

# 一种 BIST 测试激励的聚类移位压缩方法

涂吉 王子龙 李立健

(中国科学院自动化研究所 北京 100190)

**摘要** 提出一种针对内建自测试的测试激励聚类移位压缩方法。对难测故障的测试向量进行聚类压缩,将测试向量划分为若干类,每类内的向量相互之间最多只有一比特相异,从每类中只选取一个种子向量存储到 ROM 中。为了进一步提高测试向量压缩率,对聚类后的种子向量再进行移位压缩。实验结果表明,聚类移位压缩具有较高的测试数据压缩率,能减少难测向量存储单元,且能以芯片频率进行测试。

**关键词** 聚类压缩,移位压缩,内建自测试,测试激励

**中图分类号** TP306.2 **文献标识码** A

## Rotation-based Test Pattern Clustering Compression Method for BIST

TU Ji WANG Zi-long LI Li-jian

(Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China)

**Abstract** This paper presented a novel test pattern compression scheme for deterministic BIST. To reduce the storage requirements for the deterministic patterns, it relies on a two-dimensional compression scheme, which combines the clustering compression and rotation-based compression. Clustering compression divides the rest random pattern resistant faults(RPRF) into several clusters in a circuit. The test patterns in each cluster are no more than one bit different from each other, and only one seed selected from each cluster is needed to be stored in ROM. In order to reduce more storage cells, rotation-based compression method is added to compress the seeds of clustering compression. Experimental results show that the proposed scheme requires less test data storage than some previously published schemes, and it has the ability of at-speed test.

**Keywords** Clustering compression, Rotation-based compression, Built-in self-test, Test stimulus

## 1 引言

随着 EDA 工具的发展和半导体制造工艺的不断进步,现代集成电路的规模越来越大,工作频率越来越高,对集成电路的测试也提出了更高的要求。一方面,随着集成电路规模的增大,对集成电路进行测试时所需要的测试数据量增多,对自动测试设备(ATE)的性能要求变高,测试成本大幅度上升。另一方面,集成电路工作频率的增长速度比 ATE 的工作频率增长速度要快,使得 ATE 以芯片工作频率进行测试变得越来越困难,对延时故障等故障进行测试变得愈加困难。国际半导体技术蓝图(ITRS)中指出,超过 1/3 的被调查者认为测试能提高产品质量和降低产品成本<sup>[1]</sup>。内建自测试技术(Build-In Self-Test, BIST)通过在电路内部集成少量的逻辑电路实现对集成电路的测试,能降低测试成本和以电路频率进行测试,因此广泛使用在现代集成电路测试中<sup>[2]</sup>。

一个好的 BIST 方案至少应该具备以下两个品质:较少的硬件开销和较短的测试时间。在确定性 BIST 设计时,通常先采用线性反馈移位寄存器(LFSR)产生伪随机测试向量,

侦测大多数的易测故障,然后对剩下的少量难测故障,用自动测试向量生成器(ATPG)产生确定性测试向量,并压缩存储在 BIST 电路的 ROM 中。嵌入到电路的测试向量能减轻集成电路测试时对 ATE 性能的要求<sup>[3]</sup>。为了减少难测向量所占的 ROM 面积开销,工业界和学术界对测试激励向量压缩方法进行了许多研究<sup>[4]</sup>。

以下是一些针对 BIST 的测试激励向量压缩方法。基于扭环计数器(Twist-Ring Counter, TRC)的重播种技术<sup>[5]</sup>,将 LFSR 接成扭环计数器产生测试向量,其硬件开销小,能产生一些 LFSR 伪随机难以产生的确定性测试向量,但每个种子均会产生大量的无效向量,增加了测试时间。基于折叠计数器(Folding Counter)的重播种技术<sup>[6,7]</sup>,配置一个可编程的约翰逊计数器为折叠计数器,将确定性测试集嵌入折叠计数器的所有生成序列中。折叠计数器从单个种子产生的序列非常短,以致要嵌入一个完整的确定性测试集到单个折叠序列不太可能,因此该技术通常结合输入压缩等技术进行使用,算术编码压缩<sup>[8]</sup>,通常将确定性难测测试集中每 8 比特或 16 比特序列看成一个二进制数,将数值出现的概率映射到 $[0, 1]$ 区间

到稿日期:2013-09-17 返修日期:2013-11-21 本文受国家自然科学基金项目(61073035)资助。

涂吉(1986—),男,博士生,CCF 会员,主要研究方向为集成电路可测性设计、测试数据压缩等, E-mail: tuji100@163.com;王子龙(1988—),男,博士生,主要研究方向为集成电路可测性设计、片上系统体系结构设计等;李立健(1960—),男,博士,研究员,博士生导师,主要研究方向为集成电路可测性设计、集成电路微体系结构等。

分成许多小段,每段的长度等于某一序列的概率,再在段内取一个二进制小数进行编码,算术编码不需要很大的码表,但计算复杂。极限熵压缩<sup>[9]</sup>,通过计算编码的平均信息量来选择编码方案以及对无关位进行填充来提高特定编码方案的压缩率,其针对每种编码压缩,压缩率会有提高,但对无关位的填充,算法复杂。哥伦布编码<sup>[10]</sup>,即变长到变长的编码方法,使用变长的数据块来编码变长的原始数据,有些码字的利用效率不高,解码电路采用循环扫描移位寄存器即可。此外,还有变长-变长编码<sup>[11]</sup>以及变长哈夫曼编码<sup>[12]</sup>等编码压缩方法。

本文提出了一个新颖的聚类移位 BIST 测试激励压缩方案,即对难测测试集先进行聚类压缩,再对聚类压缩后的向量集进行移位压缩,然后将最终的种子存放在 BIST 电路的 ROM 中。通过对基准电路的综合实验,验证了本方案的设计思想,达到了对测试数据进行压缩的目的。

## 2 聚类压缩

本文提出的聚类压缩,是基于对难测向量之间相似度的观察,发现由 ATPG 产生的部分难测向量相互之间只有少数比特是不同的,而且具有大量的无关位。

从信息论的角度看,这些大量共同的位包含了重复的信息,在理论上可以用更少的比特表示等量的信息。于是将难测向量聚类,使得每类的元素之间包含重复的信息量尽可能多,以便于进行压缩。下面给出聚类压缩的一些定义。

**定义 1** 对于向量  $g_i = t_{i,1} t_{i,2} \dots t_{i,n} \in \{0, 1, X\}^n$ , 其中, 'X' 表示无关位,即可为 1 也可为 0,将  $g_i$  与  $g_j$  不同的比特个数称为向量  $g_i$  与  $g_j$  的聚类距离,记作  $D(g_i, g_j)$ 。计算公式为:

$$D(g_i, g_j) = \sum_{k=1}^n (t_{i,k} \oplus t_{j,k}) \quad (1)$$

例如:当  $g_i = 10001X$ ,  $g_j = 110101$  时,由式(1)得,  $D(g_i, g_j) = 3$ 。

**定义 2** 对于集合  $G(m) = \{g_1, g_2, \dots, g_m\}$ , 如果存在向量  $g_0$ ,使得对任意  $g_i \in G(m)$ ,都有  $D(g_i, g_0) \leq 1$ ,则称  $G(m)$  为一阶类。其中,  $g_0$  称为该一阶类的原型。

对基准电路 s1196 进行一阶类聚类压缩,压缩结果如图 1 所示。

```

s1196难测向量集
011101011x0110xxxxxxx1lxxxxxxx g1
01110111100110xxxxxxx10xxxxxxx g2
1111111111110xxxxxxx0 g3
01110111100110xxxxxxx1lxxxxxxx g4
01100010001110xxxxxxx1xxxxxxx g5
1111111111110xxxxxxx0lxxxxxxx g6

s1196聚类压缩后测试集
1111111111110xxxxxxx0lxxxxxxx s1
01110111100110xxxxxxx1lxxxxxxx s2
01100010001110xxxxxxx1xxxxxxx s3

```

图 1 聚类压缩实例

由 ATPG 产生的基准电路 s1196 的原始难测向量共有 6 个。其中  $g_6$  和  $g_3$  组成一阶类,可由原型  $s_1$  产生; $g_1, g_2$  和  $g_4$  组成一阶类,可由原型  $s_2$  产生; $g_5$  组成一阶类,可由原型  $s_3$  产生。聚类后的测试集只存储原型。s1196 聚类压缩后的测试集共 3 个向量。

对原型的某位翻转一次,其它位保持不变,即得到一个新测试向量。测试时,将  $n$  比特的原型依次翻转  $n$  次,即得到  $n$  个不同的向量。原始难测向量即包含在这由原型导出的  $n$  个向量中。此即聚类解压电路的思想。

假设原始难测测试集  $T_X$  共  $a$  个向量,聚类压缩集  $T_Y$  共  $b$  个向量,则聚类压缩的压缩率为:

$$R_X = 100\% \cdot (1 - b/a) \quad (2)$$

聚类解压缩的时间复杂度是  $O(n^2)$ 。解压电路需要对原型的每一位各翻转一次,因此产生的冗余测试向量增加了测试时间和测试功耗。

聚类压缩算法的伪码如下:

```

Cluster_compression(PatternSet){
  NewSet=PatternSet;
  BestCluster[n_seed][n_pattern]=∅;
  for(i_seed=0;NewSet! = ∅;i_seed++){
    Find BestCluster[i_seed] from NewSet;
    Generate BenchSeed[i_seed] from
    BestCluster[i_seed];
    NewSet=NewSet-BestCluster[i_seed];
  }
  return BenchSeed & i_seed;
}

```

## 3 移位压缩

移位压缩对种子向量进行循环移位,每个测试周期左移或右移一位,得到一个新的测试向量。如种子向量  $t = 101$  通过移位可得到新向量:  $t_1 = 110, t_2 = 011$ 。对电路 s1196 聚类后的测试集  $T_Y$  进行移位压缩,得到的移位压缩后测试集  $T_Z$  如图 2 所示。

```

s1196聚类压缩后测试集
1111111111110xxxxxxx0lxxxxxxx s1
01110111100110xxxxxxx1lxxxxxxx s2
01100010001110xxxxxxx1xxxxxxx s3

s1196移位压缩后测试集
1111111111110xxx011000100011100 z1
x01110111100110xxxxxxx1lxxxxxxx z2

```

图 2 循环移位压缩实例

$T_Y$  中的  $s_1$  和  $s_3$  压缩为  $T_Z$  中的  $z_1$ ,  $T_Y$  中的  $s_2$  压缩为  $T_Z$  中的  $z_2$ 。解压电路中将移位压缩集中的  $z_1$  向左或向右循环移位,能得到向量  $s_1$  和  $s_3$ 。

若聚类压缩前的测试集  $T_X$  为  $a$  行,移位压缩后的测试集  $T_Z$  为  $c$  行,则聚类移位压缩的压缩率为:

$$R_Y = 100\% \cdot (1 - c/a) \quad (3)$$

循环移位压缩的优点是可以利用现成的扫描链或 LFSR 等寄存器组<sup>[5]</sup>,只需增加一个计数器,即可产生伪随机向量。缺点是存在很多冗余移位,这些移位操作占用测试时间,增加测试功耗。

## 4 解压电路硬件实现

在测试的第一阶段,进行确定性测试向量测试。从 BIST 内部的 ROM 中,将预先存储的难测向量压缩集载入到聚类移位寄存器中。聚类解压缩和移位解压缩交织进行。在第二阶段,用 LFSR 伪随机产生测试向量对电路剩下的故障进行

测试。

测试第一阶段的解压电路(见图3),由 $\lceil \log_2 b \rceil$ 位的比特计数器、向量计数器、移位计数器,以及 $\lceil \log_2 w \rceil$ 位的地址计数器等组成。

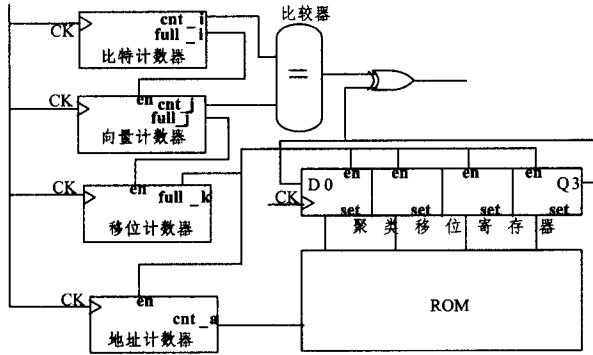


图3 聚类移位压缩解压电路原理图

图3中,  $cnt$  代表计数器的计数值,  $en$  代表使能信号,  $full$  代表计数器的最高位的值。聚类移位寄存器中预置初始种子向量。比特计数器和向量计数器控制聚类基准向量翻转第  $cnt_i$  位。比特计数器计满时, 向量计数器的计数值  $cnt_j$  增1。向量计数器计满, 也即  $full_j$  为高电平时, 使能移位计数器进行计数。当移位计数器的  $full_k$  为低电平时, 聚类移位寄存器的置位使能信号  $en$  无效, 聚类移位寄存器中的向量循环移位。移位计数器计满, 也即  $full_k$  为高电平时, 每个种子向量聚类解压和移位解压均已完毕, 此时, 使能地址计数器取ROM的下一个地址值, 同时使能聚类移位寄存器载入ROM中的新种子向量。聚类解压缩和移位解压缩共用一个  $n$  位的聚类移位寄存器组, 使得本文方案中的移位解压缩只需额外的一个移位计数器。

聚类移位解压电路的时间复杂度是  $O(n^3)$  个时钟周期。本方案的 BIST 电路采用 Verilog 语言编写, 在 Modelsim SE6.5 中已验证。

## 5 实验结果及分析

在伪随机测试模式下, 连续 256 个伪随机向量测不到故障时, 则停止伪随机测试。把剩下未检测到的故障视为难测故障, 采用美国弗吉尼亚大学教育网上公开的 ATPG 工具 Atalanta<sup>[13]</sup> 产生确定性的难测向量。本文选取了部分 ISCAS-85 和 ISCAS-89 基准电路的难测向量集, 在 2.8GHz 双核、4G 内存的 PC 机中, 采用 C 语言编写了聚类移位输入压缩的程序。

实验结果如表1和表2所列。表1是本文方案压缩率与文献[9-12]中所选方案实验结果的比较。从表1中可以看出, 极限熵方案的压缩率均高于哥伦布编码、变长-变长编码和变长哈夫曼编码的压缩率。电路 s5378、s9234、s15850 以及 s38584 的聚类移位压缩率要比极限熵压缩率高很多。电路 s13207 的聚类移位压缩率稍逊于极限熵压缩。电路 s38417 的聚类移位压缩率比文献[9-12]中所选方案压缩率都要低, 这是因为聚类和移位压缩针对完整测试向量进行处理, 而文献[9-12]中编码方案是将测试向量分成许多小段进行压缩, 分成小段后, 可重复利用的信息量增多, 但文献[9-12]中编码方案的解压电路硬件开销相应地要比本文方案大。对于输入个数较多且无关位个数较少的电路 s38417, 聚类移位压

缩的压缩效果有限。

表1 本文方案与文献[9-12]的比较

电路名称	难测向量个数	输入个数	本文聚类移位压缩	极限熵压缩 <sup>[9]</sup>	哥伦布编码 <sup>[10]</sup>	变长-变长编码 <sup>[11]</sup>	变长哈夫曼编码 <sup>[12]</sup>
s5378	6	214	83.3%	52.4%	40.7%	48.0%	51.8%
s9234	142	247	80.3%	47.8%	43.3%	43.6%	47.2%
s13207	6	700	83.3%	83.7%	74.8%	81.3%	83.5%
s15850	179	611	79.3%	68.2%	47.1%	66.2%	67.9%
s38417	441	1664	31.1%	54.5%	44.1%	43.3%	53.4%
s38584	58	1464	91.4%	62.5%	47.7%	60.9%	62.3%

电路 s38584 中无关位所占的比例在本文实验的 15 个电路中最高, 故理论上可以用更少的信息表征原始信息。从表1中可以看出, 文献[9-12]中编码方案对电路 s38584 的压缩效果比本文方案要差, 对无关位的利用率没有本文方案高。

表2 聚类移位压缩的压缩过程及压缩效率

电路名称	难测向量个数	输入个数	无关位所占比例	聚类压缩		聚类移位压缩	
				向量个数	压缩率	向量个数	压缩率
s420	20	35	49.71%	8	60.00%	8	60.00%
s641	7	54	61.11%	1	85.71%	1	85.71%
s713	9	54	61.11%	1	88.89%	1	88.89%
s838	72	67	54.42%	53	26.39%	47	34.72%
s953	10	45	68.67%	6	40.00%	3	70.00%
s1196	6	32	51.04%	4	33.33%	2	66.67%
s1238	6	32	51.04%	4	33.33%	2	66.67%
c2670	85	233	83.36%	56	34.12%	12	85.88%
s5378	6	214	92.06%	3	50.00%	2	66.67%
c7552	107	207	70.71%	80	25.23%	51	52.34%
s9234	142	247	86.74%	94	33.80%	28	80.28%
s13207	6	700	97.76%	3	50.00%	1	83.33%
s15850	179	611	96.17%	64	64.25%	37	79.33%
s38417	441	1664	97.94%	348	21.09%	304	31.07%
s38584	58	1464	98.52%	20	65.52%	5	91.38%

表2中, 第4列为各电路测试向量中无关位所占的比例, 为无关位的个数与测试向量总比特数之比。第5列为聚类压缩后测试集的向量个数。第6列为聚类压缩方法的压缩率, 由式(2)计算得出。经过聚类, 所有电路的压缩率均有提高。其中小电路 s713 的聚类压缩率达到了 88.89%, 大电路 s15850 和 s38584 的压缩率也都超过了 60%。这些实验结果说明, 聚类压缩利用向量之间的重复信息进行压缩, 效果明显。

表2中第7列为在聚类压缩的基础上再移位压缩得到的测试集的向量个数。第8列的移位压缩率由式(3)计算得出。其中, 有些电路压缩率仍有提高, 大电路 s38584 的压缩效果更加显著, 聚类移位后的压缩率高达 91.38%。另一些电路, 由于聚类压缩已充分压缩了测试集, 因此再在与聚类垂直的维度进行移位压缩时, 压缩率不变。

从表2的第4、6、8列可知, 本文方案压缩率与电路无关位所占的比例不是线性的关系。电路 s38417 的无关位所占比例仅次于 s38584, 但压缩率却是 15 个所选基准电路中最低的。

电路 s38417 的聚类压缩率和移位压缩率都不高, 这是因为该电路的输入个数较多且难测向量个数较多。在实际应用中, 应该综合分析时间开销和硬件开销来决定是否在聚类压缩的基础上再结合移位压缩。

结束语 提出了一种聚类压缩方法并将之与移位压缩方

法巧妙结合,充分共用同一个解压电路能够达到减少 ROM 开销和降低测试成本的目的。聚类移位压缩解压电路结构简单,易于实现。实验结果显示了聚类移位压缩的有效性。未来将研究容错移位算法,更充分地利用聚类压缩的特性来增加移位压缩的效率。

### 参考文献

[1] Semiconductor Industry Association (SIA). Test and Test Equipment, International Technology Roadmap for Semiconductors (ITRS) 2011 Update [OL]. <http://www.itrs.net/Links/2011ITRS.htm>

[2] 李立健,赵瑞莲. 减少多种子内建自测试方法硬件开销的有效途径[J]. 计算机辅助设计与图形学学报, 2003, 15(6): 662-666

[3] Han Yin-he, Li Xiao-wei, Li Hua-wei, et al. Embedded Test Resource for SoC to Reduce Required Tester Channels Based on Advanced Convolutional Codes [J]. IEEE Transactions on Instrumentation and Measurement, 2006, 55(2): 389-399

[4] Conroy Z, Li Hui, Balangue J. Built In Self Test (BIST) Survey-an industry snapshot of HVM component BIST usage at board and system test [C] // Electronic Manufacturing Technology Symposium (IEMT), 34th IEEE/CPMT International. Melaka, 2010: 1-6

[5] Swaminathan S, Chakrabarty K. On Using Twisted-Ring Counters for Test Set Embedding in BIST [J]. Journal of Electronic Testing, 2001, 17(6): 529-542

[6] Hellebrand S, Liang H-G, Wunderlich H-J. A Mixed Mode BIST Scheme Based on the Reseeding of Folding Counters [C] // At-

lantic City. Proceedings IEEE International Test Conference, 2000: 778-784

[7] Liang Hua-guo, Hellebrand S, Wunderlich H-J. Two-Dimensional Test Data Compression for Scan-Based Deterministic BIST [J]. Journal of Electronic Testing, 2002, 18(2): 159-170

[8] Hashempour H, Lombardi F. Compression of VLSI test data by arithmetic coding [C] // Cannes, France. Defect and Fault Tolerance in VLSI Systems, 2004: 150-157

[9] Balakrishnan K J, Toubia N A. Relating entropy theory to test data compression [C] // Test Symposium, ETS 2004. Proceedings. Ninth IEEE European. IEEE Press, Corsica, France, 2004: 94-99

[10] Chandra A, Chakrabarty K. System-on-a-chip test data compression and decompression architectures based on Golomb Codes [J]. IEEE Trans. on Computer-Aided Design, 2001, 20(3): 355-368

[11] Chandra A, Chakrabarty K. Frequency-Directed Run-Length Codes with Application to System-on-a-chip Test Data Compression [C] // Proc. VLSI Test Symposium. IEEE Computer Society, California, 2001: 42-47

[12] Gonciari P T, Al-Hashimi B M, et al. Variable-Length Input Huffman Coding for System-on-a-chip Test [J]. IEEE Trans. on computer-Aided Design, 2003, 22(6): 783-796

[13] Lee H K, Ha D S. On the Generation of Test Patterns for Combinational Circuits [R]. Technical Report No. 12\_93. Virginia: Virginia Polytechnic Institute and State University, Department of Electrical Engineering, 1997

(上接第 23 页)

和 HCNF 算法相比, HIPLTS 算法加速比 Speedup 分别提高约 13.75% 和 8.49%, 调度下界比 SLR 分别降低约 14.02% 和 9.25%。HIPLTS 算法较高地提升了任务调度效率。

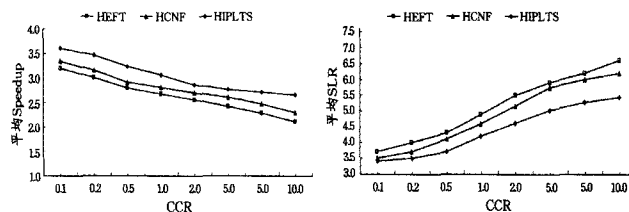


图 9 不同 CCR 下的加速比对比图 图 10 不同 CCR 下的 SLR 对比

**结束语** 如何将任务分配到处理器内核上进行处理是能否充分发挥多核处理器性能的首要问题。本文通过对异构多核处理器任务调度问题的深入分析与研究,提出一种改进的优先级列表任务调度算法,该算法采用具有异构平台匹配性的优先级排序方式和全局处理器选择方式,克服现有优先级列表任务调度算法中存在的缺点和不足,并在任务分配时采取任务复制技术和区间插入技术对调度过程进行进一步优化。实验验证结果表明,该算法有效减少了任务调度长度,提高了处理器利用率。

### 参考文献

[1] 张建军,宋业新,旷文. 基于异构环境的 Out\_Tree 任务图的调度算法[J]. 计算机科学, 2013, 40(4): 107-111

[2] Topcuoglu H, Hariri S, Wu Min-you. Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing [J]. IEEE Transactions on parallel and distributed systems, 2002, 13(3): 260-274

[3] Prashanth C, Sai Ran-ga. Algorithms for task scheduling in heterogeneous computing environments [D]. Alabama: Auburn University, 2006

[4] Ullman J D. Np-complete scheduling problem [J]. Journal of Computer and System Sciences, 1975, 10(3): 384-393

[5] Hagra T, Janecek J. A high performance low complexity algorithm for compile-time task scheduling in heterogeneous systems [J]. Parallel computing, 2005, 31(7): 653-670

[6] Daolud M I, Kharna N. A high performance algorithm for static task scheduling in heterogeneous distributed computing systems [J]. Journal of parallel and distributed computing, 2008, 68(4): 399-409

[7] 何琨,赵勇,黄文奇. 基于任务复制的分簇与调度算法[J]. 计算机学报, 2008, 31(5): 733-740

[8] Darbha S, Agrawal D P. Optimal scheduling algorithm for distributed memory machines [J]. IEEE transactions on parallel and distributed systems, 1998, 9(1): 87-95

[9] 王小非,方明. 一种基于调度簇树的周期性分布实时任务调度算法[J]. 计算机科学, 2007, 34(3): 256-261

[10] 曹仰杰,钱德沛,伍卫国,等. 众核处理器系统核资源动态分组的自适应调度算法[J]. 软件学报, 2012, 23(2): 240-252