三维扫描树叶子节点和 TSVs 数量的优化方法

刘军^{①②*}, 吴玺^{①②}, 梁华国^③, 任福继^④

① 合肥工业大学计算机与信息学院, 合肥 230009

② 情感计算与先进智能机器安徽省重点实验室, 合肥 230009

③ 合肥工业大学电子科学与应用物理学院, 合肥 230009

④ Department of Information Science & Intelligent Systems, Faculty of Engineering, The University of Tokushima, Tokushima 7708502, Japan

* 通信作者. E-mail: liujun@ict.ac.cn

收稿日期: 2014-05-14; 接受日期: 2014-08-20

国家自然科学基金 (批准号: 61306049, 61274036, 61106037, 61204046, 61432004)、国家高技术研究发展计划 (863)(批准号: 2012AA011103) 和安徽省自然科学基金 (批准号: 1208085QF127)

摘要 扫描树结构能够有效地减少集成电路的测试数据量和测试时间, 降低电路的测试成本. 为减少三维电路中扫描树的叶子节点和硅通孔数量, 首先得出了扫描树中叶子节点的最小数量为最大相容组中所含扫描单元数量的结论, 然后进一步得到了叶子节点取得最小值的充分必要条件. 并在此基础上, 提出了一种启发式算法来确定扫描树中相容组的连接顺序, 使得叶子节点数量取得最小值的同时能够优化硅通孔的数量. 实验结果表明了所提方法的有效性.

关键词 三维集成电路 扫描树 测试数据 测试时间 测试成本

1 引言

随着电路规模的不断增大, 基于扫描的可测试性设计方法造成了电路测试数据和测试时间的急剧增加, 极大地提高了测试成本^[1~3]. 测试数据压缩技术和扫描树结构能够有效地减少测试数据量和测试时间, 是降低电路测试成本的有效方式^[4~8]. 在测试压缩中, 由于解压电路是需要额外设计的, 所以测试数据压缩技术将会造成电路硬件开销的增大, 而扫描树结构能够有效地克服这一缺点^[9~11]. 扫描树将扫描单元以树的形式进行连接, 位于树的同一层的扫描单元组成一个相容组. 在同一个相容组中, 任何两个扫描单元之间都没有相同的后继, 这样是为了不增加新的不可测故障, 保证故障覆盖率. 因为没有共同的后继单元, 测试生成的时候就不会造成逻辑赋值的冲突和故障覆盖率的损失.

基于硅通孔 (through silicon vias, TSVs) 的三维集成电路通过 TSVs 将多层硅片进行垂直互连, 能够有效地克服二维集成电路的缺点, 被认为是延续“摩尔定律”的一种有效方法^[12]. 与二维电路相比, 三维电路具有更高的集成度, 这也意味着三维电路的测试数据量更大, 测试成本也更高^[13]. 测试压缩技术和扫描树结构对降低三维电路的测试成本也同样有效^[14,15]. 文献[14]提出的自适应扫描结构和文献[15]提出的三维电路扫描树的构建方法都能够有效地减少电路的测试数据量和测试时间, 并降低测试成本. 针对三维电路扫描树, 文献[16]还进一步研究了如何在三维电路扫描树中施加测试向

引用格式: 刘军, 吴玺, 梁华国, 等. 三维扫描树叶子节点和 TSVs 数量的优化方法. 中国科学: 信息科学, 2014, 44: 1203–1215, doi: 10.1360/N112014-00133

图 1 扫描树中 TSVs 的数量

Figure 1 The number of TSVs in scan tree. (a) TSVs=10; (b) TSVs=7

量以满足热量约束的要求, 避免测试时因芯片温度过高而损坏芯片. 虽然学者们针对三维电路扫描树已进行了一些研究, 然而已有的研究工作都未考虑如何最小化三维电路扫描树叶子节点的数量.

扫描树中的叶子节点可以被直接连到输出引脚上, 也可以通过测试响应压缩器连接到输出引脚^[9~11]. 然而无论采用何种连接方式, 叶子节点的增多都将导致测试输出引脚的增加, 而测试输出引脚的增加会造成芯片面积和制造成本增大. 同时, 测试输出引脚的增加也意味着测试时需要更多的 ATE 测试通道, 将导致电路测试成本的提高. 此外, 三维电路的输入和输出引脚只能位于最底层, 而三维电路中的任一层都含有扫描单元, 因而在三维电路扫描树中需要使用大量的 TSVs 来连接这些扫描单元. 而 TSVs 不仅占用芯片面积, 增加布局布线的难度, 还会降低三维电路的成品率^[17~19]. 因此, 在三维电路中构建扫描树时, 也需要尽量地减少 TSVs 的数量.

为此, 本文首先得出并证明了扫描树中叶子节点的最小值为最大相容组所含的扫描单元数量的结论, 并给出了叶子节点取得最小值的充分必要条件. 以此为基础, 本文进一步提出了构建三维电路扫描树的启发式算法, 以达到最小化叶子节点和优化 TSVs 数量的目的.

2 研究动机

将 n 个相容组连接成一棵扫描树, 有 $n!$ 种连接方式. 在这 $n!$ 种连接方式中, 有些连接方式会产生相同的叶子节点数量, 但所需的 TSVs 数量却相差很大, 如图 1 所示.

在图 1 所示的三维电路中, 每层电路都含有 2 个相容组. 第一层电路中 2 个相容组所含的扫描单元数量分别为 1 和 3, 第二层电路相容组中的扫描单元数量为 2 和 4. 图 1(a) 和图 1(b) 这两种连接方式所产生的叶子节点数量相同 (图中灰色矩形框所示), 但所用的 TSVs 数量不同. 图 1(a) 中使用了 10 根 TSVs, 而图 1(b) 使用了 7 根 TSVs. 三维电路扫描树中这种叶子节点相同而 TSVs 数量不同的现象促成了本文的研究工作, 即叶子节点的最小值是多少, 在叶子节点取得最小值时, 采取何种连接方式可以使得 TSVs 数量最少.

本文在第 3 节中证明了叶子节点数量的最小值为最大相容组中所含的扫描单元数量, 并给出了叶子节点取得最小值的充分必要条件, 在第 4 节中具体介绍了本文所提的最小化叶子节点和优化 TSVs 数量的启发式算法.

图 2 扫描树中的叶子节点数量

Figure 2 The number of leaf nodes in scan tree. (a) $|C_i| = 3, |C_j| = 4$; (b) $|C_i| = 4, |C_j| = 3$

3 扫描树的性质

本文发现了扫描树的两个特性, 为最小化叶子节点和优化 TSVs 的数量提供了基础. 为证明扫描树的这两个特性, 本文定义了一个函数 $f(x, y)$:

$$f(x, y) = \begin{cases} 0, & x < y, \\ x - y, & \text{其他}. \end{cases} \quad (1)$$

函数 (1) 可以用来计算扫描树中叶子节点的数量. 如图 2 所示的两棵扫描树, 其叶子节点数量都可以由式 (2) 来计算

$$f(|C_i|, |C_j|) + |C_j|. \quad (2)$$

式 (2) 中 $|C_i|$ 和 $|C_j|$ 分别表示相容组 C_i 和 C_j 所含的扫描单元数量, $f(|C_i|, |C_j|)$ 为相容组 C_i 与 C_j 进行连接, C_i 产生的叶子节点最小数量. 函数 $f(x, y)$ 为本文的证明扫描树的两个特性提供了有力的工具. 下面具体介绍扫描树的这两个特性以及其证明过程.

特性 1 将 n 个相容组 $C_1, C_2, \dots, C_i, \dots, C_n$ 连接成一棵扫描树, 其叶子节点数量的最小值为 $\max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_n|)$.

本文通过数学归纳法来证明特性 1.

证明

步骤 1 当 $n = 1$ 时, 扫描树中只有 C_1 一个相容组, 所以其叶子节点数量为 $|C_1| = \max(|C_1|)$.

步骤 2 当 $n = k$ 时, 扫描树中含有的相容组为 $C_1, C_2, \dots, C_i, \dots, C_k$, 假设其叶子节点数量的最小值为 $\max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|)$.

步骤 3 当 $n = k + 1$ 时, 扫描树中含有的相容组为 $C_1, C_2, \dots, C_i, \dots, C_k, C_{k+1}$, 需要证明其叶子节点数量的最小值为 $\max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|)$.

假设将上述相容组按照 $C'_1, C'_2, \dots, C'_i, C_{k+1}, C'_{i+1}, \dots, C'_k$ 的顺序连接成扫描树, 其产生的叶子节点数量记为 leaf, 其中 $C'_i \in \{C_1, C_2, \dots, C_i, \dots, C_k\}$ 且 $C_i \in \{C'_1, C'_2, \dots, C'_i, \dots, C'_k\}$. 本文首先证明 $\text{leaf} \geq \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|)$, 然后再找出一种相容组的连接顺序使得等号能够成立.

相容组 C_{k+1} 可能位于扫描树中的第一层, 也可能位于 C'_j 和 C'_{j+1} ($1 \leq j \leq k - 1$) 之间, 还有可能位于扫描树的最后一层, 因而需要分 3 种情况来计算 leaf 的值.

(1) 若相容组 C_{k+1} 位于扫描树的第一层, 即将相容组按照 $C_{k+1}, C'_1, C'_2, \dots, C'_i, \dots, C'_k$ 的顺序进

行连接形成扫描树. 此时其叶子节点的数量为

$$\text{leaf} = f(|C_{k+1}|, |C'_1|) + \sum_{i=1}^{k-1} f(|C'_i|, |C'_{i+1}|) + |C'_k|. \quad (3)$$

式 (3) 中, $\sum_{i=1}^{k-1} f(|C'_i|, |C'_{i+1}|) + |C'_k|$ 为 $C'_1, C'_2, \dots, C'_i, \dots, C'_k$ 这 k 个相容组连接而成的扫描树所产生的叶子节点数量, 由步骤 2 可知, 其最小值为 $\max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|)$. 代入式 (3) 可得

$$\text{leaf} \geq f(|C_{k+1}|, |C'_1|) + \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|). \quad (4)$$

针对式 (4) 又需要分两种情况进行讨论:

(1.1) 当 $|C_{k+1}| < |C'_1|$ 时, 由式 (1) 可知, 式 (4) 中 $f(|C_{k+1}|, |C'_1|) = 0$. 又因为 $C'_1 \in \{C_1, C_2, \dots, C_i, \dots, C_k\}$, 所以 $|C_{k+1}| < \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|)$, 即

$$\max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|) = \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|).$$

将上式和 $f(|C_{k+1}|, |C'_1|) = 0$ 代入式 (4) 可得

$$\text{leaf} \geq \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|).$$

(1.2) 当 $|C_{k+1}| \geq |C'_1|$ 时, 由式 (1) 可知, 此时式 (4) 中 $f(|C_{k+1}|, |C'_1|) = |C_{k+1}| - |C'_1|$, 代入式 (4) 可得

$$\text{leaf} \geq |C_{k+1}| - |C'_1| + \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|). \quad (5)$$

当 $|C_{k+1}| \geq \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|)$ 时, 有

$$|C_{k+1}| = \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|). \quad (6)$$

又因为 $\max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|) - |C'_1| \geq 0$, 将此式和式 (6) 代入式 (5) 可得

$$\text{leaf} \geq \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|).$$

当 $|C_{k+1}| < \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|)$ 时, 有

$$\max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|) = \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|). \quad (7)$$

又因为 $|C_{k+1}| - |C'_1| \geq 0$, 将此式和式 (7) 代入式 (5) 可得

$$\text{leaf} \geq \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|).$$

由此可知, 当 $|C_{k+1}| \geq |C'_1|$ 时, 无论 $|C_{k+1}|$ 与 $\max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|)$ 之间的大小如何, $\text{leaf} \geq \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|)$ 都是成立的.

而综合 (1.1) 和 (1.2) 可知, 当相容组 C_{k+1} 位于扫描树的第一层时, $\text{leaf} \geq \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|)$ 是成立的. 下面证明 C_{k+1} 位于 C'_j 和 C'_{j+1} 之间的情况, 即第二种情况.

(2) 当 C_{k+1} 位于 C'_j 和 C'_{j+1} ($1 \leq j \leq k-1$) 之间时, 即相容组按照 $C'_1, C'_2, \dots, C'_j, C_{k+1}, C'_{j+1}, \dots, C'_k$ 的顺序连接. 此种情况下, 叶子节点的数量为

$$\text{leaf} = \sum_{i=1}^{j-1} f(|C'_i|, |C'_{i+1}|) + f(|C'_j|, |C_{k+1}|) + f(|C_{k+1}|, |C'_{j+1}|)$$

$$\begin{aligned}
& + \sum_{i=j+1}^{k-1} f(|C'_i|, |C'_{i+1}|) + |C'_k| \\
= & \sum_{i=1}^{j-1} f(|C'_i|, |C'_{i+1}|) + f(|C'_j|, |C'_{j+1}|) + \sum_{i=j+1}^{k-1} f(|C'_i|, |C'_{i+1}|) \\
& + |C'_k| + f(|C'_j|, |C_{k+1}|) + f(|C_{k+1}|, |C'_{j+1}|) - f(|C'_j|, |C'_{j+1}|) \\
= & \sum_{i=1}^{k-1} f(|C'_i|, |C'_{i+1}|) + |C'_k| + f(|C'_j|, |C_{k+1}|) + f(|C_{k+1}|, |C'_{j+1}|) - f(|C'_j|, |C'_{j+1}|). \quad (8)
\end{aligned}$$

式 (8) 中, $\sum_{i=1}^{k-1} f(|C'_i|, |C'_{i+1}|) + |C'_k|$ 为 $C'_1, C'_2, \dots, C'_i, \dots, C'_k$ 这 k 个相容组连接而成的扫描树所产生的叶子节点数量, 其最小值为 $\max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|)$. 代入式 (8) 可得

$$\begin{aligned}
\text{leaf} \geq & \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|) + f(|C'_j|, |C_{k+1}|) \\
& + f(|C_{k+1}|, |C'_{j+1}|) - f(|C'_j|, |C'_{j+1}|). \quad (9)
\end{aligned}$$

针对式 (9), 根据 $|C_{k+1}|, |C'_j|$ 和 $|C'_{j+1}|$ 的大小, 需要按以下 6 种情况进行讨论:

$$\begin{aligned}
& \textcircled{1} |C_{k+1}| \leq |C'_j| \leq |C'_{j+1}|, \quad \textcircled{2} |C'_j| \leq |C_{k+1}| \leq |C'_{j+1}|, \quad \textcircled{3} |C'_j| \leq |C'_{j+1}| \leq |C_{k+1}|, \\
& \textcircled{4} |C_{k+1}| \leq |C'_{j+1}| \leq |C'_j|, \quad \textcircled{5} |C'_{j+1}| \leq |C_{k+1}| \leq |C'_j|, \quad \textcircled{6} |C'_{j+1}| \leq |C'_j| \leq |C_{k+1}|.
\end{aligned}$$

在以上 6 种情况下, $\text{leaf} \geq \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|)$ 都是成立的. 其证明过程与 (1.1) 和 (1.2) 类似, 在此将过程省略.

以上证明了相容组 C_{k+1} 位于扫描树第一层和中间层时的情况, 下面证明相容组位于扫描树最后一层的情况.

(3) 当相容组 C_{k+1} 位于扫描树最后一层时, 即按照 $C'_1, C'_2, \dots, C'_i, \dots, C'_k, C_{k+1}$ 的顺序进行连接, 其叶子节点数量为

$$\begin{aligned}
\text{leaf} = & \sum_{i=1}^{k-1} f(|C'_i|, |C'_{i+1}|) + f(|C'_k|, |C_{k+1}|) + |C_{k+1}| \\
= & \sum_{i=1}^{k-1} f(|C'_i|, |C'_{i+1}|) + |C'_k| + f(|C'_k|, |C_{k+1}|) + |C_{k+1}| - |C'_k|. \quad (10)
\end{aligned}$$

同理, 式 (10) 中 $\sum_{i=1}^{k-1} f(|C'_i|, |C'_{i+1}|) + |C'_k|$ 的最小值为 $\max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|)$, 代入式 (10) 可得

$$\text{leaf} \geq \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|) + f(|C'_k|, |C_{k+1}|) + |C_{k+1}| - |C'_k|. \quad (11)$$

针对式 (11), 也需要分两种情况进行讨论:

$$\textcircled{1} |C'_k| < |C_{k+1}|, \quad \textcircled{2} |C'_k| \geq |C_{k+1}|.$$

在上述两种情况下, $\text{leaf} \geq \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|)$ 也都是成立的. 其证明过程与 (1.1) 和 (1.2) 类似, 在此省略.

综上所述, 当 $n = k + 1$ 时, 扫描树的叶子节点数量是大于等于 $\max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|)$. 采用如下的连接方式就可以使得等号成立.

将相容组 $C_1, C_2, \dots, C_i, \dots, C_k, C_{k+1}$ 按照其所含扫描单元的数量进行升序排列, 假设排列的顺序为 $C'_1, C'_2, \dots, C'_i, \dots, C'_k, C'_{k+1}$, 其中 $|C'_1| \leq |C'_2| \leq \dots \leq |C'_i| \leq \dots \leq |C'_{k+1}|$. 按此顺序将相容组连接成扫描树, 此时叶子节点数量为

$$\text{leaf} = \sum_{i=1}^k f(|C'_i|, |C'_{i+1}|) + |C'_{k+1}|. \quad (12)$$

因为 $|C'_i| \leq |C'_{i+1}|$, 由式 (1) 可知, 式 (12) 中 $f(|C'_i|, |C'_{i+1}|) = 0$. 所以式 (12) 可变为

$$\text{leaf} = |C'_{k+1}| = \max(|C_1|, |C_2|, \dots, |C_i|, \dots, |C_k|, |C_{k+1}|).$$

至此, 完成了对扫描树特性 1 的证明. 由证明过程可知, 将相容组按照其所含扫描单元的数量进行从小到大的排序, 按照此顺序连接而成的扫描树可以取得最少的叶子节点. 但这只是叶子节点取得最小值的充分条件, 而非必要条件. 扫描树叶子节点取得最小值时相容组连接顺序的充分必要条件如特性 2 所述. 为后面描述方便, 记 $|C_j| = \max(|C_1|, |C_2|, \dots, |C_{j-1}|, |C_j|, |C_{j+1}|, \dots, |C_{n-1}|, |C_n|)$.

特性 2 将 n 个相容组按照 $C_1, C_2, \dots, C_{j-1}, C_j, C_{j+1}, \dots, C_{n-1}, C_n$ 顺序连接成一棵扫描树, 此扫描树叶子节点取得最小值 $|C_j|$ 的充分必要条件是 $|C_1| \leq |C_2| \leq \dots \leq |C_{j-1}| \leq |C_j|$, $|C_j| \geq |C_{j+1}| \geq |C_{j+2}| \geq \dots \geq |C_{n-1}| \geq |C_n|$.

证明

充分性. 已知条件是 $|C_1| \leq |C_2| \leq \dots \leq |C_{j-1}| \leq |C_j|$, $|C_j| \geq |C_{j+1}| \geq |C_{j+2}| \geq \dots \geq |C_{n-1}| \geq |C_n|$. 要证明将相容组按照 $C_1, C_2, \dots, C_{j-1}, C_j, C_{j+1}, \dots, C_{n-1}, C_n$ 的顺序连接形成的扫描树, 其叶子节点数量为 $|C_j|$.

相容组按照 $C_1, C_2, \dots, C_{j-1}, C_j, C_{j+1}, \dots, C_{n-1}, C_n$ 的顺序连接, 其叶子节点数量 leaf 为

$$\text{leaf} = \sum_{i=1}^{j-1} f(|C_i|, |C_{i+1}|) + \sum_{i=j}^{n-1} f(|C_i|, |C_{i+1}|) + |C_n|. \quad (13)$$

因为 $|C_1| \leq |C_2| \leq \dots \leq |C_{j-1}| \leq |C_j|$, 由式 (1) 可知

$$\sum_{i=1}^{j-1} f(|C_i|, |C_{i+1}|) = 0. \quad (14)$$

同理, 因为 $|C_j| \geq |C_{j+1}| \geq |C_{j+2}| \geq \dots \geq |C_{n-1}| \geq |C_n|$, 所以

$$\sum_{i=j}^{n-1} f(|C_i|, |C_{i+1}|) = \sum_{i=j}^{n-1} (|C_i| - |C_{i+1}|) = |C_j| - |C_n|. \quad (15)$$

将式 (14) 和 (15) 代入式 (13), 可得 $\text{leaf} = |C_j|$.

至此, 充分性证明完毕, 下面证明必要性.

必要性. 已知条件是将相容组按照 $C_1, C_2, \dots, C_{j-1}, C_j, C_{j+1}, \dots, C_{n-1}, C_n$ 的顺序连接成一棵扫描树, 此扫描树可以取得最小的叶子节点数量 $|C_j|$, 要证明 $|C_1| \leq |C_2| \leq \dots \leq |C_{j-1}| \leq |C_j|$, $|C_j| \geq |C_{j+1}| \geq |C_{j+2}| \geq \dots \geq |C_{n-1}| \geq |C_n|$.

本文证明其逆否命题成立, 即在以下 3 种情况下, 按照 $C_1, C_2, \dots, C_{j-1}, C_j, C_{j+1}, \dots, C_{n-1}$ 的顺序连接而成的扫描树, 其叶子节点数量大于 $|C_j|$:

① $|C_1| \leq |C_2| \leq \dots \leq |C_{k-1}| \leq |C_k|$, $|C_k| > |C_{k+1}|$, $|C_{k+1}| \leq \dots \leq |C_{j-1}| \leq |C_j|$, $|C_j| \geq |C_{j+1}| \geq \dots \geq |C_{n-1}| \geq |C_n|$, 其中 $1 \leq k \leq j-1$.

② $|C_1| \leq |C_2| \leq \dots \leq |C_{j-1}| \leq |C_j|$, $|C_j| \geq |C_{j+1}| \geq \dots \geq |C_{t-1}| \geq |C_t|$, $|C_t| < |C_{t+1}|$, $|C_{t+1}| \geq \dots \geq |C_{n-1}| \geq |C_n|$, 其中 $j \leq t \leq n-1$.

③ $|C_1| \leq |C_2| \leq \dots \leq |C_{k-1}| \leq |C_k|$, $|C_k| > |C_{k+1}|$, $|C_{k+1}| \leq \dots \leq |C_{j-1}| \leq |C_j|$, $|C_j| \geq |C_{j+1}| \geq \dots \geq |C_{t-1}| \geq |C_t|$, $|C_t| < |C_{t+1}|$, $|C_{t+1}| \geq \dots \geq |C_{n-1}| \geq |C_n|$, 其中 $1 \leq k \leq j-1$, $j \leq t \leq n-1$.

(1) 在第 1 种情况下, 叶子节点数量为

$$\begin{aligned} \text{leaf} &= \sum_{i=1}^{k-1} f(|C_i|, |C_{i+1}|) + f(|C_k|, |C_{k+1}|) + \sum_{i=k+1}^{j-1} f(|C_i|, |C_{i+1}|) \\ &\quad + \sum_{i=j}^{n-1} f(|C_i|, |C_{i+1}|) + |C_n| \\ &= 0 + |C_k| - |C_{k+1}| + 0 + |C_j| - |C_n| + |C_n| \\ &= |C_k| - |C_{k+1}| + |C_j|. \end{aligned} \quad (16)$$

在式 (16) 中, 因为 $|C_k| > |C_{k+1}|$, 所以 $\text{leaf} > |C_j|$.

(2) 第 2 种情况下叶子节点的数量为

$$\begin{aligned} \text{leaf} &= \sum_{i=1}^{j-1} f(|C_i|, |C_{i+1}|) + \sum_{i=j}^{t-1} f(|C_i|, |C_{i+1}|) + f(|C_t|, |C_{t+1}|) \\ &\quad + \sum_{i=t+1}^{n-1} f(|C_i|, |C_{i+1}|) + |C_n| \\ &= 0 + |C_j| - |C_t| + 0 + |C_{t+1}| - |C_n| + |C_n| \\ &= |C_j| + |C_{t+1}| - |C_t|. \end{aligned} \quad (17)$$

在式 (17) 中, 因为 $|C_{t+1}| > |C_t|$, 所以 $\text{leaf} > |C_j|$.

(3) 在第 3 种情况下, 叶子节点的数量为

$$\begin{aligned} \text{leaf} &= \sum_{i=1}^{k-1} f(|C_i|, |C_{i+1}|) + f(|C_k|, |C_{k+1}|) + \sum_{i=k+1}^{j-1} f(|C_i|, |C_{i+1}|) \\ &\quad + \sum_{i=j}^{t-1} f(|C_i|, |C_{i+1}|) + f(|C_t|, |C_{t+1}|) + \sum_{i=t+1}^{n-1} f(|C_i|, |C_{i+1}|) + |C_n| \\ &= 0 + |C_k| - |C_{k+1}| + 0 + |C_j| - |C_t| + 0 + |C_{t+1}| - |C_n| + |C_n| \\ &= |C_j| + |C_k| - |C_{k+1}| + |C_{t+1}| - |C_t|. \end{aligned} \quad (18)$$

在式 (18) 中, 因为 $|C_k| > |C_{k+1}|$, $|C_{t+1}| > |C_t|$, 所以 $\text{leaf} > |C_j|$.

至此, 完成了必要性的证明, 确定了扫描树叶子节点取得最小值的充分必要条件. 以此为基础, 本文进一步提出了构建三维电路扫描树的优化算法, 能够在叶子节点取得最小值时优化 TSVs 的数量.

4 构建扫描树

在构建扫描树时, 首先要确定三维电路中含有哪些相容组, 然后再确定这些相容组的连接顺序, 以达到最小化叶子节点并优化 TSVs 数量的目的.

4.1 相容组的构建

本文采用图论文的方法来构建相容组. 首先将三维电路中的扫描单元作为图中的点, 若两个扫描单元的测试数据在测试期间没有冲突, 则相应的两点间有一条边相连. 图构造完成后, 采用图的划分算法将此图划分为多个完全子图. 一个完全子图就代表一个相容组, 子图所含的点就是该相容组所含的扫描单元. 构建相容组的具体过程如下:

(1) 将每个扫描单元都作为图中的一个点, 若两个扫描单元的测试数据在测试期间没有冲突, 则相应的两个点之间有条无向的边相连, 以此构造一个无向图 G ;

(2) 将图 G 复制到图 G_1 , 即令 $G_1 = G$;

(3) 在图 G_1 中找到度数最大的点 V ;

(4) 建立一个子图 G_V , 子图 G_V 由 G_1 中所有与点 V 相连的其他点构成;

(5) 将点 V 加入到集合 C ;

(6) 将图 G_V 复制到图 G_1 , 即令 $G_V = G_1$;

(7) 若 G_1 非空, 返回步骤 4 执行; 若 G_1 为空, 则集合 C 中新增的点构成一个相容组;

(8) 将 $G - C$ 复制到图 G_1 , 即令 $G_1 = G - C$, 返回步骤 (4) 执行, 直到图 G 中所有的点都在集合 C 中.

本文针对三维电路的每一层都使用上述算法来求得此层电路所含的相容组, 因而一个相容组所含的扫描单元都位于电路中的同一层. 在相容组确定后, 接下来的任务是确定相容组的连接顺序, 使得叶子节点最小并尽可能地减少 TSVs 的数量.

4.2 相容组的连接顺序

假设在 4.1 小节中, 求得三维电路中有 n 个相容组. 根据扫描树的特性 2, 采用如下的策略可以使扫描树中的叶子节点取得最小值: 将 n 个相容组分成两部分 L 和 R , 将 L 中的相容组按照从小到大的顺序连接, 将 R 中的相容组按照从大到小的顺序进行连接, 再将 L 中最大的相容组与 R 中最大的相容组进行连接. 以此种方法连接而成的扫描树其叶子节点数量可以取得最小值. 为了在叶子节点取得最小值时使得 TSVs 数量也最小, 关键问题是确定集合 L 和 R 中所含的相容组. 这属于集合划分的问题, 为了在多项式时间内得到解, 本文提出了一种启发式算法, 如图 3 所示.

算法的基本思想是首先随机地分配 $i(0 \leq i \leq n)$ 个相容组到集合 L 中, 余下的 $n - i$ 个相容组分配到集合 R 中. 然后再将 L 中 locked 值为 0 相容组依次与 R 中 locked = 0 的相容组交换, 得到一个所需 TSVs 数量较小的解. 交换过程一直重复, 直到 L 或 R 中所有相容组的 locked 值为 1. 变量 locked = 0 的作用是表明相容组是否可以交换. 若某个相容组的 locked 值为 0, 则表明该相容组可以参加交换; 若 $L(R)$ 中一个相容组的 locked = 1, 则表明该相容组是从 $R(L)$ 中交换过来的, 无需再次参加交换.

图 3 步骤 9 中的变量 t 是用来记录 L 中的相容组 $L[i]$ 与 R 中哪一个相容组进行交换可以使得 TSVs 值数量最小, 其初始值为 0. 在步骤 16 中, 若 t 的值还为 0, 则表明 $L[j]$ 无需与 R 中的相容组进行交换, 因而只需将 $L[j]$ 的 locked 值设为 1. 步骤 3, 4 和 12 中, 函数 $\text{Cal_TSV}(L, R, i)$ 的功能是计算将集合 L 和 R 中的相容组连接成扫描树所需的 TSVs 的数量. 即将集合 L 中的相容组按照从小到大的顺序排列, 集合 R 中的相容组按照从大到小的顺序排列, 然后按此顺序将相容组连接成扫描树所需的 TSVs 数量.

The proposed heuristic algorithm:

```

for( $i = 0; i \leq n; i++$ ) //  $n$  is the number of compatible groups
{
1  Randomly assign  $i$  compatible groups to  $L$ ;
2  Assign the rest  $n - i$  compatible groups to  $R$ ;
3  If ( $i == 0$ ) then  $\min_1 = \text{Cal.TSV}(L, R, 0)$ ;
4  if( $i == n$ ) then  $\min_2 = \text{Cal.TSV}(L, R, n)$ ;
5  else
    {
6    Set the value of locked for all compatible groups in  $L$  and  $R$  to 0;
7     $\min_3 = \text{Cal.TSV}(L, R, i)$ ;
8    while(there exists compatible groups with locked=0 both in  $L$  and  $R$ )
        {
9         $t = 0$ ;
10       Find the first compatible group with locked=0 in  $L$ ; // assume the compatible group is  $L[j], 1 \leq j \leq i$ ;
11       Exchange  $L[j]$  and the compatible groups with locked=0 in  $R$ ; // assume the exchanged
           // compatible group in  $R$  is represented as  $R[k], 1 \leq k \leq n - i$ ;
12        $N_{\text{tsv}} = \text{Cal.TSV}(L, R, i)$ ;
13       if  $N_{\text{tsv}} < \min_3$ , then  $\min_3 = N_{\text{tsv}}, t = k$ ;
14       Place  $L[j]$  and  $R[k]$  in the original position in  $L$  and  $R$ ;
15       repeat steps 11–14 until  $L[j]$  is exchanged with each compatible group with locked=0 in  $R$ ;
16       If ( $t == 0$ ) then the set the value of locked to 1 for  $L[j]$ ;
           else set the value of locked to 1 for  $L[j]$  and  $R[t]$ , and exchange  $L[j]$  and  $R[t]$ ;
        } // end while
    } // else end
} // end for
17  $\min_4 = \min\{\min_1, \min_2, \min_3\}$ ; the compatible groups in  $L$  and  $R$  is the solution when achieving
    the value of  $\min_4$ ;

```

图 3 确定 L 和 R 中的相容组

Figure 3 Determine the compatible groups in L and R

5 实验结果

本文针对 ISCAS'89 基准电路进行了实验, 采用 hMETIS 工具对电路进行划分, 将二维的电路转为三维电路^[20]. 并采用 C++ 编程实现了本文所的方法. 表 1 是本文方法与文献 [15] 方法的叶子节点数量的比较, 两者采用的都是 MINTEST 测试集^[21]. 表中第 1 列是电路名, 第 2~4 列是将基准电路划分为两层时叶子节点数量的比较. 其中, 第 2 列和第 3 列分别是本文方法和文献 [15] 方法叶子节点的数量, 第 4 列是叶子节点减少数量的百分比. 表 1 中第 5~7 列是三层电路时的比较结果, 第 8~10 是 4 层电路时的比较结果. 从表 1 中可以看出, 本文方法的叶子节点数量明显要少于文献 [15] 的方法.

表 1 本文方法与文献 [15] 叶子节点数量的比较

Table 1 Comparison results of leaf nodes between the proposed technique and [15]

Circuits	Two layers			Three layers			Four layers		
	prop.	[15]	red.(%)	prop.	[15]	red.(%)	prop.	[15]	red.(%)
s5378	7	20	65	6	18	66.67	6	16	62.50
s9234	15	18	16.67	12	18	33.33	10	17	41.18
s13207	88	106	16.98	74	127	41.73	49	130	62.31
s35932	139	140	0.71	94	165	43.03	76	197	61.42
s38584	34	39	12.82	24	45	46.67	23	61	62.30
avg.	56.6	64.6	22.43	42	74.6	46.29	32.8	84.2	57.94

而且随着电路层数的增多, 本文方法叶子节点的数量是减少的. 这是因为随着电路层数的增多, 每层电路所含的扫描单元数量随之减少, 造成最大相容组所含的扫描单元数量也减少. 由于本文方法叶子节点的数量为最大相容组所含的扫描单元数量, 所以随着电路层数的增加, 本文方法产生的叶子节点数量逐渐减少.

由于文献 [15] 中未给出 TSVs 的数量, 而且对于如何确定相容组之间连接顺序的细节也没有介绍清楚, 本文无法与文献 [15] 进行 TSVs 数量的比较. 因为在二维相容组连接时, 一般都采用将相容组按照从小到大或从大到小的顺序进行连接. 因而在表 2 中将本文方法所用的 TSVs 述量与将相容组按照从小到大以及从大到小的顺序进行连接所用 TSVs 的数量进行了比较. 由特性 2 可知, 因为将相容组按照从小到大或从大到小的顺序连接, 产生的叶子节点数量也是最小值, 与本文方法的叶子节点数量相同, 因此在表 2 中就没有给出叶子节点的数量, 仅对 TSVs 数量进行了比较. 表 2 中第 1 列是电路名, 第 2~4 列是 2 层电路时 TSVs 数量的比较. 第 2 列是本文方法所用的 TSVs 数量, 第 3 列 “s to b” 表示将相容组按照从小到大的顺序进行连接所用的 TSVs 数量, 第 4 列 “b to s” 表示将相容组按照从大到小的顺序进行连接所用的 TSVs 数量. 第 5~7 列和 8~10 列分别表示 3 层电路和 4 层电路时 TSVs 数量的比较. 从表 2 中可看出, 本文方法所用的 TSVs 数量要少于其他两种连接方法所用的 TSVs 数量.

表 3 是本文方法的压缩率和面积开销的结果. 表中的第 1 列是电路名, CR 表示压缩率, Area 表示面积开销. 由于本文针对 MINTTEST 测试集来求相容组, 而此测试集的前面几个测试向量几乎全部是确定位, 所以求得的相容组数量也较多, 压缩率也较低. 由于相容组只是本文构建扫描树算法的输入条件, 因而也可以采用根据电路结构来求相容组的方法, 这样会极大的提高压缩率. 从表中可看出, 随着电路层数的增多, 压缩率是降低的. 这是因为随着层数的增多, 三维电路相容组的数量会增加, 从而导致压缩率的降低. 表 3 中的面积开销是在 45 nm 工艺下, TSV 面积为 $2.47 \mu\text{m} \times 2.47 \mu\text{m}$ 时得到的结果 [22]. 从表中可看出, 随着电路层数的增加, 本文方法的面积开销是增大的. 这是因为随着电路层数的增多, 扫描树中所需的 TSVs 数量是增加的, 因而面积开销也是增大的.

6 结论

本文首先证明了扫描树叶子节点的最小值为扫描树中最大相容组中扫描单元的数量, 然后给出了叶子节点取得最小值时相容组连接顺序的充分必要条件. 最后根据此充分必要条件, 提出了一种在叶子节点最小化时优化 TSVs 数量的扫描树构建方法. 此方法将相容组分在两个集合中, 一个集合中的

表 2 TSVs 数量的比较

Table 2 Comparison results of TSVs number

Circuits	Two layers			Three layers			Four layers		
	prop.	s to b	b to s	prop.	s to b	b to s	prop.	s to b	b to s
s5378	8	22	32	24	48	48	30	63	66
s9234	19	40	46	34	59	80	41	83	100
s13207	89	315	352	190	337	390	432	686	762
s35932	259	1114	1416	864	1766	1872	914	2015	2162
s38584	24	98	150	96	221	244	161	321	312

表 3 压缩率和面积开销的结果

Table 3 Results of compression ratio and area overhead

Circuits	Two layers		Three layers		Four layers	
	CR(%)	Area(μm^2)	CR(%)	Area(μm^2)	CR(%)	Area(μm^2)
s5378	28.50	97.61	26.64	292.84	24.77	366.05
s9234	29.55	231.83	26.32	414.86	24.297	500.27
s13207	64.00	1085.96	61.86	2318.34	59.86	5271.18
s35932	96.60	3160.27	95.24	10542.36	93.76	11152.45
s38584	44.95	292.84	41.87	1171.37	40.23	1964.49

相容组按照其所含扫描单元的数量进行升序排列, 另一个集合进行降序排列, 并通过交换两个集合中的相容组来减少 TSVs 的数量. 实验结果表明, 与国际上已有方法相比, 本文所提方法有效地减少了叶子节点和 TSVs 的数量. 只要三维电路中各相容组确定后, 就可以利用本文所提的启发式算法来确定各相容组的连接顺序, 在扫描树叶子节点取的最小值的同时来优化 TSVs 的数量. 在本文中利用测试集来求得相容组只是为验证启发式算法提供数据, 也可以通过分析电路的结构, 将没有共同后继的扫描单元组成相容组. 通过这种求相容组的方法, 能够使得测试数据得到有效地压缩.

参考文献

- 1 Wang L T, Wu C W, Wen X Q. VLSI Test Principles and Architectures: Design for Testability. San Francisco: Morgan Kaufmann, 2006
- 2 Needham W M. Nanometer technology challenges for test and test equipment. IEEE Comput, 1999, 32: 52–57
- 3 Vranken H, Hapke F, Rogge S, et al. ATPG padding and ATE vector repeat per port for reducing test data volume. In: Proceedings of the IEEE International Test Conference, New York, 2003. 1069–1076
- 4 Touba N A. Survey of test vector compression techniques. IEEE Des Test Comput, 2006, 23: 294–303
- 5 Chandra A, Chakrabarty K. Test data compression and test resource partitioning for system-on-a-chip using frequency-directed run-length (FDR) codes. IEEE Trans Comput, 2003, 52: 1076–1088
- 6 Pandey A R, Patel J H. Reconfiguration technique for reducing test time and test volume in Illinois scan architecture based designs. In: Proceedings of the IEEE VLSI Test Symposium, Piscataway, 2002. 9–15
- 7 Rajski J, Tyszer J, Kassab M, et al. Embedded deterministic test. IEEE Trans Comput Aid, 2004, 23: 776–792
- 8 Han Y H, Hu Y, Li X W, et al. Embedded test decompressor to reduce the required channels and vector memory of tester for complex processor circuit. IEEE Trans VLSI Syst, 2007, 15: 531–540
- 9 Li K S M, Huang J Y. Synthesizing multiple scan trees to optimize test application time. IEEE Des Test Comput, 2011, 28: 62–69
- 10 Xiang D, Li K, Sun J G, et al. Reconfigured scan forest for test application cost, test data volume, and test power

- reduction. *IEEE Trans Comput*, 2007, 56: 557–562
- 11 Cheng Y S, You Z Q, Kuang J S. Test response data volume and wire length reductions for extended compatibilities scan tree construction. In: *Proceedings of the IEEE International Symposium on Electronic Design, Test & Applications*, Piscataway, 2008. 308–313
 - 12 Knickerbocker J U, Andry P S, Dang B, et al. Three-dimensional silicon integration. *IBM J Res Dev*, 2008, 52: 553–569
 - 13 Lee H H S, Chakrabarty K. Test challenges for 3D integrated circuits. *IEEE Des Test Comput*, 2009, 26: 26–35
 - 14 Syed U S, Chakrabarty K, Chandra A, et al. 3D-scalable adaptive scan(3D-SAS). In: *Proceedings of the IEEE International 3D Systems Integration Conference*, Piscataway, 2011. 1–6
 - 15 Li K S M, Liao Y Y. Layout-aware multiple scan tree synthesis for 3D SoCs. *IEEE Trans Comput Aid*, 2012, 31: 1930–1934
 - 16 Xiang D, Shen K L. A thermal-driven test application scheme for pre-bond and post-bond scan testing of three-dimensional ICs. *ACM J Emerg Tech Com*, 2014, 10: 1–19
 - 17 Kim D H, Mukhopadhyay S, Lim S K. Through-silicon-via aware interconnect prediction and optimization for 3D stacked ICs. In: *Proceedings of the IEEE/ACM International Workshop on System Level Interconnect Prediction*, New York, 2009. 85–92
 - 18 Pan D Z, Lim S K, Athikulwongse K, et al. Design for manufacturability and reliability for TSV-based 3D ICs. In: *Proceedings of the IEEE/ACM Asia and South Pacific Design Automation Conference*, New York, 2012. 750–755
 - 19 Hsieh A C, Hwang T, Chang M T, et al. TSV redundancy: architecture and design issues in 3D IC. In: *Proceedings of the IEEE/ACM Design, Automation, and Test in Europe*, New York, 2010. 166–171
 - 20 Karypis G, Kumar V, Shekhar S. Multi-level hypergraph partitioning: applications in VLSI design. In: *Proceedings of the IEEE/ACM Design Automation Conference*, New York, 1997. 536–539
 - 21 Hamzaoglu P, Patel J H. Test set compaction for combinational circuits. In: *Proceedings of the IEEE/ACM International Conference Computer-Aided Design*, New York, 1989. 283–289
 - 22 Yang C C, Chou C W, Li J F. A TSV repair scheme using enhanced test access architecture for 3-D ICs. In: *Proceedings of the IEEE Asian Test Symposium*, Piscataway, 2013. 7–12

Optimizing the number of leaf nodes and TSVs in three dimensional scan tree

LIU Jun^{1,2*}, WU Xi^{1,2}, LIANG HuaGuo³ & REN FuJi⁴

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 *School of Electronic Science and Applied Physics, Hefei University of Technology, Hefei 230009, China;*

4 *Department of Information Science & Intelligent Systems, Faculty of Engineering, The University of Tokushima, Tokushima 7708502, Japan*

*E-mail: liujun@ict.ac.cn

Abstract Scan tree architecture can effectively reduce test data volume, test time and test cost for integrated circuits. To reduce the number of leaf nodes and TSVs(through silicon vias) in scan tree for three dimensional integrated circuits, this paper firstly draws the conclusion that the minimum number of leaf nodes is the number of scan cells contained in the maximal compatible group. Then, the necessary and sufficient condition achieving the minimum number of leaf nodes is presented. On the basis above, a heuristic algorithm is proposed, which can minimize the number of leaf nodes and reduce consumed TSVs as many as possible. Experimental results demonstrate the effectiveness of the proposed technique.

Keywords three dimensional integrated circuits, scan tree, test data, test time, test cost

LIU Jun received his Ph.D degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He is now an Associate Professor in Hefei University of Technology. He is a member of Fault-Tolerant Computing Technical Committee, China Computer Federation. His research interests include 3D IC test, test data compression and built-in self-test (BIST).

WU Xi received his B.E. degree in 2000 and M.E. degree in 2003 from department of communication engineer, Hefei University of Technology. She is now an Assistant Professor in Hefei University of Technology. She is currently working toward the Ph.D. degree on computer science at Hefei University of Technology. Her research interests include ICs design and test, signal processing.

LIANG HuaGuo received B.S. degree in electronic engineering and M.S. degree in computer application technology from Hefei University of Technology, China in 1982 and 1989 respectively. He worked in University of Stuttgart as a senior visiting scholar and received Ph.D. degree from the University of Stuttgart, Germany, in 1998–2003. He is now a Professor and the dean of School of Electronic Science and Applied Physics, Hefei University of Technology. His research interests include synthesis and test of embedded system, built-in self test (BIST), automatic test pattern generation (ATPG) algorithm, fault tolerant computing etc.

REN FuJi received his B.E. degree in 1982 and M.E. degree in 1985 from the Department of Computer Sciences, Beijing University of Posts and Telecommunications, Beijing, China. He also received his Ph. D. degree in 1991 from the Faculty of Engineering, Hokkaido University, Sapporo, Japan. he is now a Professor of the Faculty of Engineering, The University of Tokushima. His current research interests include natural language processing, artificial intelligence etc. He is a senior member of IEEE, a Vice president of CAAI, and a Fellow of The Japan Federation of Engineering Societies.