

一种双游程交替编码的测试数据压缩方法

程一飞¹ 詹文法²

(安庆师范学院计算机与信息学院 安庆 246001)¹ (安庆师范学院科技处 安庆 246001)²

摘要 SoC测试面临的挑战之一是测试数据量过大,而测试数据压缩是应对这一挑战行之有效的方法。因此,提出了一种新的双游程交替的测试数据压缩方法,该方法对测试集中0游程和1游程交替编码,并且后一游程类型可以根据前一游程类型转变得到。这样在代码字中不需要表示游程类型,减少了游程所需代码字的长度。实验结果表明,该方法能够取得比同类方法更高的压缩率,而且解压结构简单,因此能够达到降低测试成本的目标。

关键词 测试数据压缩,双游程,无关位

中图法分类号 TP391.76 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.11.005

Test Data Compression Method of Dual Run Length Alternating Coding

CHENG Yi-fei¹ ZHAN Wen-fa²

(School of Computer and Information, Anqing Normal College, Anqing 246011, China)¹

(Department of Science Research, Anqing Normal College, Anqing 246011, China)²

Abstract The large amount of test data is one of the challenges in SoC test, and the test data compression is an effective method to deal with this challenge. A new test data compression method of a dual run length alternating compression was presented. 0 run length and 1 run length can be coded alternately, and the type of next run length can be obtained according to the previous run length type. So the type of run length is not represented in code words, and hence the length of the needed code word is reduced. Experimental results show that the method can achieve higher compression ratio compared with the similar method, and decompression structure is very simple, so the goal of reducing the cost of test can be achieved.

Keywords Test data compression, Dual run length, Don't care bit

随着工艺水平的提高,系统芯片 SoC(System-on-a-chip)上集成的晶体管数目急剧增加。同时,为了加速 SoC 的设计过程,通常会广泛采用 IP(Intellectual Property)核复用的设计技术。为了保证产品无缺陷,必须对芯片进行测试。目前,SoC 测试面临的困难越来越多,而测试数据量过大是当前面临的困难之一。为了缓解测试数据急速增长的压力,通常行之有效的方法是采用压缩技术对测试数据进行压缩。测试数据压缩技术首先将测试集 T_D 按照一定的编码方法进行编码,编码后的结果记为 T_E ,并将 T_E 存储到自动测试仪(Automatic Test Equipment, ATE)中。当对电路进行测试时,先将 T_E 通过 ATE 的传输通道传送到芯片上的解压电路,解压电路将 T_E 解压得到原测试集 T_D ,并将 T_D 施加到待测电路完成测试。采用压缩技术既降低了 ATE 存储空间的需求,又降低了 ATE 传输通道的压力,同时也减少了传输时间。

依据所用编码的原理,可以将测试数据压缩方法分为基于统计编码、基于字典编码和基于游程编码等。在基于统计编码的压缩技术中, Huffman 编码虽然能够得到最短的平均码字,但是它的解码电路较为复杂。基于字典编码的方案首先需要存储字典,再者由于需要根据字典索引去找原始数据,增加了访问存储器的时间。基于游程的编码方案具有较高的

数据压缩率和较小的解码电路开销, Golomb^[1] 编码、FDR^[2] 编码和 AFR^[3] 编码等都属于基于游程的编码方案。

本文提出一种新的变长到变长的压缩方法,它是一种双游程交替的编码方法。这种方案有以下特点:(1)对测试集中0游程和1游程的长度进行编码;(2)相同长度的0/1游程使用相同的代码字;(3)0/1游程交替出现,后一游程类型可以根据前一游程类型转变得到。这样在代码字中不需要表示游程类型,减少了游程所需代码字的长度,因此能够达到进一步提高压缩效率、降低测试成本的目标。

1 双游程交替编码

游程长度,是指在一个由任意个0、1组成的数据流中,连续0或连续1的数目的个数。如000001是0游程,游程长度为5;1110是1游程,游程长度为3。

本文编码的基本思路是:对测试数据中连续0以及连续1的长度进行编码,并且假设连续0数据后必定是连续1数据;反之亦然。故引入交替编码的思想对0、1游程编码,且0、1游程采用一套编码表,若0游程之后出现的仍然是0游程或者1游程之后出现的仍然是1游程,则在这两个游程中间插入分隔符01。

到稿日期:2014-01-13 返修日期:2014-03-17 本文受国家自然科学基金项目(61306046)资助。

程一飞(1976-),男,硕士,副教授,主要研究方向为测试数据压缩, E-mail:chengyif@aqtc.edu.cn;詹文法(1978-),男,博士,教授,主要研究方向为测试数据压缩。

编码由前缀和后缀两部分构成,前缀表示组号,第一组的前缀为 10,其它组前缀有两种形式,分别是 $\underbrace{0\cdots 01}_k$ 和 $\underbrace{1\cdots 10}_k$,换种说法第 1 组的前缀是长度为 1 的 1 游程,在第 $k(k>1)$ 组的前半段,组前缀以长度为 k 的 0 游程对应的编码表示;在第 k 组的前半段,组后缀由最小的 k 位二进制数至最大的 k 位二进制数依次排列;在第 k 组的后半段,组前缀以长度为 k 的 1 游程对应的编码表示;在第 k 组的后半段,组后缀同样由最小的 k 位二进制数至最大的 k 位二进制数依次排列,因此第一组包含两种长度的游程,其它组包含 $2 \times 2^k = 2^{k+1}$ 种长度的游程。长度为 l 的游程分配到 k 组,其中 $k = \lceil \log_2^{l+5} - 1 \rceil$ 。例如 $l=26$,则 $k = \lceil \log_2^{26+5} - 1 \rceil = 3$,即在第 3 组。依此方法可得具体编码表,部分编码表如表 1 所列。

表 1 双游程交替编码表

组	游程长度	组前缀	组后缀	码字
A1	1	10	0	100
	2		1	101
	3	001	00	00100
	4		01	00101
	5		10	00110
A2	6	11	11	00111
	7		00	11000
	8	110	01	11001
	9		10	11010
	10		11	11011
A3	11	0001	000	0001000
	12		001	0001001
	...	...	...	...
	18		111	0001111
	19	1110	000	1110000
	20		001	1110001
	...		...	...
...	26	...	111	1110111
...	...	...	...	...

对本文提出的这种编码说明如下:

- (1) 因为 0/1 游程交替出现,所以长度相同的 0/1 游程采用相同的编码;
- (2) 采用 01 作为同 0 或同 1 游程之间的分隔符;
- (3) 除了第一组以外,其余各组中组前缀有两种;
- (4) 第 k 组代码字前缀长度为 $k+1$,后缀长度为 k ,代码字长度为 $2k+1$;
- (5) 每组码字中,组前缀比组后缀长一位;
- (6) 除第一组外,第 k 组包含 2^{k+1} 个游程;
- (7) 第 $k+1$ 组比第 k 组代码字长度长 2,其中前缀增加 1 位,后缀增加 1 位。

对一个测试集编码的过程如图 1 所示。

- (1) 初始化标志位 flag 为 0, flag 作为当前期望的游程类型标志;
- (2) 获取当前游程长度,并求得对应编码;
- (3) 判断当前流程类型与期望游程类型是否相同,若相同,flag 取反;否则,需要输出分隔符;
- (4) 输出对应编码;
- (5) 重复(2)~(4)编码下一个游程直到整个测试集编码结束;
- (6) 对于测试集中最后一个游程,若它不是完整的游程,则需要是在所获得的编码后面加上游程结束标志,让其构成一个游程。例如最后得到的结果是 111111,则在其后添加 1 位 0,变成 1111110。

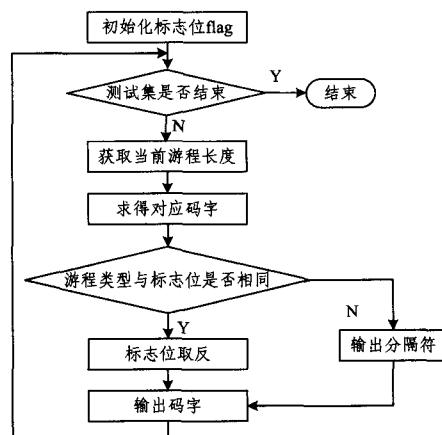


图 1 双游程交替编码流程图

2 编码举例

原始测试数据:00001 11111111111111111110 1110 0000 0001(41bits)

编码后的数据:00101 1110000 01 00100 11000(24bits), 其中 00101、1110000、00100、11000 为相应游程对应的编码,由于 11111111111111111110、1110 同于 1 游程,因此在它们之间插入 01 作为分隔符。

3 解码器设计

从表 1 可以看出游程的长度 i 可通过将 1+前缀起始位+所有后缀组成的二进制数换算为十进制再减 5 得到,即 $i = (1Xt)_2 - 5$,其中 X 为对应编码前缀起始位。如,游程长度 $i=5$,对应的码字为 00110, $(1010)_2 - 5 = 10 - 5 = 5$;游程长度 $i=8$,对应的码字为 11001, $(1101)_2 - 5 = 13 - 5 = 8$ 。因此使用一个 $k+2$ 位计数器就可以输出相应的位串。

本方案解压结构简单,独立于被测电路且大小可变,仅需要一个 FSM(有限状态机)、一个 $k+2$ 位计数器、一个 $\log_2(k+2)$ 位计数器和一个异或门。解压结构框图如图 2 所示,其中的信号名称和功能描述如表 2 所列。

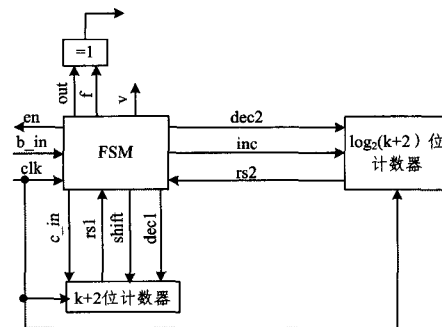


图 2 解压电路框图

表 2 解码器中信号及对应功能

信号	功能
b_in	数据输入端,需解码的数据通过此端移入 FSM
en	使能端,1 表示解码器已准备就绪,等待接受需解码的数据
c_in	FSM 向 $k+2$ 位计数器输入编码数据的通道
shift	控制编码数据移入 $k+2$ 位计数器
dec1	控制 $k+2$ 位计数器作减 1 操作
rs1	$k+2$ 位计数器复位标志
inc	控制 $\log_2(k+2)$ 计数器的加 1
dec2	控制 $\log_2(k+2)$ 计数器减 1 操作,
rs2	$\log_2(k+2)$ 计数器的复位标志
out	FSM 输出的解码数据
f	标志位,0 表示 0 游程,1 表示 1 游程

下面结合框图及流程图(见图 3)介绍本方法解码基本过程。

- (1)初始化 $en=1, f=0$;
- (2)FSM 接收编码的前缀并将前缀移入到 $k+2$ 位计数器。 $en=shift=inc=1$;若 $\log_2(k+2)$ 位计数器为 1,且前缀第 1 位为 0,即前缀为 01,跳至(8);
- (3)移入 1 位 1 到 $k+2$ 位计数器,继续移入前缀开始位到 $k+2$ 位计数器;
- (4)将尾部移入 $k+2$ 位计数器,dec2=1,表示每进入一位 $\log_2(k+2)$ 位计数器减 1,直到 $\log_2(k+2)$ 位计数器减为 1;
- (5) $k+2$ 位计数器进行减 1 操作。 $k+2$ 位计数器每减 1,out 输出 1 位 0, $v=1$,直到 $k+2$ 位计数器减为 5;
- (6)将 out 输出与 f 异或得到最终输出;
- (7)将 out 输出的 1 与 f 异或得到最终输出;
- (8)f 取反;
- (9)重复(2)–(8)直到解码结束。

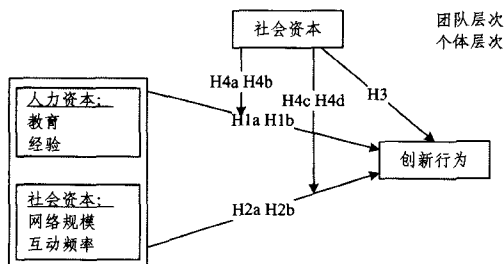


图 3 解码流程图

4 无关位填充

测试集中不仅包含 0 和 1,还包含大量的无关位(95%~99%)^[4]。无关位的取值不影响故障覆盖率,因此,可以选择性地对无关位进行填充,来提高编码的压缩效率。

本文采用如下无关位填充策略:

- 1)形如 AXXXXX 或 XXXA 这种形式的序列,X 填充为 A;
- 2)形如 AXXXXA 这种形式的序列,X 填充为 A;
- 3)形如 AAAAAXXXXXBBBBXXXXAAAA 这种形式的序列,采用动态规划的思想来确定多少 X 填充为 A,多少 X 填充为 B。

分别用 $A=\{a_i\}, B=\{b_i\}$ 来记录确定位的长度及确定位后无关位的长度。

例如,对于 $T_D=111XXXXXX00000XXXX1111XXXX0$,它可以表示为 $A=\{3,5,4,1\}, B=\{6,4,4,0\}$ 。

X 填充问题实际上就是一个将 B 序列中的 b_i 分配到 a_i 和 a_{i+1} ,使得所有游程对应的编码总长度最小。即 $Y(i,j)=\min\{Y(i-1,k)+f(a_i+j+b_{i-1}-k)\}$,其中 i 表示第 i 个游程, j 表示第 i 个游程尾部取 j 个无关位, $f(i)$ 表示长度为 i 的游程对应的编码长度, k 取值为从 0 依次到 b_{i-1} 。

5 实验结果

为了验证本文提出的编码方法的有效性,将其应用于 ISCAS 89 标准电路中几个规模较大的时序电路,采用了美国

Duke 大学提供的 MinTest ATPG 测试生成工具产生的测试向量集。

下面给出本方法的实验结果,如表 3 所列。第 1 列是电路名称,第 2 列是原始测试数据位数,第 3 列是压缩后数据位数,第 4 列是压缩率。

表 3 本方法压缩效果

电路名称	原始测试数据 位数(bit)	压缩后数据 位数(bit)	压缩率 (%)
s5378	23754	11938	49.74
s9234	39273	20542	47.69
s13207	165200	28174	82.95
s15850	76986	24606	68.04
s35932	28208	5063	82.05
s38417	164736	61813	62.48
s38584	199104	71797	63.94
平均			65.27

为了验证本方法的压缩效果,将本方法与国内外相关成果进行比较,结果如表 4 所列。

表 4 几种压缩方法效果比较

电路名称	Golomb 码 ^[1] 压缩率	FDR 码 ^[2] 压缩率	交替连续 长度码 ^[5] 压缩率	相对游程 长度码 ^[6] 压缩率	本文方法 压缩率
s5378	40.70	48.02	45.12	36.19	49.74
s9234	43.34	43.59	42.79	42.94	47.69
s13207	74.78	81.30	80.43	80.77	82.95
s15850	47.11	66.22	65.13	64.49	68.04
s35932	4.69	19.37	79.06	79.43	82.05
s38417	44.12	43.26	56.52	53.56	62.48
s38584	54.86	60.91	60.57	60.28	63.94
平均	44.23	51.81	61.37	59.67	65.27

第 1 列为电路名称,第 2 列是 Golomb 码压缩效果,第 3 列是 FDR 码压缩效果,第 4 列是交替连续长度码压缩效果,第 5 列是相对游程长度码^[6]压缩效果,第 6 列是本文方法压缩效果。从表 4 可以看出本文方法比 Golomb 码压缩效果平均提高了 21.04%,比交替连接长度码压缩效果平均提高了 3.9%,比相对游程长度码压缩效果平均提高了 5.6%。双游程交替的压缩效率要好于其它几种编码方法,其原因是测试集中除了有大量的 0 游程外,还有大量的 1 游程,因此同时对 0/1 游程进行编码可以获得更理想的压缩效果。

结束语 本文提出了一种双游程交替的测试数据压缩方法,即一种变长到变长的压缩方法,它以 FDR 码为基础,根据测试集中除了有大量的 0 游程外,还有大量的 1 游程,提出了对 0/1 游程交替编码的方法,后一游程类型可以根据前一游程类型转变得到,这样在代码字中不需要表示游程类型,减少了游程所需代码字的长度,从而有效提高了压缩率。同时该方法中解码电路简单且独立于被测电路。基于此,本方法具有极好的应用前景。

参考文献

- [1] Chandra A, Chakrabarty K. System-on-a-chip Test Data Compression and Decompression Based on Internal Scan Chains and Golomb Coding[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits& Systems, 2002, 21(6): 715-722

(下转第 55 页)

相似度匹配^[10]直接利用图的矩阵表示法对图的节点进行相似度计算,其缺陷在于这种设计模式的识别方法只注重各个节点之间的相似程度,而忽略了整张模型图的相似程度,这样只能保证各个节点的匹配而不能保证整幅模型图的匹配,因而准确度不够理想。

模板匹配^[10]则是根据不同的关系来构造多个关系矩阵,然后关系矩阵利用特征值合并归一的方式得到一个特征矩阵,进而对两个图的特征矩阵使用例如 NCC 算法 (Normal Cross Correlation)^[14]等相似性计算算法来计算两个图的相似性,然而这种方式虽然准确度较相似度匹配高,但是它仍然注重于节点相似而不是全图相似,只是其携带的结构信息更多。

DNIT (Depth-Node-Input Table)^[11]匹配也是一种图匹配算法,它将图匹配的过程分解成为 k 步, k 的值为两幅图中的一幅经过变换后若干个节点可以变得相同的节点个数。首先利用一个由祖先节点的数目、子女节点的数目、兄弟节点的数目组成的三元组(t_1, t_2, t_3)来表示模型图中的一个节点,然后再根据三元组使用一些算法来计算出每一个节点的特征值,这样便将整张模型图的节点信息转换成了该图的节点特征值表,然后再计算设计模式的模型图和用户系统模型图对应的特征值表之间的距离,得到一个距离矩阵 P ,然后使用迭代的方法迭代 K 步,每次迭代的过程都找出 P 中每行的最小值,最终得到两幅图的相似度,借此来实现设计模式的识别。

利用控制流 (CFG) 图进行设计模式识别^[12,13]的方法是利用分析数据流的方式来分析系统的行为模式,得到一个个的基本块,将这些分析得到的基本块串联起来便得到了控制流图,利用控制流图对设计模式进行识别。

传统的设计模式识别的方法计算矩阵相似度过程非常繁琐,对于设计模式的识别准确度也不高,而且将设计模式的模型转化成数字矩阵难以给人一个直观的识别过程的认识。文本提出的这种基于结构驱动的模式查询技术的设计模式的识别方法很好地克服了以上这些问题,识别过程清晰明了,识别的准确性也比较高。

结束语 本文基于之前提出的 UML 模型查询技术,进一步提出了一种基于结构查询的 UML 设计模式识别方法。针对之前模型查询技术中匹配算法由于使用递归而导致时间开销过大的问题,进行了基于 UML 的定向优化改进,有效提高了查询的效率。本文提出的针对 UML 的设计模式识别方法,能够灵活有效地查询 UML 模型中的特定结构,从而识别出相应的设计模式。此外,由于本文的设计模式结构特征是通过自定义的方式给出的,因此它具有良好的可扩展性。

下一步,我们还可以尝试对算法进行进一步的优化,使算法的运行效率更高,同时我们还打算将该工作完全移植到

EMF 框架下。

参考文献

- [1] Liu Hai-yan, Liang Jian-long, Suo Zhi-hai, et al. Design pattern and their applications to software design[J]. Journal of Xi'an Jiaotong University, 2005, 39(10): 1043-1047
- [2] Lu Bo, Chai Yue-ting. On Unified Modeling Language—UML [J]. Computer Engineering and Science, 2000, 22(4): 58-60
- [3] Subgraph Isomorphism Problem [EB/OL]. [2013-07-09]. http://en.wikipedia.org/wiki/Subgraph_isomorphism_problem
- [4] Document Object Model (DOM) [EB/OL]. [2009-01-06]. <http://www.w3.org/DOM/>
- [5] Wang Fang, Li Zheng-fan. The Realization Method of Parsing XML Document by SAX[J]. Journal of East China Jiaotong University, 2004, 21(1): 84-86
- [6] Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to Algorithms (Second Edition)[M]. Beijing, China: China Machine Press, 2006
- [7] Zhang Xue-lin, Zhang Tian, Li Xuan-dong. Query by Drawing Examples of UML Model[C]//Proceedings of the Software Engineering Conference (APSEC), 2012 19th Asia-Pacific. Hongkong, China, 2012: 154-157
- [8] Gamma E, Helm R, Johnson R, et al. Design Patterns: Elements of Reusable Object-Oriented Software[M]. Beijing, China: China Machine Press, 1995
- [9] Java API for XML Processing (JAXP) Tutorial [EB/OL]. [2008-07]. <http://www.oracle.com/technetwork/java/sax-138988.html>
- [10] Dong Jing, Sun Yong-tao, Zhao Ya-jing. Design pattern detection by template matching[C]//ACM Symposium on Applied Computing-SAC. 2008: 765-769
- [11] Pandel A, Gupta M, Tripathi A K. DNIT—A new approach for design pattern detection[C]//International Conference on Computer and Communication Technology-ICCCCT. 2010
- [12] Gupta M, Rao R S, Tripathi A K. Design Pattern Detection using inexact graph matching[C]//Proceedings of the International Conference on Communication and Computational Intelligence. 2010: 211-217
- [13] Shi Ni-ja, Olsson R A. Reverse Engineering of Design Patterns from Java Source Code[C]//Automated Software Engineering-ASE. 2006: 123-134
- [14] Lewis J P. Fast Template Matching[J]. Vision Interface, 1995, 5: 120-123
- [15] Schmidt D C. Guest Editor's Introduction: Model-Driven Engineering[J]. IEEE Computer, IEEE CS, 2006, 39: 25-31

(上接第 24 页)

- [2] Chandra A, Chakrabarty K. Frequency-directed Run-length (FDR) Codes with application to system-on-a-chip test data compression[C]//Proceedings of IEEE VLSI Test Symposium, 2001. Washington, DC, USA: IEEE Computer Society, 2001: 42-47
- [3] Chandra A, Chakrabarty K. Reduction of SOC Test Data Volume, Scan Power and Testing Time Using Alternating Run

Length Codes [C] // Design Automation Conference, 2002. Washington, DC, USA: IEEE Computer Society, 2002: 673-678

- [4] Toubia N A. Survey of Test Vector Compression techniques[J]. IEEE Design & Test of Computers, 2006, 23(4): 294-303
- [5] 梁华国, 蒋翠云. 基于交替与连续长度码的有效测试数据压缩和解压[J]. 计算机学报, 2004, 27(4): 548-554
- [6] 韩建华, 詹文法, 查怀志. 一种相对游程长度编码方案[J]. 计算机科学, 2012, 39(5): 295-299