

基于单元故障模型的树型加法器的测试

李兆麟¹⁾ 盛世敏²⁾ 吉利久²⁾ 王阳元²⁾

¹⁾(清华大学计算机科学与技术系高性能研究所 北京 100084)

²⁾(北京大学微电子学研究所 北京 100871)

摘 要 首先分析了树型加法器的原理,总结了其运算特性.其次在介绍单元故障模型的基础上分析了树型加法器的测试向量生成.分析结果表明, $5n-1$ 个测试向量可以实现树型加法器中所有单元故障的检测.这些测试向量具有很好的规则性,能够利用片上测试向量生成器实现,适合于应用内建自测试技术测试.基于此,作者提出了一种内建自测试的测试结构,测试时只需存储 7 个籽测试向量,其它测试向量可以在这 7 个籽测试向量的基础上通过循环移位实现.最后给出了实验分析结果.

关键词 树型加法器;单元故障模型;测试向量生成;内建自测试

中图法分类号 TP332

Testing of Tree Adder Based on Cell Fault Model

LI Zhao-Lin¹⁾ SHENG Shi-Min²⁾ JI Li-Jiu²⁾ WANG Yang-Yuan²⁾

¹⁾(Institute of High Performance Computing, Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

²⁾(Institute of Microelectronics, Peking University, Beijing 100871)

Abstract Based on cell fault model, the paper studies test pattern generation and self test of tree adder, which is frequently used in the high performance processors. Firstly, the paper analyzes the tree adder's principle and concludes its properties. Secondly, based on the introduction of cell fault model, which is fit for the testing of regular circuits, the paper presents a complete solution to tree adder testing. In conclusion, $5n-1$ test patterns are enough to exhaustively test all detectable cell faults and any modification of tree adder is not required. Experiment results show that these patterns can guarantee that all cell faults are tested. These patterns have high regularity and can be effectively produced on-chip as required for BIST. Furthermore, a kind of test structure suited for BIST is presented in the paper. When testing, only 7 seed patterns are memorized and all other test patterns can be realized by loop shifting.

Keywords tree adder; cell fault model; test pattern generation; BIST

1 引 言

集成电路进入了高度集成的系统芯片时代,集

成电路测试面临着更大的考验.例如,传统的固定型故障模型由于计算量过大或不可复用等不足,已经不适应系统芯片的测试.随着被测电路规模的增大,ATPG 费用越来越高,而且由 ATPG 生成的测试向

收稿日期:2001-08-16;修改稿收到日期:2003-03-03.本课题得到国家“八六三”高技术研究发展计划重点项目(2002AA1Z030)和国家自然科学基金(60106004)资助.李兆麟,男,1973年生,博士,讲师,主要研究方向为微处理器体系结构、微处理器设计及测试、集成电路设计及测试方法学等. E-mail: lzl73@tsinghua.edu.cn. 盛世敏,男,1946年生,教授,博士生导师,主要研究方向为 SoC 设计技术、IP 嵌入技术及混合信号 IC 设计. 吉利久,男,教授,博士生导师,主要研究方向为 VLSI 设计. 王阳元,教授,博士生导师,中国科学院院士,主要研究方向为微电子学.

量规则性差,测试时存储在 ROM 中,耗费大量的存储费用.因此对于一些规则电路,研究特定的测试向量生成方法具有十分重要的意义.

在处理器设计中,高速的加法运算多采用先行进位加法器^[2].早期的先行进位加法器由于信号的扇入、扇出数目的限制,不能实现多位数的相加.为了改善对信号的扇入、扇出数目的限制,Brent 和 Kung 提出了一种基于树型先行进位结构的加法器^[3].在此基础上,为了进一步提高运算速度,文献^[4,5]中介绍了一种具有紧密结构的树型加法器.它的运算时间为 $O(\log_2 n)$,设计结构优化,适合于 VLSI 实现,具有 IP 实现价值.本论文主要是针对这种树型加法器研究测试向量生成.

文献^[6]中介绍了一种基于单固定型故障的测试向量生成方法,但这种方法不能实现加法器中单固定型故障的 100% 检测.文献^[7]提出了一种从固定型故障引申出的新型对故障模型(pair-fault model)及基于该故障模型的测试向量生成方法,但这种方法也不能实现树型加法器的完全检测^[8~10].鉴于树型加法器的结构十分规则,文献^[11]介绍了一种故障模型——单元故障模型(cell fault model).该故障模型能够有效地反映规则电路中的物理缺陷.本论文就是基于单元故障模型来研究树型加法器的测试向量生成.

本论文首先分析了树型加法器的原理,总结了其运算特性;其次在介绍单元故障模型的基础上讨论了树型加法器的测试向量生成.理论分析的结果表明, $5n-1$ 个测试向量就可以实现树型加法器中所有单元的单元故障的检测,无需增加任何硬件开销.这些测试向量具有很好的规则性,能够利用片上测试向量生成器实现,适合于应用内建自测试技术实现.基于此,本论文提出了一种内建自测试的测试结构,测试时只需存储 7 个籽测试向量,其它测试向量可以在这 7 个籽测试向量的基础上通过循环移位实现.

2 树型加法器的原理分析

$A = a_{n-1}a_{n-2}\cdots a_0$ 和 $B = b_{n-1}b_{n-2}\cdots b_0$ 分别表示 n 位的被加数和加数, $S = s_{n-1}s_{n-2}\cdots s_0$ 表示相加和, $c_j (0 \leq j \leq n-1)$ 表示进位.基于上述符号定义,和 S 与各位进位的计算见式(1).

$$\begin{cases} s_j = a_j \oplus b_j \oplus c_{j-1}, \\ c_j = a_j \cdot b_j + c_{j-1} \cdot (a_j \oplus b_j), \quad 1 \leq j \leq n-1 \\ s_0 = a_0 \oplus b_0, \quad c_0 = a_0 \cdot b_0 \end{cases} \quad (1)$$

本论文是以无 c_{-1} 进位加的树型加法器为研究对象.研究结果对于具有相同树型结构的有 c_{-1} 进位加的树型加法器、减法器等同样适用.

提高加法器运算速度的关键是如何快速生成进位.定义 $g_j = a_j \cdot b_j$ 和 $p_j = a_j \oplus b_j (0 \leq j \leq n-1)$, g_j 为进位生成因子, p_j 为进位传播因子.式(1)变为式(2).

$$\begin{cases} s_j = p_j \oplus c_{j-1}, \\ c_j = g_j + c_{j-1} \cdot p_j, \quad 1 \leq j \leq n-1 \\ s_0 = p_0, \quad c_0 = g_0 \end{cases} \quad (2)$$

树型加法器的进位生成基于一种针对二元组的运算“ O ”^[3~5].具体的 O 运算见式(3).

$$(g_{in}, p_{in})O(g'_{in}, p'_{in}) = (g_{in} + p_{in}g'_{in}, p_{in}p'_{in}) \quad (3)$$

定义 1. $(G_{j,i}, P_{j,i}) = (g_j, p_j)O(g_{j-1}, p_{j-1})O \cdots O(g_i, p_i), 0 \leq i < j \leq n-1$.

定义 2. $(G_j, P_j) = (G_{j,0}, P_{j,0}) = (g_j, p_j)O(g_{j-1}, p_{j-1})O \cdots O(g_0, p_0), 0 \leq j \leq n-1$.

基于定义 2,进位的计算变换为式(4).

$$c_j = G_j, \quad 0 \leq j \leq n-1 \quad (4)$$

基于 O 运算,树型加法器可以分成三个部分: g, p 生成部分、树型先行进位链及最后和生成部分,具体见图 1.

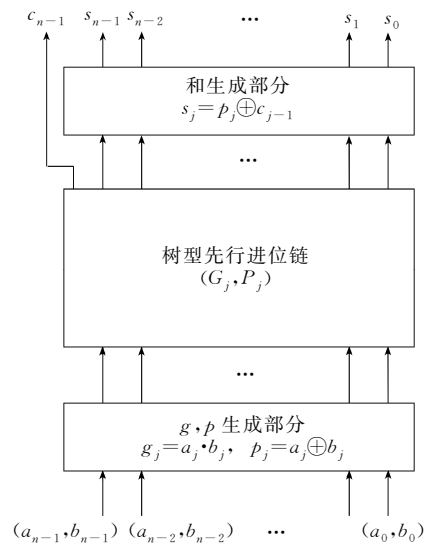


图 1 树型加法器的结构图

图 2 显示的是一个 8 位加法器的树型先行进位结构.图中有两种基本单元:黑单元和白单元,它们的

基本信号连接见图 3,功能分别见式(5)和(6). 经过分析可知树型先行进位链的信号延迟为 $O(\log_2 n)$, 共有 $n \lceil \log_2 n \rceil - \sum_{i=0}^{\lceil \log_2 n \rceil - 1} 2^i$ 个黑单元和 $n + \sum_{i=0}^{\lceil \log_2 n \rceil - 1} 2^i$ 个白单元. 白单元的功能相当于两个缓冲器, 因此对树型进位链的延迟起决定作用的是黑单元. 为了便于说明, 我们定义了树型进位链的层次及列, 用 $H_{j,i}$ 表示第 j 列、第 i 层的黑单元.

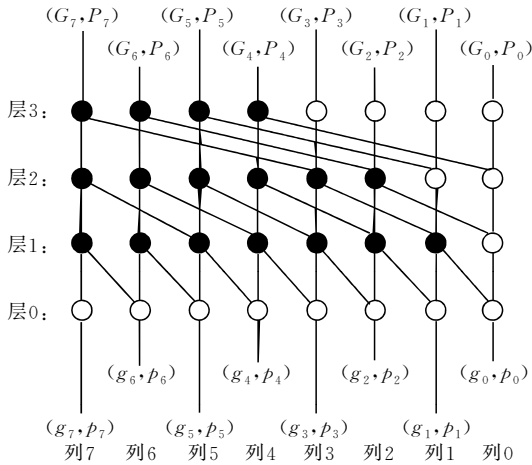


图 2 8位加法器的树型进位结构

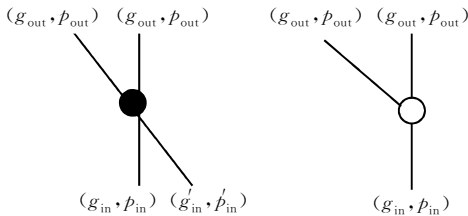


图 3 黑单元和白单元的信号连接示意图

黑单元: $(g_{out}, p_{out}) = (g_{in} + p_{in}g'_{in}, p_{in}p'_{in})$ (5)

白单元: $(g_{out}, p_{out}) = (g_{in}, p_{in})$ (6)

基于上述的分析和定义, 本文将 O 运算的特性分析总结如下.

特性 1. $g_j \cdot p_j = 0, G_{j,i} \cdot P_{j,i} = 0, G_j \cdot P_j = 0, 0 \leq i < j \leq n-1$.

根据特性 1 我们得知 g_j 和 p_j 不可能同时为 1, (g_j, p_j) 的取值集合为 $\varphi = \{(0,0), (0,1), (1,0)\}$. 对于 $(G_{j,i}, P_{j,i})$ 和 (G_j, P_j) , 我们得到同样的结论.

这里需要说明的是, $p_j = a_j + b_j$ 也可以实现同样的树型进位功能. 此时特性 1 则不成立. 本论文是以 $p_j = a_j \oplus b_j$ 为研究对象, 研究方法对 $p_j = a_j + b_j$ 情况下也同样适用.

特性 2. O 运算具有封闭性, 即 $X \in \varphi, Y \in \varphi, XOY = Z \in \varphi$.

为了方便说明 O 运算, 我们定义 $A = (0,0), B = (0,1), C = (1,0)$, 不同值之间的 O 运算的运算结果见表 1.

表 1 O 运算

	A	B	C
A	A	A	A
B	A	B	C
C	C	C	C

特性 3. $B = (0,1)$ 是 O 运算的交换因子.

$BO(g_j, p_j) = (g_j, p_j)OB = (g_j, p_j)$ (7)

$BO(G_{j,i}, P_{j,i}) = (G_{j,i}, P_{j,i})OB = (G_{j,i}, P_{j,i})$ (8)

特性 4. $A = (0,0)$ 和 $C = (1,0)$ 是 O 运算的支配因子.

$AO(g_j, p_j) = A, AO(G_{j,i}, P_{j,i}) = A$ (9)

$CO(g_j, p_j) = C, CO(G_{j,i}, P_{j,i}) = C$ (10)

在特性 4 基础上, 我们可以进一步得知如果 $(g_j, p_j) = A$ 或 C , 则 $(G_{j,i}, P_{j,i})$ 和 (G_j, P_j) 的值一定相应地为 A 或 C .

特性 5. O 运算具有结合性.

$(G_{j,i}, P_{j,i}) = (G_{j,m}, P_{j,m})O(G_{m-1,i}, P_{m-1,i})$ (11)

$(G_j, P_j) = (G_{j,i+1}, P_{j,i+1})O(G_i, P_i)$ (12)

为了描述黑单元的输入与输出之间的关系, 依据特性 5, 本论文定义了一个模型, 见图 4.

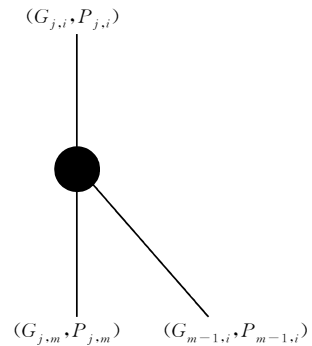


图 4 黑单元模型

定义 3. $(G_{j,m}, P_{j,m})$ 和 $(G_{m-1,i}, P_{m-1,i})$ 是 $(G_{j,i}, P_{j,i})$ 的生成因子; 反之, $(G_{j,i}, P_{j,i})$ 是 $(G_{j,m}, P_{j,m})$ 和 $(G_{m-1,i}, P_{m-1,i})$ 的集成因子; 相对于 $(G_{j,i}, P_{j,i}), (G_{j,m}, P_{j,m})$ 是 $(G_{m-1,i}, P_{m-1,i})$ 的左等价因子, $(G_{m-1,i}, P_{m-1,i})$ 是 $(G_{j,m}, P_{j,m})$ 的右等价因子.

定义 4. (G_j, P_j) 是树型进位电路的根单元, (g_j, p_j) 是树型进位电路的叶单元.

特性 6. 同一集成因子的两个生成因子互不

相关.

特性 6 的含义是指同一集成因子的两个生成因子的赋值互不相关. 这是因为同一集成因子的两个生成因子所包含的叶单元集合相交为空. 例如, $\{(g_j, p_j) \cdots (g_m, p_m)\} \cap \{(g_{m-1}, p_{m-1}) \cdots (g_i, p_i)\} = \emptyset$.

3 单元故障模型

为了实现不同的设计目标(如面积、速度等),每一种规则电路都有多种实现算法. 不同的算法对应实现不同的电路结构,如加法器有行波进位、选择进位及先行进位等多种实现算法. 设计规则电路通过基本单元的选取和连接关系的确定以实现不同的设计目标,其中每一个基本单元都采用行为级或 RTL 级描述. 规则电路测试分析的实质就是对其基本单元和单元间连接关系的测试分析.

由于树型加法器的结构规则,用单固定型故障模型已经不能有效地描述规则电路中的物理缺陷,我们需要一种更为复杂的故障模型. 为了使测试向量具有通用性,要求这种故障模型必须与具体工艺库或门级电路无关. 文献[11]介绍了一种单元故障模型. 该故障模型能够满足上述要求. 由于单元故障模型独立于具体电路实现,因此该故障模型具有通用性,可以应用在加法器、乘法器等规则电路的测试向量生成中.

定义 5. 满足下面两个条件的故障模型为单元故障模型.

- (1) 规则电路中一次最多只能有一个基本单元发生永久性故障;
- (2) 故障只影响基本单元的组合行为,即故障只改变了故障单元的真值表,并不能使之具有时序行为.

其实单元故障模型只是一个分析模型,其包含电路中所有可能影响电路功能的固定型故障. 由于单元故障模型与基本单元的内部电路实现无关,所以要完成基本单元级的故障检测,要求必须实现电路中任一基本单元的所有可能的输入组合,即穷举测试.

如果要求一个测试向量集能够实现单元故障的检测,必须要求实现两个条件:

- (1) 故障激励. 通过原始输入端能够实现电路中每个基本单元的所有可能的输入组合.
- (2) 故障信号传播. 在保证每个基本单元的所有

可能的输入组合的同时,必须要保证每个单元的任何可能的故障信号至少可以传播到电路的一个原始输出端.

其中第(1)条保证了单元故障的激发,第(2)条保证了故障信号的观察.

对于单元故障我们也可以像单固定型故障一样来定义其故障覆盖率. 文献[11]已经给出了单元故障覆盖率的计算公式,但是在该文献中单元故障覆盖率的定义仅考虑了故障激励,没有考虑故障信号传播对故障覆盖率的影响. 因此本论文在充分考虑故障激励和故障信号传播的情况下将文献[11]给出的单元故障覆盖率的计算公式做如下改进.

假设一个规则电路中包含有 N 个相同的基本单元,则该电路的单元故障覆盖率定义为 CFC (Cell Fault Coverage),见式(13),其中单元故障的检测必须满足上述两个条件.

$$CFC = \frac{\text{已检测的基本单元数 } N_t}{\text{所有基本单元数 } N} \times 100\% \quad (13)$$

但如果一个规则电路不只由一种基本单元构成,而是由 K 种基本单元有规律地构成,则每种基本单元的检测难易程度与该基本单元的电路规模成正比. 各种基本单元占电路的比例 P_j ($1 \leq j \leq K$) 见式(14),其中 N_j 为第 j 种基本单元的个数. 整个规则电路的单元故障覆盖率为 CFC_T ,见式(15),其中 CFC_j 为第 j 种基本单元的单元故障覆盖率.

$$P_j = \frac{N_j 2^{m_j}}{\sum_{j=1}^K N_j 2^{m_j}}, \quad 1 \leq j \leq K \quad (14)$$

$$CFC_T = \sum_{j=1}^K P_j \cdot CFC_j, \quad 1 \leq j \leq K \quad (15)$$

定理 1. 能够实现规则电路中所有单元故障检测的测试向量集一定能够实现单固定型故障的 100% 故障覆盖率.

这个定理可以从下面两点很容易证明.

(1) 可测的单固定型故障集合是单元故障集合的一个子集.

(2) 属于任一基本单元的单固定型故障一定能够被该单元的一种输入组合激励,同时测试向量集又能保证该故障信号在至少一个原始输出端可以被观察.

4 树型加法器的测试向量生成

树型加法器主要有三种部分: g, p 生成部分、树

型先行进位链和生成部分.

n 位加法器的 g, p 生成部分包含 n 个 2 输入、2 输出的基本单元, 测试时只需 4 个测试向量就可以实现其中每一个基本单元的所有 4 种输入组合. 以

4 位加法器为例, 4 个测试向量分别为 $\overbrace{0000}^A \overbrace{0000}^B$, $\overbrace{0000}^A \overbrace{1111}^B$, $\overbrace{1111}^A \overbrace{0000}^B$, $\overbrace{1111}^A \overbrace{1111}^B$. 表 2 给出每个基本单元的输入与输出的对应关系.

表 2 g, p 生成部分中基本单元的输入与输出的对应关系

(a_i, b_i)	(g_i, p_i)	(a_i, b_i)	(g_i, p_i)
(0, 0)	A	(1, 0)	B
(0, 1)	B	(1, 1)	C

n 位加法器的和生成部分同样包含 n 个相同的互相独立的基本单元, 很容易讨论其测试向量生成. 而树型先行进位链采用的是树状结构, 测试向量生成比较复杂, 因此树型加法器的测试关键在于树型先行进位链的测试向量生成.

树型先行进位链包括黑单元和白单元. 测试向量生成主要考察黑单元的测试情况, 原因如下:

(1) 白单元的功能相当于两个缓冲器, 可以被视为黑单元的输入或输出. 如果黑单元被检测了, 则白单元一定也同时被检测了.

(2) 电路中存在许多扇出线, 但是这些扇出线没有重汇聚现象, 因此所有的黑单元都被测试了, 则整个电路就被检测了.

黑单元包含有 4 个输入端和 2 个输出端, 根据 O 运算的特性 1, 任一黑单元最多只能实现 9 种输入组合. 但是当电路中存在故障时黑单元的输出可能是 (1, 1), 因此为了考察故障信号的传播情况, 我们也必须考察输入为 (1, 1) 时 O 运算的情况. 在表 1 基础上我们可以得到一个扩展表, 见表 3, 其中 $D = (1, 1)$. 我们已知 $\varphi = \{A, B, C\}$, 在 O 运算扩展表的基础上新的输入集合为 $\varphi' = \{A, B, C, D\}$.

表 3 O 运算的扩展表

	A	B	C	D^*
A	A	A	A	A^*
B	A	B	C	D^*
C	C	C	C	C^*
D^*	C^*	D^*	C^*	D^*

引理 1. 如果一个测试向量集能够实现每一黑单元的所有可能的输入组合, 且能够同时保证故障信号至少传播到一个原始输出端, 则该测试向量

集能够实现树型进位链的 100% 单元故障覆盖率.

该引理是显然的. 基于 O 运算的特性 6, 对于任一黑单元而言, 能够很容易地实现 9 种输入组合. 所以测试的重点是考察故障信号传播情况. 本论文定义 $X \in \varphi, \psi'(X) \in \{\varphi' - X\}$.

引理 2. 故障信号的传播条件为 $XOY = Z(X, Y, Z \in \varphi), \psi'(X)OY = \psi'(Z)$ 或 $XO\psi'(Y) = \psi'(Z)$ 同时成立.

定义 6. $\tau(X) = \{Y | XOY = Z, \psi'(X)OY = \psi'(Z), X \in \varphi, Y \in \varphi, Z \in \varphi\}$ 为输出信号为 X 的黑单元发生故障时故障信号传播的右传播条件; 同时 $\lambda(X) = \{Y | YOX = Z, YO\psi'(X) = \psi'(Z), X \in \varphi, Y \in \varphi, Z \in \varphi\}$ 为输出信号为 X 的黑单元发生故障时故障信号传播的左传播条件.

下面以 $X = A$ 为例, 说明如何求得 $\tau(A)$ 和 $\lambda(A)$.

表 4 输出信号为 A 的黑单元正常时和发生故障时的信号传播情况

$XOY = Z$	$\psi'(X)OY = \psi'(Z)$
$AOY = A$	$BOA = A, BOB = B, BOC = C$ $COA = C, COB = C, COC = C$ $DOA = C, DOB = D, DOC = C$
$YOX = Z$	$YO\psi'(X) = \psi'(Z)$
$AOA = A, BOA = A, COA = C$	$AOB = A, BOB = B, COB = C$ $AOC = A, BOC = C, COC = C$ $AOD = A, BOD = D, COD = C$

由表 4 可以推出 $\tau(A) = \{B, C\}$ 和 $\lambda(A) = \{B\}$. 同理可得 $\tau(B) = \lambda(B) = \{B\}$ 及 $\tau(C) = \lambda(C) = \{B\}$.

定理 2. $5n - 1$ 个测试向量能够实现树型先行进位电路中所有单元的单元故障检测.

证明. 树型先行进位电路的检测实际上就是电路中所有黑单元的检测. 黑单元的 9 种可能的输入组合为 $\{(A, A), (A, B), (A, C), (B, A), (B, B), (B, C), (C, A), (C, B), (C, C)\}$. 要证明实现树型先行进位电路中所有单元的单元故障检测必须证明 $5n - 1$ 个测试向量可以实现每一黑单元的 9 种输入组合, 同时还要保证每一黑单元发生故障时故障信号可以至少传播到一个原始输出端.

我们将 $5n - 1$ 个测试向量分为 6 组加以分析.

第 1 组: 1 个测试向量为 $\overbrace{(BB \cdots BB)}^n$. 树型进位电路中所有黑单元的输入均为 (B, B) , 输出为 B . 假设 $H_{j,i}$ 黑单元发生故障时, 如果其输出是原始输出端, 则故障信号可以直接观察; 如果其输出不是原始输出端, $\tau(B) = B$, 且其等价因子的值一定为 B , 因此故障信号可以传播到同列、下一层的黑单元的输

出,以此类推,故障信号一定可以从原始输出端(G_j , P_j)处观察.

第 2 组: n 个测试向量为 $\{(A \overbrace{BB \cdots BB}^{n-1}) (A \overbrace{BB \cdots BB}^{n-2}) \cdots (B \overbrace{BB \cdots BB}^{n-3}) (B \overbrace{BB \cdots BB}^{n-2}) (B \overbrace{BB \cdots BB}^{n-1})\}$.

测试向量 $(A \overbrace{BB \cdots BB}^{n-1})$ 可以实现第 $n-1$ 列黑单元的 (A, B) 输入组合. 第 $n-1$ 列上的每个黑单元的输出的右等价因子均为 B , $\tau(A) = \{B, C\}$, 则其中任一黑单元的故障信号均可以传播到原始输出端 (G_{n-1}, P_{n-1}). 测试向量 $(A \overbrace{BB \cdots BB}^{n-2})$ 进一步可以实现第 $n-2$ 列黑单元的 (A, B) 输入组合. 第 $n-2$ 列上的每个黑单元的输出的右等价因子均为 B , 则其中任一黑单元的故障信号均可以传播到原始输出端 (G_{n-2}, P_{n-2}). 对其他测试向量可以以此类推, 直至测试向量 $(B \overbrace{BB \cdots BB}^{n-3})$ 实现第 1 列的黑单元的 (A, B) 输入组合及故障信号观察.

测试向量 $(A \overbrace{BB \cdots BB}^{n-2})$ 可以实现 $H_{n-1,1}$ 黑单元的 (A, A) 输入组合, 且该单元的故障信号可以传播到原始输出端 (G_{n-1}, P_{n-1}). 下一个测试向量 $(B \overbrace{BB \cdots BB}^{n-3})$ 进一步可以实现 $H_{n-2,1}$ 和 $H_{n-1,2}$ 两个黑单元的 (A, A) 输入组合, 且能够保证这些单元的故障信号可以分别传播到原始输出端 (G_{n-2}, P_{n-2}) 和 (G_{n-1}, P_{n-1}). 对其他测试向量可以以此类推, 直至测试向量 $(B \overbrace{BB \cdots BB}^{n-2})$ 实现 $H_{1,1}, H_{2,2}, H_{4,3}$ 等每层最右边的黑单元的 (A, A) 输入组合且故障信号传播.

总之, 上述 n 个测试向量可以同时实现所有黑单元的 (A, A) 和 (A, B) 两种输入组合.

第 3 组: n 个测试向量为 $\{(A \overbrace{BB \cdots BB}^{n-2}) (B \overbrace{BB \cdots BB}^{n-3}) \cdots (B \overbrace{BB \cdots BB}^{n-2}) (B \overbrace{BB \cdots BB}^{n-1})\}$. 这 n 个测试向量可以实现每一个黑单元的 (B, A) 和 (A, C) 两种输入组合及故障信号观察. 推导方法与第二组的相同.

第 4 组: $n-1$ 个测试向量为 $\{(C \overbrace{BB \cdots BB}^{n-2}) (B \overbrace{BB \cdots BB}^{n-3}) \cdots (B \overbrace{BB \cdots BB}^{n-2}) (B \overbrace{BB \cdots BB}^{n-1})\}$. 这 $n-1$ 个测试向量可以实现每一个黑单元的 (C, A) 输入组合.

第 5 组: n 个测试向量为 $\{(C \overbrace{BB \cdots BB}^{n-1}) (B \overbrace{BB \cdots BB}^{n-2}) \cdots (B \overbrace{BB \cdots BB}^{n-1})\}$.

$\{(C \overbrace{BB \cdots BB}^{n-2}) \cdots (B \overbrace{BB \cdots BB}^{n-1}) (B \overbrace{BB \cdots BB}^{n-1})\}$. 这 n 个测试向量可以实现每一个黑单元的 (B, C) 和 (C, B) 两种输入组合.

第 6 组: $n-1$ 个测试向量为 $\{(C \overbrace{BB \cdots BB}^{n-2}) \cdots (B \overbrace{BB \cdots BB}^{n-3}) (B \overbrace{BB \cdots BB}^{n-2}) (B \overbrace{BB \cdots BB}^{n-1})\}$. 这 n 个测试向量可以实现每一个黑单元的 (C, C) 输入组合.

综上所述, $5n-1$ 个测试向量就可以实现每一黑单元的 $\{(A, A), (A, B), (A, C), (B, A), (B, B), (B, C), (C, A), (C, B), (C, C)\}$ 9 种输入组合, 同时还要保证每一黑单元的故障信号可以传播到至少一个原始输出端.

证毕.

上述定理中 $5n-1$ 个测试向量可以树型进位链中所有黑单元的所有输入组合. 和生成部分包含 $n-1$ 并行的异或门. $5n-1$ 个测试向量中下述 $2n$ 个就可以实现每个异或门的 4 种输入组合. 依据和生成的输出 $s_{n-1} s_{n-2} \cdots s_2 s_1$ 定义异或门 $EO_{n-1}, EO_{n-2}, \cdots, EO_2, EO_1$. 4 种输入组合实现如下.

(1) $(0, 0)$ 输入组合

测试向量 $(A: \overbrace{00 \cdots 00}^n, B: \overbrace{00 \ 11 \cdots 11}^{n-2})$ 可以实现 EO_{n-1} 和 EO_{n-2} 的 $(0, 0)$ 输入组合; 测试向量 $(A: \overbrace{00 \cdots 00}^n, B: \overbrace{100 \ 11 \cdots 11}^{n-3})$ 可以实现 EO_{n-3} 的 $(0, 0)$ 输入组合; 以此类推, 测试向量 $(A: \overbrace{00 \cdots 00}^n, B: \overbrace{11 \cdots 11}^{n-2} 00)$ 可以实现 EO_1 的 $(0, 0)$ 输入组合.

(2) $(0, 1)$ 输入组合

测试向量 $(A: \overbrace{01 \ 00 \cdots 00}^{n-2}, B: \overbrace{0 \ 11 \cdots 11}^{n-1})$ 可以实现 EO_{n-1} 的 $(0, 1)$ 输入组合; 测试向量 $(A: \overbrace{001 \ 00 \cdots 00}^{n-3}, B: \overbrace{10 \ 11 \cdots 11}^{n-2})$ 可以实现 EO_{n-2} 的 $(0, 1)$ 输入组合; 以此类推, 测试向量 $(A: \overbrace{00 \cdots 001}^{n-1}, B: \overbrace{11 \cdots 1101}^{n-2})$ 可以实现 EO_1 的 $(0, 1)$ 输入组合.

(3) $(1, 0)$ 输入组合

测试向量 $(A: \overbrace{00 \cdots 00}^n, B: \overbrace{11 \cdots 11}^n)$ 可以实现所有 $n-1$ 个异或门的 $(1, 0)$ 输入组合.

(4) $(1, 1)$ 输入组合

测试向量 $(A: \overbrace{00 \cdots 001}^{n-1}, B: \overbrace{11 \cdots 11}^n)$ 可以实现所有 $n-1$ 个异或门的 $(1, 1)$ 输入组合.

结论 1. 对于 n 位树型加法器而言, $5n-1$ 个测试向量可以实现所有单元的单元故障检测.

5 树型加法器的测试

通过观察,表中 $5n-1$ 个测试向量具有很好的规律性,其中大部分测试向量可以在少数几个测试向量基础上通过循环移位完成.例如,测试向量 $(A:00\dots00$

$888, B:100 \overbrace{11\dots11}^{n-3})$ 就是在测试向量 $(A: \overbrace{00\dots00}^n, B: \overbrace{00 \ 11\dots11}^{n-2})$ 循环向右移一位的结果.经过观察,我们可以发现输入的测试向量中,有 7 个籽测试向量,其他测试向量均可以在这 7 个籽测试向量基础上循环移位生成.被加数和加数除了前两位外,其他位分别固定为 $A_0^{n-3}: \overbrace{00\dots00}^{n-2}$ 和 $B_0^{n-3}: \overbrace{11\dots11}^{n-2}$. 7 个籽测试向量分别见表 5.

表 5 7 个籽测试向量及其对应的树型先行进位链的输入

籽测试向量	对应的树型先行进位链的输入
$A:00 \overbrace{00\dots00}^{n-2} B:00 \overbrace{11\dots11}^{n-2}$	$AA \overbrace{BB\dots BB}^{n-2}$
$A:00 \overbrace{00\dots00}^{n-2} B:01 \overbrace{11\dots11}^{n-2}$	$AB \overbrace{BB\dots BB}^{n-2}$
$A:01 \overbrace{00\dots00}^{n-2} B:01 \overbrace{11\dots11}^{n-2}$	$AC \overbrace{BB\dots BB}^{n-2}$
$A:00 \overbrace{00\dots00}^{n-2} B:11 \overbrace{11\dots11}^{n-2}$	$BB \overbrace{BB\dots BB}^{n-2}$
$A:10 \overbrace{00\dots00}^{n-2} B:10 \overbrace{11\dots11}^{n-2}$	$CA \overbrace{BB\dots BB}^{n-2}$
$A:10 \overbrace{00\dots00}^{n-2} B:11 \overbrace{11\dots11}^{n-2}$	$CB \overbrace{BB\dots BB}^{n-2}$
$A:11 \overbrace{00\dots00}^{n-2} B:11 \overbrace{11\dots11}^{n-2}$	$CC \overbrace{BB\dots BB}^{n-2}$

一般的加法器结构需要 3 个寄存器分别存储被加数、加数及相加的结果.在此结构的基础上,本文提出了一种内建自测试结构,见图 5.其中加法器的输出连接到一个 $n+1$ 位的多输入特征寄存器 MISR.事实上,MISR 是一个具有极低故障混淆率 ($\approx \frac{1}{2^k}$) 的测试响应压缩器.输入寄存器由两个循环移位寄存器构成.籽测试向量生成有两种方式:一种

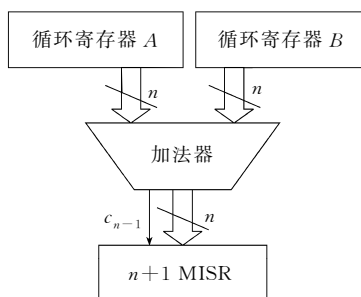


图 5 树型加法器的测试结构

方式是可以利用 ROM 存储 7 个籽测试向量,事实上由于除了最高两位外其他位均固定为 0 或 1,所以只需存储前两位即可;另一种方式利用一个 4 位的计数器生成被加数和加数的前两位的值.测试时,只需 7 次赋值及 $5n-4$ 次循环移位就可以实现树型加法器的完全测试.

本论文以 8 位、16 位、32 位和 64 位加法器为例,说明测试向量生成效果及对单固定型的检测,见表 7.表中第 2 行显示的是应用 Alcatel 公司 $0.35\mu\text{m}$ CMOS 工艺实现的加法器的运算速度.第 3 行显示的是测试向量数,第 4 行显示的是原电路的等效门数,第 5 行显示的是测试电路增加的电路比(测试电路的等效门数/原电路的等效门数).第 6 行和第 7 行显示的是分别针对 Alcatel $0.35\mu\text{m}$ 工艺和 CLASS 工艺应用 SYNOPSIS 中的故障模拟器实现的故障模拟结果 FC_1 和 FC_2 .结果显示, $5n-1$ 个测试向量可以实现 100% 单固定型故障的故障覆盖率,且与综合目标工艺无关.

表 7 应用 SYNOPSIS 的实验结果

加法器的规模	运算速度 (ns)	测试向量数	原电路规模	测试电路规模	FC_1	FC_2
8 位	2.81	39	234	32.9	100	100
16 位	3.26	79	515	28.7	100	100
32 位	3.70	159	1123	26.0	100	100
64 位	4.22	319	2461	23.6	100	100

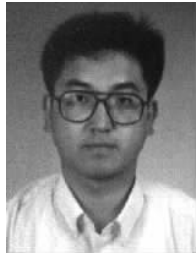
6 结 论

树型加法器采用树型先行进位结构、电路规则,适合于 VLSI 实现.本文首先分析了树型加法器的原理,总结 O 运算的运算特性.此外还分析了单元故障模型,给出了单元故障的故障覆盖率.在此基础上本文主要研究了树型加法器的测试向量生成及内建自测试结构.结论是 $5n-1$ 个测试向量可以实现 100% 单元故障覆盖率.实现结果还表明, $5n-1$ 个测试向量可以实现单固定型故障的 100% 的故障覆盖率.本文基于单元故障模型研究树型加法器的测试向量生成方法对研究其他规则电路的测试向量也适用.

参 考 文 献

1 Blanton R D, Hayes J P. On the design of fast, easily testable ALU's. IEEE Transactions on Very Large Scale Integration

- (VLSI) Systems, 2000, 8(2):220~223
- 2 Han Kyung-Nam, Han Sang-Wook, Yoon Euisik . A new adder scheme with reduced P , G signal generations using redundant binary number system. In: Proceedings of the 2000 IEEE International Symposium on Circuits and Systems, Geneva Switzerland, 2000. 633~636
 - 3 Brent R P, Kung H T. A regular layout for parallel adders. IEEE Transactions on Computers, 1982, C-31(3):260~264
 - 4 Hsu J, Bair O. A compiler for optimal adder design. In: Proceedings of the IEEE 1992 Custom Integration Circuits Conference, Boston Massachusetts, 1992. 25.6.1~25.6.4
 - 5 Kazumasa Suzuki, Masakazu Yamashina, Takashi Nakayama, *et al.* A 500MHZ, 32bit, 0.4 μ m CMOS RISC Processor. IEEE Journal of Solid State Circuits, 1994, 29(12):1464~1473
 - 6 Becker B. Efficient testing of optimal time adders. IEEE Transactions on Computer, 1988, 37(9):1113~1120
 - 7 Arjhan C, Deshmukh R G. A novel fault-model for regular and irregular parallel-prefix adders. In: Proceedings of Southeastcon'98, Orlando Florida, 1998. 397~400
 - 8 Wagh M, Chen C-I H. The high-level design synthesis with redundancy for high speed testable adders. In: Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, Orlando Florida, 1999. 358~361
 - 9 Wagh M, Chen C-I H. Testability Synthesis for Jumping Carry Adder. In: Proceedings of the 12th Annual IEEE International ASIC/SOC Conference, Washington DC, 1999. 130~134
 - 10 Blanton R D, Hayes J P. Testability of convergent tree circuits. IEEE Transactions on Computer, 1996, 45(8):950~963
 - 11 Gizopoulos D, Paschalis A, Zorian Y. An effective built-in self-test scheme for parallel multipliers. IEEE Transactions on Computers, 1999, 48(9): 936~950



LI Zhao-Lin, born in 1973, Ph. D., lecturer. His research interests include high performance computer architecture, design and test of SoC.

SHENG Shi-Min, born in 1946, professor, Ph. D. supervisor. His current research interests include SoC design

methodology, IP embed technology and Mixed signal IC design.

JI Li-Jiu, professor, Ph. D. supervisor. His current research interests include VLSI design.

WANG Yang-Yuan, professor, Ph. D. supervisor, member of Chinese Academy of Sciences, IEEE Fellow, IEE Fellow, His research focuses on novel device, novel technique & novel structure circuits in microelectronics.