

基于对角线的硅通孔容错设计

刘 军^{1,2}, 董 鹏^{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 合肥工业大学 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, 日本 德岛 7708502)

摘 要: 三维集成电路通过使用硅通孔(through-silicon vias, TSV)作为垂直方向上芯片的通信链路, 具有高密度, 高带宽, 低功耗等优点. 由于 TSV 在制造和使用过程中可能会出现故障, 导致整个三维芯片的故障. 为了提高三维芯片的良品率, TSV 的良品率必须尽可能的提高. 本文提出了一种新的基于对角线的 TSV 容错方案, 并提出了基于最大流算法的故障修复算法, 在 TSV 出现故障时使用对角线中的冗余 TSV 来修复故障 TSV 以提高整个三维芯片的良品率. 实验结果表明, 相比基于路由的容错方案, 本文提出的基于对角线的 TSV 容错方案, 芯片修复率可以达到 98.38% 至 98.96%, 方案造成的面积开销降低了 70% 左右。

关键词: 硅通孔; 对角线; 故障修复

中图分类号: TP391

文献标识码: A

文章编号: 1000—7180(2018)11—0073—06

Diagonal-Based for TSV Fault Tolerance Design

LIU Jun^{1,2}, Dong Peng^{1,2}, REN Fu-ji³

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

2 Anhui Province Key Laboratory of Affective Computing and Advanced Intelligent Machine, Hefei University of Technology, Hefei 230009, China; 3 Department of Information Science & Intelligent Systems,

Faculty of Engineering, The University of Tokushima, Tokushima 7708502, Japan)

Abstract: Three dimensional integrated circuits, using through-silicon vias (TSVs) as the communication link between vertical chip, with the advantages of high density, high bandwidth and low power consumption. As the TSV during processing and using process may be failure, will lead to the entire three-dimensional chip failure. In order to improve the yield of 3D chips, the yield of TSV must be raised as much as possible. This paper presents a novel diagonal-based redundant TSV repair scheme and proposes a repair algorithm based on maximum flow algorithm to tolerance the faulty TSV with redundant TSVs in the diagonal to improve the yield of the entire three-dimensional chip. Experimental results show that the yield of the proposed repair strategy is 98.38% to 98.96% and the hardware overhead can be reduced by up to 70% compared to router-based technique.

Key words: through-silicon vias; diagonal-based; fault tolerance

1 引言

良品率问题是三维集成电路的设计和制造过程中面临的一个重要的挑战^[1-2], 影响三维集成电路的大规模应用. 三维电路的良品率主要受到堆叠阶段和封装阶段的影响. 提高堆叠阶段的良品率需要进

行绑定前测试, 将测试过无故障的裸片进行堆叠可以极大地提高良品率. 封装阶段包括很多复杂的工艺, 如晶元削薄, TSV 绑定等, 这个阶段会出现不同类型的 TSV 故障, 导致相邻层芯片之间的信号传输失败^[3-4]. 为了提高三维集成电路的良率, TSVs 的良品率必须尽可能地提高. 在 TSVs 块中添加冗余

收稿日期: 2018—01—30; 修回日期: 2018—02—20

基金项目: 国家自然科学基金(61432004;61306049;61474035)

TSVs 进行 TSV 容错结构设计是比较有效和流行的方式之一。

本文提出了一种基于对角线的 TSV 容错方案,将冗余 TSV 布置在整个 TSV 块的对角线上,TSVs 块被分成两个不相交的子块.在 TSV 发生故障时,将故障 TSV 的信号连通到对角线上的冗余 TSV 上.实验结果表明,该方案整体效果优于以往的容错方案.

国内外针对 TSV 容错设计已经做了很多相关的工作.文献[5]提出了基于信号转移(Signal-shift)的容错结构,4 个功能 TSVs 和 1 个冗余 TSV 形成一个 TSVs 链,多路选择器将故障 TSV 的信号向着冗余 TSV 的方向逐步传递,可以修复出现一个故障 TSV 的情况.如图 1 所示,左图为基于信号转移的容错结构,右图为出现一个故障时 TSVs 的修复结果.这种方案的优点是结构简单,对于均匀分布的故障修复率较高.

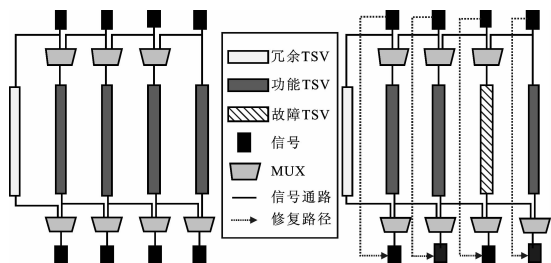


图 1 Signal-shift 容错方案

针对 TSV 故障存在聚类分布效应,Li Jiang 等人提出了一种基于路由的冗余 TSV 修复方案^[6],TSV 块中的每一个 TSV 都与一个转换器相连,转换器主要由三个 3 路选择器组成.当信号 TSV 发生故障时,转换器会将故障 TSV 负责传输的信号连接到对应的冗余 TSV 上.这种结构使得故障 TSV 可以被“远端”的冗余 TSV 修复.因此这种方案可以修复聚类 TSV 故障.然而,由于 TSV 块中的每个 TSV 都需要一个转换器来负责信号的重新路由,导致这种方案硬件开销较大.图 2 所示为基于路由的容错结构,图中出现了 6 个聚类 TSV 故障,尽管这种严重的聚类故障很少发生,基于路由的容错方案仍然可以修复.

文献[4]的容错结构虽然简单,但是对于聚类 TSV 故障的修复效果较差,整体修复率较低.文献[6]通过使用“远端”TSV 修复故障,取得了极高的修复率,但是带来了较高的硬件开销.本文综合考虑了修复率和硬件开销,提出了基于对角线的 TSV 容错结构,以较低的硬件开销达到了较高的修复率.

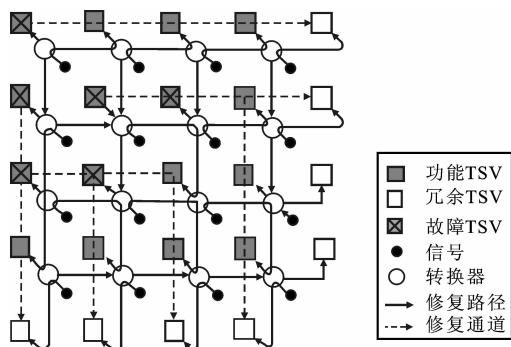


图 2 基于路由 TSV 修复示例

2 基于对角线的 TSV 容错方案

这部分首先介绍基于对角线的 TSV 容错结构,然后介绍基于最大流算法的修复算法,并举例说明.

2.1 TSV 冗余修复结构

TSV 在制造过程中一般有两种布局方式,有规律分布和无规律分布^[7].有规律分布即 TSV 布局如同一个整齐的阵列,无规律分布指 TSV 相互之间的位置没有规律.这两种布局方式在不同领域的集成电路都有着广泛的应用.本文提出的基于对角线的 TSV 容错方案主要针对的是 TSV 有规律布局,对于 TSV 的无规律分布,可以通过将 TSV 的物理分布映射成有规律的逻辑分布来解决这个问题.

图 3 所示是本文提出的基于对角线的 TSV 容错结构.本方案将冗余 TSV 布置在 TSV 块的对角线上.对角线上的冗余 TSV 将整个 TSV 块分成两个不相交的 TSV 子块,我们称之为上三角子块和下三角子块.图中是一个 4×4 的 TSV 块,对角线上的灰色圆圈表示冗余 TSV,其他位置的白色圆圈表示信号 TSV.当信号 TSV 出现故障时,通过将信号 TSV 原先负责传输的信号向着 TSV 块的对角线的方向传播,直到冗余 TSV 上来实现修复.规定修复方向只能沿着对角线进行.即上三角子块中的 TSV 信号传播方向只能向左和向下传播,下三角子块中的 TSV 信号传播方向只能向上和向右传播.

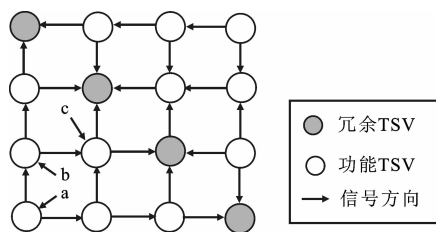


图 3 基于对角线冗余 TSV 修复结构

为了实现故障 TSV 信号的重新路由,需要为每一个信号 TSV 提供一个多路选择器(MUX),但是不同位置上的 TSV 所需的选择器的结构不同.我们以图 3 中的三个典型位置的 TSV 为例说明一下多路选择器的结构. a 处的 TSV,只需要接收来自本身的信号,因此不需要配备选择器. b 处的 TSV 需要接收来自下方的信号和本身的信号,因此需要配备一个 2 路选择器. c 处的 TSV 需要接收来自下方,左方和本身的信号,因此需要配备一个 3 路选择器. 因此,对于图 3 中的信号 TSV 来说,共需要 8 个 2 路选择器和 2 个 2 路选择器. 对于冗余 TSV 来说,由于冗余 TSV 在初始化时不需要负责信号传输,所以对角线两端的两个冗余 TSV 需要 2 路选择器,其他位置的冗余 TSV 则需要使用 4 路选择器. 三种类型的多路选择器结构如图 4 所示.

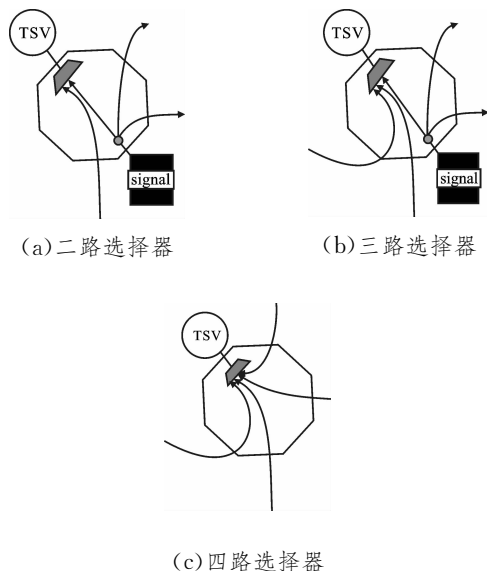


图 4 多路选择器结构

分析可以推得,一个 $N \times N$ 的 TSV 块,冗余 TSV 的数量为 N 个,其中信号 TSV 一共需要 $(N-2) \times 4$ 个 2 路选择器, $N^2 - N - (N-2) \times 4 - 2$ 个 3 路选择器. 例如, 8×8 的 TSV 块中,冗余 TSV 的个数为 8 个,则信号 TSVs 需要 24 个 2 路选择器和 30 个 3 路选择器. 而 N 个冗余 TSV 共需要 2 个 2 路选择器和 $N-2$ 个 4 路选择器.

本文提出的基于对角线的 TSV 容错方案有如下几点好处:

- (1)从物理布局上将一个 TSV 块分成两个不相交的子块,降低了发生聚类故障的概率;
- (2)对于每一个子块,对角线上的冗余 TSV 均可用来修复故障,提高了局部“冗余率”;
- (3)上三角子块和下三角子块可以分别应用冗

余修复算法,降低了算法的复杂度.

2.2 TSV 修复算法

在进行 TSV 修复之前,首先需要进行测试获得 TSV 块的故障表. 这个故障表中记录了故障 TSV 的信息. 冗余修复算法则是用来分析 TSV 块是否可以修复,如果 TSV 块是可修复的,冗余修复算法产生故障 TSV 的修复路径. 当 TSV 发生故障时,需要将故障 TSV 原本负责传输的信号重新找到一条路径到达一个无故障的信号 TSV 或者冗余 TSV. 因此,可以将修复算法总结为:为每一个信号找到一条不相交的修复路径. 本文采用最大流算法(Maximum Flow method)来解决这个问题^[8].

在使用最大流算法的过程中,还需要考虑由于信号重新选择传输路径可能带来的信号时延的问题. 由于信号在 TSV 中传输时需要保证时序正确性,当信号的传输时延超过了其时序正确性的阈值,芯片也会发生故障. 因此,在为每一个信号确定其传输路径时不能超过其时序正确性的阈值. 实际情况中,每个信号的时序正确性阈值可能各不相同. 简单起见,本文将这个阈值规定为一个统一的阈值 T ,这个 T 是整个 TSV 块中最关键的一条信号传输路径的时序正确性阈值. 在使用最大流算法确定信号的修复路径时,一旦传输路径长度超过了阈值 T ,就抛弃这一条路径,重新查找其他修复路径.

在详细介绍本文提出的 TSV 冗余修复算法之前,先提出几个有助于提高修复效率的规则:

(1)分别处理 TSV 子块. 本文提出的 TSV 容错结构中,冗余 TSV 布置在 TSV 块的对角线上,整个 TSV 块被分成上三角子块和下三角子块,这两个子块中的信号传输路径没有交集. 因此,可以对这两个子块分别使用最大流算法产生信号的修复路径,降低了算法的复杂度.

(2)TSV 故障表预处理. 从故障表中得出两个子块中各自的 TSV 故障信息. 在使用最大流算法处理时,先处理 TSV 故障数多的子块. 如果 TSV 故障多的子块不可修复,那么整个 TSV 块即是不可修复的,可以提前发现 TSV 块是否可以修复,避免了很多无用功.

(3)设置优先搜索策略. 在使用最大流算法搜索信号传输路径时可以使用横向优先搜索或纵向优先搜索. 但对于两个子块,需要使用同一种搜索策略. 如图 5 所示,图(a)中下三角子块使用了纵向优先搜索,故障 TSV a 最后找到的冗余 TSV 位于左上角,

上三角子块使用了横向优先搜索,故障 TSV b 找到的冗余 TSV 也位于左上角,两个故障 TSV 的修复需要用到同一个冗余 TSV,产生了冲突. 而图(b)中两个子块同时使用了横向优先搜索,两个故障 TSV 修复使用的 TSV 分别位于左上角和右下角,没有产生冲突. 本文规定,在使用最大流算法时一律采用横线优先搜索策略.

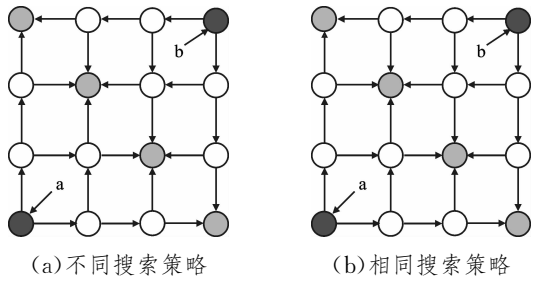


图5 搜索策略对修复方案的影响

本文提出的基于最大流的 TSV 冗余修复算法如下.

输入:有向流图 G

输出:冗余修复方案

步骤一:初始化有向流图 G,删除故障 TSV 及其关联边

步骤二:将 TSV 块分为两个子块,按故障 TSV 个数递减排列

步骤三:分别处理每一个子块. 对于每一个子块中的信号,按照横向优先搜索策略,找出一条不相交的信号修复路径,该路径长度不能超过阈值 T,若找不到这样的路径,则该 TSV 块不可修复,转向步骤五

步骤四:将两个子块的修复路径所使用的冗余 TSV 做交集运算,若结果为空集,说明 TSV 块可以修复,输出修复方案,转向步骤五;若结果不为空集,说明两个子块的修复方案产生冲突,此修复方案不可行,转向步骤三重新进行搜索

步骤五:修复结束.

下面以图 6 所示的 4×4 TSVs 块说明本文的修复结构和修复算法的工作流程. 首先,将 TSVs 块划分成上三角子块和下三角子块,从故障表可知,上三角子块中有 3 个故障 TSVs,下三角子块中有 1 个故障 TSV. 因此,先使用最大流算法确定上三角子块中故障 TSVs 的修复路径. 将上三角子块抽象成一个有向流图,即 $G(V, E)$,顶点集合 V 中包括开始节点、信号集合、选择器集合、TSVs 集合和目标节点. 开始节点和目标节点分别是最大流算法中的起始节点和结束节点. 信号集合包含子块中的所有信号. 选择器集合包含子块中的所有多路选择器. TSVs 集合包含子块中所有的功能 TSVs 和冗余 TSVs. 边集合 E 包括从开始结点到信号结点的边,信号结点到选择器的边,选择器到无故障的功能 TSVs 和冗余 TSVs 节点

的边以及无故障的功能 TSVs 和冗余 TSVs 节点到目标节点的边. 有向流图构建完成后,使用最大流算法计算出有向流图的最大流结果. 图 6 中上三角子块的有向流图如图 7(a)所示,其最大流为 6,等于子块中信号的个数,说明每个信号都有不相交的传输路径,即上三角子块是可以修复的,修复路径如图 6 实线箭头所示,4 号 TSV 使用 3 号 TSV 来修复. 由于 3 号 TSV 被用来传输 4 号 TSV 的信号,因此,3 号 TSV 也需要修复,3 号 TSV 使用 2 号 TSV 来修复,同理 2 号 TSV 使用 1 号冗余 TSV 来修复. 7 号 TSV 使用 6 号冗余 TSV 修复,8 号 TSV 使用 12 号 TSV,12 号 TSV 使用 11 号冗余 TSV 修复. 接着按照同样方法处理下三角子块. 下三角子块的有向流图如图 7(b)所示,修复路径如图 6 实线箭头所示,13 号 TSV 使用 14 号 TSV 修复,14 号 TSV 使用 15 号 TSV 修复,15 号 TSV 使用 16 号冗余 TSV 修复. 两个子块的修复方案使用的冗余 TSVs 集合分别是(1,6,11)和(16). 将两个集合交集运算结果为空集,说明两个子块的修复方案没有冲突,则整个 TSVs 块是可以修复的.

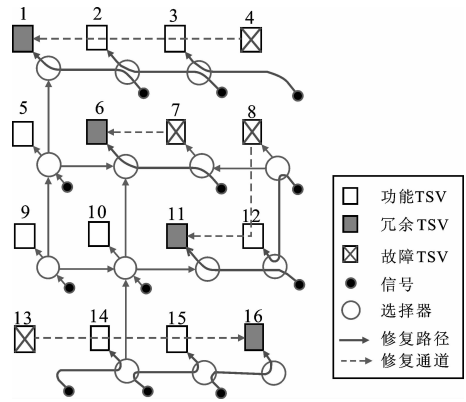


图6 基于对角线 TSV 块修复示例

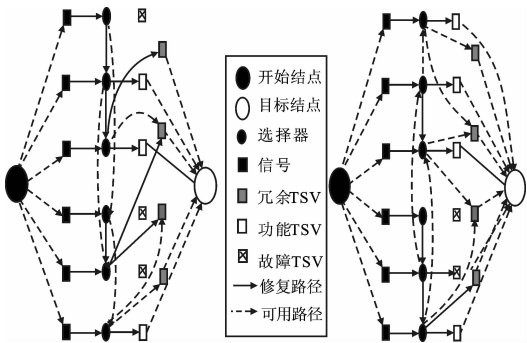


图7 基于对角线修复流图

3 实验结果与分析

本部分对本文提出的 TSV 容错方案进行了相关的实验,并与文献[6]提出的基于路由的 TSV 容错方案进行对比. 实验结果表明,本文提出的 TSV 容错方案极大地降低了硬件开销,同时保持极高的修复率.

由于本文提出的方案对 TSV 块的尺寸有较好的适应性,在实验中使用了两种规格的 TSV 块,分别表示为 Diagonal-4 和 Diagonal-8. 其中,Diagonal-4 表示 TSV 块为 4×4 ,对角线上的 4 个 TSV 为冗余 TSV,最多可以修复 4 个故障 TSV. Diagonal-8 表示 TSV 块为 8×8 ,对角线上的 8 个 TSV 为冗余 TSV,最多可以修复 8 个故障 TSV.

3.1 修复率分析

由于本方案主要是修复聚类 TSV 故障,实验中使用复合泊松分布来模拟 TSV 故障的聚类分布,聚类效应参数 α 设置为 1,单个 TSV 故障率设置为 1%. 图 8 所示为本文提出的方案与文献[6]的容错方案的修复率对比柱状图,表 1 是本文方案与文献[6]方案在不同故障数量情况下的详细修复率. 由表可知,本方案的修复率基本达到了文献[6]的方案的修复率. Diagonal-4 方案由于仅有 4 个冗余 TSV,只能修复故障 TSV 少于或等于 4 个的情况,对于出现 4 个以上故障 TSV 的情况修复率为 0,总体的修复率为 98.55%. Diagonal-8 方案由于有 8 个冗余

TSV,最多可以修复 8 个故障,总体的修复率达到了 99.13%. 本文提出的冗余修复结构中有两个信号 TSV(左下角和右上角)不参与故障 TSV 信号的重新路由,所以修复过程中的解空间比文献[6]中的稍小,带来的修复率上的下降也是符合实验结果的.

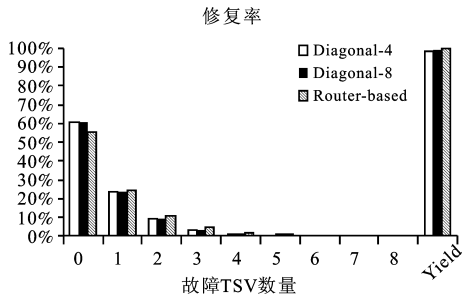


图 8 实验修复率结果

3.2 硬件开销分析

TSV 容错方案中的硬件开销主要是冗余 TSV 和多路选择器. 本实验中,选择器的技术参数来源于 Nangate 45 nm 开源库. 其中,2 路选择器的面积开销为 $1.33\times 1.4 = 1.862\text{ }\mu\text{m}^2$. 三路选择器是由两个 2 路选择器构成,其面积开销为 $1.862\times 2 = 3.724\text{ }\mu\text{m}^2$,4 路选择器由三个 2 路选择器构成,其面积开销为 $1.862\times 3 = 5.586\text{ }\mu\text{m}^2$. 对于文献[6]的方案和本文的方案分别设置了 4×4 和 8×8 两种规格的 TSV 块.

表 1 文献[6]方案与本文方案修复率

	0	1	2	3	4	5	6	7	8	Yield
Diagonal-4	60.79%	23.84%	9.35%	3.66%	0.92%	0.00%	0.00%	0.00%	0.00%	98.55%
Diagonal-8	60.79%	23.84%	9.35%	3.66%	0.92%	0.36%	0.14%	0.06%	0.02%	99.13%
Router-based	55.56%	24.69%	10.97%	4.88%	1.98%	0.91%	0.34%	0.15%	0.05%	99.53%

由前面的分析可知,Diagonal-4 的方案一共需要 10 个 2 路选择器,2 个 3 路选择器和 2 个 4 路选择器. Diagonal-8 的方案一共需要 26 个 2 路选择器,30 个 3 路选择器和 6 个 4 路选择器. 四种方案所使用的 TSV 数量,多路选择器的数量和多路选择器的面积开销如表 2 所示. 图 9 是四种方案的多路选择器面积开销的柱状图. 由图可直观得出,本文的方案相较于文献[6]的方案,极大地降低了硬件开销. 由于两种方案中的信号 TSV 数量不相同,本文使用单个信号 TSV 硬件开销(多路选择器面积与信号 TSV 数量的比值)来直观地比较两种方案的硬件开销. Router-4 和 Router-8 的单个信号 TSV 硬件开销均为 11.17,而本文提出的 Diagonal-4 和 Diagonal-8 的单个信号 TSV 硬件开销分别为 3.10 和

3.46,相较于文献[6]的容错方案分别降低了 69.02%和 72.25%. 由此可见,本文提出的基于对角线的冗余 TSV 修复方案在保证较高的故障 TSV 修复率的同时,极大地降低了硬件开销.

表 2 文献[6]与本文方案的硬件开销对比

	Router-4	Router-8	Diagonal-4	Diagonal-8
信号 TSV	16	64	12	56
冗余 TSV	8	16	4	8
冗余率	1;2(50%)	1;4(25%)	1;3(33%)	1;7(14%)
2 路选择器数量	0	0	10	26
3 路选择器数量	48	192	2	30
4 路选择器数量	0	0	2	6
2 路选择器面积	0	0	18.62	48.41
3 路选择器面积	178.75	715.01	7.45	111.72
4 路选择器面积	0	0	11.17	33.52
多路选择器总面积	178.75	715.01	37.24	193.65
单个信号 TSV 硬件开销	11.17	11.17	3.10	3.46

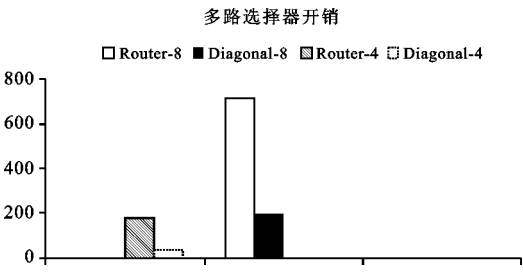


图 9 四种方案的多路选择器面积对比图

4 结束语

本文提出了一种基于对角线的 TSV 容错方案, 该方案通过将冗余 TSV 布置在 TSV 块的对角线上, TSVs 块划分成上三角子块和下三角子块. 在信号 TSV 出现故障时, 通过基于最大流算法的修复方法确定 TSV 的修复路径. 本方案不仅可以修复随机 TSV 故障, 对于聚类 TSV 故障也有极高的修复率. 实验结果表明, 与文献[6]提出的基于路由的 TSV 容错方案相比, 本方案保证了较高的故障 TSV 修复率, 同时极大地减少了多路选择器的使用数量, 降低了修复方案带来的硬件开销.

参考文献:

[1] 刘军, 李进, 王伟. 基于相邻多层冗余共享的三维存储器堆叠方法[J]. 微电子学与计算机, 2017, 34(7):1-6.

[2] 孟真, 张兴成, 刘谋. TSV 封装中互连结构的差分串扰建模研究 [J]. 微电子学与计算机, 2017, 34(9):1-6.

[3] Zhang Ling, Wang Weizheng, Mei Junjin. Fault-tolerant design for TSVs of 3D stacked ICs [J]. Computer Engineering and Applications, 2015, 51(14):11-16.

[4] Shijie Zhang, Xiaole Cui, Qiang Zhang, et al. ATSV repair method for clutered faults [J]. High Voltage Apparatus, 2011, 47 (12) :96-99.

[5] Marinissen E J, Zorian Y. Testing 3D chips containing throughsilicon vias [C]//Proc. IEEE Int'l Test Conf. Austin, 2009:1-11.

[6] Jiang L, XU Q, Eklow B. On effective TSVs repair for 3D-stacked ICs [J]. IEEE/ACM Proc. Design, Automation, and Test in Europe, 2011, 58(11):1-10.

[7] Shengcheng Wang, Mehdi B, Tahooriand Krishnendu Chakrabarty. Defect clustering-aware spare-TSV allocationfor 3DICs [D]. Duke University, Durham, USA, 2015:307-314.

[8] Goldberg A V, Rao S. Beyond the flow decomposition barrier [J]. Journal of the ACM, 1998, 45 (5): 783-797.

作者简介:

刘 军 男, (1978-), 博士, 副教授, 硕士生导师. 研究方向为数字系统设计自动化、集成电路测试、嵌入式系统等.

董 鹏 (通讯作者) 男, (1992-), 硕士研究生. 研究方向为三维集成电路故测试与修复等.

E-mail: 1796800781@qq. com.

任福继 男, (1959-), 博士, 教授, 博士生导师. 研究方向为人工智能、情感机器人、自然语言理解等.