

面向测试的SOC核间互连网络约简算法

尚玉玲^{1,2}, 李玉山¹

(1. 西安电子科技大学 电路CAD研究所, 陕西 西安 710071;

2. 桂林电子科技大学 电子工程学院, 广西 桂林 541004)

摘要: 随着集成电路的制造工艺和工作频率已经进入了深亚微米和吉赫兹时代, 片上系统核间互连总线的串扰测试已经成为不容忽视的问题. 通过对片上系统核间互连总线特征的研究, 提出了一种面向测试的片上系统核间互连总线的约简算法. 该算法首先对核间互连的拓扑结构进行描述, 建立互连关系矩阵, 并以受害线为根结点构建互连关系分级树, 根据测试精度去掉多余的攻击线. 其次, 根据互斥算法对包含三态双向驱动源的互连线进行筛选, 确定施加激励的互连线. 经过约简后的互连网络可根据测试矢量生成算法生成测试矢量集. 该算法使生成的测试矢量集大大缩小, 并提高了测试效率.

关键词: 片上系统; 核间互连; 三态双向网络

中图分类号: TN407 **文献标识码:** A **文章编号:** 1001-2400(2009)05-0871-06

Study of the reduction algorithm of SOC inter-core interconnects for testing

SHANG Yu-ling^{1,2}, LI Yu-shan¹

(1. Research Inst. of Electronic CAD, Xidian Univ., Xi'an 710071, China;

2. School of Electronic Eng., Guilin Univ. of Electronic Tech., Guilin 541004, China)

Abstract: With the manufacturing technology and operation frequency of VLSI entering the era of DSM and GHz, the crosstalk of SOC inter-core interconnects can not be ignored anymore. A reduction algorithm for core-external interconnect is presented based on characteristics of the interconnect bus of SOC. Any topology of inter-core interconnects is described first and an interconnect relationship tree is built. Unwanted aggressive interconnects could be cut off according to the accuracy of test. Interconnects with the tri-state and bi-direction driving source are sifted according to the mutex algorithm. Test patterns are generated for these interconnects after sifting according to the TPG algorithm. The test set is decreased and test efficiency is improved.

Key Words: system on chip; inter-core interconnects; tri-state bi-directed nets

随着集成电路制造技术和微电子技术的快速发展, 片上系统(System On Chip, 缩写为SOC)的制造工艺和工作频率已经进入了深亚微米和吉赫兹时代. 知识产权(IP)核间的互连总线已成为具有互连效应的传输线, 信号传输线间的耦合电容和耦合电感导致传输信号畸变, 出现了信号完整性问题. SOC的测试前期主要针对IP核^[1-2], 而深亚微米工艺的应用, 促使人们开始研究SOC核间互连总线的串扰故障测试模型, 测试和保证信号完整性就成为需要迫切解决的问题. 目前, 国内外针对串扰故障模型及测试问题已经开展了一些研究, 并取得了一定的成果, 比较有影响的测试模型有最大攻击者故障模型^[3]、多跳变故障模型^[4]、半跳变故障模型^[5]、最大化控制信号完整性故障模型^[6]等. 然而, 上述的故障模型均建立在3个假设的基础上: (1)互连线为理想传输线, 即互连线的宽度、耦合长度、线间距等均相等; (2)单向信号传输; (3)信号驱动端为双态驱动. 然而, 通过对SOC核间互连总线的研究可知, SOC中的互连线存在着大量三态双向的驱动源, 且互

收稿日期: 2009-03-18

基金项目: 国家自然科学基金资助(60871072, 60861003); 广西区教育厅项目资助(200808LX130)

作者简介: 尚玉玲(1977-), 女, 讲师, 西安电子科技大学博士研究生, E-mail: syl@guet.edu.cn.

连的拓扑结构比传统的理想互连线结构复杂. 鉴于此, 笔者以 SOC 核间互连总线的信号特征分析为基础, 以互连总线的几何结构为对象, 在保证故障覆盖率 100% 的前提下, 提出了 SOC 核间互连总线的约简算法, 对约简后的互连网络进行测试矢量生成, 可大大减少测试矢量集.

1 SOC 互连总线特征

SOC 设计技术是以模块设计方法为基础, 以 IP 核集成与复用为核心的芯片设计技术, 形成高集成、高速、低功耗的集成电路. 典型的 SOC 系统由多个 IP 核、用户定义逻辑 (User Defined Logic, 缩写为 UDL) 及核间互连总线组成, 而互连总线又由多条任意拓扑结构的互连线组成, 如图 1 所示.

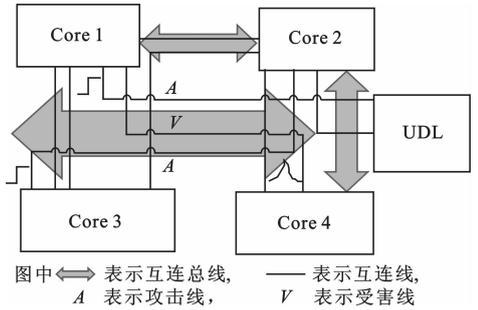


图 1 SOC 核间互连结构图

1.1 信号的三态及双向特征

在一般的理想互连线分析中, 假设传输线为双态 (低电平或高电平) 特征, 信号的传递方向是同侧单向, 在进行串扰测试时, 只需在传输线的同侧同时加载测试矢量. 然而, 在 SOC 中, 由于总线复用特点, 存在着大量的三态驱动源, 具有高电平、低电平与高阻抗三态特征, 因此, 施加控制信号可以使驱动源处于高阻抗状态或工作状态 (使能). 在高阻抗状态下, 传输线不传递信号. 同时, SOC 互连线中的信号传递方向具有双向特点, 可以通过控制信号控制双向传输线的传递方向, 如图 2 所示.

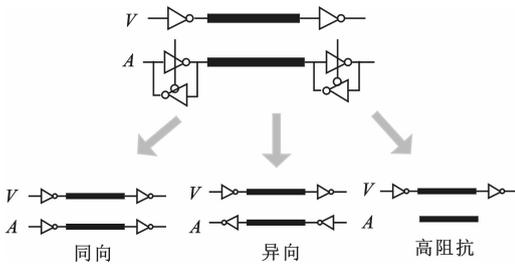


图 2 SOC 中的双向互连线

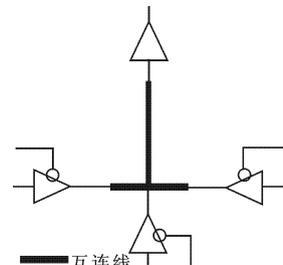


图 3 SOC 中的三态互连线

对于三态驱动的 SOC 串扰测试, 具有性质 1^[7-10].

性质 1 攻击线的同时异向串扰测试激励比同时同向测试激励可以在受害线上产生更强的攻击性.

1.2 信号的多驱动特征

在 SOC 中, 互连线存在着多个三态驱动源驱动设计, 如图 3 所示. 根据三态的特征可知, 在一个总线上同时只能有一个端口作输入, 这时其他端口必须在高阻态. 在串扰测试时, 施加的串扰故障测试激励应具有下列性质^[6].

- 性质 2** 受害线的三态驱动源至少有一个使能.
- 性质 3** 攻击线的三态驱动源至多有一个使能.
- 性质 4** 使所有攻击线的三态驱动源最大集使能.

1.3 驱动器尺寸特征

在 SOC 中, 存在着多种尺寸的驱动源, 为了研究不同驱动源尺寸对串扰大小的影响, 在 HSPICE 下进行了仿真, 仿真结果表明: 随着驱动源尺寸的变化, 信号的跳变时间会增大/减小, 进而影响驱动强度及串扰强度, 如图 4 所示. 从而得出性质 5.

性质 5 随着攻击线驱动源尺寸的增加, 驱动强度增强, 在受害线上产生的串扰增强.

1.4 耦合位置特征

在一般理想互连线的分析中, 假定攻击线与受害线等长, 起始位置相同, 也即耦合位置在攻击线的驱动源处. 然而, 在 SOC 中, 由于互连线间为任意互连拓扑结构, 长度不等, 耦合位置也会发生变化, 如图 5 所示.

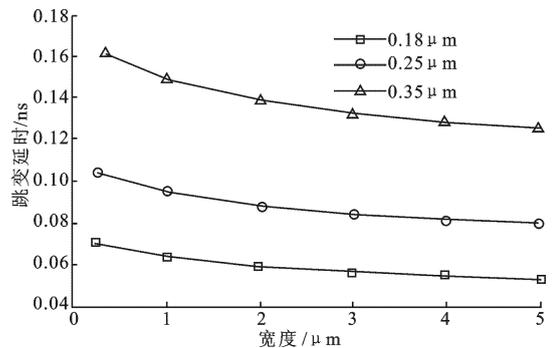


图 4 驱动器尺寸对延时的影响

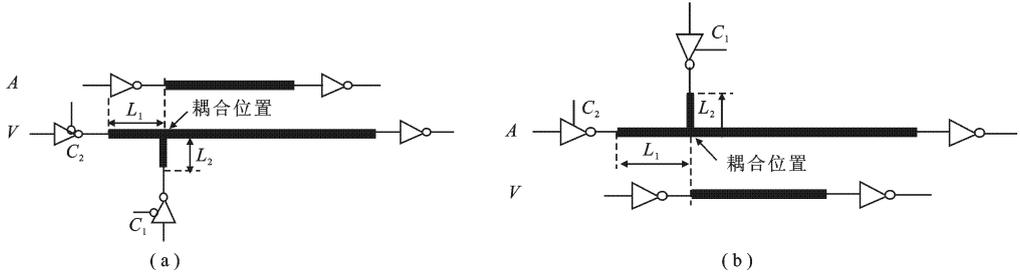


图 5 不同耦合位置的互连线

性质 6 当受害线长度较长,多驱动源攻击线中距离耦合位置较大的驱动源施加串扰测试激励.

性质 7 当受害线长度较短,多驱动源攻击线中距离耦合位置较小的驱动源施加串扰测试激励.

2 SOC 核间互连总线的互斥约简算法

SOC 的核间互连总线的互斥约简算法的基本思想是:首先对 SOC 核间任意拓扑结构的互连线进行描述,建立其互连关系矩阵,并以受害线为根结点构建其分级互连关系树,进行互连拓扑结构的约简.对约简后的互连网络分析其三态双向特征,根据互斥算法对双向三态的互连线进行筛选,去掉互斥的互连线,确定最终需要施加激励的互连网络,从而大大减少了测试矢量的数目.

2.1 互连拓扑结构的描述及约简

2.1.1 互连拓扑结构表达

设 SOC 的任意互连结构如图 6 所示.

设有 n 条互连线,对于任意互连拓扑结构,可以建立关联矩阵 $\mathbf{K}_{n \times n}$. 其中

$$K_{ij} = \begin{cases} 0 & , \quad i = j \vee L_i \text{ 与 } L_j \text{ 不相邻,} \\ 1 & , \quad L_i \text{ 与 } L_j \text{ 相邻.} \end{cases} \quad (1)$$

对于图 6 的互连拓扑结构,关联矩阵 $\mathbf{K}_{10 \times 10}$ 为

$$\mathbf{K}_{10 \times 10} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (2)$$

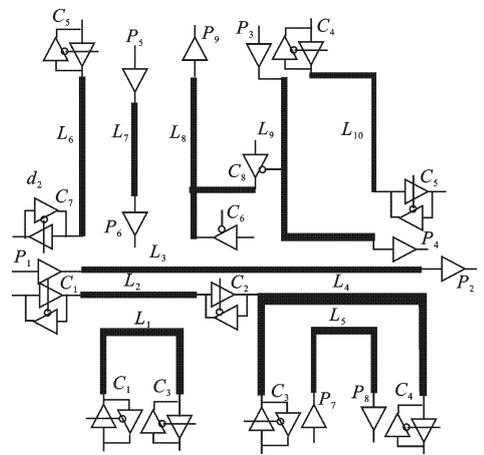


图 6 包含三态双向网络的互连网络示意图

2.1.2 互连拓扑结构的约简

设互连拓扑结构约简集合为 $V = [V_0, V_1, V_2, \dots, V_s]$, 其中 $V_i (i = 0, \dots, s)$ 为第 i 级关联集合.

设受害线为 L_i , 以 L_i 为根结点构建表达互连拓扑关系的分级关系树, 并记下每一级的互连线序号, 用于测试矢量生成时的互连拓扑结构的约简.

该生成过程的伪代码描述如下:

互连拓扑的分级关系树的生成函数 $I(k)$.

- (1) k 为受害线的序号变量;
- (2) {定义整型变量及级别: $\text{int } i, j, \text{level}=0$;
- (3) 置空队列 Q ;
- (4) 输出当前受害线的序号 k 及所在的级别 level ;
- (5) 并将该结点的访问标记设为真, $V[k]=\text{TRUE}$;
- (6) 将访问过的结点序号 k 入队列 Q : $\text{ENQUEUE}(Q, k)$;
- (7) 将“-1”作为 level 的标识入队列 Q : $\text{ENQUEUE}(Q, -1)$;
- (8) While(队列 Q 不为空)
- (9) { 从队列 Q 取出元素赋值给 i ;
- (10) if ($i=-1$) && (队列 Q 非空)
- (11) { $\text{ENQUEUE}(Q, -1)$;
- (12) Level++;
- (13) for($j=0$; $j<n$; $j++$)
- (14) {if($(K[i][j]=1)$ && (! $v[j]$))
- (15) {输出该互连线的序号 j 及所在的级别 level ;
- (16) 将访问过的结点序号 j 入队列 Q ;}}}
- (17) }

以图 6 为例,若以 L_1 为受害线构建互连分级树如图 7 所示,每级的互连线集合为:

$$V_0 = \{L_1\}; V_1 = \{L_2, L_4\}; V_2 = \{L_3, L_5\};$$

$$V_3 = \{L_8, L_9\}; V_4 = \{L_7, L_{10}\}; V_5 = \{L_6\}$$

根据实际测试需求的精度,选择相应的 level 级别,如选择 $\text{level}-3$,则约简集合 V 为

$$V = V_0 \cup V_1 \cup V_2 \cup V_3 = \{L_1, L_2, L_4, L_3, L_5, L_8, L_9\}$$

2.1.3 三态双向互斥约简

定义 1 互斥.

在 SOC 中,若存在两条互连线 A, B ,如果任一三态双向驱动源为 A 互连线的驱动源,又为 B 互连线的接收端,则称 A 与 B 互斥.

在 SOC 中,一般同时存在双向互连及单向互连.对于单向互连,由于驱动源确定,所以,正确的设计中已经进行了约束,因此不需要进行互斥约简.然而,对于三态双向互连,基于三态特性及上述性质 2~性质 4,可以根据受害线中的驱动源激励方向,对需要进行互斥的三态驱动源进行约简.互斥约简算法如下:

设有 n 条互连线 L_1, L_2, \dots, L_n ,均为三态双向互连线,互连线的两个端点分别为 (a, b) .

步骤 1 建立双向互连线.

对于信号传递的方向为 $a \rightarrow b$ 时,建立有序对 $L: \langle a, b \rangle$;

对于信号传递的方向为 $b \rightarrow a$ 时,建立异向互连线有序对 $L': \langle a', b' \rangle$,此时 $a = b'$ 且 $b = a'$,称 L' 与 L 为互补互连线.

设集合: $Z_1 = \{L_1, L_2, \dots, L_n\}, Z_2 = \{L'_1, L'_2, \dots, L'_n\}$,双向互连线集合为 $S = Z_1 \cup Z_2$.

以图 6 为例,其中三态双向互连线为 $L_1, L_2, L_4, L_6, L_{10}$,其异向互连线对为:

$$L_1: (1, 3) \leftrightarrow L'_1: (3, 1) \quad , \quad L_2: (1, 2) \leftrightarrow L'_2: (2, 1) \quad , \quad L_4: (3, 4) \leftrightarrow L'_4: (4, 3) \quad ,$$

$$L_6: (5, 7) \leftrightarrow L'_6: (7, 5) \quad , \quad L_{10}: (4, 5) \leftrightarrow L'_{10}: (5, 4) \quad ,$$

则 $Z_1 = \{L_1, L_2, L_4, L_6, L_{10}\}, Z_2 = \{L'_1, L'_2, L'_4, L'_6, L'_{10}\}$.

双向互连线集合 $S = Z_1 \cup Z_2 = \{L_1, L_2, L_4, L_6, L_{10}, L'_1, L'_2, L'_4, L'_6, L'_{10}\} = \{S_1, S_2, \dots, S_{10}\}$. (3)

步骤 2 建立互斥矩阵 Q .

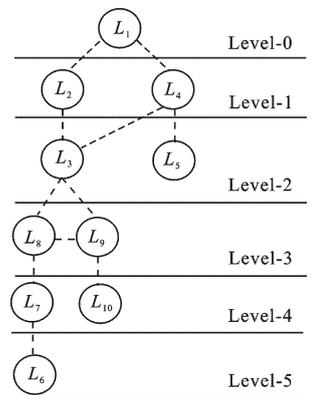


图 7 以 L_1 为根的互连关系树

互斥矩阵 Q 为互连线及其异向互连线之间的关联关系,对于 n 条互连线,有 m 条双向互连线 $S, m = 2n$, 互斥矩阵 $Q_{m \times m}$ 为

$$Q_{m \times m} = \begin{matrix} & L_1 & \cdots & L_n & L'_1 & \cdots & L'_n \\ \begin{matrix} L_1 \\ \vdots \\ L_n \\ L'_1 \\ \vdots \\ L'_n \end{matrix} & \left[\begin{array}{cccccc} q_{11} & \cdots & q_{1n} & q_{1(n+1)} & \cdots & q_{1m} \\ \vdots & & \vdots & \vdots & & \vdots \\ q_{n1} & \cdots & q_{nm} & q_{n(n+1)} & \cdots & q_{nm} \\ q_{(n+1)1} & \cdots & q_{(n+1)n} & q_{(n+1)(n+1)} & \cdots & q_{(n+1)m} \\ \vdots & & \vdots & \vdots & & \vdots \\ q_{m1} & \cdots & q_{mm} & q_{m(n+1)} & \cdots & q_{mm} \end{array} \right] & = & \begin{matrix} S_1 & \cdots & S_m \\ \vdots & & \vdots \\ S_m & \left[\begin{array}{ccc} q_{11} & \cdots & q_{1m} \\ \vdots & & \vdots \\ q_{m1} & \cdots & q_{mm} \end{array} \right] & \end{matrix} \end{matrix} \quad (4)$$

设两条双向互连线 $S_i: (a_i, b_i)$ 与 $S_j: (a_j, b_j), q_{ij}$ 取值如下:

$$q_{ij} = \begin{cases} -1 & , \quad \exists a_i \exists b_j ((a_i = b_j) \vee (a_j = b_i) \vee ((a_i = a_j) \wedge (b_i = b_j))) \\ 1^+ & , \quad \exists a_i \exists b_j (a_i = a_j) \wedge (b_i \neq b_j) \\ 1^- & , \quad \exists a_i \exists b_j (b_i = b_j) \wedge (a_i \neq a_j) \\ 0 & , \quad \exists a_i \exists b_j (a_i \neq a_j) \wedge (b_i \neq b_j) \end{cases} \quad (5)$$

对于图 6,可建立如下 Q 阵:

$$Q_{10 \times 10} = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \\ S_{10} \end{matrix} & \left[\begin{array}{cccccccccc} -1 & 1^+ & -1 & 0 & 0 & -1 & -1 & 1^- & 0 & 0 \\ 1^+ & -1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & -1 & 1^+ & 0 & -1 & 0 & 1^- \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & -1 & 1^+ \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 & 1^+ & 1^- & -1 \\ -1 & -1 & 1^+ & 0 & 0 & -1 & 1^- & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 1^- & -1 & 0 & 0 & 0 \\ 1^- & 0 & -1 & 0 & 1^+ & -1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1^- & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1^+ & -1 & 0 & 0 & 0 & -1 & -1 \end{array} \right] & \end{matrix} \quad .$$

步骤 3 指定受害线 S_k .

步骤 4 建立互斥集合.

定义 2 出度集合 C_1 指对于受害线 S_k ,若互连线 S_i 与 S_k 具有相同的起始端,即 $a_i = a_k$,则 $S_i \subset C_1$.

定义 3 入度集合 C_2 指对于受害线 S_k ,若互连线 S_i 与 S_k 具有相同的终止端,即 $b_i = b_k$,则 $S_i \subset C_2$.

定义 4 自由集合 C_3 指对于受害线 S_k ,若互连线 S_i 与 S_k 的起始端与终止端均不相同,即 $a_i \neq a_k$ 且 $b_i \neq b_k$,则 $S_i \subset C_3$.

设 $C_1 = C_2 = C_3 = \emptyset$,在 Q 阵中,对于受害线 S_k ,有

$$C_1 = C_1 + \{S_j\} \quad , \quad \text{当且仅当 } q_{kj} = 1^+ \quad , \quad j = \{1, 2, \dots, m\} \quad , \quad (6)$$

$$C_2 = C_2 + \{S_j\} \quad , \quad \text{当且仅当 } q_{kj} = 1^- \quad , \quad j = \{1, 2, \dots, m\} \quad , \quad (7)$$

$$C_3 = C_3 + \{S_j\} \quad , \quad \text{当且仅当 } q_{kj} = 0 \quad , \quad j = \{1, 2, \dots, m\} \quad . \quad (8)$$

当图 6 中 $L_1: (1, 3)$ 为受害线, $C_1 = \{S_2\} = \{L_2\}, C_2 = \{S_8\} = \{L'_4\}, C_3 = \{S_4, S_5, S_9, S_{10}\} = \{L_6, L_{10}, L'_6, L'_{10}\}$.

步骤 5 互斥筛选.

定理 1 出度集合中的元素间不存在互斥.

定理 2 入度集合中的元素间不存在互斥.

(1) 出度-入度集合互斥筛选. 对于出度集合 C_1 中的任意元素 C_{1i} 与入度集合任意元素 $C_{2j}, C_2 =$

$C_2 - \{C_{2j}\}$, 当且仅当 $q_{ij} = -1$;

(2) 出度-自由集合互斥筛选. 对于出度集合 C_1 中的任意元素 C_{1i} 与自由集合任意元素 C_{3j} , $C_3 = C_3 - \{C_{3j}\}$, 当且仅当 $q_{ij} = -1$;

(3) 入度-自由集合互斥筛选. 对于入度集合 C_2 中的任意元素 C_{2i} 与自由集合任意元素 C_{3j} , $C_3 = C_3 - \{C_{3j}\}$, 当且仅当 $q_{ij} = -1$;

(4) 自由集合内部互斥筛选. 对于自由集合 C_3 中的任意两元素 C_{3i} 与 C_{3j} , 若 $q_{ij} = -1$, 则

(a) 基于性质 4 进行筛选;

(b) 若基于性质 4 无法筛选, 且 C_{3i} 与 C_{3j} 为互补互连线时, 即该互连线的驱动源不受约束, 可以加载双向信号, 则 C_{3i} 与 C_{3j} 同时保留.

步骤 6 获得三态双向互斥约简集合 $C = C_1 \cup C_2 \cup C_3$. (9)

以图 6 中的 $L_1: (1, 3)$ 为受害线时, 其互斥约简后的互连线集合如下:

$$C_1 = \{S_2\} = \{L_2\}, \quad C_2 = \{S_8\} = \{L_4\}, \quad C_3 = \{S_5, S_9\} = \{L_{10}, L_6\}, \\ C = C_1 \cup C_2 \cup C_3 = \{S_2, S_8, S_5, S_9\} = \{L_2, L_4, L_{10}, L_6\} .$$

3 算法分析

上述的互斥约简算法对 SOC 核间互连网络进行了约简, 大大减少了可测的网络数目. 以图 6 的互连网络为例, 其中包含了 10 条互连线, 受害线为 L_1 , 如果不经任何约简, 则除受害线外, 所有的攻击线均需要施加测试激励. 以 MT 模型为例^[4], MT 模型所需的测试矢量数目为 $N_{\text{pattern}} = N \times 4^{2N-1} = N \times 2^{N+1}$ (N 为互连网络的数目). 因为包含了双向的互连网络, 则需要施加的测试矢量全集为 MT 测试矢量集的两倍, 即 40 960 条双矢量, 经过约简后, 只需要 640 条双矢量. 对于不同的测试矢量生成算法, 生成的测试矢量集将得到不同程度的压缩. 由此可见, 该互斥约简算法对 SOC 的核间互连网络进行约简可大大缩小测试矢量集, 进而提高测试效率.

4 总 结

以上提出了一种 SOC 核间互连总线的互斥约简算法, 分析了 SOC 总线的特征, 得出三态双向网络、驱动器尺寸及耦合位置对串扰的影响. 在此基础上提出了互斥及约简算法, 对被测的互连网络进行化简, 对化简后的互连网络进行渐进式地测试矢量生成. 该方法对于进一步优化测试矢量集具有一定的效果.

参考文献:

- [1] 张弘, 杨莉, 李玉山. 通用乘法器 IP 的可测性设计[J]. 西安电子科技大学学报, 2003, 30(3): 65-68.
Zhang Hong, Yang Li, Li Yushan. Research on the Design for the Testability of the General Multiplier[J]. Journal of Xidian University, 2003, 30(3): 65-68.
- [2] 雷绍充, 邵志标, 梁峰. 一种新颖的乘法器核内建自测试设计方法[J]. 西安电子科技大学学报, 2006, 33(5): 153-157.
Lei Shaochong, Shao Zhibiao, Liang Feng. A Novel BIST Technique for Multipliers Cores[J]. Journal of Xidian University, 2006, 33(5): 153-157.
- [3] Cuvillo M, Dey S, Xiaoliang Bai. Fault Modeling and Simulation for Crosstalk in System-on-chip Interconnects [C]// IEEE/ACM International Conference on Computer-Aided Design. Los Alamitos: IEEE Computer Society, 1999: 297-303.
- [4] Tehranipour M H, Ahmed N, Nourani M. Multiple Transition Model and Enhanced Boundary Scan Architecture to Test Interconnects for Signal Integrity [C]//IEEE International Conference on Computer Design: VLSI in Computers and Processors. Los Alamitos: IEEE Computer Society, 2003:554-559.