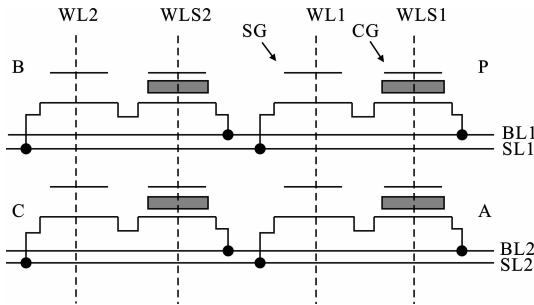


图 2 2T-SONOS 的 2×2 存储阵列模型

编程扰动(program disturb, PD)是指在编程操作下,存储单元发生的扰动失效.编程扰动又可以分为三种扰动失效模型,分别为抑制扰动(inhibit disturb)、位线扰动(bit-line disturb)以及桥接扰动(bridge fault).编程扰动的具体操作电压情况如表 1 所示.

表 1 编程扰动下的电压施加情况^[3]

单元	WLS	WL	BL	扰动失效类型	失效现象
P 单元	V_{pos}	V_{neg}	V_{neg}	无	无
A 单元	V_{pos}	V_{neg}	V_{bl}	抑制扰动失效 桥接扰动失效	误编程(0→1) 误编程(0→1)
B 单元	V_{neg}	V_{neg}	V_{neg}	无	无
C 单元	V_{neg}	V_{neg}	V_{bl}	位线扰动失效	误擦除(1→0)

注: $V_{bl}=1.2\text{ V}$; $V_{pos}=7.2\text{ V}$; $V_{neg}=-3.5\text{ V}$ ^[3]

抑制扰动失效(inhibit disturb)发生在单元 A 中,在存储阵列擦除后,对 P、A 单元所在的行进行编程操作时,对 P 单元进行编程“1”操作,对 A 单元进行编程“0”操作,由于 P 和 A 单元共用相同的字线(WLS),而相比较 P 单元,A 单元的位线(BL)施加更小的电压,使得 A 单元栅极与衬底间所产生的电场强度不足以让电子产生隧穿效应进入电荷俘获层,从而使 A 单元仍然保持之前的擦除状态“0”.但由于隧穿氧化层的电荷隔离性较差或编程时间过长等因素的影响下,A 单元中部分电子产生不必要的隧穿效应,电子进入电荷俘获层,发生抑制扰动(inhibit disturb),导致 A 单元被误编程为状态“1”^[6].

桥接扰动失效(bridge fault)也可能在单元 A 中发生,随着存储阵列的最小线宽不断减小,存储器在制造过程中会出现不同类型的缺陷,其中由于制造缺陷导致两条位线的漏极之间通过电桥或通过电阻相连接,从而单元间漏极电压发生扰动,这种失效类型称为桥接失效^[7].如图 2 所示,若单元 P 与单元 A 发生桥接失效,单元 A 受到单元 P 漏极处 V_{pos} 电

压的影响而发生扰动,导致单元 A 中衬底间电子受纵向电场影响发生隧穿效应,进入电荷俘获层,单元 A 被误编程为状态“1”.

位线扰动(bit-line disturb)也称为漏极扰动(drain disturb),扰动发生在单元 C 中,若单元 C 为已编程器件,发生扰动前状态为“1”,由于单元 C 和单元 A 共用相同的位线,单元 C 受到漏极电压 V_{bl} 的影响,使得其栅极和漏极间产生电场,电子在电场的作用下发生隧穿效应,从电荷俘获层流入漏极,导致单元 C 发生位线扰动.随着扰动效应的持续发生,电荷俘获层中电子数量逐渐减少,最终单元 C 被误擦除.

读扰动(Read Disturb, RD)是指单元在读操作时,存储单元发生的扰动失效,该扰动可以分为单元自身的读扰动和单元间的读扰动,读扰动的失效效应为误擦除.在 2T-SONOS 结构中,不论存储单元的选择管是否导通,读扰动都会发生^[3].同时需要注意,存储单元上施加的读电压较小,并且一次读操作的时间较短,仅一次读操作可能无法造成存储单元发生读扰动失效,因此想要激活并检测读扰动失效需要较长的操作时间或者多次连续读操作^[8].产生读扰动的操作电压如表 2 所示.

表 2 读扰动下的电压施加情况^[3]

单元	WLS	WL	BL	扰动失效类型	失效现象
P 单元	0 V	1.8 V	1.1 V	自身读扰动失效	误擦除(1→0)
A 单元	0 V	1.8 V	0 V	无	无
B 单元	0 V	0 V	1.1 V	单元间读扰动	误擦除(1→0)
C 单元	0 V	0 V	0 V	无	无

注: $V_{GND}=0\text{ V}$ ^[3]

在对单元 P 进行读操作时,单元 P 和单元 B 中发生了读扰动(Read disturb).首先,在 2T-SONOS 单元阵列中,单元 P 所在的 WL 上施加了 1.8 V 的电压,目的为将其所在行的选择管导通.其次,单元 P 的位线上施加了 1.1 V 的读电压,导致单元 P 在漏极处受到 1.1 V 电压的影响,栅极和漏极间产生一个纵向电场,电子受电场影响发生隧穿效应,从电荷俘获层中脱离,进入漏极区域,导致单元 P 发生误擦除现象.而单元 B 同理,虽然单元 B 的选择管并未导通,但其栅极和漏极处同样存在纵向的电场,导致电子发生隧穿效应,单元 B 被误擦除.因为单元 P 是在读取自身单元存储状态时发生失效,故称为自身读扰动失效,而单元 B 是在单元 P 进行读操作时发生了失效,故称为单

元间的读扰动失效。

3 SONOS 型 flash 的扰动失效测试算法研究

3.1 March-FT 测试算法

March-FT 测试算法在 flash 测试中最为常用,这种测试算法不仅可以测试 flash 存储器特有的扰动失效,同时也可以对 RAMs 存储器的故障失效进行检测,提高失效覆盖率的同时没有增加过多的硬件和时间开销. March-FT 算法结构^[4]如下:

$(E); \uparrow(R_1, P_0, R_0); \downarrow(R_0); (E); \downarrow(R_1, P_0, R_0);$
 $\downarrow(R_0)$

E 代表擦除操作,擦除后存储单元逻辑状态均为“1”, R 代表读操作,其中 R_1 代表读“1”操作, R_0 代表读“0”操作, P_0 代表编程操作,编程后存储单元的逻辑状态为“0”,“ $\uparrow$ ”表示地址增序,“ $\downarrow$ ”表示地址降序. 算法包含 6 个元素,元素 2 和元素 5 中分别包含 3 种亚元素. 在测试时,算法中的每种元素需要对所有存储单元遍历一遍,如元素 2 所示 $\uparrow(R_1, P_0, R_0)$,表示该操作从末首地址开始,地址依次递减减 1,每个地址按序依次进行读“1”操作;编程“0”操作以及读“0”操作. 编程操作可以激活存储单元发生扰动失效,而进行读“0”或读“1”时,若读出相反的状态,则判定存储单元发生扰动,起到检查失效的作用. 上述测试算法为单 bit 数据传输,文献[4]同时提出了位宽为 4 bit 的 March-FT 测试算法,该测试算法可以面向字进行测试.

3.2 两种基于 March-FT 算法优化的衍生算法

随着存储技术的不断发展,传统的 March-FT 测试算法受到数据传输位宽较小的限制,不再满足新型 flash 存储器的检测. 文献[9]基于 4 bit 的 March-FT 测试算法进行改进,提出了一种棋盘格 March-like 测试算法,算法可以检测 flash 存储器的全部扰动失效,同时缩短了算法结构的长度,算法复杂度降低,测试效率提升,该测试算法面向位宽为 16 bit 的存储器使用. 同时文献[10]中也将 March-FT 测试算法扩展到面向位宽为 16 bit 的测试算法. 这两种基于 March-FT 改进后的测试算法均可以对位宽为 16 bit 的 flash 存储器进行扰动失效测试,但 SONOS 型 flash 存储器的最小操作单元为行 (page),故上述两种测试算法无法直接用于 SONOS 型 flash 存储器测试,需要对其进行扩展. 本文分别将两种基于 16 bit 数据传输的 March-FT 测试算法在 SONOS 型 flash 存储器平台进行扩展,扩展内容如下:(1)更新擦写操作后存储单元的逻辑状态. 即

原算法中,存储单元在擦除操作后,单元逻辑状态为“1”,编程操作后逻辑状态为“0”. 而 SONOS 型 flash 存储单元在擦除操作后逻辑状态为“0”,编程操作后逻辑状态为“1”或“0”. (2)扩展测试算法中的最小操作单元,将原有数据位宽为 16 bit 的测试算法扩展到 32 bit,且最小操作单元为行 (page). 扩展后的测试算法如下所示.

棋盘格 March-like 测试算法^[9]扩展后如下所示,并将其命名为 March-FT 扩展算法 1.

$(E); \uparrow(R_{\text{page}}(00000000), P_{\text{page}}(FFFFFFF),$
 $R_{\text{page}}(FFFFFFF)); \downarrow(R_{\text{page}}(FFFFFFF));$
 $(E); \downarrow(R_{\text{page}}(00000000), P_{\text{page}}(FFFFFFF),$
 $R_{\text{page}}(FFFFFFF)); \downarrow(R_{\text{page}}(FFFFFFF));$
 $(E); \uparrow(P_{\text{page}}(AAAAAAAA), P_{\text{page}}(55555555));$
 $\downarrow(R_{\text{page}}(AAAAAAAA), R_{\text{page}}(55555555));$
 $(E); \downarrow(P_{\text{page}}(55555555), P_{\text{page}}(AAAAAAAA));$
 $\downarrow(R_{\text{page}}(55555555), R_{\text{page}}(AAAAAAAA));$

March-FT 测试算法^[10]扩展后如下所示,并将其命名为 March-FT 扩展算法 2.

$(E); \uparrow(R_{\text{page}}(00000000), P_{\text{page}}(FFFFFFF),$
 $R_{\text{page}}(FFFFFFF)); \downarrow(R_{\text{page}}(FFFFFFF));$
 $(E); \downarrow(R_{\text{page}}(00000000), P_{\text{page}}(FFFFFFF),$
 $R_{\text{page}}(FFFFFFF)); \downarrow(R_{\text{page}}(FFFFFFF));$
 $(E); \uparrow(P_{\text{page}}(0000FFFF)); \downarrow(R_{\text{page}}(0000FFFF));$
 $(E); \downarrow(P_{\text{page}}(FFFF0000)); \downarrow(R_{\text{page}}(FFFF0000));$
 $(E); \uparrow(P_{\text{page}}(0F0F0F0F)); \downarrow(R_{\text{page}}(0F0F0F0F));$
 $(E); \downarrow(P_{\text{page}}(F0F0F0F0)); \downarrow(R_{\text{page}}(F0F0F0F0));$
 $(E); \uparrow(P_{\text{page}}(33333333)); \downarrow(R_{\text{page}}(33333333));$
 $(E); \downarrow(P_{\text{page}}(CCCCCCCC)); \downarrow(R_{\text{page}}(CCCCCCCC));$
 $(E); \uparrow(P_{\text{page}}(55555555)); \downarrow(R_{\text{page}}(55555555));$
 $(E); \downarrow(P_{\text{page}}(AAAAAAAA)); \downarrow(R_{\text{page}}(AAAAAAAA));$

在上述算法中, R_{page} 为读取行 (read page) 操作, P_{page} 为行编程 (page program) 操作,数据位宽均为 32 bit. 下面举例说明两种算法中操作符的含义, March-FT 扩展算法 1 的 $(P_{\text{page}}(AAAAAAAA), P_{\text{page}}(55555555))$ 操作符为偶数行中全部地址写入测试向量 32'hAAAAAAAA,奇数行中全部地址写入测试向量 32'h55555555. March-FT 扩展算法 2 中 $P_{\text{page}}(F0F0F0F0)$ 操作符表示对每行中的各个存储地址写入 32 bit 测试向量 F0F0F0F0; $R_{\text{page}}(F0F0F0F0)$ 操作符表示对每行中的各个存储地址读取 32 bit 数据 F0F0F0F0. 观察扩展后的测试算法可以发现,算法通过写入不同的测试向量,达到对每个存储单元进行编程“0”和编程“1”的操作. 编程

“0”操作可以激发编程扰动中的抑制扰动失效和位线扰动失效,而编程“1”操作可以激发编程扰动中的桥接扰动失效,并且与之对应的读操作可以检测出单元自身与单元间是否发生上述扰动失效.但扩展后的 March-FT 衍生测试算法对 SONOS 型 flash 存储器测试仍然存在不适用性,总结如下:(1)算法无法检测 SONOS 型 flash 存储器的读扰动失效.由于读扰动失效需要多次连续读操作才能激活,而上述测试算法中的读操作次数明显不满足激活读扰动失效的条件,故上述两种 March-FT 衍生算法无法对读扰动失效进行检测.(2)March-FT 衍生算法时间复杂度高,测试时间较长.特别是 March-FT 衍生算法 2 中的测试向量种类丰富,如 32'hCCCCCCC、32'h33333333、32'h0F0F0F0F 等.研究发现该测试向量均可由棋盘格测试向量代替,如奇数行中地址均写入 32'hAAAAAAAA,偶数行中地址均写入 32'h55555555,或者偶数行中地址均写入 32'hAAAAAAAA,奇数行中地址均写入 32'h55555555,类似这样的棋盘格式测试向量,可以最大程度的激发 SONOS 型 flash 存储器中的扰动失效.

3.3 March-SONOS 算法提出

根据上述分析,通过总结 March-FT 算法的测试思想,并与 SONOS 型 flash 的扰动机制相结合,将棋盘格向量与 March-FT 算法融合,提出了一种名为 March-SONOS 的测试算法,该测试算法可以检测 SONOS 型 flash 的全部扰动失效. March-SONOS 算法结构精简,时间复杂度低,算法结构如下:

```
E; ↑(R(00000000),P(55555555),
R(55555555)"); ↑(R(55555555));
E; ↓(R(00000000),P(AAAAAAAAA),
R(AAAAAAAAA)"); ↓(R(AAAAAAAAA));
E; ↑(P(AAAAAAAAA),R(AAAAAAAAA)");
↑(R(AAAAAAAAA));
E; ↓(P(55555555),
R(55555555)"); ↓(R(55555555));
```

March-SONOS 算法中 P(55555555)代表对奇数行中全部地址写入 32'h55555555,偶数行中全部地址写入 32'hAAAAAAAA;P(AAAAAAAAA)代表对奇数行中全部地址写入 32'hAAAAAAAA,偶数行中全部地址写入 32'h55555555;且读操作同理,需要注意的是 March-SONOS 测试算法中增加了多次读操作,如 R(55555555)ⁿ表示对奇数行中所有地址读 32'

h55555555,对偶数行中所有地址读 32'hAAAAAAAA,读取次数均为 n 次,具体的读取次数需针对不同性能的存储器特殊设定,目的是通过多次连续读操作激发并检测出读扰动失效.

4 测试算法的性能与评价

在文献[5]中,研究人员设计了一款模拟 flash 存储器扰动失效的算法评估系统,名为 RAMSES-FT.该算法评估系统适用于对面向字操作的 flash 存储器,且不包含 SONOS 型 flash 存储器的特有扰动失效,从而不适用于 SONOS 型 flash 存储器架构.本文基于 RAMSES-FT 算法评估系统,设计了一款适用于 SONOS 型 flash 测试算法的评估系统.算法评估系统使用 verilog 硬件描述语言设计,主要包含 3 个模块,分别为失效描述模块(fault descriptor,FD)、失效模拟引擎模块(fault simulation engine)以及失效覆盖率计算模块(Fault coverage calculation).

失效描述模块将 flash 的扰动失效机制通过四点要素描述出来,分别为激发失效的操作(aggres-sor,AGR)、扰动单元的位置(suspect,SPT)、检测扰动失效的操作(victim,VTM)以及覆盖扰动失效的操作(recoverer,RCV),AGR 为激活存储单元扰动失效的操作,比如编程“1”操作、编程“0”操作以及读操作等;SPT 为产生扰动与被扰动单元间的位置关系,比如存储单元自身发生扰动,以及同一位线或字线上的相邻存储单元间发生扰动;VTM 为检测到扰动失效的操作,比如读“0”或者读“1”操作;RCV 为覆盖扰动失效的操作,比如编程或擦写操作,这类操作若比 VTM 提前发生,则会覆盖掉扰动对应的失效效应.基于这四点元素,建立了 SONOS 型 flash 存储器的扰动失效模型,如表 3 所示.以抑制扰动失效是为例,它是一种单元自身的扰动失效,编程“0”操作可以激发抑制扰动,通过对扰动单元进行读“0”操作可以检测到该失效,若在检测操作前对失效单元进行擦除或者编程操作会将该失效覆盖.

表 3 SONOS 型 flash 存储器扰动失效模型

	抑制扰动	位线扰动	桥接扰动	自身读扰动	单元间读扰动
AGR	编程“0”	编程“0”	编程“1”	读操作	读操作
SPT	单元自身	位线	字线	单元自身	位线
VTM	读“0”	读“1”	读“0”	读“1”	读“1”
RCV	编程 & 擦除	编程 & 擦除	编程 & 擦除	编程	编程

失效模拟引擎模块可以根据算法中的操作以及扰动失效的描述,判断出是否可以检测到该失效,并

对检测到的失效单元计数,之后通过失效覆盖率的计算模块计算,并将结果输出到用户主机中,算法评估系统的工作流程如图 3 所示.

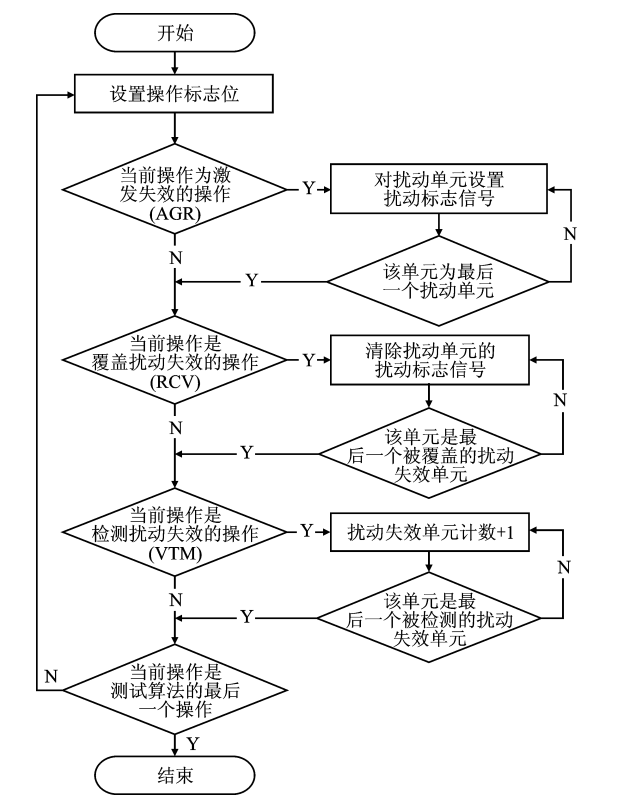


图 3 失效模拟引擎模块工作流程图

本文通过 SONOS 型 flash 的算法评估系统对三种测试算法的扰动失效覆盖率进行评估,测试算法评估报告如表 4 所示.

表 4 算法评估报告

失效覆盖率	算法列表	March-FT 扩展算法 1	March-FT 扩展算法 2	March-SONOS
	算法时间复杂度	$4E+4NP+8NR$	$10E+10NP+14NR$	$4E+4NP+(8+2n)NR$
	抑制扰动	100%	100%	100%
	位线扰动	100%	100%	100%
	桥接扰动	100%	100%	100%
	单元自身读扰动	0%	0%	100%
	单元间读扰动	0%	0%	100%

本文基于四种不同规格大小的 flash 存储器进行实验分析,通过表 4 中算法的时间复杂度表达式可知测试所需的时间长度,其中 E 代表存储器整片擦除所需的时间; P 代表对一行进行编程所需的时间; R 代表对一行进行读取所需的时间; N 代表 SONOS 型 flash 存储器的行数; n 代表读取操作的次数(以连续读取 50 次为例),最终将三种测试算法分

别在 4 种不同规格大小的 SONOS 型 flash 存储器中进行测试,测试时间结果如图 4 所示.

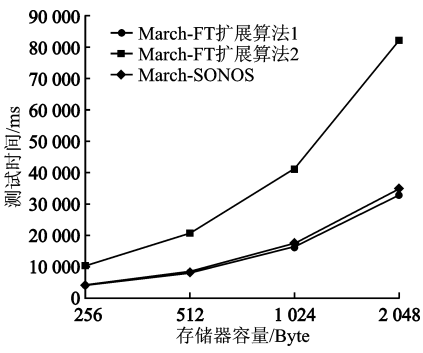


图 4 算法测试时间对比图

通过失效覆盖率评估报告可以看出,两种扩展后的 March-FT 测试算法均无法对读扰动失效进行检测,而 March-SONOS 测试算法可检测到全部扰动失效,且失效覆盖率达到 100%. 同时对比不同算法的测试时间可以看出, March-SONOS 测试算法相比较 March-FT 扩展算法 1,测试时间只增加了 0.062%,就可以将全部失效覆盖率达到 100%,而相比较 March-FT 扩展算法 2,测试时长更是减少了 55.97%,测试效率提升一倍之多.

5 结束语

本文从 SONOS 器件的结构和工作机制入手,分析了 SONOS 型 flash 存储器的扰动失效原理和机制. 基于对两种 March-FT 衍生测试算法的扩展与分析,提出了优化的 SONOS 型 flash 存储器扰动失效测试算法——March-SONOS. 设计并使用 SONOS 型 flash 存储器的算法评估系统,将 March-SONOS 测试算法与 March-FT 衍生测试算法进行对比和评估. 结果表明, March-SONOS 测试算法对 SONOS 型 flash 存储器的读扰动失效检测效果明显,且失效覆盖率达到 100%. 同时相比较 March-FT 衍生测试算法,测试时间减少 55.97%,测试效率更高.

参考文献:

[1] HIDAKA H. Evolution of embedded flash memory technology for MCU[C]//Proceedings of 2011 IEEE International Conference on IC Design & Technology. Kaohsiung, China: IEEE, 2011: 1-4. DOI: 10.1109/ICICDT.2011.5783209.

[2] IEEE. IEEE Std 1005-1991 IEEE standard definitions and characterization of floating gate semiconductor arrays[S]. IEEE, 1991. DOI: 10.1109/IEEESTD.

1991. 101070.
- [3] LIU L J, SHEN G F, CAO G, et al. Study of the disturb in SONOS memory[C]//Proceedings of the IEEE 11th International Conference on Solid-State and Integrated Circuit Technology. Xi'an, China: IEEE, 2012: 1-3. DOI: 10.1109/ICSICT.2012.6466695.
- [4] YE H J C, CHENG K L, CHOU Y F, et al. Flash memory testing and built-in self-diagnosis with march-like test algorithms[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2007, 26(6): 1101-1113. DOI: 10.1109/TCAD.2006.885828.
- [5] CHENG K L, YE H J C, WANG C W, et al. RAM-SES-FT: a fault simulator for flash memory testing and diagnostics[C]//Proceedings of the 20th IEEE VLSI Test Symposium (VTS 2002). Monterey, CA, USA: IEEE, 2002: 281-286. DOI: 10.1109/VTS.2002.1011153.
- [6] TSUJI Y, TERA I M, FUJIEDA S, et al. New degradation mode of program disturb immunity of sub 90-nm node split-gate SONOS memory[C]//Proceedings of 2008 IEEE International Reliability Physics Symposium. Phoenix, AZ, USA: IEEE, 2008: 699-700. DOI: 10.1109/RELPHY.2008.4558995.
- [7] ZHANG Q W, ZENG Z M, ZHANG Y X, et al. EFA method of AA-bridge-type-defect of SONOS NOR flash[C]//Proceedings of 2019 China Semiconductor Technology International Conference (CSTIC). Shanghai, China: IEEE, 2019: 1-4. DOI: 10.1109/CSTIC.2019.8755717.
- [8] MARTIROSYAN S, HARUTYUNYAN G, SHOUK-OURIAN S, et al. An efficient testing methodology for embedded flash memories[C]//Proceedings of 2017 IEEE East-West Design & Test Symposium (EWDTS). Novi Sad: IEEE, 2017: 1-4. DOI: 10.1109/EWDTS.2017.8110074.
- [9] 杨丽婷, 王琴, 倪昊, 等. pFlash 故障测试算法[J]. 微电子学与计算机, 2019, 36(9): 55-60. DOI: 10.19304/j.cnki.issn1000-7180.2019.09.011.
- YANG L T, WANG Q, NI H, et al. Fault test algorithm based on pFlash[J]. Microelectronics & Computer, 2019, 36(9): 55-60. DOI: 10.19304/j.cnki.issn1000-7180.2019.09.011.
- [10] 郭进杰, 王瑜, 李婷, 等. 基于 FLASH-March 算法的 FLASH 缺陷检测系统[J]. 微电子学与计算机, 2014, 31(10): 81-85. DOI: 10.19304/j.cnki.issn1000-7180.2014.10.019.
- GUO J J, WANG Y, LI T, et al. A flash fault detection system based on FLASH-March algorithm[J]. Microelectronics & Computer, 2014, 31(10): 81-85. DOI: 10.19304/j.cnki.issn1000-7180.2014.10.019.

作者简介:

乌勒 男, (1994-), 硕士研究生. 研究方向为数字集成电路设计与 flash 存储器测试. E-mail: wule_up@sjtu.edu.cn.

刘武 男, (1976-), 博士, 助理研究员. 研究方向为微机电系统.

洪亮 男, (1980-), 博士, 高级工程师. 研究方向为大规模数字集成电路设计.