

相邻层冗余共享的 三维存储器成品率优化方法

刘 军^{1,2}, 朱承强^{1,2}, 吴 奎^{1,2}, 王 伟^{1,2}, 任福继³

(1. 合肥工业大学计算机与信息学院, 安徽合肥 230009;
2. 情感计算与先进智能机器安徽省重点实验室, 安徽合肥 230009;
3. 日本德岛大学先端技术科学教育部, 德岛常三岛町 7708502)

摘 要: 存储裸片堆叠方案和冗余共享策略对提高三维存储器成品率有重要影响. 为提高三维存储器的成品率并且减少行列冗余所需的 TSVs 数量, 提出了一种相邻层冗余共享结构. 该冗余共享结构使得每层存储裸片的行列冗余不仅能被本层使用, 而且能被相邻层使用. 并在此结构的基础上, 提出了一种新的存储裸片堆叠方案. 该方案通过构建存储裸片的选择限制条件, 每次选中适合的存储裸片来堆叠三维存储器以充分利用行列冗余. 实验结果表明, 与国际上同类方法相比, 所提方案有效地提高了三维存储器的成品率, 并且减少了行列冗余所需的 TSVs 数量.

关键词: 三维存储器; 冗余共享; 成品率; 存储裸片

中图分类号: TP306+.2

文献标识码: A

文章编号: 0372-2112 (2018)03-0629-07

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2018.03.017

Yield Optimization Technique for Three Dimensional Memory Based on Redundancy Sharing Among Adjacent Layers

LIU Jun^{1,2}, ZHU Cheng-qiang^{1,2}, WU Xi^{1,2}, WANG Wei^{1,2}, REN Fu-ji³

(1. School of Computer and Information, Hefei University of Technology, Hefei, Anhui 230009, China;

2 Anhui Province Key Laboratory of Affective Computing and Advanced Intelligent Machine, Hefei, Anhui 230009, China;

3 Department of Information Science & Intelligent Systems, Faculty of Engineering,
The University of Tokushima, Minamijosanjima, Tokushima 7708502, Japan)

Abstract: Memory dies stacking strategy and redundancy sharing structure have great effects on the yield of three-dimensional (3D) memory. To improve 3D memory yield and reduce the number of TSVs consumed by row and column redundancies, this paper proposed a redundancy sharing structure among adjacent layers. In the proposed sharing structure, the redundancies in each layer not only can be used to repair the faults in the layer where the redundancies reside, but also can be utilized by adjacent layers. On the basis of this proposed structure, a new die-stacking strategy is presented. Through formalized selection conditions for memory dies, the presented strategy can choose suitable memory dies to stack 3D memory. In this way, the row and column redundancies are fully used. Experimental results illustrate the proposed redundancy sharing structure and die-stacking strategy can effectively improve 3D memory yield and reduce the number of TSVs consumed by spare rows and columns.

Key words: 3D memory; redundancy sharing; yield; memory dies

1 引言

三维集成电路通过 TSVs (Through Silicon Vias) 在垂直方向上将芯片进行堆叠能够有效地提升电路的性

能, 减少互连的延迟和功耗, 具有广泛的应用前景^[1,2]. 基于三维集成电路技术的存储器已成功的应用于商业中, 在高性能计算等领域有着重要的应用价值^[3,4]. 三维存储器通过 TSVs 来实现各层存储裸片 (memory

dies)间的信号传输,有效的提高了信号的传输带宽^[4].但由于三维存储器存储容量以及存储密度较大,使得三维存储器在制造过程中很容易产生故障.此外,在三维存储器的绑定(bonding)过程中也会额外的引入故障,因而三维存储器的成品率很低.成品率问题也成为三维存储器设计和制造过程中需要解决的一个关键问题.

为了提高存储器的成品率,二维存储器通常采用行列冗余修复的方式^[5].行列冗余修复是指当存储单元出现故障时,将此故障单元所在的行或列的地址映射到冗余的行或列,从而达到修复存储器的目的.在二维存储器中,为了充分利用行列冗余进一步提高成品率,可以采用存储块与存储块之间冗余共享的方式^[6].这是因为存储裸片中的故障不是均匀分布的,有的存储块故障较多,所含的行列冗余不够用;而有的存储块故障较少,剩余大量的行列冗余.通过存储块间冗余共享的方式可以充分的利用行列冗余,提高存储器的成品率,但此方法不仅增加了信号的传输延迟,还增大了布线的难度,硬件开销很大.

但在三维存储器中,由于各层存储裸片是通过TSVs进行互连的,而TSVs的传输延迟较低,因而在三维存储器的修复中可以采用层间冗余共享的策略(即任一存储裸片的冗余不仅能被本层使用,还可以被其它层的存储裸片使用),并且对三维存储器的性能影响很小.Chou等人提出了三维存储器层间冗余共享的内建自修复(Built-In Self-Repair)方案和存储裸片匹配策略,提高了三维存储器的成品率^[7,8].Xu等人根据存储裸片的故障位图,将存储裸片的堆叠问题转化成二分图的匹配问题,有效的提高了三维存储器成品率^[9].虽然文献[7~9]的方案能够提高三维存储器的成品率,但这些方案仅适用于两层冗余共享结构(即存储裸片的冗余仅能被本层和另外一层存储裸片使用).为此, Lee等人提出了全局冗余共享修复方案(即存储裸片的冗余不仅能被本层使用,而且能被其它所有层使用),进一步提高了三维存储器的成品率^[10].但是该方案极大地增加了TSVs的数量.

为提高三维存储器的成品率,同时减少TSVs数量,本文提出了一种相邻层冗余共享方案(Redundancy Sharing among Adjacent Layers),简称RSAL方案.相邻层冗余共享是指存储裸片的冗余不仅能被本层使用,而且可以被上下相邻层的存储裸片使用.并且针对相邻层冗余共享结构,本文提出了一种新的存储裸片选择方案.该方案通过构建存储裸片的选择限制条件,每次选中适合的存储裸片来堆叠三维存储器以充分利用行列冗余.与已有方案相比,RSAL方案有效的提高了三维存储器的成品率,并且减少了行列冗余共享所需的TS-

Vs数量.

2 背景知识

本部分首先介绍了存储裸片的分类,然后介绍了三维存储器的堆叠流程.

2.1 存储裸片分类

根据存储裸片是否能被修复,存储裸片可分为四种基本类型^[10,11]:

(1)无故障存储裸片:是指没有任何故障的存储裸片,此类存储裸片不需要使用其所含的行列冗余来修复;

(2)可自修复存储裸片:可以通过自身的行列冗余来修复,此类存储裸片还有可能剩余部分行列冗余;

(3)层间冗余共享可修复存储裸片:通过自身的行列冗余不能被修复,但可以通过借用其它层存储裸片的剩余冗余来完成修复;

(4)不可修复存储裸片:通过自身的行列冗余不能被修复,通过借用其它层的行列冗余也不被修复的存储裸片.

如果三维存储器不使用层间冗余共享策略,那么只能选择无故障存储裸片和可自修复的存储裸片进行堆叠.若使用层间冗余共享策略,那么就可以选择无故障存储裸片、可自修复的存储裸片以及冗余共享可修复存储裸片来堆叠三维存储器,使得存储裸片的行列冗余得到了充分利用,提高了三维存储器的成品率.

2.2 三维存储器堆叠流程

三维存储器可以采用三种堆叠方式:晶元对晶元(W2W)、存储裸片对晶元(D2W)和存储裸片对存储裸片(D2D)^[12-14].每种集成方法都有自身的优缺点,W2W集成技术制造工艺简单,一次可生产成千上百个存储器.但是,W2W集成技术堆叠的三维存储器成品率很低.D2W和D2D集成技术与W2W相比制造工艺较复杂,然而成品率相对较高.本文提出的RSAL方案适用于D2D集成技术.

基于D2D集成技术的三维存储器堆叠流程图如图1所示.在绑定前测试与修复过程中,将存储裸片分为四种类型,即无故障存储裸片、可自修复存储裸片、冗余共享可修复存储裸片和不可修复存储裸片.在堆叠过程中可以选择无故障存储裸片、可自修复存储裸片和冗余共享可修复存储裸片进行堆叠.而在堆叠的过程中,将哪些存储裸片进行堆叠对于成品率有很大的影响.例如有四个存储裸片(A,B,C,D),其中存储裸片A可自修复,且剩余两个行冗余;存储裸片B和C可自修复,但没有剩余的行列冗余;存储裸片D不能自修复,还需要两个行冗余才能被修复.如果选择存储裸片A与B堆叠,存储裸片C与D堆叠,那么最终的三维存储器

成品率只有 50%。然而,如果选择存储裸片 A 与 D 堆叠,存储裸片 B 与 C 堆叠,那么三维存储器成品率为 100%。

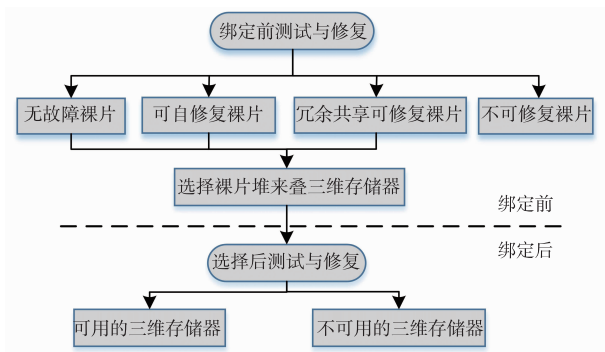


图1 三维存储器绑定流程图

3 RSAL 方案的结构及流程

本节首先介绍 RSAL 方案的冗余共享结构,然后详细介绍 RSAL 方案的完整流程。

3.1 RSAL 方案的冗余共享结构

在 RSAL 方案中,每个存储裸片的行列冗余不仅被本层存储裸片使用,而且可以被上下两层的存储裸片使用,如图 2 所示。RSAL 方案的优势是其成品率要高于两层冗余共享,与全局冗余共享结构的成品率接近。而连接行列冗余所需的 TSVs 数量虽然略多于两层冗余共享,但要比全局冗余共享结构少很多。如在图 2 中,四个存储裸片的相邻层冗余共享需要使用 6 根 TSVs,而两层冗余和全局冗余共享结构则分别需要 4 根和 18 根 TSVs。

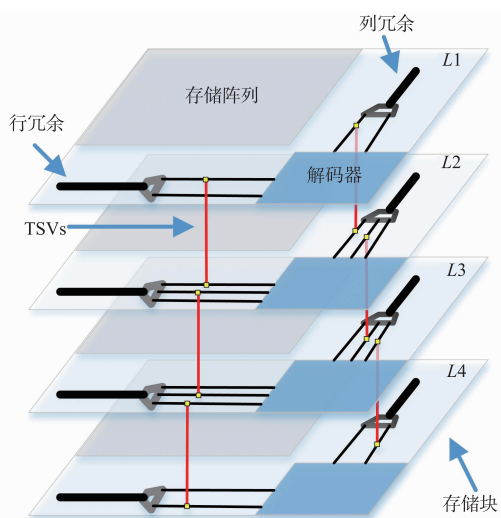


图2 RSAL方案共享结构

当三维存储器含有两层存储裸片,则 RSAL 方案与两层冗余共享以及全局冗余共享所需的 TSVs 数量相同。但若三维存储器堆叠层数大于或等于 3 时,RSAL 方

案需要使用 TSVs 数量虽然略多于两层冗余共享方案,但仅占全局冗余共享方案的 $1/(L-1)$,其中 L 为三维存储器的层数。

3.2 RSAL 方案的流程

RSAL 方案首先对存储裸片进行绑定前的测试与修复,然后再通过构建选择条件选择合适的存储裸片进行堆叠,以充分利用行列冗余提高三维存储器的成品率。

3.2.1 绑定前测试与修复

此步对存储裸片进行绑定前的测试与修复,得到每个存储裸片需要多少行冗余和列冗余才能被修复,所需的行冗余和列冗余数量分别记为 RF 和 CF 。然后根据 RF 和 CF 的值对存储裸片进行分类,抛弃不可修复的存储裸片。

假设每个存储裸片有两个行冗余和两个列冗余,在 RSAL 方案中只有相邻层的冗余才能被共享,因此一个存储裸片最多能够使用 6 个行冗余和 6 个列冗余。若一个存储裸片 $RF \geq 7$ 或 $CF \geq 7$,则意味着该存储裸片通过借用相邻层的存储冗余也不能够被修复,此类存储裸片将被抛弃。

3.2.2 选择存储裸片进行堆叠

绑定前测试与修复结束后,接着就是如何选择存储裸片进行堆叠。假设每个存储裸片中含有 S_r 个行冗余和 S_c 个列冗余,经过绑定前测试和修复后还剩余 N 个存储裸片,其中第 i ($1 \leq i \leq N$) 个存储裸片需要 RF_i 个行冗余和 CF_i 个列冗余才能够被修复,每个三维存储器含有 L 层存储裸片。RSAL 方案的存储裸片堆叠流程如图 3 所示,下面按照图 3 所示流程来具体介绍。

图 3 中首先将 N 个存储裸片按照 RF 和 CF 的数值之和进行降序排列,并将存储裸片依次记录到状态表中。状态表的作用是存放存储裸片的状态信息,若一个存储裸片被选中作为三维存储器的一层,则此存储裸片处于“选中”状态,否则此存储裸片处于“未选中”状态。初始时,还未开始选择存储裸片,因而所有存储裸片的状态都是“未选中”状态。

存储裸片降序排列后,RSAL 方案每次选中一个满足条件的存储裸片作为三维存储器的第 i ($1 \leq i \leq L$) 层。若没有满足条件的存储裸片,那么就将 RF 和 CF 数值之和最大的存储裸片从状态表中删除。当有 L 个存储裸片被选中时,意味着三维存储器堆叠成功,需要从状态表中删除被选中的这 L 个存储裸片。接下来判断状态表中存储裸片的数量是否小于 L ,若小于 L ,表明剩余的存储裸片不够堆叠成一个三维存储器,因此程序结束。否则,重新开始下一次堆叠。

在三维存储器堆叠过程中,为了使行列冗余能够得到充分的利用,存储裸片的选取必须满足一定的条

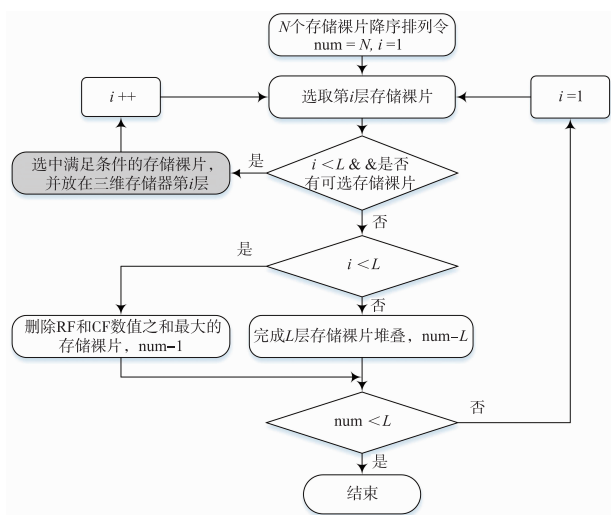


图3 存储裸片堆叠流程

件,下面来介绍存储裸片的选择条件.由于第一层存储裸片只能使用自身及第二层存储裸片的行列冗余,因此 RF_1 和 CF_1 要满足以下条件:

$$RF_1 \leq 2S_r \quad (1)$$

$$CF_1 \leq 2S_c \quad (2)$$

RSAL 方案在降序排列的状态表中选中第一个满足式(1)和式(2)条件的存储裸片作为三维存储器的第一层.由于首先选取的是需要行列冗余较多的存储裸片,因此剩下存储裸片所需的行列冗余较少,能够堆叠成一个三维存储器的概率就更大,从而使得三维存储器成品率更高.

当选择第 i ($2 \leq i \leq L-1$) 层存储裸片时,由于第 i 层存储裸片不仅能使用自身的冗余,还能使用第 $i-1$ 层及第 $i+1$ 层存储裸片的行列冗余,因此三维存储器的第 1 层到第 i 层总共能够使用的行冗余和冗余数量分别为 $(i+1) \times S_r$ 和 $(i+1) \times S_c$. 在选择第 i 层存储裸片时, RF_i 和 CF_i 的值要满足如下条件:

$$RF_1 + RF_2 + \dots + RF_i \leq (i+1) \times S_r \quad (3)$$

$$CF_1 + CF_2 + \dots + CF_i \leq (i+1) \times S_c \quad (4)$$

同样的,RSAL 方案在降序排列的状态表中选中第一个满足式(3)和式(4)条件的存储裸片作为三维存储器的第 i 层.

当选择第 L 层存储裸片时,由于 L 个存储裸片总共含有 $L \times S_r$ 个行冗余和 $L \times S_c$ 个列冗余,为保证 L 个存储裸片都能被修复,因此需要满足如下条件:

$$RF_1 + RF_2 + \dots + RF_L \leq L \times S_r \quad (5)$$

$$CF_1 + CF_2 + \dots + CF_L \leq L \times S_c \quad (6)$$

下面通过图 4 的例子来讲述存储裸片选择的过程.假设有 6 个存储裸片,每个存储裸片有两个行冗余和两个列冗余,需要从 6 个存储裸片中选出 4 个存储裸片来堆叠一个三维存储器.在绑定前测试与修复过程中,得

到每个存储裸片的 RF 和 CF 值,按照存 RF 和 CF 之和进行降序排序,如图 4 (a) 所示.根据图 4 所描述的存储裸片堆叠流程,当选择三维存储器的第 1 层存储裸片时,如图 4 (b) 所示,有 5 个存储裸片 (B, C, D, E, F) 满足条件,由于存储裸片 B 所需的行列冗余数量最多,因此选中存储裸片 B 放在第 1 层.当选择第 2 层存储裸片时,如图 4 (c) 所示,有 5 个存储裸片 (A, C, D, E, F) 满足条件,由于存储裸片 A 所需的行列冗余数量最多,因此存储裸片 A 被选中.接下来选择第 3 层时,如图 4 (d) 所示,满足条件的有 C, D, E 和 F 这 4 个存储裸片,存储裸片 C 被选中.最后选择第四层存储裸片时,有存储裸片 E, F 满足条件,因此选中存储裸片 E 放在第 4 层,如图 4 (e) 所示.最终,存储裸片 B, A, C, E 被堆叠成一个三维存储器.

Die A	Die A	Die C	Die D	Die E	Die F
RF=5	RF=1	RF=2	RF=1	RF=0	RF=0
CF=2	CF=4	CF=1	CF=1	CF=1	CF=0

(a) 存储裸片的存储修复信息

裸片分类图							状态表		裸片分类图							状态表					
CF	RF	0	1	2	3	4	5	6	Die	状态	CF	RF	0	1	2	3	4	5	6	Die	状态
0	F	E							A	排除	0	F	E						A	选中	
1		D			B				B	选中	1		D						C	排除	
2			C						C	排除	2			C					D	排除	
3									D	排除	3								E	排除	
4									E	排除	4								F	排除	
5				A					F	排除	5				A						
6											6										

(b) 选择第1层存储裸片

(c) 选择第2层存储裸片

裸片分类图							状态表		裸片分类图							状态表					
CF	RF	0	1	2	3	4	5	6	Die	状态	CF	RF	0	1	2	3	4	5	6	Die	状态
0	F	E							C	选中	0	F	E						D	排除	
1		D							D	排除	1		D						E	选中	
2			C						E	排除	2								F	排除	
3									F	排除	3										
4											4										
5											5										
6											6										

(d) 选择第3层存储裸片

(e) 选择第4层存储裸片

图4 存储裸片选择方案实例

4 实验结果与分析

本部分对 RSAL 方案、文献[8]的两层冗余共享方案和文献[10]的全局冗余共享方案进行了比较.实验中使用了 1000 个存储裸片,在三维存储器的层数分别为 4 和 8 这两种情况下对三维存储器的成品率以及冗余行列共享所需的 TSVs 数量进行了比较.实验中假定每个存储裸片有 1024×1024 bit, 存储裸

片的故障分布服从泊松分布.泊松分布有两种情况:(1) 聚集故障分布,其中 $\lambda = 8, \alpha = 2.382$; (2) 均匀故障分布,其中 $\lambda = 8, \alpha = 0.6232$. 由于三维存储器堆叠层数为 2 时情况一样,因此本文不考虑堆叠层数为 2 的情况.

表 1 给出了 RSAL 方案与文献[8]方案在不考虑 TSVs 成品率情况下三维存储器成品率的实验结果. 由表 1 可知,RSAL 方案的成品率在任何情况下都要高于文献[8]方案,随着堆叠层数的增加,文献[8]方案的成品率变化不明显,然而 RSAL 方案的成品率提高幅度很明显. 就冗余数量对三维存储器成品率的影响而言,冗余数量越多,RSAL 方案的成品率与文献[8]方案差距越大. 此外,由表 1 数据可知,故障分布呈现均匀分布的情况下 RSAL 方案的成品率更高. 表 2 给出了 RSAL 方案与文献[10]方案在不考虑 TSVs 成品率情况下三维存储器成品率的比较结果. 由表 2 可知,RSAL 方案的成品率与文献[10]方案接近. 就冗余数量对三维存储器成品率的影响而言,冗余数量越多,RSAL 方案的成品率与文献[10]方案差距越小. 此外,由表 2 数据可知,

故障分布呈现均匀分布的情况下两种方案的成品率更高.

表 3 给出了 RSAL 方案与文献[8]所需 TSVs 的比较结果,由表 3 可知,RSAL 方案所需 TSVs 数量略高于文献[8]方案,当堆叠层数越多,RSAL 方案所需 TSVs 数量越多. 表 4 给出了 RSAL 方案与文献[10]所需 TSVs 的比较结果,由表 4 可知,RSAL 方案所需 TSVs 数量要远远小于文献[10]方案,随着三维存储器堆叠层数的增加,RSAL 方案所需 TSVs 数量文献[10]方案相比降低比率更大.

以上表 1 和表 2 中三维存储器的成品率只考虑了存储单元的成品率,未考虑 TSVs 成品率. 若将 TSVs 成品率考虑进去,三维存储器的成品率 Y_{stack} 可由式(7)来计算:

表 1 RSAL 方案成品率与文献[8]方案实验比较结果(不考虑 TSVs 成品率)

行列冗余 $S_r \times S_c$		三维存储器不同条件下成品率											
		$\lambda = 8, \alpha = 2.382$						$\lambda = 8, \alpha = 0.6232$					
		4 层			8 层			4 层			8 层		
S_r	S_c	文献[8]	RSAL	提高比率	文献[8]	RSAL	提高比率	文献[8]	RSAL	提高比率	文献[8]	RSAL	提高比率
2	2	31.20%	36.65%	5.45%	30.38%	40.38%	10.00%	35.25%	39.82%	4.57%	34.16%	41.62%	7.46%
2	3	41.28%	56.82%	15.54%	40.89%	61.46%	20.57%	44.71%	59.47%	14.76%	43.98%	66.08%	22.10%
3	3	52.34%	77.86%	25.52%	52.34%	83.83%	31.49%	55.76%	81.03%	25.27%	60.36%	87.87%	27.51%
3	4	60.25%	87.48%	27.23%	68.06%	97.52%	29.46%	65.26%	90.23%	24.97%	71.08%	98.35%	27.27%

表 2 RSAL 方案成品率与文献[10]方案实验比较结果(不考虑 TSVs 成品率)

行列冗余 $S_r \times S_c$		三维存储器不同条件下成品率											
		$\lambda = 8, \alpha = 2.382$						$\lambda = 8, \alpha = 0.6232$					
		4 层			8 层			4 层			8 层		
S_r	S_c	文献[10]	RSAL	降低比率	文献[10]	RSAL	降低比率	文献[10]	RSAL	降低比率	文献[10]	RSAL	降低比率
2	2	42.43%	36.65%	5.78%	44.26%	40.38%	3.88%	43.74%	39.82%	3.92%	47.82%	41.62%	6.20%
2	3	61.38%	56.82%	4.56%	66.78%	61.46%	5.32%	63.42%	59.47%	3.95%	70.25%	66.08%	4.17%
3	3	81.52%	77.86%	3.66%	85.26%	83.83%	1.43%	84.98%	81.03%	3.95%	90.43%	87.87%	2.56%
3	4	89.34%	87.48%	1.86%	98.83%	97.52%	1.31%	92.42%	90.23%	2.19%	99.64%	98.35%	1.29%

表 3 RSAL 方案与文献[8]所需 TSVs 的比较结果

行列冗余 $S_r \times S_c$		TSVs 数量					
		4 层			8 层		
S_r	S_c	文献[8]	RSAL	提高比率	文献[8]	RSAL	提高比率
2	2	8	12	50%	16	28	75%
2	3	10	15	50%	20	35	75%
3	3	12	18	50%	24	42	75%
3	4	14	21	50%	28	49	75%

表 4 RSAL 方案与文献[10]所需 TSVs 的比较结果

行列冗余 $S_r \times S_c$		TSVs 数量					
		4 层			8 层		
S_r	S_c	文献[10]	RSAL	降低比率	文献[10]	RSAL	降低比率
2	2	36	12	66.67%	196	28	85.71%
2	3	45	15	66.67%	245	35	85.71%
3	3	54	18	66.67%	294	42	85.71%
3	4	63	21	66.67%	343	49	85.71%

$$Y_{\text{stack}} = Y_{\text{c-mem}} \cdot Y_{\text{tsv-mem}} \cdot Y_{\text{tsv-red}} \quad (7)$$

式(7)中 $Y_{\text{c-mem}}$ 表示存储单元的成品率, $Y_{\text{tsv-mem}}$ 表示存储单元中 TSVs 的成品率, $Y_{\text{tsv-red}}$ 表示行列冗余中 TSVs 的成品率. $Y_{\text{tsv-red}}$ 可由式(8)来计算:

$$Y_{\text{tsv-red}} = (1 - f_{\text{tsv}})^{N_{\text{tsv}}} \quad (8)$$

式(8)中 f_{tsv} 表示 TSVs 故障率, N_{tsv} 表示冗余行列所需的 TSVs 数量.

由于三种方案所使用相同存储单元,因而 $Y_{\text{tsv-mem}}$ 相同,但由于冗余共享结构的不同,因而 $Y_{\text{c-mem}}$ 和 $Y_{\text{tsv-red}}$ 的值不相同. 考虑 TSVs 情况下 RSAL 方案分别与文献[8]方案和文献[10]方案在不同条件下成品率的比较结果如表 5 和表 6 所示,其中 TSVs 故障率 $f_{\text{tsv}} = 0.0005$. 从表 5 中可以看出,考虑 TSVs 的成品率后,RSAL 方案的成品率仍然高于文献[8]方案的成品率. 而且随着堆叠层数的增加,RSAL 方案与文献[8]相比,成品率提高的比率更大. 此外,就冗余数量对三维存储器成品率的影响而言,冗余数量越多,RSAL 方案与文献[8]方案相比,成品率提高的比率也是越来越大.

表 5 RSAL 方案成品率与文献[8]方案实验比较结果(考虑 TSVs 成品率)

行列冗余 $S_r \times S_c$		三维存储器不同条件下成品率											
		$\lambda = 8, \alpha = 2.382$						$\lambda = 8, \alpha = 0.6232$					
		4 层			8 层			4 层			8 层		
S_r	S_c	文献[8]	RSAL	提高比率	文献[8]	RSAL	提高比率	文献[8]	RSAL	提高比率	文献[8]	RSAL	提高比率
2	2	31.08%	36.43%	5.35%	30.17%	39.81%	9.64%	35.11%	35.58%	0.47%	33.92%	41.04%	7.12%
2	3	41.07%	56.42%	15.35%	40.48%	60.42%	19.94%	44.49%	59.07%	14.58%	43.54%	64.96%	21.42%
3	3	52.03%	77.16%	25.13%	51.71%	82.07%	30.36%	55.43%	80.03%	24.60%	59.71%	84.02%	24.31%
3	4	59.83%	86.61%	26.78%	67.11%	95.18%	28.07%	64.80%	89.33%	24.53%	70.84%	95.99%	25.15%

表 6 RSAL 方案成品率与文献[10]方案实验比较结果(考虑 TSVs 成品率)

行列冗余 $S_r \times S_c$		三维存储器不同条件下成品率											
		$\lambda = 8, \alpha = 2.382$						$\lambda = 8, \alpha = 0.6232$					
		4 层			8 层			4 层			8 层		
S_r	S_c	文献[10]	RSAL	提高比率	文献[10]	RSAL	提高比率	文献[10]	RSAL	提高比率	文献[10]	RSAL	提高比率
2	2	41.67%	36.43%	-5.24%	38.33%	39.81%	1.48%	42.95%	35.58%	-7.37%	43.37%	41.04%	-2.33%
2	3	60.03%	56.42%	-3.61%	59.10%	60.42%	1.32%	62.02%	59.07%	-2.95%	62.17%	64.96%	2.79%
3	3	79.32%	77.16%	-2.16%	73.58%	82.07%	8.49%	82.69%	80.03%	-2.66%	78.04%	84.02%	5.98%
3	4	86.57%	86.61%	0.04%	83.21%	95.18%	11.97%	89.55%	89.33%	-0.22%	83.90%	95.99%	12.09%

5 总结

本文提出了一种 RSAL 方案,该方案采用相邻层冗余共享结构,即存储裸片的冗余不仅能被本层使用,而且可以被上下相邻层的存储裸片使用. 然后在此结构基础上,提出了一种新的裸片选择算法,该算法适用于行列冗余修复的三维存储结构. 通过构建存储裸片的

由表 6 可知,考虑 TSVs 的成品率后,当堆叠层数为 4 层时,RSAL 方案的成品率要低于文献[10]方案. 但是在故障呈聚集型分布,且 $S_r = 3, S_c = 4$ 的条件下,RSAL 方案的成品率高于文献[10]方案. 这是因为行列冗余数增加时,文献[10]方案所需 TSVs 数量也增加,导致 TSVs 成品率下降,造成三维存储器成品率的下降. 当堆叠层数达到 8 层时,RSAL 方案的成品率要高于文献[10]方案. 这是因为三维存储器层数较多时,文献[10]方案中行列冗余所需的 TSVs 数量较多,造成 TSVs 成品率下降,从而导致三维存储器的成品率降低. 然而在故障呈均匀分布,且 $S_r = 2, S_c = 2$ 的条件下,此时 RSAL 方案的成品率要低于文献[10]方案. 主要因为冗余数较少,用来连接冗余的 TSVs 数量较少,此时 TSVs 的成品率对三维存储器成品率影响不大. 此外,就冗余数量对三维存储器成品率影响而言,随着冗余数量的增加,RSAL 方案的成品率的增长幅度要大于文献[10]方案.

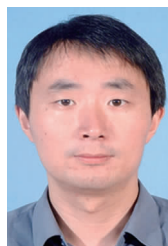
选择限制条件,每次选中适合的存储裸片来堆叠三维存储器以充分利用行列冗余,提高三维存储器的成品率. 实验结果表明,与两层冗余共享方案相比,RSAL 方案的 TSVs 数量虽然略有增加,却显著地提高了三维存储器的成品率. 与全局冗余共享方案相比,RSAL 方案所需的 TSVs 数量明显减少,并且随着三维存储器层数的增加,TSVs 数量减少的也越多. 当三维存储器的层数

较高时,综合考虑存储单元和 TSVs 的成品率,RSAL 方案的成品率要高于全局冗余共享方案的成品率。

参考文献

- [1] 常郝,梁华国,蒋翠云,等.一种 3D 堆叠集成电路中间绑定测试时间优化方案[J].电子学报,2015,43(2):393-398.
Chang Hao,Liang Hua-guo,Jiang Cui-yun,et al. Optimization scheme for mid-bond test time on 3-D stacked ICs [J]. Acta Electronic Sinica,2015,43(2):393-398. (in Chinese)
- [2] Knickerbocker J U,Patel C S,Andry P S,et al. 3-D silicon integration and silicon packaging technology using silicon through-vias [J]. IEEE Journal of Solid-State Circuits,2006,41(8):1718-1725.
- [3] Iyer S S,Kirihata T,Wordeman M R,et al. Process-design considerations for three dimensional memory integration [A]. Proceedings of IEEE Symposium on VLSI Technology [C]. Honolulu:IEEE Press,2009. 60-63.
- [4] Sun H,Liu J,Anigundi R S,et al. 3D DRAM design and application to 3D multicore systems [J]. IEEE Design & Test of Computers,2009,26(5):36-47.
- [5] Lu S K,Tsai Y C,Hsu C H,et al. Efficient built-in redundancy analysis for embedded memories with 2-D redundancy [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems,2006,14(1):34-42.
- [6] Bahl S. A sharable built-in self-repair for semiconductor memories with 2-D redundancy scheme [A]. Proceedings of IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems [C]. Rome:IEEE Press,2007. 331-339.
- [7] Chou C W,Huang Y J,Li J F. A built-in self-repair scheme for 3-D rams with inter die redundancy [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,2013,32(4):572-583.
- [8] Chou C W,Huang Y J,Li J F. Yield-enhancement techniques for 3D random access memories [A]. Proceedings of IEEE International Symposium on VLSI Design Automation and Test [C]. Hsin Chu: IEEE Press,2010. 104-107.
- [9] Jiang L,Ye R,Xu Q. Yield enhancement for 3D-stacked memory by redundancy sharing across dies [A]. Proceedings of IEEE International Conference on Computer-Aided Design [C]. San Jose:IEEE Press,2010. 230-234.
- [10] Lee J,Park K,Kang S. Yield enhancement techniques for 3D memories by redundancy sharing among all layers [J]. ETRI Journal,2012,34(3):388-398.
- [11] Lee J,Park K,Kang S. A die-selection method using search-space conditions for yield enhancement in 3D memory [J]. ETRI Journal,2011,33(6):904-913.
- [12] Reda S,Smith G,Smith L. Maximizing the functional yield of wafer-to-wafer 3-D integration [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems,2009,17(9):1357-1362.
- [13] Papanikolaou A,Dimitrios S,Riko R. Three Dimensional System Integration:IC Stacking Process and Design [M]. Berlin: Springer Science & Business Media,2010. 190-195.
- [14] Kang W,Lee C,Lim H,et al. A 3 dimensional built-in self-repair scheme for yield improvement of 3 dimensional memories [J]. IEEE Transactions on Reliability,2015,64(2):586-595.

作者简介



刘 军 男,1978 年 2 月生于江苏新沂,现为合肥工业大学计算机与信息学院副教授、硕士生导师. 主要研究方向为集成电路可测试性设计、测试数据压缩等。

E-mail: liujun@ict. ac. cn



朱承强 男,1990 年 11 月生于安徽安庆,现为合肥工业大学计算机与信息学院硕士. 主要研究方向为三维存储器。

E-mail: zcqiand159@163. com