

引用格式: 桑胜男, 张立军, 郑坚斌, 等. 一种适用于 FinFET 单存储单元的高效的动态故障测试算法 [J]. 微电子学与计算机, 2019, 36(4): 17-22. [SANG S N, ZHANG L J, ZHENG J B, et al. An efficient dynamic fault test algorithm for FinFET single memory cell [J]. Microelectronics & Computer, 2019, 36(4): 17-22.]

一种适用于 FinFET 单存储单元的高效的动态故障测试算法

桑胜男¹, 张立军¹, 郑坚斌², 彭增发²

(1 苏州大学, 江苏 苏州 215000; 2 苏州兆芯半导体科技有限公司, 江苏 苏州 215000)

摘 要: FinFET 作为现在最前沿的晶体管技术, 在嵌入式存储器上得到了广泛应用. 但是, 新的工艺也带来了新的缺陷. Synopsys 公司通过故障建模及测试发现: FinFET 存储器比平面存储器对动态故障更敏感, 而经典的测试算法仅仅针对静态故障. 目前关于动态故障的测试算法极少, 并且复杂度很高. 因此, 本文提出了一种改进的动态故障测试算法, 该算法能够覆盖所有连续两次敏化操作的单单元动态故障.

关键词: SRAM; March 算法; 动态故障; 故障原语; MBIST

中图分类号: TN432

文献标识码: A

文章编号: 1000-7180(2019)04-0017-06

An Efficient Dynamic Fault Test Algorithm for FinFET Single Memory Cell

SANG Sheng-Nan¹, ZHANG Li-Jun¹, ZHENG Jian-Bin², PENG Zeng-Fa²

(1 Soochow University, Suzhou 215000 China; 2 Megacores Technology Co. LTD, Suzhou 215000, China)

Abstract: FinFET as the most advanced transistor technology, has been widely used in embedded memory. However, the new process also brings new defects. Synopsys found through fault modeling and testing that FinFET memory is more sensitive to dynamic faults than planar memory. The classic test algorithm is only for static faults. At present, there are few test algorithms for dynamic faults, and the complexity is very high. Therefore, an improved dynamic fault test algorithm is proposed in this paper. This algorithm can cover all single-cell dynamic faults in two successive sensitization operations.

Key words: SRAM; march algorithm; dynamic fault; fault primitive; MBIST.

1 引言

随着半导体技术的不断发展, FinFET 通过多栅结构有效的抑制沟道效应和制作工艺的良好兼容性, 在嵌入式存储器上得到了广泛应用. 但是, FinFET 存储器也带来了新的设计复杂性、缺陷覆盖和良率挑战. Synopsys 关于 FinFET 工艺故障建模的部分普遍结论是: (1) FinFET 存储器比平面存储器对动态故障更敏感; (2) FinFET 存储器对制程变异

故障更稳定; (3) 静态单单元和耦合故障在两种存储器中均很常见^[1].

常用的测试算法如 MATS、March C、March C+、March C-等都仅适用于静态故障测试. 但对于动态故障的研究, 国内外都尚处于起步阶段. 目前关于动态故障测试算法的研究成果有: Said Hamdioui 提出了 RAW1(13N) 和 RAW(26N)^[2] 两种算法, 这两种算法仅能覆盖部分单单元和双单元动态故障. Benso 提出了一种复杂度为 100N 的 March 算法^[3], 该算法主要针对连续两次敏化操作的单单元

故障及耦合故障. Harutunyan 将上述 100N 算法优化成复杂度为 70N 的 March MD2 算法,且没有改变故障覆盖率,同时,还提出 March MD1_a (33N)和 March MD1_b (33N)两种用以检测所有的连续两次敏化操作的单元故障的算法^[4].

本文在 March MD1_a (33N)和 March MD1_b (33N)两种算法的基础上进行该进,提出一种复杂度为 25N 的测试算法且保证故障覆盖率不变,并应用于 16 nm 存储器芯片的自动测试电路中.

2 FinFET 技术简介

2.1 FinFET 器件

鳍式场效应晶体管 (Fin Field-Effect Transistor, FinFET)是由美籍华人科学家胡正明教授提出的一种 MOS 结构,其中的 Fin 在构造上类似鱼鳍,所以称为“鳍式”. FinFET 是三维结构,也称三栅晶体管,如图 1 所示. 沟道是由金属栅极包裹的薄鳍片,沿源漏方向鳍片的长度为沟道长度. 这种结构增强了栅极对沟道的控制能力,从而降低了漏电流,抑制短沟道效应. 在工艺制造时,通过短沟道隔离 (Shallow Trench Isolation, STI) 决定鳍片的高度 (H_{fin})^[5]. 鳍片的宽度 (W_{fin}) 决定了一些电学特性,晶体管开启电流与关断电流的比值及阈值电压及随着鳍片宽度的增加而减小,这表明更短的鳍片宽度,可以更好地控制沟道电流^[6].

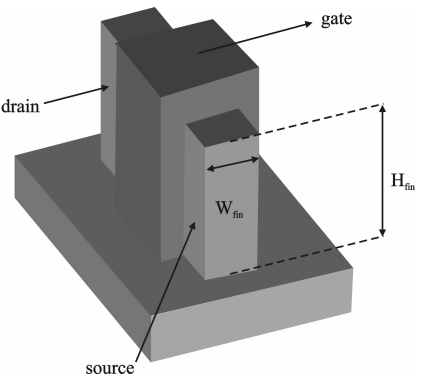


图 1 FinFET 结构

2.2 FinFET 特殊缺陷

与平面工艺相比,FinFET 表现出了一些特殊的缺陷,如图 2 所示,FinFET 主要有五种特殊缺陷^[7]:

- (a) 鳍片开路: 鳍片完全开路 and 电阻开路.
- (b) 栅极开路: 栅极完全开路 and 电阻开路.
- (c) 鳍片粘连: 源极和漏极完全短路 and 电阻短路.

(d) 栅极-鳍片短路: 栅极和鳍片完全短路 and 电阻短路.

(e) 制程变异: FinFET 参数值发生变化.

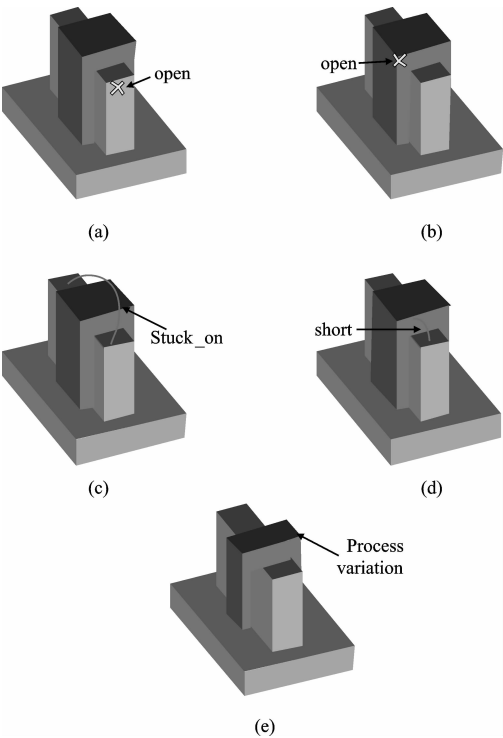


图 2 FinFET 缺陷

3 故障原语及故障模型

故障原语 (Fault Primitive, FP)^[8] 表示对存储器进行某一操作时,存储器预期行为和所观察到的行为之间的差异,记为 $FP = (S/F/R)$. S 表示敏化测试单元故障行为的状态或操作, F 表示故障单元的状态或行为, R 表示存储单元的输出值.

3.1 静态故障

静态故障表示最多对存储单元进行一次操作而导致的故障,用故障原语表示即 $S \in \{0, 1, 0w0, 0w1, 1w0, 1w1, 1r1, r0, r1\}$, 分别表示存储单元状态 0、状态 1、0 写 0、0 写 1、1 写 0、1 写 1、读 0、读 1. $F \in \{0, 1, \uparrow, \downarrow\}$, $\uparrow(\downarrow)$ 表示上升(下降)传输. $R \in \{0, 1, -\}$, 当 S 不是读操作时, R 为一. 耦合故障的故障原语表示为 $FP = (Sa; Sv/F/R)$, $Sa(Sv)$ 表示敏化施主(受主)单元的状态或操作, 此处应注意到, 若 Sa 为一个状态, 则 Sv 可能是一个状态或操作; 若 Sa 是一个操作, 那么 Sv 只能是一个状态.

静态故障可分为单元故障及耦合故障, 单元故障包括: 地址译码故障 (Address-Decoder Fault, AF/ADF)、固定故障 (Stuck-At Fault, SAF)、转换

故障 (Transition Fault, TF)、写干扰故障 (Write Disturb Fault, WDF)、读破坏故障 (Read Destructive Fault, RDF)、欺骗性读破坏故障 (Deceptive Read Destructive Fault, DRDF)、错误读故障 (Incorrect Read Fault, IRF)^[9], 其中地址译码故障和固定故障由于表现形式一致, 可以统一为状态故障 (State Fault, SF). 考虑到算法复杂度的问题, 当前的算法主要研究相邻单元耦合故障, 其故障类型主要包括: 反向耦合故障 (Inversion Coupling Fault, CFin)、幂等耦合故障 (Idempotent Coupling Fault, CFid)、状态耦合故障 (State Coupling Fault, CFs). 以上故障的功能故障模型 (functional fault model, FFM) 及故障原语如表 1 所示.

表 1 静态故障 FFM 及 FPs

FFM	FPs
SF	(0/1/−), (1/0/−)
TF	(0w1/↓/−), (1w0/↑/−)
WDF	(0w0/↑/−), (1w1/↓/−)
RDF	(r0/↑/1), (r1/↓/0)
DRDF	(r0/↑/0), (r1/↓/1)
IRF	(r0/0/1), (r1/1/0)
CFin	(0w1: 0/↑/−), (0w1: 1/↓/−), (1w0: 0/↑/−), (1w0: 1/↓/−)
CFid	(0w1: 0/1/−), (0w1: 1/0/−), (1w0: 0/1/−), (1w0: 1/0/−)
CFs	(0: 0/1/−), (0: 1/0/−), (1: 0/1/−), (1: 1/0/−)

表 2 单单元动态故障 FFM 及 FPs

FFM	FPs
dRDF	(0r0r0/↑/1), (1r1r1/↓/0), (0w0r0/↑/1), (1w1r1/↓/0), (0w1r1/↓/0), (1w0r0/↑/1)
dDRDF	(0r0r0/↑/0), (1r1r1/↓/1), (0w0r0/↑/0), (1w1r1/↓/1), (0w1r1/↓/1), (1w0r0/↑/0)
dIRF	(0r0r0/0/1), (1r1r1/1/0), (0w0r0/0/1), (1w1r1/1/0), (0w1r1/1/0), (1w0r0/0/1)
dTF	(0w0w1/↓/−), (1w1w0/↑/−), (0w1w0/↑/−), (1w0w1/↓/−), (0r0w1/↓/−), (1r1w0/↑/−)
dWDF	(0w0w0/↑/−), (1w1w1/↓/−), (0w1w1/↓/−), (1w0w0/↑/−), (0r0w0/↑/−), (1r1w1/↓/−)

3.2 动态故障

动态故障是指对单个存储单元至少进行一次操作而导致的故障, 同静态故障一样, 动态故障可分为单单元动态故障及相邻单元动态耦合故障, 本文仅对单单元动态故障进行研究. 与静态故障的故障原语不同的是, 在单单元动态故障的故障原语中, $S \in$

$\{0r0r0, 0r0w0, 0r0w1, 1r1r1, 1r1w0, 1r1w1, 0w0w0, 0w1w1, 1w0w0, 1w1w1, 0w0w1, 0w1w0, 1w0w1, 1w1w0, 0w0r0, 0w1r1, 1w0r0, 1w1r1\}$. 单单元动态故障主要包括: 动态读破坏故障 (Dynamic Read Destructive Fault, dRDF)、动态欺骗性读破坏故障 (Dynamic Deceptive Read Destructive Fault, dDRDF)、动态错误读故障 (Dynamic Incorrect Read Fault, dIRF)、动态转换故障 (Dynamic Transition Fault, dTF)、动态写破坏故障 (Dynamic Write Destructive Fault, dWDF). 其功能故障模型及故障原语如表 2 所示.

4 几种动态故障测试算法

March 算法 (齐步法) 是在 1982 年 ITC 国际测试会议上推出的^[10], March 算法具有线性的时间复杂度, 即测试时间与存储容量成正比, 并且具有较高的故障覆盖率, 是目前使用比较广泛和成熟的 RAM 测试算法之一. 每种 March 算法都包含多个 March 元素, 每个 March 元素都由地址变化顺序、读写操作和操作数据三部分组成. $\uparrow$ 表示按地址升序进行操作, $\downarrow$ 表示按地址降序进行操作, e 表示操作的地址顺序可以是升序, 也可以是降序.

常用的动态故障测试算法有:

March RAW1(13N):

$$\{ (w0); \uparrow (w0, r0); \uparrow (r0); \uparrow (w1, r1); \uparrow (r1); \uparrow (w1, r1); \uparrow (r1); \uparrow (w0, r0); \uparrow (r0) \}.$$

March RAW(26N):

$$\{ \uparrow (w0); \uparrow (r0, w0, r0, r0, w1, r1); \uparrow (r1, w1, r1, r1, w0, r0); \downarrow (r0, w0, r0, r0, w1, r1); \downarrow (r1, w1, r1, r1, w0, r0); \downarrow (r0) \}.$$

March MDI_a(33N):

$$\{ \uparrow (w0); \uparrow (w0, w1, w0, w1); \uparrow (r1, w0, w0); \uparrow (w0, w0); \uparrow (r0, w1, r1, w1, r1, r1); \uparrow (r1); \uparrow (w1, w0, w1, w0); \uparrow (r0, w1, w1); \uparrow (w1, w1); \uparrow (r1, w0, r0, w0, r0, r0); \uparrow (r0) \}.$$

March MDI_b(33N):

$$\{ \uparrow (w0); \uparrow (w0, w1, w0, w1, r1); \uparrow (w0, w0); \uparrow (w0, w0); \uparrow (r0, w1, r1, w1, r1, r1); \uparrow (r1); \uparrow (w1, w0, w1, w0, r0); \uparrow (w1, w1); \uparrow (w1, w1); \uparrow (r1, w0, r0, w0, r0, r0); \uparrow (r0) \}.$$

March RAW1 算法是基于先写后读来检测单单元动态故障的, 仅能覆盖部分 dRDF、dDRDF 及 dIRF. March RAW 算法是在 March RAW1 算法的

基础上进行拓展,使其能覆盖部分的动态耦合故障. March MDI_a 算法能覆盖所有单单元动态故障. March MDI_b 算法与 March MDI_a 算法没有本质区别,仅仅是将第三个和第八个 March 元素中的 r0 操作放到前一个 March 元素中,没有改变故障覆盖率.

5 算法改进与实现

5.1 算法的改进

March MDI_a 算法与 March MDI_b 算法对单单元动态故障的覆盖率达到了 100%,但是其复杂度却高达 33N,Harutunyan 也证明了覆盖所有单单元动态故障的最小 March 算法的复杂度是 33N^[4]. 但是该结论成立的条件是必须考虑初始化数据,可以看到在 March MDI_a 与 March MDI_b 算法中有很多重复的序列,如 $\Downarrow(w0, w1, w0, w1)$, 这个序列用来检测 $FP1 = (0w0w1/\downarrow/-)$ 与 $FP2 = (1w0w1/\downarrow/-)$, 其中在第一步写操作前的 0 或 1 就是初始化数据. 而研究发现初始化数据并不会影响写驱动的行为^[11], 因此 FP1 与 FP2 是等价的. 基于此, 本文提出了一种复杂度为 25N 的 March DS 算法, 该算法对单单元动态故障的覆盖率达到了 100%, 其算法序列如下:

March DS(25N):
 $\{ \Downarrow(w0); \Downarrow(w0, w1); \Downarrow(r1, w0, w0);$
M0 M1 M2
 $\Downarrow(r0, w1, r1, r1); \Downarrow(r1, w1, r1); \Downarrow(w1, w0);$
M3 M4 M5
 $\Downarrow(r0, w1, w1); \Downarrow(r1, w0, r0, r0); \Downarrow(r0, w0,$
r0) }.

M6 M7 M8
由上可知, March DS 算法由 M0, M1, M3, M4, M5, M6, M7, M8 等 9 个 March 元素组成, 其状态转换图如图 3 所示, 图中 rstn 为复位信号, 低电平有效, bist_en 为测试开始信号, check_adr_max 用来检测算法是否执行到最后一个地址, cycle_cnt 用来记录对单个存储单元连续操作的次数, 当全部状态完成后, 算法进入无效(IDLE)状态.

March DS 算法的故障覆盖表如表 3 所示, 下面以 dRDF 故障中的 $(0w0r0/\uparrow/1)$ 与 $(1w0r0/\uparrow/1)$ 为例说明表 3 是如何得到的, 首先忽略初始化数据, 则 $(0w0r0/\uparrow/1)$ 与 $(1w0r0/\uparrow/1)$ 是等价的, 然后 M7 中连续的写 0 读 0 操作对故障进行敏化, 最后由 M7 中的最后一次读 0 对故障进行检测.

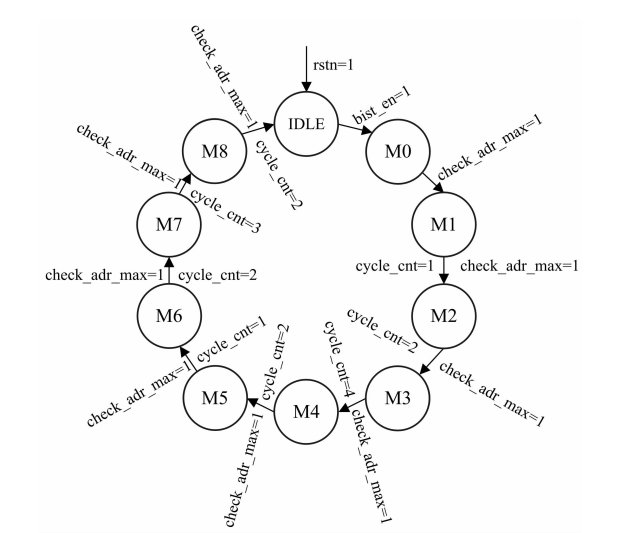


图 3 March DS 算法状态转移图

表 3 March DS 算法故障覆盖表

FFM	FPs	敏化	检测
dRDF	$(0r0r0/\uparrow/1)$	M7	M7, M8
	$(1r1r1/\downarrow/0)$	M3	M3, M4
	$(0w0r0/\uparrow/1), (1w0r0/\uparrow/1)$	M7	M7
	$(0w1r1/\downarrow/0), (1w1r1/\downarrow/0)$	M3	M3
dDRDF	$(0r0r0/\uparrow/0)$	M7	M7, M8
	$(1r1r1/\downarrow/1)$	M3	M3, M4
	$(0w0r0/\uparrow/0), (1w0r0/\uparrow/0)$	M7	M7
	$(0w1r1/\downarrow/1), (1w1r1/\downarrow/1)$	M3	M3
dIRF	$(0r0r0/0/1)$	M7	M7, M8
	$(1r1r1/1/0)$	M3	M3, M4
	$(0w0r0/0/1), (1w0r0/0/1)$	M7	M7
	$(0w1r1/1/0), (1w1r1/1/0)$	M3	M3
dTf	$(0w0w1/\downarrow/-), (1w0w1/\downarrow/-)$	M1	M2
	$(0w1w0/\uparrow/-), (1w1w0/\uparrow/-)$	M5	M6
	$(0r0w1/\downarrow/-)$	M3	M3
	$(1r1w0/\uparrow/-)$	M7	M7
dWDF	$(0w0w0/\uparrow/-), (1w0w0/\uparrow/-)$	M2	M3
	$(0w1w1/\downarrow/-), (1w1w1/\downarrow/-)$	M6	M7
	$(0r0w0/\uparrow/-)$	M8	M8
	$(1r1w1/\downarrow/-)$	M4	M4

5.2 算法实现

为实现提出的 March DS 算法,本文设计了如图 4 所示的 MBIST 电路结构,它包括算法状态机、控制信号生成器、地址生成器、数据生成器和数据比较器.图中各信号的含义如下:

- clk:MBIST 电路的工作时钟信号;
- rstn:复位信号,低电平有效;
- bist_start:MBIST 电路的启动信号,仅当 bist_start 为高电平时,MBIST 电路才会工作;
- ceb:存储器使能信号,信号为低电平时,可以对存储器进行操作;
- web:存储器写使能信号,低电平表示写操作;
- addr:存储器的地址输入;
- data_in:存储器的数据输入;
- data_out:存储器的数据输出;
- SO:MBIST 测试结果输出.

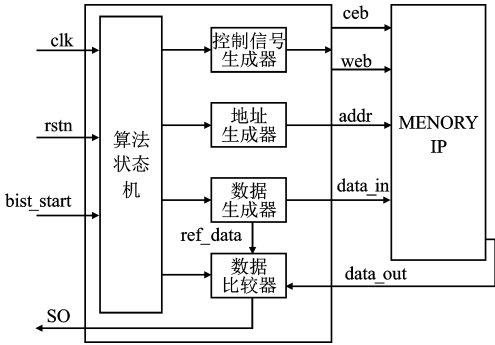


图 4 MBIST 电路结构

6 算法仿真

使用 Ncverilog 工具,结合大小为 40×8 的 SRAM 对算法进行仿真,仿真波形如图 5 所示,bist_err(外部连接至 SO pin)为低电平表示存储器无故障.

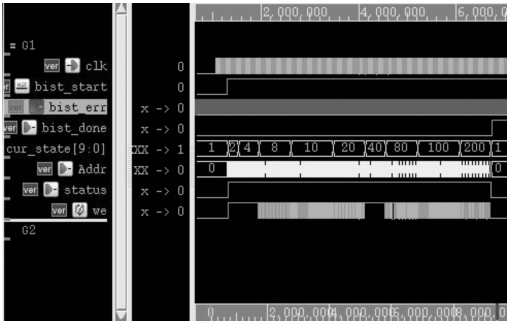


图 5 March DS 算法时序

对存储单元 8'h1a 注入 dRDF(0r0r0/↑/1),仿真波形如图 6 所示,在算法运行到 M8 时,bist_err

出现高电平,表明 March DS 算法能检测到 dRDF 故障.仿真时注入其他故障类型,仿真结果如表 4 所示,bist_err 全部出现高电平,证明 March DS 算法对单存储单元动态故障的覆盖率达到 100%.

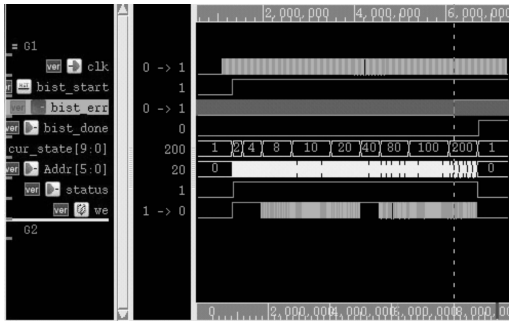


图 6 March DS 算法对 dRDF 故障检测

表 4 仿真结果

	FFM	dRDF	dDRDF	dIRF	dTF	dWDF
bist_err	1	1	1	1	1	1

7 芯片的测试结果

本文 MBIST 测试芯片的设计采用 TSMC 16 nm 先进制程,使用 ATE 机台进行测试,测试系统的结构如图 7 所示,图中 SI 表示 ATE 的配置文件输入,clk 为时钟信号,本次测试的时钟频率为 1GHz,bist_start 为 BIST 电路启动信号,rstn 为复位信号,SO 为测试结果输出.本次测试选取 5 片芯片,每片包含 5 个单端口 SRAM IP(SP)实例和 5 个单端口寄存器类型 IP(R1)实例.表 5 为各芯片的测试结果.

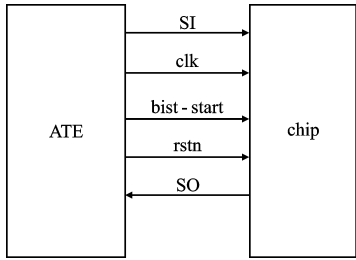


图 7 测试系统结构

8 结束语

本文基于故障原语介绍了 SRAM 静态故障及单单元动态故障.通过研究几种动态故障测试算法,提出了一种新的动态故障测试算法,该算法与 March MDI_a和 March MDI_b对单单元动态故障的覆盖率都达到 100%,但复杂度却比 March MDI_a和

March MDI_b 算法降低了约 24%，然后设计了 MBIST 电路,对 March DS 算法进行了仿真验证,结果表明该算法可以覆盖 SRAM 单存储单元动态故障,最终将设计的 MBIST 电路应用于 16 nm 存储器测试中,测试结果表明,本文提出的新的动态故障测试算法能够有效探测到 FinFET 工艺中可能出现的新型的单存储单元的动态故障.

表 5 芯片测试结果 (typical, 25℃, VDD=0.88 V)

IP 名称与容量	Die0	Die1	Die2	Die3	Die4
SPW720B160M4B1	Pass	Pass	Pass	Pass	Pass
SPW1568B160M4B1	Pass	Pass	Fail	Pass	Pass
SPW9600B24M16B2	Pass	Pass	Pass	Pass	Pass
SPW1568B144M4B2	Pass	Pass	Pass	Pass	Pass
SPW1280B13M8B1	Pass	Pass	Pass	Pass	Pass
R1W22B142M1B1	Pass	Pass	Pass	Pass	Pass
R1W256B144M1B1	Pass	Pass	Pass	Pass	Pass
R1W1024B144M1B1	Fail	Pass	Pass	Pass	Pass
R1W32B144M2B1	Pass	Pass	Pass	Pass	Pass
R1W512B144M1B1	Pass	Pass	Pass	Pass	Pass
良率	90%	100%	90%	100%	100%

参考文献:

[1] HARUTYUNYAN G, TSHAGHARYAN G, VARDANIAN V, et al. Fault modeling and test algorithm creation strategy for FinFET-based memories[C]//Vlsi Test Symposium. IEEE, 2014;1-6.

[2] HAMDIOUI S, AL-ARS Z, GOOR A J V D. Testing Static and Dynamic Faults in Random Access Memories[C]// IEEE Vlsi Test Symposium. IEEE Computer Society, 2002;395.

[3] BENSO A, BOSIO A, CARLO S D, et al. Automatic March tests generation for static and dynamic faults in SRAMs [C]//Test Symposium, 2005. European. IEEE, 2005;122-127.

[4] HARUTUNYAN G, VARDANIAN V A, ZORIAN

Y. Minimal March Tests for Detection of Dynamic Faults in Random Access Memories [J]. Journal of Electronic Testing, 2007, 23(1):55-74.

[5] DU L, ZHAO H, YANG W, et al. Optimization of STI oxide recess uniformity for FinFET beyond 20nm [C]// Semiconductor Technology International Conference. IEEE, 2015;1-4.

[6] JOSEPH J, PATRIKAR R. Impact of Fin Width and Graded Channel Doping on the Performance of 22nm SOI FinFET [M]//VLSI Design and Test. Springer Berlin Heidelberg, 2013;153-159.

[7] HARTYUNYAN G, SHOUKOURIAN S, VARDANIAN V, et al. Extending fault periodicity table for testing faults in memories under 20nm [C]//IEEE East-West Design& Test Symposium. IEEE, 2015; 1-4.

[8] HAMDIOUI S, GAYDADJIEV G N, GOOR A J V. A Fault Primitive Based Analysis of Dynamic Memory Faults[C]//IEEE 14 Annual Workshop on Circuits, Systems and Signal Processing. Veldhoven, Netherlands. IEEE, 2003;84-89.

[9] 石磊, 王小力. 一种基于存储器故障原语的 March 测试算法研究[J]. 微电子学, 2009, 39(2);251-255.

[10] MARINESCU M. Simple and Efficient Algorithms for Functional RAM Testing[C]//International Test Conference 1982, Philadelphia, Pa, USA. DBLP, 1982; 236-239.

[11] BOSIO A, DILILLO L, GIRARD P, et al. Advanced Test Methods for SRAMs [M]. Springer US, 2010.

作者简介:

桑胜男 男, (1993-), 硕士研究生. 研究方向为集成电路设计与测试. E-mail: ssnsang@163. com.

张立军 男, (1971-), 博士, 研究员, 博士生导师. 研究方向为 SOC 设计及方法学.

(上接第 16 页)

[9] 王宜怀, 朱仕浪, 郭芸. 嵌入式技术基础与实践 (第 4 版)[M]. 北京: 清华大学出版社, 2017.

[10] 姜鲜桃. 双序列比对 Needleman-Wunsch 算法研究 [D]. 呼和浩特: 内蒙古农业大学, 2017.

作者简介:

陈成 男, (1993-), 硕士研究生. 研究方向为嵌入式系统

与物联网. E-mail: 20165227004@stu. suda. edu. cn.

王宜怀 男, (1962-), 博士, 教授. 研究方向为嵌入式系统、传感网与智能控制技术.

钱涵佳 女, (1994-), 硕士研究生. 研究方向为嵌入式系统与物联网.

彭涛 男, (1989-), 博士研究生. 研究方向为物联网与图像处理.