

基于故障划分的并行测试生成算法^{*}

曾芷德 曾献君

(国防科学技术大学计算机系 长沙 410073)

摘要 文章从理论上分析了提高基于故障划分的并行测试生成算法的加速比的途径。在此基础上,提出了把相关故障识别和最短路径敏化相结合的基于输出扇入锥的逆向故障划分方法 BFPOC(backword fault partitioning of output fan-in cones),并把该方法与 Banerjee 推荐的基于输入扇出锥的正向故障划分方法 TFPIC(toward fault partitioning of input fan-out cones)和常用的基于故障顺序的等步长划分方法 EDPFS(equal distance partitioning of fault sequences)进行了实验比较。结果表明,在大规模并行处理环境下,BFPOC 方法比后两种方法有明显的优势,能获得更高的加速比。

关键词 并行测试生成,故障并行,故障划分,输出扇入锥,输入扇出锥,加速比。

中图法分类号 TP338

借助并行处理的优势来对付 VLSI 测试生成时间的指数复杂性是一种可供选择的方法^[1]。大规模并行处理机(MPP)的问世和 workstation 机群(WSG)的广泛配置,为并行测试生成的实现提供了良好的硬件环境。但是,要把测试生成这类原本属于串行处理的算法改造成可并行运行的算法并非易事。

近 10 年来,国内外学者除了继续致力于提高单机测试生成系统效率的研究^[2-3]以外,还进行了多机并行测试生成算法的研究,并先后提出了基于故障并行、启发式并行、搜索空间并行、功能并行和电路并行等多种并行测试生成方法,也报道过一些关于并行测试生成的原型系统。例如,按初级输入扇出锥对电路故障进行集束分组的并行系统^[4]、按启发式并行的系统^[5]、对难测故障按搜索空间并行的系统^[6]以及对时序电路分别按故障并行^[7]、按搜索空间并行^[8]和按电路并行^[9]的系统等等。这些系统参与并行处理的处理机数目为 5~16,属于小规模并行,它们的共同特点是比较重视处理机之间的动态负载均衡。众所周知,实现动态负载均衡是以增加处理机之间的通信开销为代价的。理论上不难证明,实现 N 个处理机的动态负载均衡所需通信开销的时间复杂性为 $O(N^2)$ 。这表明对 MPP 系统($N \geq 128$),靠实现动态负载均衡来提高加速比可能会有困难。为此,我们建议把研究重点放在如何提高静态负载均衡上,以减少依托 MPP 机的并行测试生成系统的通信开销,提高加速比。我们曾提出过按逻辑相关性的大小,把被测电路(CUT)原有的以初级输出(PO)为锥顶的 P 个输出扇入锥分配给 N 个处理机,进行并行处理的优化划分算法——最小递归二分法^[10],收到了较好的效果。但是,对那些反馈线很多、反馈深度很深的闭环电路,每个输出锥基本包含整个电路,此时再试图将电路按输出锥分块将是徒劳的。因此,本文研究依托 MPP 机的基于故障静态划分的并行测试生成算法。

1 关于故障并行的加速比

假设 $m \times n$ 阶矩阵 (a_{ij}) 为 CUT 的故障检测矩阵,其中 $j=1,2,\dots,n$ 与 CUT 的目标故障集 SF 中的 n 个故障对应, $i=1,2,\dots,m$ 与检测 SF 的完全测试集 ST 中的每个测试码相对应。

$$\begin{cases} a_{ij}=1, \text{表示测试码 } T_i \text{ 检测故障 } f_j; \\ a_{ij}=0, \text{表示测试码 } T_i \text{ 未检测故障 } f_j. \end{cases}$$

* 本文研究得到国家自然科学基金资助。作者曾芷德,1941年生,教授,主要研究领域为数字系统故障诊断与容错。
曾献君,1966年生,博士,副教授,主要研究领域为数字系统CAD。

本文通讯联系人:曾芷德,长沙 410073,国防科学技术大学计算机系

本文 1998-08-24 收到原稿,1998-12-01 收到修改稿

我们的任务是把 SF 中的 n 个故障平分给 N 个处理机进行并行测试生成, 希望获取最高的加速比. 此任务可形式化描述如下:

把矩阵 (a_{ij}) 按列均分成 N 个 $m \times n/N$ 阶的子阵 (b_{ij}^k) , 其中 $i=1, 2, \dots, m; l=1, 2, \dots, n/N$ (或 $(n/N)+1$); $K=1, 2, \dots, N$. 由于此时 (b_{ij}^k) 中包含了一些全零行, 按行压缩掉这些全零行之后, 有 $i=1, 2, \dots, m_k$, 如果假设 $\max\{m_k\}=M$, 则加速比

$$SP = m(t_g + t_f \cdot m_{fs}) / (M(t_g + t_f \cdot m_{fp})). \quad (1)$$

式(1)中, $M \leq m$, t_g 为生成一个测试码所需的平均时间, t_f 为一个测试码模拟一个故障所需的平均时间. 它们由 CUT 的复杂性及所用的测试生成(TG)和故障模拟(FS)算法决定, 与故障划分无关. m_{fs} 和 m_{fp} 分别为在串行和并行生成环境下, 一个测试码平均需要模拟的故障数. 为了便于分析, 我们假设 ST 中的 m 个测试码完全覆盖 SF 中的 n 个故障, 每个测试码都检测 n/m 个故障, 于是可以导出

$$\begin{aligned} m_{fs} &= n(m+1)/2m, \\ m_{fp} &= n(M+1)/(2N \cdot M), \\ SP &= N(m(2t_g + t_f \cdot n) + t_g \cdot n) / (M(2t_g \cdot N + t_f \cdot n) + t_g \cdot n). \end{aligned} \quad (2)$$

式(2)中, 只有 M 与故障划分方法有关. 显然, 如果在对 (a_{ij}) 按列划分时能使 M 取最小值, 则 SP 为最大值, 可获得最高的加速比.

如果在对 SF 划分时 (a_{ij}) 是已知的, 则获取最高加速比的故障划分可抽象化为纯数学问题: 从矩阵 (a_{ij}) 的 C_n^m 种列组合中选取使 M 最小的列组合. 此问题用模拟退火法或最小递归二分法都不难解决. 但是, 由于进行故障划分时尚未求出 ST , (a_{ij}) 是未知的, 上述方法暂时还用不上. 不过, 上述分析已给了我们一条极重要的启示: 具有最高加速比的划分, 必须是能使每个处理机所分得的故障集的完全测试集最小(即 M 最小)的划分. 为此, 在进行故障划分时必须充分考虑故障的等价性、相关性和独立性.

VLSI 中的故障, 按检测每个故障的完全测试集的相关性, 可以分为等价故障、相关故障和独立故障. 设 T_α , T_β 分别为固定故障 α, β 的完全测试集, 如果存在

$T_\alpha = T_\beta$, 则称 α, β 为等价故障;

$T_\alpha \neq T_\beta$, 但 $T_\alpha \cap T_\beta \neq \emptyset$, 则称 α, β 为相关故障;

$T_\alpha \cap T_\beta = \emptyset$, 则称 α, β 为独立故障.

显然, 按上述定义, 不可测故障属于独立故障. 考虑到测试生成系统(test generation system, 简称 TGS)在生成 CUT 的目标故障表时, 对等价故障已进行了识别和归并处理, 因此, 在划分故障时不再考虑等价故障.

为使并行测试生成算法获得较高的加速比, 划分故障时应遵循下列 3 条基本原则:

- (1) 相关故障优先分配给同一个处理机;
- (2) 独立故障优先分配给不同的处理机;
- (3) 各处理机处理的相关故障集的个数及故障总数基本相同.

2. 3 种故障划分方法

不言而喻, 相关故障和独立故障的识别是故障划分的关键. 本文提出 BFPOC(backward fault partitioning of output fan-in cones)方法, 并与文献[1]推荐的 TFPIC(toward fault partitioning of input fan-out cones)方法和较常用的 EDPFS(equal distance partitioning of fault sequences)方法进行比较, 分析它们的优劣, 现分述如下.

2.1 基于输出锥的逆向划分方法(BFPOC)

该方法的基本思想是把相关故障的识别与最短路径敏化^[11]的思想相结合. 具体描述如下.

(1) 功能块故障分组规则

假设功能块输出 $s-a-0$ 故障的组号为 k , $s-a-1$ 故障的组号为 $k+1$, 根据功能块的功能、故障相关性和最短路径敏化思想, 对其输入故障以及与输入相连的扇出源故障打分组标记:

(1) 或(或非)门

① 级别最低的输入和它们的先行扇出源的 $s-a-0$ 故障都标记 $k(k+1)$, $s-a-1$ 标记 $k+1(k)$;

② 设其他输入共有 i 个 ($i=1,2,\dots$), 则这些输入的 $s-a-0$ 和它们各自的先行扇出源的 $s-a-0$ 分别标记 $k+2(k+3)$, $k+4(k+5), \dots, s-a-1$ 则标记 $k+1(k)$.

(2) 与(与非)门

① 级别最低的输入及其先行扇出源的 $s-a-1$ 都标记 $k+1(k)$, $s-a-0$ 标记 $k(k+1)$;

② 设其他输入共有 i 个 ($i=1,2,\dots$), 则这些输入的 $s-a-1$ 和它们各自的先行扇出源的 $s-a-1$ 分别标记 $k+3(k+2)$, $k+5(k+4), \dots, s-a-0$ 则标记 $k(k+1)$.

(3) 驱动(非)门

此门输入的先行扇出源 $s-a-0$ 标记 $k(k+1)$, $s-a-1$ 标记 $k+1(k)$.

(4) 对称输入的复合门

① 对与或(与或非)门, 同一与门输入故障分组规则同与(与非)门, 不同与门的对称输入 $s-a-1$ 分在同一组.

② 对或与(或与非)门, 同一或门输入故障分组规则同或(或非)门, 不同或门的对称输入 $s-a-0$ 分在同一组.

(5) 其他功能块

① 对级别最低的输入 a , 检查输入/输出连通矩阵. 如果是同相连通, 则 a 及其先行扇出源线 $s-a-0$ 标记 k , $s-a-1$ 标记 $k+1$; 反相连通, 则 a 及其先行扇出源线 $s-a-0$ 标记 $k+1$, $s-a-1$ 标记 k ; 双相连通, 则随机地按同相或反相对待;

② 对其他输入 b , ($i=1,2,\dots$), 如果是同相连通, 则 b 及其先行扇出源的故障 $s-a-0(1)$ 的组号为 $k+i+1(k+i+2)$; 反相连通, 则为 $k+i+2(k+i-1)$; 双相连通, 则随机地按同相或反相对待.

(2) 电路故障分组过程

从最复杂的初级输出 (PO_i) 开始, 向初级输入 (PI) 逆向遍历该输出锥的每一个功能块. 按上述规则, 在每个功能块输出故障组号的基础上, 对它们的输入及其先行扇出源的故障分组. 逆向最短路径敏化思想已隐含在其中.

(3) 故障组号分配算法

(1) 同一个故障只分配一个组号, 不同故障的组号不允许重迭. 按此原则, 各输出锥之间公共的故障只分配一次组号.

(2) 为避免不同故障的组号重迭, 本文提出故障组号分配算法如下:

① 一条引线分配两个邻接数字作为两个故障的组号, 即使某条引线的一个或两个故障已被等价归并掉. 这里的“引线”包括 PI , 内部功能块的输出线和扇出个数大于 1 的扇出源的扇出分线.

② 按功能块故障分组原则判明属于同一组的故障, 组号向最小的看齐; 不属于同一组的故障, 保留原组号不变.

(4) 按上述规则, 编在同一组的故障构成一个相关故障集; 不同组的故障的相关性和独立性, 要想精确描述将比较复杂, 但总的趋势是独立性大于相关性.

(5) 故障向处理机分配原则: 同组故障应分配给同一个处理机, 不同组故障应分配给不同的处理机, 除非考虑处理机负载平衡需进行调整时例外. 当处理机数 \ll 故障组数时, 应使分配给不同处理机的故障组数基本相同.

(6) 故障划分举例

例 1: 图 1 的电路经等价归并后, 共设 30 个故障, 按划分规则将被分为 15 组, 结果如下.

- 组 0: x_1^0, a^0, x_2^0 组 1: x_1^1, x_8^0, x_2^1, b^0 组 2: x_9^0, x_1^1, x_6^1, f^0 组 3: x_{10}^0, x_i^1 组 4: x_{11}^0, x_3^1, j^0
- 组 5: x_3^0 组 6: c^0 组 7: g^0 组 8: d^0, x_1^0 组 9: e^0, x_6^0
- 组 10: h^0 组 11: i^0, x_4^0 组 12: k^0 组 13: x_7^0 组 14: l^0, x_5^0

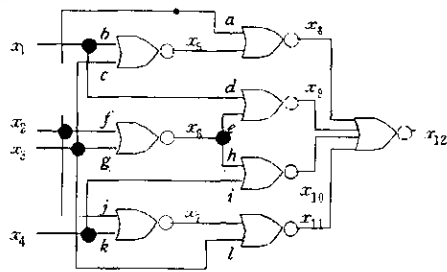


图 1

这 30 个故障,用 ATGTA 系统将生成 8 个测试码.其中有 4 个冗余故障被分在不同组.

2.2 基于输入锥的正向划分方法(TFPIC)

本方法的基本思想是:从一个 PI 出发,沿着其扇出后继线,向着有关的 PO 进行搜索,把沿途所经引线构成的以 PI 为顶的输入锥的故障,按照反向次数分成两组,每一组也构成一个相关故障集.各输入锥之间的公共故障也只分配一次组号.显然,

$$\text{CUT 故障总组号} = 2 * \text{PI 总数}.$$

当把故障向处理机分配时,仍然遵循方法 1 的原则.

2.3 基于 CUT 故障表中故障顺序的等步长划分方法(EDPFS)

CUT 故障表中故障顺序按功能块级别从低到高排列,每个功能块的故障按输出 0,1 故障和输入 0,1 故障排列,输出、输入严格按端号顺序排列.所谓等步长划分是指以处理机数目 N 为步长,把 CUT 的 n 个故障分成 N 份,每份含故障数 n/N 或 $(n/N)+1$,其中第 $i(i=1,2,\dots,N)$ 个处理机分得的故障序号为

$$j * N + i, \quad j=0,1,2,\dots,j * N + i \leq n.$$

当处理机数目达到 16 以上时,按故障顺序等步长划分,可以把绝大多数功能块的输入、输出故障分给不同的处理机,这与经等价归并之后,同一功能块中(不含异或门)留下的故障大部分是独立故障的规律相符.这大概是该方法也比较有效的重要原因.

3 实验结果及结论

3.1 实验 1

我们在单台 SUN ULTRA2 上建成了一个并行故障诊断模拟系统,一次最多可模拟 256 个处理机的并行测试生成.该系统用有限回溯测试模式产生方法生成测试码,用同时故障模拟进行故障覆盖.我们用此系统对 ISCAS-85 部分电路模拟了上述 3 种划分方法的效果,当处理机分别为 4、16、64、256 时,测试生成的加速比和故障覆盖率结果列在表 1 中.为使加速比的比较相对合理,我们要求单机串行、多机并行所得到的故障覆盖率应基本相同.方法 1 和方法 2 所用的预处理时间很少,可不予考虑.并行生成的故障覆盖率是把各处理机生成的测试码集中起来进行故障模拟得到的,采用的是测试码并行、单故障传播的快速故障模拟方法^[12].

表 1

电路名称	划分方法	4 处理机		16 处理机		64 处理机		256 处理机	
		加速比	故障覆盖率(%)	加速比	故障覆盖率(%)	加速比	故障覆盖率(%)	加速比	故障覆盖率(%)
C1355	1	2.83	98.83	7.36	99.32	18.67	99.32	53.62	99.51
	2	2.50	98.08	6.95	99.14	14.13	99.26	47.52	99.26
	3	2.61	98.27	6.06	98.83	16.34	99.44	46.47	99.26
C1908	1	2.97	99.29	7.12	99.29	17.32	99.15	36.95	99.34
	2	2.49	99.15	4.69	99.24	12.68	99.06	30.54	99.15
	3	2.51	99.01	6.26	99.01	13.65	99.06	34.22	99.06
C2670	1	3.02	95.65	10.27	95.65	27.54	95.65	49.52	95.72
	2	2.00	95.47	5.65	95.51	12.43	95.58	33.05	95.58
	3	2.75	95.51	8.73	95.51	24.91	95.51	40.62	95.65
C3540	1	2.86	96.14	9.85	96.19	20.02	96.19	37.89	96.25
	2	2.72	96.14	7.58	96.16	9.04	96.22	15.68	96.22
	3	3.36	96.03	7.71	96.16	16.84	96.19	29.96	96.16
C5315	1	2.90	98.91	10.79	98.94	20.08	98.96	37.72	98.96
	2	3.08	98.80	9.17	98.78	12.01	98.94	23.57	98.94
	3	2.82	98.75	8.32	98.94	14.60	98.96	32.70	98.96
C7552	1	3.25	98.35	7.51	98.36	16.92	98.36	40.63	98.36
	2	3.29	98.28	5.49	98.34	9.76	98.36	24.57	98.36
	3	2.95	98.33	6.87	98.36	16.49	98.33	32.60	98.36

3.2 实验 2

我们用 ATGTA 系统对一个 25000 门、有上千条反馈线的 ASIC 电路采用方法 1 进行故障划分,模拟了 4 处理机并行测试生成和故障模拟的效果.结果数据列在表 2 中,其中 TG 时间为 SUN20 的 CPU 秒,而 FS 时间为 SUN4 的 CPU 秒.

表 2

TG/FS 方式	测试码数	测试码长度	时间(s)(加速比)		故障覆盖率(%)
			TG	FS	
单处理机	4 000	18 532	17 313(1.0)	64 102(1.0)	96.65
4 处理机	4 * 1 000	19 409	4 462(3.88)	15 832(4.05)	96.75

3.3 结 论

分析表 1 和表 2 的数据,可以得出下列结论.

(1) 平均来看,在 4 处理机的情况下,3 种方法的效果差别不太大.

(2) 当处理机数达到 16 以上时,方法 1 的加速比明显优于方法 2 和方法 3,所达到的故障覆盖率与单机时几乎相同(单机覆盖率请参阅文献[3]),比方法 2 和方法 3 略好一些.方法 2 是文献[1]推荐的方法.但我们的实验结果是,当处理机数较大时,它不但不如本文提出的方法 1,甚至比方法 3 的效果还差.

(3) 方法 1 对于 6 个 ISCAS-85 电路取得了线性加速比(直线斜率在 0.2~0.7 之间)的良好效果.特别是处理机数为 256 时,平均加速比达到 42,表明该方法对大规模并行测试生成将有明显优势.

(4) 表 2 说明,方法 1 对于时序电路同样有效.对 TG 与 FS 为松散耦合的系统,4 处理机并行所获得的加速比直线斜率等于 1.

参考文献

- Banerjee P. Parallel algorithm for VLSI Computer-Aided Design. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1994
- 李忠诚,潘榆奇,闵应骅.一个基于电路结构分析的测试产生系统——SABATPG.中国科学(A辑),1993,23(2):189~196
(Li Zhong-cheng, Pan Yu-qi, Min Ying-hua. A structure analysis based test generation system——SABATPG. Science in China (series A), 1993, 23(2):189~196)
- 曾芷德. ASIC 测试生成和可测性分析系统 ATGTA. 计算机学报, 1998, 21(5):448~455
(Zeng Zhi-de. ASIC test generation and testability analysis system——ATGTA. Chinese Journal of Computers, 1998, 21(5):448~455)
- Patil S, Banerjee P. Performance trade-offs in a parallel test generation fault simulation environment. IEEE Transactions on Computer-aided Design Integrated Circuits Systems, 1991, 10(12):1542~1558
- Chandra S J, Patel J H. Test generation in a parallel processing environment. In: Lopresti P V ed. Proceedings of the International Conference on Computer Design (ICCD'88). Rye Brook, NY: IEEE C, IEEE CAS and IEEE ED, 1988. 11~14
- Patil S, Banerjee P. A parallel branch and bound approach to test generation. IEEE Transactions on Computer-aided Design Integrated Circuits Systems, 1990, 9(3):313~322
- Agrawal P, Agrawal V D, Villojo J. Sequential Circuit Test Generation on a Distributed System. In: Newton A R ed. Proceedings of the 30th Design Automation Conference (DAC'93). Anaheim: ACM/IEEE, 1993. 337~341
- Ramkumar B, Banerjee P. ProperTest: portable parallel test generation for sequential circuits. In: Jones I K ed. Proceedings of the International Conference Computer-aided Design (ICCAD'92). Washington D. C: Computer Society Press, 1992. 220~223
- Chan T. Distributed automatic test pattern generation. In: Tsai Y T ed. Proceedings of the Application Specific Integrated Circuits Conference (ASIC'92). Rochester, NY: IEEE Rochester Section, 1992. 385~388
- Tao Jing, Zeng Zhi-de. A network partitioning algorithm for parallel test generation in VLSI circuit. In: Yang Shu zi ed. Proceedings of the 4th International Conference on CAD/CG. Wuhan: SPIE(USA), 1995. 645~650
- 曾芷德. 数字系统测试与可测性. 长沙:国防科学技术大学出版社, 1992
(Zeng Zhi-de. Testing and Testability for Digital Systems. Changsha: National University of Defense Technology Press, 1992)
- 刘胜利,曾芷德. 组合电路故障模拟加速方法的研究与实现. 见:全国第 10 届 CAD & CG 会议论文集. 北京:航天总公司 23 所, 1998. A-1~A-5

(Liu Sheng-li, Zeng Zhi-de. The research and realization of accelerating fault simulation for combinational circuits. In: Proceedings of the 10th Chinese CAD/CG Conference. Beijing: Institute No. 23 of China Aerospace Corporation, 1998. A-1~A-5)

A Parallel Test Generation Algorithm Based on Fault Partitioning

ZENG Zhi-de ZENG Xian-jun

(Department of Computer Science National University of Defense Technology Changsha 410073)

Abstract In this paper, the authors analyze in theory how to increase the speed-up ratio of parallel test generation algorithm based on fault partitioning. The approach of backward fault partitioning of output fan-in cones (BFPOC) which combines the relevant fault recognition and shortest path sensitization, is presented. And BFPOC is compared via experiment with the approach of toward fault partitioning of input fan-out cones (TFPIC) proposed by Banejee and the general one, equal distance partitioning of fault sequence (EDPFS). The experimental results show that in large-scale parallel processing environment, BFPOC can reach higher speed-up ratio, obvious super to the other two approaches.

Key words Parallel test generation, fault parallelism, fault partitioning, output fan-in cones, input fan-out cones, speed-up ratio.