

VLSI 功能测试的最佳测试序列的选取*

曾成碧**

陈光禔

(四川大学电力系 成都 610065) (电子科技大学 CAT 室 成都 610054)

【摘要】 介绍了 VLSI 功能测试向量生成的 Petri 网模型和 Petri 网模拟测试序列中指令的关系, 构造了压缩存储网络的拓扑信息的 Petri 网简约矩阵。用 Petri 网简约矩阵检查测试序列的不确定度, 使测试序列不确定度最小, 从而优化了测试序列, 即从给定的指令中选出了最佳测试序列。

关键词 最佳测试序列; 功能测试; 数字电路; Petri 网

中图分类号 TN707

随着集成电路的迅速发展, 复杂程度的增加和可测点的减少使测试愈加困难。对用户来说往往只知道指令集, 因此只能进行功能测试(如微处理器的测试)。目前经常使用的系统图法、RTL 和微编程技术等, 都是较有效的功能测试方法^[1], 但存在一些不足, 如测试算法复杂、测试码欠优化、故障覆盖率不高等^[2]。衡量功能测试的质量包括测试序列、序列长度和故障覆盖率, 通过测试序列的指令维数和指令顺序, 比较测试码的不确定度和故障覆盖率来判断测试性能的好坏。本文以 VLSI 功能测试生成的 Petri 网模型为基础^[3-5], 采用 Petri 网矩阵法检查测试指令的不确定度, 对测试码进行优化, 从给定的指令集中选出最佳的测试序列。

1 测试序列的 PN 模型

定义 指令维数是执行段连续访问寄存器的次数。若微指令至少访问一次寄存器, 则维数与执行段的微指令数一致; 在测试 I' 下, 指令的测试序列由 I' 的指令序列组成; 测试过程是指令集的每一条指令的测试序列的集合。在同一测试序列中, 当不能完全测试两条以上不同指令时, 就产生不确定性, 导致故障的不确定性(或故障被掩盖)^[4]。

一个测试序列由设置序列的前任指令集、测试 I' 的指令、提供观察序列的后继指令集三部分组成。本文研究以 Petri 网为工具, 从给定的指令集中选出最佳的测试序列。

如图 1 所示, 近似模型 A 代表一条指令: 输入变迁、位置、输出变迁。位置中有标识表示完成指令测试, 连接测试序列的所有 I' 指令的输入输出变迁, 建立测试序列 I' 的 Petri 网模型^[3,4]。

定理 1 仅当所有指令测试完才能实现最小程度的不确定性(不确定度最小)。

证明 假设至少有一条指令 I'' 未测试, 属于设置序列或观察序列。首先, 不知 I'' 运行的设置数据是否正确装载, 也不知运行结果是否可观察。若运行结果与期望值不同(此时有一故障), 在 I' 和 I'' 之间故障特性不确定; 若运行结果与期望值相同, 故障被掩盖, 任意情况下不确定度将增加。已测试指令作为图形的根, 整个图形象棵树, 模拟测试序列的 PN 图, 不确定性将呈循环圈。

定理 2 维数是 k 的测试指令, 为使不确定度最小, 属于测试序列的全部指令维数不大于 k^2 。

如图 2 所示, 近似模型 B^2 代表指令集和故障区(包括寄存器和其他功能单元)对任意指令 I' : 输入寄存器(或外部引脚)集合 $R_I(I')$; 输出寄存器(或外部引脚)集合 $R_O(I')$ 。PN 模型中, 与 I' 的输入变迁相连的位置 r 代表 R_I 、 R_O 寄存器, 同时属于 R_I 和 R_O 的同一寄存器标为特殊位置 r , 因为其功能在两种情况下不同。在 $R_I(I')$ 中, 任意前任指令集 $I_p \in P(I')$ (如 $R_O(I_p) \cap R_I(I') \neq \emptyset$) 与共享寄存器位置 r

1999年10月25日收稿

* 国家“八五”重点科研项目

** 女 29岁 博士生 讲师

相连, I_p 作递归运算, 直到所有 I_p 与根重合; 在 $R_o(I')$ 中, 任意后继指令集 $I_s \in S(I')$ (如 $R_s(I_p) \cap R_o(I') \neq \emptyset$) 与共享寄存器位置 r 相连, I_s 作递归运算, 直到无后继指令。

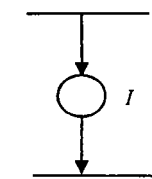


图 1 近似模型 A

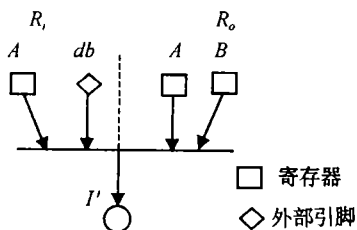


图 2 近似模型 B

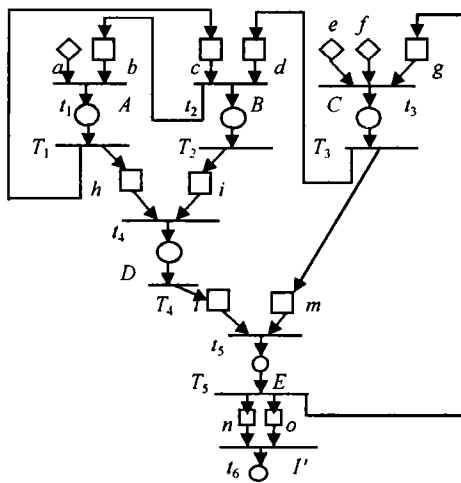


图 3 测试序列的 PN 模型

2 Petri 网的简约矩阵 D^- : 压缩信息

模型 B 中, 位置 I (指令位置) 的入度和出度总为 1, 位置 I 只将输入位置 r 的输入数据 (可能修改) 传送到下一位置 I 的位置 r 中, 不同的子图合并会影响位置 r 的入度, 但出度仍为 1, 这种特性使 $R_s(I_p)$ 和 $R_o(I_p)$ 都有标识, 标识直接转移到 I_{p+1} 的位置 r , 此特性对 D^- 矩阵很重要。任意路径 PN 由位置 r 、变迁、位置 I 、变迁组成, 如图 3 所示。通过减少变迁, 可得到沿着网络的路径由位置 r 和位置 I 交替组成。邻接矩阵 D 存储 PN 的拓扑信息, 其缺点是邻接矩阵 D 太大 ($N \times M$, N 为变迁数, M 为位置数)。构造简约矩阵 D^- 能压缩有用信息, 而且大大减小矩阵行列数。考虑矩阵 D^- , $D^- = \# [p_i, I(t_j)]$ 代表输入函数, D_{ij}^- 表示从位置 j 到变迁 i 的连接弧 (位置出度), 由于位置出度为 1, D^- 矩阵只有 1 或 0。

定义 1 $\{t_i\}$ 为位置 r 输出的变迁集 (输入位置 I); $\{T_j\}$ 为输入位置 r 的变迁集 (位置 I 输出); $\{r_k\}$ 为位置 r 集; $\{I_h\}$ 为位置 I 集; N_t 为 $\{t_i\}$ 的维数; N_T 为 $\{T_j\}$ 的维数; M_r 为 $\{r_k\}$ 的维数; M_I 为 $\{I_h\}$ 的维数; 其中 $N_t = N_T + 1$, 这是由于不考虑 I' 的输出变迁。

定义 2 任意给定位置 r 的元素 r_k , 通过输入变迁与位置 I 的元素 I_h 相连, 则定义 I_h 的维数作为 r_k 的相关维数, 给定指令集 I' , $R_s(I')$ 和 $R_o(I')$ 的所有元素的相关维数相同。另外, 同一位置不同的指令, 则相关维数不同。

矩阵 D^- 的行列值见式 (1), 其中列按维数增加的顺序排列。位置 r 按维数增加的顺序排列 (即从 $1 \sim M_r$ 列), 位置 I 按维数增加的顺序排列 (即从 $M_r + 1 \sim M_r + M_I$ 列); 行以相同的标准排列 $\{t_i\} \cup \{T_j\}$, 将变迁 t_i 按维数增加的顺序排行 (即从 $1 \sim N_t$ 行), 然后变迁 T_j 按维数增加的顺序排行 (即从 $N_t + 1 \sim N_t + N_T$ 行)。这样组成了矩阵 D^- : 左上子矩阵 $D_{11}^- (N_t \times M_r)$ 包含了全部有用信息; 右上子矩阵 $D_{12}^- (N_t \times M_I)$ 为 0; 左下子矩阵 D_{21}^- 为 0; 右下子矩阵 $D_{22}^- (N_T \times M_I)$ 是单位阵。

矩阵 D^+ 的行列按相关维数增加的顺序排列, $D^+ = \# [p_i, O(t_j)]$ 代表输出函数, 见式 (2)。同理矩阵 D^+ 左上子矩阵 $D_{11}^+ (N_T \times M_r)$ 包含了全部有用信息; 右上子矩阵 $D_{12}^+ (N_T \times M_I)$ 为 0; 左下子矩阵 D_{21}^+ 为 0; 右下子矩阵 $D_{22}^+ (N_T \times M_I)$ 是单位阵。

邻接矩阵 $D = D_{22}^- D^+$ (见式 (3)), 则矩阵 D 的特性为左上子矩阵 $D_{11} (N_t \times M_r)$ 包含了全部有用信息, 子矩阵 D_{12} 、 D_{21} 、 D_{22} 都为 0, 这样可以获得约简矩阵 $D^- = D_{11}$ (见式 (4))。

$$D^- = \begin{pmatrix} & a & b & c & d & e & f & g & h & i & l & m & n & o & A & B & C & D & E \\ t_1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_2 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_3 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ T_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ T_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ T_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ T_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

$$D^+ = \begin{pmatrix} & a & b & c & d & e & f & g & h & i & l & m & n & o & A & B & C & D & E \\ T_1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ t_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ t_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ t_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ t_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

D' 的列对应于位置 r , 行对应位置 l 的变迁(T_j-t_j)。矩阵 D' 的元素 $d_{ij} \in D'$, $d_{ij} = -1$: r_j 是 l_i 的前趋指令; $d_{ij} = 0$: r_j 与 l_i 无直接联系; $d_{ij} = 1$, r_j 是 l_i 的后继指令。

$$D = \begin{pmatrix} & a & b & c & d & e & f & g & h & i & l & m & n & o & A & B & C & D & E \\ T_1-t_1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_2-t_2 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_3-t_3 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_4-t_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_5-t_5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_1-T_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_2-T_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_3-T_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_4-T_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_5-T_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

$$D' = \begin{pmatrix} & a & b & c & d & e & f & g & h & i & l & m & n & o \\ T_1-t_1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ T_2-t_2 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ T_3-t_3 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ T_4-t_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 \\ T_5-t_5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (4)$$

3 用矩阵 D' 优化测试序列

衡量不同测试序列的两个属性是覆盖率和不确定度。通过检查单个测试序列可以诊断出不确定度。本文只考虑测试序列的不确定度, 从给定的指令中寻求最佳测试序列。考察矩阵 D' 的一列: 若任意列 j 的和 $\sum p_{ij} = -1$, r_j 是位置 r 的根, 位置 r 的根是测试序列的可控点; $\sum p_{ij} = 0$, r_j 是连接两个位置(I_1, I_2)的位置 l , 即 $r_j \in R_r(I_1) \cap R_r(I_2)$ 。 $\sum p_{ij} > 0$, 则有两种可能: 1) j 列只有一个元素是 1, r_j 是位置 r 的叶, r_j 是直接是测试 I' 的前趋指令; 2) j 列多个元素是 1, r_j 是交换测试序列的合并位置 r , 若无负元素仍位置 r 的叶(即不同测试序列的合并仅是测试 I' 的前趋指令)。

研究位置 r 的叶, 判断 Petri 网的初始标识, 初始标识的转移产生终态标识。若初始标识覆盖所有的叶, 测试没有不确定性; 否则, 修正 Petri 网, 激活变迁, 产生循环圈。循环圈意味着不确定度, 则需进一步检查 D' 。 D' 的正负元素在位置 r 和位置 l 之间类似“生产者-消费者”的关系。这样 Petri 网的循环圈与 D' 中沿行列正负元素交替的闭合路径相对应。

为使不确定度最小, 遵循打开循环圈的条件是测试指令维数最低, 并且循环圈无标识的输入位置 r 数最小。可见, 测试指令的最小不确定度(即最优的测试序列)必须符合考察维数最低指令, 选择最小输入无标识的位置 r 数。

$$\begin{bmatrix} & a & b & c & d & e & f & g & h & i & l & m & n & o \\ T_1-t_1 & -1 & -1 & \underbrace{1 \ 0 \ 0 \ 0 \ 0 \ 1}_{\text{循环圈}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_2-t_2 & 0 & \underbrace{1 \ -1 \ -1}_{\text{循环圈}} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ T_3-t_3 & 0 & 0 & 0 & \underbrace{1 \ -1 \ -1 \ -1}_{\text{循环圈}} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ T_4-t_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \underbrace{1 \ -1 \ -1}_{\text{循环圈}} & 0 & 0 & 0 & 0 & 0 \\ T_5-t_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \underbrace{0 \ 0 \ -1}_{\text{循环圈}} & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

图 4 循环圈代表不定性

$$\begin{bmatrix} & a & b & c & d & e & f & g & h & i & l & m & n & o \\ T_1-t_1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ T_2-t_2 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ T_3-t_3 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ T_4-t_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 \\ T_5-t_5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 1 & 1 \end{bmatrix}$$

图 5 循环圈的打开

维数最低指令与 D' 的循环圈中负元素对应, 其对应的位置 r 反馈输入变迁到考察指令, 然后判断维数相同的指令间输入无标识的位置 r 数大小, 再讨论每条考察指令在循环圈中的负元素数, 由于循环圈, 位置 r 不装载标识, 如图 4 所示, 元素数最小的行与测试最佳指令对应。一旦选定测试指令, 循环圈可打开。此时, 在无标识的位置 r 加入标识, 激活输入变迁。循环圈中位置 r 的标识复位 0, 产生新根, 循环圈不存在, 如图 5 所示。由此可以得到给定指令的不同测试序列的比较和循环圈的诊断算法。这样从循环圈数、循环圈长度和测试序列长度等参数比较, 选择出模糊度最小的最佳测试序列。

4 结束语

本文以 VLSI 功能测试生成的 Petri 网模型为基础, 讨论 Petri 网的经典矩阵 D' 和 D' 推导邻接矩阵 D' 以及由矩阵 D' 得到简约矩阵 D' 的过程, 简约矩阵 D' 压缩有用信息, 大大减小矩阵行列数。利用简约矩法检查测试指令的不确定度, 对测试码进行优化, 从给定的指令中选出不确定度最小的最佳测试序列。

参 考 文 献

- 1 陈光福, 张世冀. 数据域测试及仪器. 北京: 电子工业出版社, 1994
- 2 Distanto F. A Petri net matrix approach in VLSI functional testing. EUROMICRO, 1985: 347~355
- 3 Bedina M, Distanto F, Gsami M. A petri net model for microprocessor functional test. Proc Journess d'Electronique, 1983: 149~160
- 4 Sami M G, Bedina M. A formal approach to computer-assisted generation functional test patterns for VLSI device. Proc ISCAS 84, 1984(1): 19~23
- 5 曾成碧, 陈光福. 可达树分析法及其在测试中的应用. 电子科技大学学报, 1999, 28(5): 524~527

Choice of Best Test Sequence for VLSI Functional Testing

Zeng Chengbi

Chen Guangju

(Dept. of Electrical Power Engineering, Sichuan University Chengdu 610065) (CAT Lab., UEST of China Chengdu 610054)

Abstract This paper introduces Petri net model of functional test pattern generation for VLSI. Relationships between operators involved in test sequence are accomplished via Petri. Reduced matrix compacting storage of net topology is constructed. Ambiguity of test sequence is checked via Petri net matrix. Test pattern is optimized and the best test sequence is got.

Key words digital circuits; best test sequence; functional test; Petri net