

# 一种邻近层资源共享的 三维堆叠存储器内建自修复策略

崔小乐<sup>1)</sup> 张世界<sup>1)</sup> 张 强<sup>1)</sup> 金玉丰<sup>1),2)</sup>

<sup>1)</sup>(北京大学深圳研究生院集成微系统重点实验室 广东 深圳 518055)

<sup>2)</sup>(北京大学微纳电子研究院 北京 100871)

**摘 要** 在摩尔定律面临终结的趋势下,三维集成电路技术被认为是继续提升集成电路性能和集成度的重要技术途径之一.由于采用堆叠的结构,三维集成电路适用于高密度以及异质集成应用领域,存储器是高密度集成电路的典型代表,是三维集成电路的重要应用方向之一.三维存储器技术可同时提高存储密度与访存通路带宽,是解决“存储墙”问题的一种可行技术途径.然而,由于存储器件尺寸的微缩,存储阵列中的故障存储单元数量呈增加趋势.对存储器产品,基于冗余存储资源的内建自修复技术是提高其可靠性的重要方法.三维存储器的层与层之间可通过硅通孔等技术实现互连,使得垂直方向上的冗余资源共享成为可能,从而改善三维存储器的可靠性.该文提出一种邻近层共享冗余的三维存储器修复策略,用以提高现有三维存储器内建自修复技术的故障修复能力.该策略不会引发死锁现象,具有良好的冗余资源利用率和较小的硅通孔面积代价.仿真结果表明,与已有的结对冗余策略相比,该文所提出的冗余共享策略具有更高的故障修复率,且故障修复率不随存储器层数的增加而显著下降,更适用于大规模三维存储器.

**关键词** 三维存储器;内建自修复;层间冗余共享;邻近层冗余共享

中图法分类号 TP393 DOI号 10.11897/SP.J.1016.2017.02030

## A BISR Scheme for 3D Stacked Memory by Sharing Adjacent Redundancy Cells Across Dies

CUI Xiao-Le<sup>1)</sup> ZHANG Shi-Jie<sup>1)</sup> ZHANG Qiang<sup>1)</sup> JIN Yu-Feng<sup>1),2)</sup>

<sup>1)</sup>(Key Laboratory of Integrated Microsystems, Peking University Shenzhen Graduate School, Shenzhen, Guangdong 518055)

<sup>2)</sup>(Institute of Microelectronics, Peking University, Beijing 100871)

**Abstract** Three-dimensional (3D) integrated circuit (IC) is regarded as an important way for the enhancement on the performance and integration density of IC to cope with the challenge of “the ending of the Moore’s law”. The 3D IC is suitable for the high density or heterogenous integration applications for its multi-layer stacked structure. Memory is one of the important applications of 3D integration technology for it is a typical high density product. 3D memory, which increases memory density and bandwidth of memory access path simultaneously, is a promising solution of the “memory wall problem”. With the scaling down of memory devices, the amount of faulty cells in memory array increases rapidly, it gives rise to the requirement of the build in self-repair (BISR) technique for acceptable memory reliability. The through silicon vias

收稿日期:2015-07-13;在线出版日期:2016-04-04.本课题得到国家“九七三”重点基础研究发展规划项目基金(2015CB057201)、广东省自然科学基金(2015A030313147)、深圳市基础研究项目(JCYJ20140417144423194, JCYJ20140417144423198)资助.崔小乐,男,1975年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为芯片测试技术、容错计算. E-mail: cuixl@pkusz.edu.cn.张世界,男,1990年生,硕士研究生,主要研究方向为芯片容错设计.张 强,男,1990年生,硕士研究生,主要研究方向为芯片测试.金玉丰,男,1961年生,博士,教授,主要研究领域为芯片封装技术、三维集成电路、MEMS技术.

(TSVs), which works as the vertical signal paths in 3D memory, enable the inter-die sharing techniques of redundant resources for higher utilization ratio. This work proposes a sharing strategy of adjacent redundant cells across dies for better repair capability. This strategy is deadlock free, and it has good utility of redundant resources and relatively small area overhead of through silicon vias (TSVs). Simulation results show that, comparing with the published die-pair sharing strategy, the proposed strategy obtains higher repair rate. Furthermore, the proposed BISR has no obvious drop of repair rate with respect to the increase of the layer count, implying a good applicability in the large scale 3D memories.

**Keywords** 3D memory; built-in self-repair; inter-die redundancy sharing; sharing adjacent redundancy cells across dies

## 1 引言

硅通孔(Through Silicon Via, TSV)是当前备受关注的三维集成电路层间互连技术。基于 TSV 技术,可以实现堆叠芯片在垂直方向的信号互连,大大缩短互连延迟,提高系统性能和集成度<sup>[1]</sup>。存储器是高集成度芯片的代表,是三维集成技术的重要应用方向之一。三维存储器可同时提高访存带宽和存储密度,可有效应对因计算机性能提升所面临的“存储墙”问题,故受到业界和学界的关注。三星公司已经利用 TSV 技术制造出 8 层堆叠的 DRAM 存储器<sup>[2-4]</sup>,推出后立即引起了业界的巨大反响。2012 年,北京大学也完成了国内首款基于 TSV 技术的三维 SRAM 样片的设计与制造,成倍提高了存储密度<sup>[5]</sup>。

由于存储单元的密度高,存储器件尺寸小,在制造中引入的物理缺陷容易产生故障,导致部分存储单元无法正常工作。自修复是目前常用的提升存储器整体可靠性的技术手段。备用的冗余存储器行/列被加入到存储阵列中。当发现存储单元故障后,故障单元可以被备用存储器单元所替代,从而实现修复功能。存储器的修复问题已经被证明是一个 NP 难问题,虽然可通过穷尽搜索获得最优解,但是需消耗大量时间<sup>[6]</sup>。基于对面积开销以及性能的权衡考虑,通常冗余替换是以行或列为单位进行的。加入到芯片中的备用资源数量是有限的,为了使得存储器获得有效的修复,研究者们已开发多种修复方法<sup>[7-8]</sup>,以期在修复率约束下减少修复时间。其中,ESP(Essential Spare Pivoting algorithms)算法<sup>[8]</sup>由于无需使用全局故障点阵信息,节省存储故障信息所占用的面积,目前被普遍应用。

由于驱动电压的限制,在二维存储器产品中一般包含有多个存储器块。若每个存储块都备有独享的备用资源,则备用资源的利用率可能较低。若使相邻的存储块可共享其冗余存储资源进行修复,可提高备用资源的利用率,但是该策略的布线开销大,较难实施于传统的二维存储电路<sup>[9]</sup>。在三维存储器中,连接垂直方向的 TSV 较二维芯片的布线短很多,这不但可使层间信号互连性能得以改善,同时也会使备用资源的层间共享变得可行,可使修复率大为提高<sup>[10]</sup>。

三维存储器冗余修复的总体策略分为 4 类:层内冗余(Intra-Layer Redundancy)自修复、层间冗余(Inter-Layer Redundancy)自修复、整层冗余(Layer Redundancy)自修复和全局冗余(Global Redundancy)自修复。其中层内冗余自修复策略与二维存储器的冗余自修复方式相同,仅利用同层的备用资源修复故障单元,不同的层之间不能共享冗余资源,因此修复能力受限于本层冗余资源数量,且易造成冗余资源的浪费<sup>[11]</sup>;层间冗余自修复是指本层的备用资源不仅仅供给本层的故障单元使用,也可以被别的层使用,冗余资源利用率较高,同时提高了存储器可靠性。典型案例如文献<sup>[10]</sup>中所提出的结对层间冗余共享方案。整层冗余自修复策略设置冗余的备用层,当某一层出现故障时,该层可被备用层整层替代。显然这种策略会造成较大的资源浪费,在商用化的三维存储器产品中很难应用,目前主要用来修复使用晶圆堆叠方式(Wafer to Wafer, W2W)下的三维存储器<sup>[12]</sup>。全局冗余自修复是指存储器中的所有冗余资源都集中设置在一个备用层中,其余层共享这一备用层的冗余资源,可以得到近似 100%的高修复率<sup>[13-16]</sup>。需注意的是,由于需要利用 TSV 将所有的备用存储单元连接到每一层,全局冗余自修复策略

所需的 TSV 的数量比较多,面积开销较大. 在 22 nm 的 MOS 工艺技术下,6 管的 SRAM 标准单元的面积已经小至  $0.081 \mu\text{m}^2$ <sup>[17]</sup>,而晶体管数量更少的 DRAM、Flash 存储器所占面积更小. 然而,根据国际半导体技术发展路线图(International Technology Roadmap for Semiconductors, ITRS)的预测,2018 年 TSV 直径最低仅可达到  $0.5 \mu\text{m}$ <sup>①</sup>,圆形 TSV 在芯片上所占面积最小为  $0.785 \mu\text{m}^2$ . 这说明 TSV 尺寸微缩的速度相对比较慢,故相比于存储单元,TSV 的面积占比较大. 这意味着 TSV 数量对于存储器的面积有较大的影响. 因此,虽然全局冗余策略的修复能力强,但是面积开销决定了全局冗余策略在目前并不适用. 权衡故障修复率、资源利用率和面积代价等因素,层间冗余自修复策略是一类具有优势的策略.

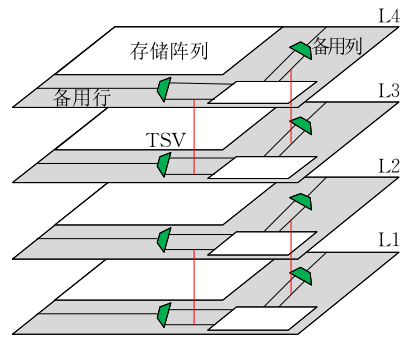
基于层间冗余存储资源共享的思路,本文提出一种邻近层冗余共享的修复策略,并开展实验验证工作. 论文结构如下:第 2 节介绍邻近层冗余共享策略;第 3 节介绍基于该策略的内建自修复(Build-In Self Repair, BISR)设计,并论证其有效性;第 4 节给出仿真实验的结果;第 5 节为本文的结论.

## 2 邻近层冗余共享策略

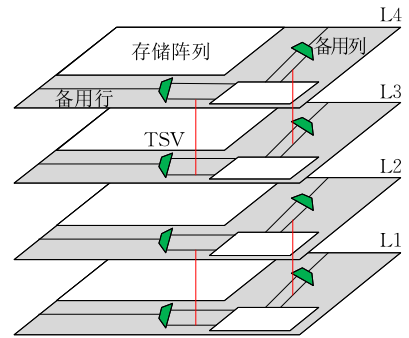
### 2.1 研究动机

层间冗余策略可有效利用冗余资源,获得较高的故障修复率,而其面积开销相对较小,是一类具有应用价值的三维存储器自修复策略. 然而,目前采用这类策略的具体方案尚不多见,只有文献[10]给出一种层间冗余共享的具体修复方案,如图 1(a)所示. 三维存储器中的单层芯片两两结成一组,两个芯片之间可以进行冗余资源的共享,但组间并无冗余共享机制,我们可称其为结对共享策略. 图 1(a)中, L1 层与 L2 层, L3 层与 L4 层分别结成冗余资源共享组,其中 L1、L2、L3 和 L4 分别代表三维存储器中的第 1 层到第 4 层存储器芯片.

然而,文献[10]着重讨论了堆叠前的芯片选取匹配(Die Matching)问题,故障修复在组堆叠之后,整体堆叠之前进行,未给出三维存储器整体堆叠完成后的故障修复方案. 文献[10]将同组的两片芯片的故障点阵图进行合并,通过错误信息共享形成组内存储器故障点阵图,从而将双层芯片的修复问题转换成单片存储器的修复问题. 通常基于存储器故障点阵图的修复方法需要消耗较多的存储资源来保



(a) 结对冗余共享的示意图<sup>[10]</sup>



(b) 邻近层冗余共享的示意图

图 1

存故障信息,实现代价较大. 该策略每一组所能共享的冗余资源仅是两层上的冗余资源. 假如 L1 层和 L2 层没有故障存在,而 L3 层和 L4 层的冗余资源不足以修复芯片上的故障,此时虽然 L1 和 L2 层上的冗余资源并未使用,但仍会导致存储器的整体不可被修复. 这说明结对冗余共享修复策略的故障修复能力取决于组内冗余资源的配置. 要提高其故障修复能力需要增加组内冗余资源配置,但这又有可能造成较低的冗余资源利用率. 究其原因,主要还是由于组间没有冗余共享机制造成冗余资源的低利用率问题.

针对以上问题,本文提出一种新的冗余资源共享策略,通过增加组间共享来增加对冗余资源的利用率,从而达到提高修复能力的目的.

### 2.2 邻近层冗余共享策略

增加层间冗余资源共享需要增加 TSV 数量,会造成面积上的代价. 综合权衡面积开销和整体修复能力,本文提出一种邻近层冗余共享策略,将相邻层上的冗余资源进行连接,并使每一层存储器都可以获得邻近层的冗余资源,如图 1(b)所示. L1 层与 L2 层共享冗余资源, L2 层可以与 L1 层和 L3 层共享

① International Technology Roadmap for Semiconductors (ITRS)- Interconnect, <http://www.itrs.net>, 2013

冗余资源,L3层可以与L2层和L4层共享冗余资源,L4层与L3层共享资源.最上层和最下层芯片的邻近层只有一层,可获得的冗余资源数量与结对冗余共享策略相同.处在中间层的芯片除了本层的冗余资源外,还可以获得相邻两层存储器芯片的冗余资源,以期提高存储器的修复率.

邻近层共享冗余共享策略相比于结对冗余共享策略添加了组间连接的 TSV,增加了每一层芯片可共享的平均资源数量.由于邻近层冗余共享策略中所有邻近层都可以共享冗余资源,因此该策略对在堆叠前(Pre bonding)、部分堆叠(Partial stack)和整体堆叠后(Post bonding)所发现的故障存储单元的修复均可适用,在适用阶段范围方面明显优于结对冗余共享策略.

TSV 在芯片上所占的面积是一个重要的层间冗余共享成本.假设每个存储块的备用行数量为  $m$ ,备用列数量为  $n$ ,三维存储器的堆叠芯片层数为  $L$ ,每个 TSV 在芯片上所占面积为  $S$ ,结对冗余共享策略、全局冗余修复策略及本文提出的邻近层冗余共享策略的冗余资源互连所需的 TSV 总数和 TSV 在每一层所占的面积如表 1 所示.由于不同层间的 TSV 位置相互对准,对于一层芯片,一面设置 TSV 和两面同时设置 TSV 所占的芯片面积是相同的.虽然邻近层冗余共享策略使用 TSV 总数多于结对冗余共享策略,但是在芯片面积上与结对冗余共享策

略相比并没有增加,故该策略并非通过增大芯片面积换取冗余资源利用率的提高,可用性良好.

表 1 用于共享资源的 TSV 数量和面积的对比

	结对冗余	全局冗余	邻近层冗余
TSV 数	$L(m+n)/2$	$(L-1)L(m+n)$	$(L-1)(m+n)$
每层上 TSV 所占面积	$(m+n)S$	$L(m+n)S$	$(m+n)S$

### 3 邻近层冗余共享策略的 BISR

#### 3.1 三维存储器的 BISR

三维存储器中的每一层设置相同的 BISR 电路结构,如图 2 所示,包含内建自测试、故障分析、故障列表等模块.内建自测试模块对存储器芯片中的存储单元进行测试,故障分析模块完成故障的分类,故障信息存入故障列表中.故障分为行故障(Row Fault)、列故障(Column Fault)和正交单个单元故障(Orthogonal Fault)<sup>[18]</sup>.其中行故障使用备用行替代,列故障使用备用列替代,正交单个单元故障可使用备用行或备用列进行替代.为了提高修复能力,先施行故障、列故障修复,然后再进行正交单个单元故障修复.该 BISR 结构与二维存储器的 BISR 结构类似,重要的区别之处在于其中的冗余资源包括本层的冗余资源以及当前可用的邻近层中的冗余资源,通过 TSV 实现层间冗余资源信息共享.

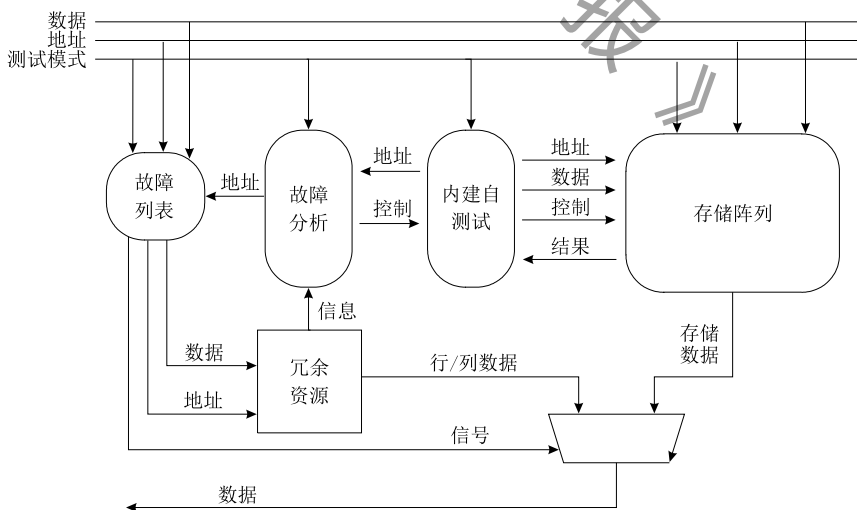


图 2 每一层的 BISR 结构

存储器每次上电时,都会进入测试模式进行自检并完成故障修复,然后转入正常工作模式.在正常工作模式下时,当访存操作指向故障单元时就会通过查询故障列表,用冗余资源进行替代,从而将对故障单元的读写操作重定位到对应的冗余单元.本文

所用 BISR 结构与传统的二维芯片的结构相似,不同之处在于邻近层冗余资源的共享.

共享信息的硬件开销主要表现在 TSV 在芯片上所占面积.邻近层冗余共享修复方式所需 TSV 分为 3 类:

第 1 类 TSV 用于冗余资源互连,如表 1 中所分析的结果,每一层所需要的数量为  $m+n$ .

第 2 类 TSV 传输表示本层冗余资源是否足够用于修复本层的故障存储单元的标记信号. 当该 TSV 信号值为 0 时,表示本层冗余资源不够,对于与之连接的邻近层来说,这意味着这一层对其的请求信号. 当信号值为 1 时,表示本层冗余资源够用,邻近层根据此信号可知这一层有冗余资源剩余,即可向这一层申请冗余资源. 冗余行和冗余列各需要 1 个标志位,故共需两根 TSV.

第 3 类 TSV 用于表示冗余资源申请数量或者剩余冗余资源数量. 当本层冗余资源是否够用的标志信号值为 1 时,第 3 类 TSV 上的信号值表示本层剩余冗余资源的数量;反之,可用于表示向相邻层申请冗余资源的数量. 每 1 层接收到的冗余资源申请总数为所有相邻层发来的冗余资源的申请数量之和. 冗余资源申请信息包含行修复申请、列修复申请和正交单个单元故障修复申请. 每 1 层向外发出的行修复申请数量不大于冗余行数量  $m$ ,列修复申请数量不大于冗余列数量  $n$ ,正交单个单元故障修复申请数量不大于  $m+n$ .

因此,除去用于冗余资源互连的 TSV,基于邻近层修复策略 BISR 结构共需要的 TSV 数量为  $\log_2 m + \log_2 n + \log_2(m+n) + 2$ . 关于结对冗余和全局冗余的相关文献中并未分析这类 TSV 开销,但由其原理可知,结对冗余策略的 BISR 在组内层间也需要与邻近层冗余相同数量的第 2 类和第 3 类 TSV 开销,但组间层之间无需设置 TSV. 对于全局冗余策略而言,由于需要标定冗余资源申请来自于哪一层,因此所需的第 2 类和第 3 类 TSV 开销必然大于本案的开销.

### 3.2 邻近层冗余共享的修复流程

基于邻近层冗余共享策略的存储器测试修复流程如图 3 所示. 每一层芯片同时测试,对检测出的故障单元进行故障分类,统计出每一层有多少故障行、故障列和正交单个单元故障,然后进行冗余分析,并通过地址重置操作完成修复. 本文提出的邻近层冗余共享策略使得在修复时的可用备用单元可能在本层中,也可能在邻近层中. 因此在修复流程中规定,当前层有修复需要时,优先使用下一层的冗余资源,其次使用本层的冗余资源,最后再申请使用上一层的冗余资源. 这种基于层优先级的顺序申请方式可有效避免冗余资源申请中的死锁现象,提高冗余资源的使用效率.

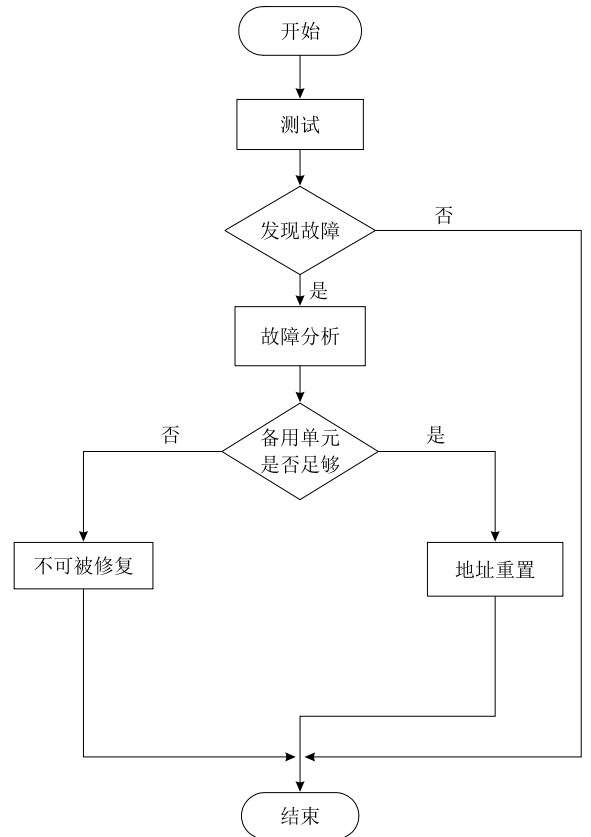


图 3 存储器测试修复流程

邻近层冗余共享修复算法请见算法 1.

#### 算法 1. 邻近层共享冗余资源算法.

输入: 每层芯片的冗余资源数  $Redundancy[level]$  和故障数  $Fault[level]$ , 其中  $level = \{1, 2, \dots, L\}$

输出: 地址映射关系  $Addressmapping[level]$ , 其中  $level = \{1, 2, \dots, L\}$

```

IF  $Redundancy[1] > Fault[1]$  THEN
  //冗余资源数大于故障数
  {  $Counter[1] = Redundancy[1] - Fault[1]$ ;
    //计算剩余冗余资源数
     $Flag[1] = 1$ ; //标志设置为可修复
     $Repair(1)$ ; //对第一层存储器进行修复
     $Addressmapping[1]$ ;
    //故障地址和冗余单元形成重映射
  }
ELSE
  {  $Flag[1] = 0$ ; //标志设置为不可修复
     $Counter[1] = Fault[1] - Redundancy[1]$ ;
    //计算所需申请的冗余资源数
  }
ENDIF
FOR  $level = 2:L$  //对第 2 至第 L 层存储器进行修复
  { IF  $Flag[level-1] = 0$  THEN
    //下一层不可被修复
  
```

```

{Repair(level-1); //对下一层进行修复
Addressmapping[level-1];
//形成地址重映射
Redundancy[level]=Redundancy[level]-
Counter[level-1];
//计算本层余剩冗余资源数
}
ELSE
{IF (level>2) THEN
{Redundancy[level]=Redundancy[level]+
Counter[level-1]-
Redundancy[level-2];
//计算本层的可用冗余资源数,本层为第3层
//及以上层
}
ELSE
{
Redundancy[level]=Redundancy[level]+
Counter[level-1];
//计算第2层的可用冗余资源数
}
ENDIF
}
ENDIF //更新本层资源数操作结束
IF Redundancy[level]>fault[level] THEN
//本层冗余资源数大于故障数
{Flag[level]=1; //标志设置为可修复
Repair(level); //修复本层存储器
Addressmapping[level]; //形成地址重映射
Counter[level]=Redundancy[level]-
Fault[level];
//计算剩余冗余资源
IF level==L THEN
{Return(1);} //全局可修复
ENDIF
}
ELSE
{Flag[level]=0; //本层不可被修复
Counter[level]=Fault[level]-
Redundancy[level];
//需要借用的资源数量
IF level==L THEN
{Return(0);} //全局不可修复
ENDIF
}
ENDIF
}
ENDFOR

```

### 3.3 修复策略的有效性

首先讨论不同修复策略的不可被修复条件. 假设各层的行故障数为  $Fm_i$ , 列故障数为  $Fn_i$ , 正交单个单元故障数为  $Fs_i$ , 其中  $i$  取值为  $[1 \sim L]$ , 备用行、列数量分别为  $m, n$ . 结对冗余和邻近层冗余策略的不可被修复的最宽松条件如表 2 所示. 根据表 2 可知, 相对于结对冗余修复策略, 邻近层冗余修复策略的单层最大可修复的故障数较多. 其原因是, 结对冗余修复策略在两层芯片故障总数大于两层的冗余资源总数时必然不可被修复, 但是邻近层修复策略在此情况下可以从第 3 层存储器中申请冗余资源, 故尚有一定修复的可能. 可见, 邻近层冗余修复策略的修复能力优于结对冗余修复策略.

表 2 不可被修复的最宽松条件

	结对冗余	邻近层冗余
行不可被修复条件	$Fm_i > 2m$	$Fm_i > 3m$
列不可被修复条件	$Fn_i > 2n$	$Fn_i > 3n$
正交单个单元故障不可被修复条件	$\sum_{i=1}^2 Fs_i > 2(m+n)$	$\sum_{i=1}^L Fs_i > 3(m+n)$

全局冗余共享方式的单层不可修复条件与多层不可修复条件相同, 都是  $\sum_{i=1}^L (Fn_i + Fm_i + Fs_i) > L(m+n)$ . 因此当故障数小于冗余资源数时, 采用全局冗余策略可以修复所有故障. 邻近层冗余共享策略的修复能力劣于全局冗余策略. 但是全局冗余策略所需的 TSV 数量与邻近冗余共享策略相比高出数倍, 因此实际上是用面积代价换取修复能力的提升, 对于面积约束严格的存储器类产品而言, 具有相当大的设计实现难度.

BISR 机制的性能指标可用存储器内建自修复时间说明. 内建自修复时间包含了存储器测试时间  $T_{\text{test}}$  和存储器修复时间  $T_{\text{repair}}$  两部分. 测试时间由测试算法的算法复杂度决定. 为了节省测试时间, 可对每一层存储芯片同时进行测试, 故三维存储器的测试时间等同于一层存储器芯片的测试时间, 即三维存储器总测试时间为  $T_{\text{test}}$ . 在邻近层冗余共享策略下, 故障修复过程是从下层向上层串行进行的, 故总修复时间为  $LT_{\text{repair}}$ . 因此, 邻近层冗余共享策略的内建自修复时间为  $T_{\text{test}} + LT_{\text{repair}}$ . 对于结对冗余共享而言, 其修复时间在组内是串行执行的, 在组间可并行进行, 因此内建自修复的总时间为  $T_{\text{test}} + 2T_{\text{repair}}$ , 性能优于邻近层冗余共享. 然而, 在内建自修复的总时间中, 测试时间远大于修复时间<sup>[19]</sup>, 即  $T_{\text{test}} \gg T_{\text{repair}}$ , 存储器的自修复时间主要由测试时间决定. 因此虽然邻近层冗余共享策略的性能略逊于

结对冗余共享,但相差不是很大.

## 4 实验与结果分析

### 4.1 实验方案

如式(1)所示的修复率是已修复的故障数与总故障数的比率,反映了冗余修复策略在存储器故障修复能力方面的优劣.本文以修复率为指标,比较不同冗余共享修复策略的有效性.

$$\text{修复率} = \frac{\text{已修复故障数}}{\text{总故障数}} \times 100\% \quad (1)$$

实验方案具体步骤如下.

#### 步骤 1. 故障注入

Polya-Eggenberger 分布是进行集成电路良率模型分析时的常用概率分布<sup>[20-21]</sup>,本实验采用此分布类型在存储器阵列中进行故障注入.故障注入采用固定故障模型(Stuck-At Fault Model),每一片存储器阵列中注入故障的数量是随机的,故障的数量符合 Polya-Eggenberger 分布.故障在存储阵列中的位置采用均匀分布进行随机化处理.参照文献<sup>[22-23]</sup>,故障注入时使正交单个单元故障数占总故障数的 70%,行故障占比为 15%,列故障占比为 15%.

#### 步骤 2. 冗余分析

由于故障注入后,故障行、故障列和正交单个单元故障的位置已知,故本实验省去测试过程.采用表 2 设置的条件,开展冗余分析,明确当前冗余资源配置是否可以修复注入的故障.

#### 步骤 3. 冗余替换与修复率计算

分别采用本文所提出的邻近层冗余共享策略和结对冗余共享策略进行故障单元修复.为节省故障信息存储空间的大小,本文选用 ESP 算法进行故障修复,对应于算法 1 中的 Repair(level)操作.对修复结果进行统计,并根据式(1)计算修复率.

### 4.2 实验结果分析

修复结果与存储器阵列规模、冗余资源配置数量、三维存储器堆叠层数以及故障注入数量均相关.本文分别考虑上述变化因素,对邻近层冗余共享策略与结对冗余共享策略的故障修复结果进行仿真,结果如表 3 所示.其中,  $F_{\max}$  和  $F_{\text{avg}}$  分别表示各层中注入的最大故障数量和平均故障数量,  $R$  和  $C$  表示各层中的冗余行数量和冗余列数量.考虑到故障注入的随机性对修复结果的影响,在每种条件下都进行了 10000 组实验,取平均结果填入表 3 中.由于两种共享策略在堆叠层数为 2 时并无区别,所以本文未对 2 层堆叠情形进行修复仿真.为了方便与结对冗余共享策略进行对比,实验中三维存储器堆叠层数取为偶数.

表 3 三维存储器故障修复仿真结果

存储阵列 规模	故障数				不同层数下的修复率/%							
	故障数		备用资源数		4层		6层		8层		10层	
	$F_{\max}$	$F_{\text{avg}}$	$R$	$C$	结对冗余	本文方法	结对冗余	本文方法	结对冗余	本文方法	结对冗余	本文方法
32×32	5	1.86	1	1	87.18	91.18	85.99	91.56	85.47	91.63	85.25	91.66
	8	2.92	1	1	66.75	69.78	66.63	69.83	64.07	68.04	63.77	67.93
	8	2.92	2	2	95.99	97.70	95.67	98.16	95.44	98.36	95.33	98.49
64×64	10	3.66	2	2	89.11	92.20	88.05	92.50	87.63	92.59	87.48	92.72
	10	3.66	3	3	98.94	99.70	98.79	99.76	98.75	99.81	98.66	99.84
	15	5.14	3	3	92.22	94.57	91.63	95.01	91.41	95.39	91.16	95.41
128×128	15	5.14	4	4	98.75	99.48	98.52	99.57	98.54	99.66	98.33	99.62
	20	6.55	4	4	93.87	95.77	93.29	96.17	93.21	96.58	92.99	96.77
	25	7.65	4	4	88.45	90.52	87.74	90.99	87.28	91.20	87.26	91.48
	25	7.65	5	5	95.96	97.21	95.56	97.58	95.31	97.84	95.28	97.99
	30	8.71	5	5	92.50	93.94	91.89	94.41	91.52	94.55	90.78	94.64
256×256	30	8.71	6	6	97.28	98.00	97.00	98.37	96.89	98.56	96.03	98.65
	33	11.04	6	6	90.04	92.50	89.27	92.88	88.71	92.93	88.49	93.02
	35	11.49	6	6	88.34	90.89	87.65	91.24	87.09	91.28	86.77	91.37
	35	11.49	7	7	93.92	95.86	93.36	96.26	93.22	96.62	92.89	96.61
	40	12.60	7	7	91.20	93.17	90.62	93.51	90.17	93.86	89.86	93.81
512×512	40	12.60	8	8	95.26	96.74	94.84	97.07	94.58	97.41	94.45	97.50
	45	17.38	9	9	87.79	91.10	87.08	91.49	86.65	91.66	86.35	91.68
	45	17.38	10	10	91.76	94.76	91.23	95.24	90.92	95.44	90.82	95.66
	48	18.21	10	10	90.24	93.27	89.55	93.73	89.22	93.88	88.82	93.91
	48	18.21	11	11	93.49	96.14	92.89	96.48	92.66	96.80	92.51	96.82
	53	19.56	10	10	87.45	90.57	86.54	90.67	85.99	90.60	85.68	90.63
	53	19.56	11	11	91.29	94.02	90.60	94.42	90.19	94.63	90.09	94.76

(续 表)

存储阵列 规模	故障数		备用资源数		不同层数下的修复率/%							
	$F_{\max}$	$F_{\text{avg}}$	R	C	4层		6层		8层		10层	
					结对冗余	本文方法	结对冗余	本文方法	结对冗余	本文方法	结对冗余	本文方法
1024×1024	55	21.07	11	11	89.12	92.28	88.37	92.70	87.94	92.75	87.75	92.80
	55	21.07	12	12	92.45	95.22	91.90	95.63	91.69	96.04	91.37	96.04
	60	22.50	12	12	89.91	93.01	89.48	93.53	89.10	93.65	88.87	93.75
	60	22.50	13	13	91.01	93.85	90.39	94.28	90.20	94.50	89.96	94.59
	65	23.77	12	12	87.74	90.82	86.92	90.95	86.67	91.30	86.30	91.20
2048×2048	65	23.77	13	13	91.17	93.88	90.46	94.27	90.17	94.49	90.01	94.55
	80	24.95	13	13	89.91	92.82	89.35	93.36	88.99	93.52	88.71	93.61
	80	24.95	14	14	92.98	95.49	92.47	95.83	92.23	96.05	92.07	96.26
	100	29.29	15	15	89.55	92.47	88.81	93.00	88.57	93.26	88.33	93.35
	100	29.29	16	16	91.99	94.61	91.64	95.34	91.44	95.63	91.21	95.78

由表 3 可知,本文所提出邻近层冗余共享策略的修复率在任何情况下都大于结对冗余共享策略的对应结果,说明该策略的修复能力优于结对冗余共享策略.就冗余资源数量的影响而言,当备用资源数量小于平均故障数量时,邻近层冗余共享策略的修复率高于结对冗余共享策略.相同故障数量条件下,备用资源数量越多,两种共享方式的修复率之差也就越小.此外,结果数据表明,随着堆叠层数的增加,本文所提出的策略相比于结对冗余策略的修复率也会有更大的提高.

根据 ITRS 预测,2018 年的三维 DRAM 存储器的堆叠层数将会达到 8~16 层,因此本文进一步对 1024×1024 的存储阵列分别进行了 4~16 层情形的修复率仿真,如图 4 所示.设置每一层的最大故障数为 55,备用行数量和备用列数量均为 12,经过 10000 次仿真后统计得到的平均每层的故障数为 21.07 个.由图 4 可见,随着堆叠芯片层数的增加,邻近层冗余共享策略的修复率随之增加,而结对冗余共享策略的修复率随之下降,说明邻近层冗余共享策略更加适合于大规模的三维存储器集成应用.

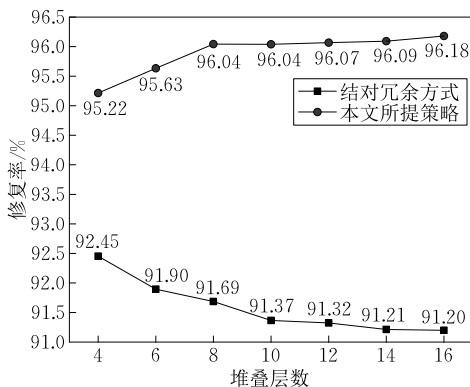


图 4 1024×1024 存储器阵列在备用行列数为 12 时在不同层数条件下的修复率

在上述实验中,每层冗余资源总数为 24,大于每层平均故障数 21.07,满足最宽松的可修复条件.

为了考察每层冗余资源数量小于平均故障数的情况下两种共享策略的优劣,本文将仿真条件中的备用行列数目都改设为 10,保持其它条件不变,再次进行 10000 次仿真,获取平均修复率结果,如图 5 所示.图 5 中的数据 displays,当冗余资源数量小于平均故障数量时,两种共享策略的修复率均有所下降.但随着层数的增加,邻近层冗余共享策略修复率的下降幅度小于结对冗余共享策略,两者的差值随着层数增加而拉大.由此结果可得,邻近层冗余共享策略明显优于结对冗余共享策略,亦反映出两种策略对冗余资源利用率的差别.

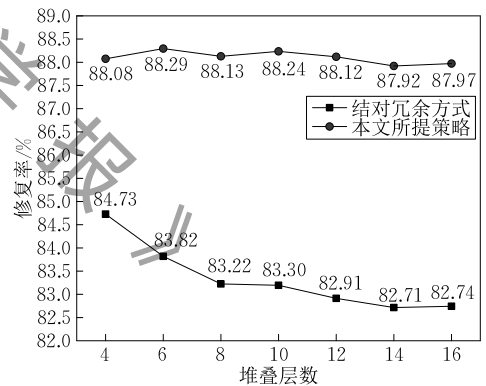


图 5 1024×1024 存储器阵列在备用行列数为 10 时在不同层数条件下的修复率

## 5 结 论

本文提出一种邻近层冗余共享的三维存储器故障修复方法.由于可在层间共享冗余备份的存储行/列,因此本方法具有较高的故障修复率.实验结果表明,无论冗余资源数量是否大于平均故障数量,本方法的修复率结果均优于结对冗余共享策略,且本方法的修复率水平随着三维存储器堆叠层数的增加而增加,可以适用于未来不断增长的三维存储器集成密度.本方法所需的每层 TSV 面积开销与结对冗余



共享策略相当,具有良好的可实现性.此外,本策略中的冗余资源共享方案是动态调整的,故既能用于堆叠前的单片存储器故障修复,也能用于堆叠后的三维存储器故障修复,可覆盖三维存储器开发的全阶段.

### 参 考 文 献

- [1] Davis W R, Wilson J, Mick S, et al. Demystifying 3D ICs: the pros and cons of going vertical. *IEEE Design & Test of Computers*, 2005, 22(6): 498-510
- [2] Kang U, Chung H J, Heo S, et al. 8GB 3-D DDR3 DRAM using through-silicon-via technology. *IEEE Journal of Solid-State Circuits*, 2010, 45(1): 111-119
- [3] Sekiguchi T, Ono K, Kotabe A, et al. 1-Tbyte/s 1-Gbit DRAM architecture using 3-D interconnect for high-throughput computing. *IEEE Journal of Solid-State Circuits*, 2011, 46(4): 828-837
- [4] Kim J S, Oh C S, Lee H, et al. A 1-2 V 12.8GB/s 2GB Mobile Wide-I/O DRAM with 4 128 I/Os using TSV based stacking. *IEEE Journal of Solid-State Circuits*, 2012, 47(1): 107-116
- [5] Ma S, Sun X, Zhu Y, et al. Design and process development of a stacked SRAM memory chip module with TSV interconnection//*Proceedings of the IEEE Electronic Components and Technology Conference (ECTC)*. San Diego, America, 2012: 1925-1929
- [6] Kuo S Y, Fuchs W K. Efficient spare allocation for reconfigurable arrays. *IEEE Design & Test of Computers*, 1987, 4(1): 24-31
- [7] Huang R F, Li J F, Yeh J C, et al. A simulator for evaluating redundancy analysis algorithms of repairable embedded memories//*Proceedings of the IEEE International Workshop on Memory Technology, Design and Testing (MTDT)*. Isle of Bendor, France, 2002: 68-73
- [8] Huang C T, Wu C F, Li J F, et al. Built-in redundancy analysis for memory yield improvement. *IEEE Transactions on Reliability*, 2003, 52(4): 386-399
- [9] Yamagata T, Sato H, Fujita K, et al. A distributed globally replaceable redundancy scheme for sub-half-micron ULSI memories and beyond. *IEEE Journal of Solid-State Circuits*, 1996, 31(2): 195-201
- [10] Jiang L, Ye R, Xu Q. Yield enhancement for 3D-stacked memory by redundancy sharing across dies//*Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*. San Jose, America, 2010: 230-234
- [11] Chang Y J, Huang Y J, Li J F. A built-in redundancy-analysis scheme for RAMs with 3D redundancy//*Proceedings of the IEEE VLSI Design, Automation and Test (VLSI-DAT)*. Hsinchu, China, 2011: 1-4
- [12] Taouil M, Hamdioui S. Layer redundancy based yield improvement for 3D wafer-to-wafer stacked memories//*Proceedings of the IEEE European Test Symposium (ETS)*. Trondheim, Norway, 2011: 45-50
- [13] Wang X, Vasudevan D, Lee H H S. Global built-in self-repair for 3D memories with redundancy sharing and parallel testing//*Proceedings of the IEEE 3D Systems Integration Conference (3DIC)*. Osaka, Japan, 2012: 1-8
- [14] Chi C C, Chou Y F, Kwai D M, et al. 3D-IC BISR for stacked memories using cross-die spares//*Proceedings of the VLSI Design, Automation, and Test (VLSI-DAT)*. Hsinchu, China, 2012: 1-4
- [15] Lee C, Kang W, Cho D, et al. A new fuse architecture and a new post-share redundancy scheme for yield enhancement in 3-D-stacked memories. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2014, 33(5): 786-797
- [16] Lee J, Park K, Kang S. Yield enhancement techniques for 3D memories by redundancy sharing among all layers. *ETRI Journal*, 2012, 34(3): 388-398
- [17] Smith K C, Wang A, Fujino L C. Through the looking glass: Trend tracking for ISSCC 2012. *IEEE Solid-State Circuits Magazine*, 2012, 4(1): 4-20
- [18] Wang L T, Wu C W, Wen X Q. *VLSI Test Principles and Architectures Design for Testability*. San Francisco, USA: Elsevier Inc., 2006: 530
- [19] Kang W, Lee C, Lim H, et al. A 3 dimensional built-in self-repair scheme for yield improvement of 3 dimensional memories. *IEEE Transactions on Reliability*, 2015, 64(2): 586-595
- [20] Stapper C H. On a composite model to the IC yield problem. *IEEE Journal of Solid-State Circuits*, 1975, 10(6): 537-539
- [21] Stapper C H, McLaren A N, Dreckmann M. Yield model for productivity optimization of VLSI memory chips with redundancy and partially good product. *IBM Journal of Research and Development*, 1980, 24(3): 398-409
- [22] Huang R F, Li J F, Yeh J C, et al. A simulator for evaluating redundancy analysis algorithms of repairable embedded memories//*Proceedings of the Memory Technology, Design and Testing (MTDT)*. Los Alamitos, USA, 2002: 68-73
- [23] Huang R F, Li J F, Yeh J C, et al. Raisin: Redundancy analysis algorithm simulation. *IEEE Design & Test of Computers*, 2007, 24(4): 386-396



**CUI Xiao-Le**, born in 1975, Ph. D., associate professor. His research interests include VLSI testing and fault tolerant computing.

**ZHANG Shi-Jie**, born in 1990, M. S. candidate. His research interest is fault tolerant design of 3D ICs.

**ZHANG Qiang**, born in 1990, M. S. candidate. His research interest is testing of 3D ICs.

**JIN Yu-Feng**, born in 1961, Ph. D., professor. His research interests include 3D ICs, VLSI packaging and MEMS technology.

## Background

This work studies the problem of build in self-repair (BISR) technique for 3D memory. This topic is very fresh since the 3D memory is an emerging memory technology. Although BISR for traditional 2D memory can be transplanted into the 3D counterpart, researchers are finding new methods taken the advantages of 3D memory. Till now, only few publications provide solutions about BISR for 3D memory, a large search space is still waiting to be explored. This paper proposes a sharing strategy of adjacent redundancy cells across dies for 3D memory, to obtain better repair rate of the BISR.

This work is supported by the National Basic Research Program (973 Program) of China "Fundamental research on wafer-level 3D integration for 20/14nm ICs" (No. 2015CB057201). This project is aiming at laying a manufacturing technology

foundation for 3D IC. 3D memory is one of the sampling chips to demonstrate the semiconductor technologies developed in this project. The work in this paper is aiming at a reliable design of the 3D memory. Besides that, the work is also supported by the Nature Science Foundation of Guangdong Province (No. 2015A030313147) and the R&D Projects of Shenzhen Government (Nos. JCYJ20140417144423194, JCYJ20140417144423198), the targets are developing test and design technology of 3D ICs respectively.

Our group has worked in the field of 3D ICs for years. We have mastered the TSV technology, interposer technology, and have designed the first 3D memory sample chip in China. The work in this paper can provide a good solution of the reliability enhancement of the 3D memory.