

# 混合定变长码的测试数据压缩方案

詹文法<sup>1),2)</sup> 梁华国<sup>1)</sup> 时 峰<sup>1)</sup> 黄正峰<sup>1)</sup>

<sup>1)</sup>(合肥工业大学计算机与信息学院 合肥 230009)

<sup>2)</sup>(安庆师范学院教育科学与技术系 安徽 安庆 246011)

**摘 要** 文章提出了一种混合定变长码的测试数据压缩方案,该方案可以有效压缩芯片测试数据量,此压缩方案将代码字拆分为固定长度的首部 and 可变量长度的尾部两部分.首部固定使解压过程简单,硬件开销小;尾部可变使编码灵活.同时采用了将尾部最高位隐藏的方法来进一步提高压缩率,还使用了特殊的计数器来进一步简单化解压电路.对 ISCAS 89 部分标准电路的实验结果显示,文中提出的方案在压缩效率和解压结构方面都明显优于同类压缩方法,如 Golomb 码、FDR 码、VIHC 码、v9C 码等.

**关键词** 测试数据压缩;编码;内建自测试;定长码;变长码

**中图法分类号** TP306

## A Test Data Compression Scheme Based on Mixed Fixed and Variable Length Coding

ZHAN Wen-Fa<sup>1),2)</sup> LIANG Hua-Guo<sup>1)</sup> SHI Feng<sup>1)</sup> HUANG Zheng-Feng<sup>1)</sup>

<sup>1)</sup>(School of Computer and Information, Hefei University of Technology, Hefei 230009)

<sup>2)</sup>(Department of Educational Technology, Anqing Normal College, Anqing, Anhui 246011)

**Abstract** A test data compression scheme based on fixed and variable length coding (FAVLC) is presented, by using which the test data can be compressed efficiently. In the scheme code words is divided into fixed length head, which eases the control of decompression, and variable length tail, which adds the feasibility of encoding. In order to obtain further compression effect, the highest bit of the tail is reduced from the code words. In addition, a special shift counter is also used, which further ease the control circuit. Experimental results show that the proposed scheme obviously outperforms the traditional coding methods in the compression ratio and the implementation of decompression, such as Golomb, FDR, VIHC, v9C coding.

**Keywords** test data compression; coding; built-in self-test; fixed length coding; variable length coding

## 1 引 言

随着集成电路制造工艺的不断发展,单个芯片

上集成的晶体管数目越来越多.单个芯片上集成晶体管数目的增多和芯片工作频率的提升,给测试带来了新的挑战,其中庞大的测试数据量是测试必须面对的挑战之一.

收稿日期:2007-03-06;最终修改稿收到日期:2008-08-15. 本课题得到国家自然科学基金重点项目(60633060)、国家自然科学基金重大研究计划(90407008)和安徽省教育厅自然科学基金项目(KJ2008B031)资助. 詹文法,男,1978年生,博士研究生,副教授,主要研究方向为测试数据压缩,可测性设计等. E-mail: zhanwenfa@gmail.com. 梁华国,男,1959年生,博士,教授,博士生导师,主要研究领域为嵌入式系统综合与测试、内建自测试(BIST)、ATPG 算法、分布式控制等. 时 峰,男,1984年生,硕士研究生,主要研究方向为内建自测试、测试数据压缩等. 黄正峰,男,1978年生,博士研究生,主要研究方向为嵌入式系统综合与测试、数字系统设计自动化.

用一个可接受的代价,来解决急剧增长的测试数据量和有限的自动测试设备(Automated Test Equipment, ATE)传输带宽之间的矛盾,正变得愈来愈困难,测试资源划分(Test Resource Partition, TRP)提供了一种颇有前途的解决方案<sup>[1-3]</sup>. 在 TRP 技术的研究方面,总体上可分为 3 个方面:(1) 测试集紧缩(Test Set Compaction, TSC)<sup>[4]</sup>. 该技术主要通过紧缩部分带有无关系位(Don't care bits)的测试立方(test tubes),在故障覆盖率不变的情况下减少测试向量的个数来减少测试数据量,其优点是不需要投入附加的硬件开销,其缺点是其非模型故障的覆盖率要受到影响.(2) 内建自测试(Built-in Self-Test, BIST)<sup>[5-6]</sup>. BIST 的基本思想是利用芯片本身所带有的测试模式生成器(Test Pattern Generator, TPG),在片上直接生成测试向量,以降低对 ATE 的要求. 由于 BIST 生成的多是伪随机测试向量,测试时通常存在着抗随机故障(Random Resistant Fault, RRF),故 BIST 存在故障覆盖率不高、测试序列较长的弊端. 虽然可以通过加权或采用混合模式的 BIST 等方法来进一步提高测试效率,但随着电路规模的扩大,RRF 的增多,要付出的硬件开销将显著增加.(3) 测试数据压缩(Test Data Compression, TDC)<sup>[1]</sup>. TDC 主要采用的是无损数据压缩的方法,来压缩预先计算的测试数据,然后通过片上解压器进行解压. 它同样是将一些测试资源从 ATE 移入到芯片中,以达到减少测试数据量、缩短测试时间的目的,并能允许使用低速 ATE 而不降低测试质量. 该方法不需要了解被测设计(Design Under Test, DUT)的具体内部结构,可以很好地保护知识产权,因而得到了广泛的应用.

编码压缩方法是测试数据压缩最常用的方法之一,其根据原始数据和压缩后的数据体积的变化关系可分为 4 大类:定长到定长、定长到变长、变长到定长和变长到变长<sup>[7]</sup>. 定长到定长编码是使用定长的小数据块来编码定长的原始数据块,常见的有字典编码<sup>[8]</sup>和 LFSR 重播种编码方法<sup>[9]</sup>;定长到变长的编码使用一个变长的数据块来编码定长的原始数据,常见的有哈夫曼编码方法<sup>[1]</sup>;变长到定长的编码使用定长的数据来编码变长的原始数据,经典的游程编码属于这种类型<sup>[1]</sup>;变长到变长的编码使用变长的数据块来编码变长的原始数据,常见的有 Golomb 编码<sup>[10]</sup>、FDR 编码<sup>[2]</sup>、交替编码<sup>[11]</sup>、

EFDR 码<sup>[12]</sup>、交替连续码(AARLC)<sup>[13]</sup>、混合游程码(HRC)<sup>[14]</sup>、SVIC 码<sup>[15]</sup>、Variable-Tail 码<sup>[16]</sup>、变游程码<sup>[17]</sup>、VIHC<sup>[18]</sup>、v9C 码<sup>[19]</sup>等. 在以上 4 类方法中,定长到定长的编码方法压缩效果最差,但解压结构最简单、控制协议也最简单;变长到变长的编码方法可以取得很好的压缩效果,但其硬件开销、解压结构和控制协议也最复杂;变长到定长和定长到变长的编码方法其压缩效果、硬件开销、解压结构等处于定长到定长与变长到变长之间.

变长到变长的编码方法虽然可以取得较好的压缩效果,但该方法通常无法解决压缩率和硬件开销之间的矛盾,同时控制协议也比较复杂,FDR 码及其变种编码,如 EFDR 码、交替连续码、混合游程码、Variable-Tail 码和变游程码等在这个方面表现尤为突出. 为了弥补和克服这个缺点,本文提出了一种混合定变长码的测试数据压缩方案. 该方案同时具有定长码解压结构、控制协议简单和变长码编码灵活、压缩效果好的优点. 理论分析和实验结果都表明本方案具有很高的测试数据压缩率.

本文第 2 节介绍本方案的算法和理论基础、首部位宽的选取算法以及与其它方法的比较;第 3 节是相应的解压器实现;第 4 节给出了具体的实验数据;最后总结全文.

## 2 混合定变长码编码算法

混合定变长码编码方案将代码字分为首部和尾部两个部分,首部使用固定长度的位宽,尾部使用变长位宽;为了进一步提高压缩率,对代码字的尾部采用藏头的方法,因为所有代码字的尾部最高位都是数值 1,所以该 1 不需要编码,只需在解压时自动生成.

### 2.1 混合定变长编码

混合定变长编码是一种不等间距的变长到定长加变长的编码方式. 所谓不等间距编码方式是指在分组时每个组的大小根据其出现的频率进行了适当的调整. 表 1 是定长为 2 的混合定变长码的编码表,第 1 列为游程长度,第 2 列为组号,第 3 列和第 4 列分别为代码字的首部和尾部,第 5 列为原始代码字,第 6 列为将原始代码字的尾部最高位隐藏后的藏头代码字. 从表 1 可以看到从  $A_1$  到  $A_4$ ,每个分组的容量是依次递增的,这样的分组更符合实际数据中游程的分布<sup>[2]</sup>.

表 1 混合定变长码编码表(定长为 2)

游程长度	组号	首部	尾部	代码字	藏头代码字
0	A <sub>1</sub>	01	10	0110	010
1			11	0111	011
2	A <sub>2</sub>	10	100	10100	1000
3			101	10101	1001
4			110	10110	1010
5			111	10111	1011
6	A <sub>3</sub>	11	1000	111000	11000
7			1001	111001	11001
...			...	...	...
12			1110	111110	11110
13	A <sub>4</sub>	00	1111	111111	11111
14			10000	0010000	000000
15			10001	0010001	000001
...			...	...	...
28			11110	0011110	001110
29			11111	0011111	001111

分析 FDR 码表<sup>[2]</sup>,可知 FDR 码表所表示的实际游程长度  $x$ (其所在组数为  $k$ )等于其对应的 FDR 的首部  $a$  与尾部  $b$  之和,即有  $x=a+b$ ,如对于首部和尾部分别为 1110 和 0000 的游程长度为 14 的 FDR 码有  $(1110)_2+(0000)_2=(14)_{10}$ ,考虑到对任一游程的 FDR 码,其首部总是以 111...1110 形式出现,其中 1 的个数比该游程所在组数少 1,而且 FDR 码的尾部长度与首部长度相同.对上述  $x=a+b$  变形有

$$x=(a+2+b)-2=$$
$$(((\underbrace{11\cdots 110}_k)_2+(\underbrace{10}_2)_2)+(\underbrace{b'}_k)_2)-2=$$
$$(((\underbrace{100\cdots 00}_{k+1})_2+(\underbrace{b'}_k)_2)-2)=(\underbrace{1b'}_{k+1})_2-2,$$

即在 FDR 码尾部最高位前面加一位 1 得到编码所代表的游程长度比对应的 FDR 码所代表的实际游程长度多 2. 因此可以使用将 1 和 FDR 尾部级连的方法来编码,可以省去 FDR 首部的开销.

混合定变长编码正是采用了将 1 和 FDR 尾部级连的方法来编码.分析表 1 第 4 列混合定变长码的尾部,发现其正好是将 1 与对应游程的 FDR 尾部级连,因此尾部所表示的数值比原始测试数据的游程长度多 2,如  $(10000)_2-(14)_{10}=(16)_{10}-(14)_{10}=(2)_{10}$ . 通过这个规律,尾部的编码过程就变得非常简单,只用将测试数据的游程长度加上 2 再转化二进制数就可以了.

再一次分析表 1 第 4 列尾部,可以发现尾部的最高位都是数值 1,根据这个规律,我们在编码时就可以将尾部的最高位省去而不必编码,即采用藏头

的方法,藏头的代码字如表 1 第 6 列所示.如上述长度为 6 的游程采用藏头的编码方法,其代码字为 11000.采用藏头的方法可以使每个代码字又节省一位,进一步提高了压缩效率.在解压时,只需在尾部的最高位自动生成数据 1 就可以了.

进一步分析表 1 可知,代码字的首部主要用来区分不同的代码字,解压时,先读入首部(定长为  $k$  位就读入  $k$  位),再根据首部的数据来确定需要再读入的尾部的数据长度,如对于  $k=2$  的游程长度为 6 的代码字:11000,在解压时,先读入首部 11,再由解压控制电路译码知需要再读入的尾部为 3 位,即 000.分析任意藏头的代码字,发现该代码字首部对应值正好是该代码字除去最高位后的长度,如  $A_2$  组首部的值:  $(10)_2=2$ ,而该组代码字所对应的尾部除去最高位后长度也是 2,使用这种代码字的设计是为了使解压非常方便,即后缀的长度直接在代码字中给出.需要指出的是,解压器在读入尾部数据时,应先读入尾部数据再让首部计数器减 1 操作,即让存储首部长度的计数器先做减 1 操作,再判断该计数器是否为 0.这样设计的目的,使  $A_k$  组(定长为  $k$  位就是  $2^k$  组)首部的值也能直接表示需要读入的尾部长度.

需要指出的是首部位宽的选择,决定了测试数据的压缩效率,首部位数过多会导致每个代码字长度增加,不利于测试数据压缩,首部位数过少又会导致代码字不能表示的游程数目增加,即不能用一个代码字来编码的游程数目增加,这样也不利于测试数据压缩.容易得出,对于定长为  $k$  的方案,每个代码字首部总是  $k$  位,而尾部可以有  $2^k$  种不同的组合.本方案所能最多表示的测试数据的游程长度受到首部位宽  $k$  的限制,对于确定的  $k$ ,其最多所能表示的游程长度为  $2^{k+1}-3$ ,对于不能表示的游程则将其划分成多个游程来分别编码,具体划分方法将在 2.2 节叙述.

图 1 所示为一个应用藏头代码字编码的实例,其中代码字的首部位宽为 2.以测试向量中数据 0 的个数作为游程长度进行编码,编码表如表 1 所示,对于图 1 中的一个测试向量片断 0000000001,该片断中数据 0 的游程长度为 9,对其对应的混合定变长码的编码就为 11011,编码后的测试向量片断相对于原测试片断节省了 5 位.

原始测试向量: 0000001 001 1 000000001 0001 0000000001 00001 01 000000001  
编码后测试向量:  $T_E=11000$  1000 010 11010 1001 11011 1010 011 11010  
原始测试向量共 50 位,  $T_E$  共 38 位,节省了 12 位

图 1 测试向量进行藏头代码字编码实例

## 2.2 编码算法的理论分析与首部位宽选择

从编码表可知,混合定变长码根据测试数据中“0”或“1”的游程分布进行分组,  $A_1, A_2, A_3, \dots, A_n$ , 其中  $n=2^k$  ( $k$  为首部位宽), 即分组数由  $k$  确定,  $k$  不同, 分的组数也不同. 对于确定的  $k$ , 所能表示最长游程为  $2^{2^k+1}-3$ . 如果测试数据的游程超过  $2^{2^k+1}-3$ , 则将测试数据分成连续若干个等于  $2^{2^k+1}-3$  的游程和 1 个小于  $2^{2^k+1}-3$  的游程来分别进行编码, 这样划分是为了解压时可以根据当前游程是否为  $2^{2^k+1}-3$  来判断后续游程是否与当前游程在编码前是同一游程. 如果解压时的当前游程和后续游程在编码前属于同一游程, 可以设置一个标志, 即当前游程长度为  $2^{2^k+1}-3$  时就设置一个标志, 这样在解压时可以根据该标志判断当然解压的游程与下一个需要解压的游程是否属于同一个游程, 从而可以避免输出 1 (如果是对 0 类型游程编码) 或 0 (如果是对 1 类型游程编码).

首先, 我们分析一下, 对应于  $k$  的不同取值时, 测试数据的压缩效率变化情况. 为了便于比较, 引入压缩增益  $\beta$  这个概念,  $\beta$  定义为待压缩序列的平均游程长度  $\bar{R}$  与编码后的平均码字长度  $\bar{C}$  之比<sup>[2]</sup>.

测试数据在编码时, 所有无关位全部被填充为确定位 (具体为 0 或 1 由算法决定), 不失一般性, 假设测试数据中 0 出现的概率为  $p$ , 则 1 出现的概率为  $(1-p)$  ( $0 \leq p \leq 1$ ).

对于确定的首部  $k$ , 若  $i < 2^{2^k+1}-3$ , 游程  $i$  出现的概率 (记为  $p_i$ ) 对应于连续出现  $i$  个 0 和 1 个 1 的概率, 即  $p_i = p^i(1-p)$ , 若  $i = 2^{2^k+1}-3$ , 游程  $i$  出现的概率对应于连续出现  $i$  个 0 的概率, 即  $p_i = p^i$ .

$k=1$  时, 压缩增益为

$$\beta_1 = \frac{\bar{R}_1}{\bar{C}_1} = [1 \times (1-p) + 2 \times p \times (1-p) + 3 \times p^2(1-p) + 4 \times p^3 \times (1-p) + 5 \times p^4 \times (1-p) + 5 \times p^5] / [2 \times (1-p) + 2 \times p \times (1-p) + 3 \times p^2(1-p) + 3 \times p^3 \times (1-p) + 3 \times p^4 \times (1-p) + 3 \times p^5],$$

化简得

$$\beta_1 = \frac{1-p^5}{(1-p)(2+p^2)}.$$

同理可求得  $k=2$  时, 压缩增益为

$$\beta_2 = \frac{1-p^{29}}{(1-p)(3+p^2+p^6+p^{14})}.$$

$k=3$  时, 压缩增益为

$$\beta_3 = \frac{1-p^{509}}{(1-p)(4+p^2+p^6+p^{14}+p^{30}+p^{62}+p^{126}+p^{254})},$$

...

$k=i$  时, 压缩增益为

$$\beta_i = \frac{1-p^{2^{2^i+1}-3}}{(1-p)(i+1+\sum_{r=1}^{2^i-1} \sum_{j=1}^r p^{2^j})},$$

...

为了进一步分析  $k$  取不同值时, 压缩增益的变化, 我们画出相关的压缩增益图, 如图 2 所示, 横坐标为概率  $p$ , 纵坐标为压缩增益  $\beta$ . 由图 2 可以看出,  $p < 0.62$  时, 不能压缩数据;  $0.62 < p \leq 0.80$ ,  $k=1$  时压缩效果最好;  $0.80 < p \leq 0.95$ ,  $k=2$  时压缩效果最好;  $p > 0.95$ ,  $k=3$  时压缩效果最好. 实际上随着  $p$  的增加,  $k$  的取值也需要增加才能取得更好的压缩效果, 如  $p > 0.996$  时,  $k$  取 4 时压缩效果就比  $k$  取 3 时压缩效果好, 但考虑到实际情况中,  $p > 0.996$  的概率非常小, 而且即使此种情况,  $k$  取 3 时仍然能取得比较好的压缩效率, 因此图 2 就没有画出  $k > 3$  的情况.

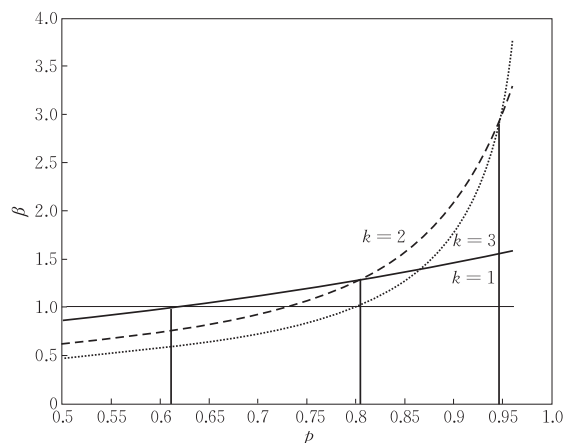


图 2  $k$  取不同值时的压缩增益图

根据以上分析, 我们不难得出  $k$  的选择方法, 即首先统计原始测试数据中 0 和 1 的比例, 若原始测试数据中 0 的数据位少于 1 的数据位, 则将所有无关位指定为 0, 按 0 游程编码; 否则按 1 游程编码. 再由式 (1) 根据测试数据中 0 游程的概率  $p$  来确定. 即如果测试数据中 0 游程的概率  $0.62 < p \leq 0.80$ ,  $k$  取 1; 如果测试数据中 0 游程的概率  $0.80 < p \leq 0.95$ ,  $k$  取 2; 如果测试数据中 0 游程的概率  $p > 0.95$ ,  $k$  取 3; 如果测试数据中 0 游程的概率  $p < 0.62$ , 可以选择 1 游程进行压缩, 其  $k$  值的选择与对应 0 游程时  $k$  值选择类似. 另外, 测试向量中有许多无关位, 可以填充为 1 来增加 1 的概率.

$$k = \begin{cases} 1, & 0.62 < p \leq 0.80 \\ 2, & 0.80 < p \leq 0.95 \\ 3, & 0.95 < p \end{cases} \quad (1)$$

为了更加形象地说明本方案的压缩效率, 可以

进一步画出本方案和 FDR 码的相关增益曲线图,如图 3 所示,横坐标为概率  $p$ ,纵坐标为压缩增益  $\beta$ . 由于 FDR 算法存在着无限求和的因素,画图时取前 1024 项作为近似逼近.

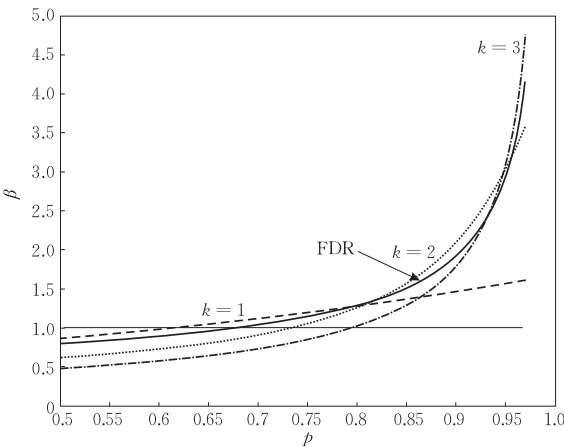


图 3 本方案与 FDR 码的压缩增益曲线比较

从图 3 可以看出,对于不同的概率  $p$ ,本方案总有一段曲线在 FDR 压缩增益曲线上. 因此,对于确定的概率  $p$ ,选择合适的  $k$ ,总可以使本方案的压缩增益效果优于 FDR 码的效果,如对于  $p=0.9$  时,取  $k=2$ ,本方案就可以取得比 FDR 码好的压缩增益.

3 解压电路的设计

传统的解压电路多采用基于有限状态机的电路,该种方式随着状态数的增加,控制电路也变得复杂,硬件开销也将增加. 如果采用有限状态机的方法,混合定变长码的解压电路也可以实现,但随着  $k$  值的增加,电路的状态数将呈指数规律增长,电路硬件开销显著增加.

本方案解压电路的基本框图如图 4 所示. 该解压电路解压编码的测试集为  $T_E$ ,输出为  $T_D$ . 该解压电路结构简单,仅需 3 个计数器(1 个  $\log_2 k$  位计数器,1 个  $k$  位计数器和 1 个  $i=2^k+1$  位计数器( $k$  为首部位宽))和控制单元组成.

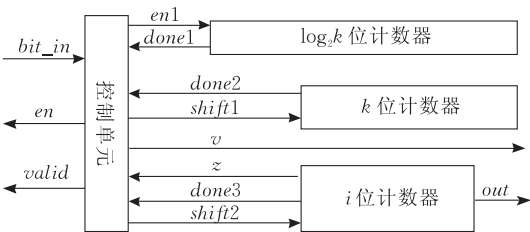


图 4 解压器的结构方框图

该解压器嵌入在芯片上,其结构简单,规模小,不会明显地引入硬件成本,其既独立于被测试电路,又独立于预先计算的测试集.

本方案不同于其它解压器的设计就是采用一种特殊的  $i$  位移位计数器. 该  $i$  位移位计数器相对于传统的移位计数器有两个方面的特征:(1)采用将最低位置数值 1 的方法,且该数值在移位时作为最高位随其它数据一起,向  $i$  位移位计数器高位移位;(2)采用非零下限,传统的计数器向下计数通常计到零为止,而本方案向下计数到数值 2(即计数器内容“10”)时,就开始下一个解码操作. 上述两个功能可以通过简单的组合电路实现,且不会显著增加硬件开销.

上述特殊的  $i$  位移位计数器,主要是用于恢复混合定变长码在编码过程中的尾码最高位的数据 1,首先将 1 直接用硬件的方法存于  $i$  位计数器的最高位,同时在向下计数时只计到数值 2,是因为从编码表中,尾部(包括最高位“1”)对应二进制数比其所代表实际游程长度多 2.

该解压器的基本工作原理为:

(1)首先有限状态机发出使能信号“en”为 1, valid 为低电平,表明输出 out 无效,此时由  $\log_2 k$  位计数器控制读入  $k$  位数据(即首部)移入  $k$  位计数器;

(2)从前面分析,可知此时  $k$  位计数器的数值即为需要读入数据的尾部的位宽. 信号 done1 输出高电平控制数据由 bit\_in 移入  $i$  位计数器,同时  $k$  位计数器开始加 1 计数;

(3)bit\_in 和  $i$  位计数器本身预置的数据 1 一起移向  $i$  位计数器高位,直到  $k$  位计数器向上计数到 0,此时  $k$  位计数器使 done2 为高电平,从而表明尾部已全部移入  $i$  位计数器.

(4)判断此时  $i$  位计数器是否为全 1. 如果为全 1,通过信号  $z$  置  $v$  信号为高电平,否则  $v$  为低电平. 信号  $v$  是用来判断当前游程与下一个游程是否在编码前属于同一游程,通过前面的划分方法可知, $v$  为 1 一定表示下一次游程在编码前与当前游程属于同一游程. 因此可以通过信号  $v$  的标志来确定当前游程结束时输不输出“1”(对 0 游程解压时)或“0”(对 1 游程解压时).

(5)最后由  $i$  位计数器控制 valid 为高电平,表明输出 out 有效,此时输出连续“0”,直至该计数器为数值“2”,这时若  $v$  为高电平,使 en 为高电平, valid 为低电平,否则输出高电平,同时使 en 为高电平, valid 为低电平.

(6)完整的序列解压输出过程与上面类似.

4 实验结果

该部分将通过实验数据来说明本方案中  $k$  的选取和压缩效率. 为了增加实验结果的可比性,本文采用美国 Duke 大学提供的 MinTest 产生的测试向量集(与文献[2]中采用的测试向量数据一致)并分别对 ISCAS 89benchmark<sup>[20]</sup> 电路中几个规模较大的时序电路进行了实验.

由于 MinTest 测试集中 0 的数据位少于 1 的

数据位,故本实验方案全部采用 0 游程编码.

当测试数据中 0 的概率确定时,  $k$  值对压缩效率的影响如表 2 所示. 表 2 第 1 列为 ISCAS 89 标准电路中几个规模较大的时序电路名称,第 2 列为原测试集的数据长度,最后 1 列为测试数据中 0 的概率,中间 6 列分别为  $k$  取不同值时所对应的压缩数据长度和相应压缩率,采用文献[2]的计算公式. 从表 2 可以看出,  $p$  和  $k$  之间的关系能很好地满足式 (1),即  $k$  的选取只需根据该测试数据中 0 出现的概率  $p$  由式(1)来选取.

表 2  $p$  确定时  $k$  值对压缩效率的影响

电路名称	原始数据 位数/bit	$k=3$		$k=2$		$k=1$		$p(0)$
		数据位数/bit	压缩率/%	数据位数/bit	压缩率/%	数据位数/bit	压缩率/%	
s5378	23754	16547	30.34	11366	<b>52.15</b>	15730	33.78	85.11
s9234	39273	25300	35.58	21278	<b>45.82</b>	26325	32.97	87.74
s13207	165200	30435	<b>81.58</b>	45660	72.36	58349	64.68	96.96
s15850	76986	28868	62.50	24868	<b>67.70</b>	30150	60.84	93.08
s38417	164736	115123	30.12	93796	<b>43.06</b>	132695	19.45	82.11
s38584	199104	69608	65.04	55175	<b>72.29</b>	134859	32.27	91.56

本方案与其它方案压缩效率的比较如表 3 所示,可以看出,对同样的测试数据,本方案的压缩效率的要比 Golomb 码<sup>[10]</sup>、FDR 码<sup>[2]</sup>、EFDR 码<sup>[12]</sup>、交替连续码(AARLC)<sup>[13]</sup>等具有明显较好的效果. 在表 3 中,原始的测试数据没有进行差分处理,只是简单地将无关位填充为数据 0,从表 3 可以发现,对于

6 种不同的时序电路,本方案都有不同程度的改善,本方案与 Golomb 编码的方案平均差值压缩效率(两种算法针对 6 个不同电路的压缩率差值的平均值)为 10.81%;与 FDR 码的平均差值压缩效率为 3.22%;与 EFDR 码的平均差值压缩效率为 5.10%;与交替连续码(AARLC)的平均差值压缩效率为 2.01%.

表 3 混合定变长码的压缩方法与其它算法比较(Mintest,无差分)

电路名称	原始数据位数/bit	本方案		Golomb <sup>[10]</sup>	FDR <sup>[2]</sup>	EFDR <sup>[12]</sup>	AARLC <sup>[13]</sup>
		$k$	$\alpha$ /%	$\alpha$ /%	$\alpha$ /%	$\alpha$ /%	$\alpha$ /%
s5378	23754	<b>2</b>	<b>52.15</b>	40.70	48.02	50.81	45.12
s9234	39273	<b>2</b>	<b>45.82</b>	43.34	43.59	37.83	42.79
s13207	165200	<b>3</b>	<b>81.58</b>	74.78	81.30	79.38	80.43
s15850	76986	<b>2</b>	<b>67.70</b>	47.11	66.22	56.29	65.13
s38417	164736	<b>2</b>	43.06	44.12	43.26	52.35	<b>56.52</b>
s38584	199104	<b>2</b>	<b>72.29</b>	47.71	60.91	—	60.57
平均值			<b>60.43</b>	49.63	57.21	55.33	58.43

一些传统的算法采用压缩差分集的方法来提高压缩效果,即原始的测试数据差分后,再进行压缩. 为了进一步比较本方案与其它方法,本方案也使用差分集的方法. 实验结果如表 4 所示. 表 4 中的最后 3 列是对应的 FDR 码、VIHC 码、 $v9C$  码、HRC、SVIC、Variable-Tail 码的改进方案,与它们相比,本方案的压缩效率仍占优势,本方案与差分的 FDR 码的平均差值压缩效率为 13.86%;与差分的 VIHC 码的平均差值压缩效率为 13.54%;与差分的  $v9C$  码的平均差值压缩效率为 15.96%;与差分的 HRC

码的平均差值压缩效率为 14.66%;与差分的 SVIC 码的平均差值压缩效率为 18.86%;与差分的 Variable-Tail 码的平均差值压缩效率为 13.52%.

需要指出的是使用差分集的方法在解压时需要一个与扫描链等长的环形移位寄存器(CSR),这将大幅度牺牲其它性能,显著增加硬件开销. 结合表 3 和表 4,本方案在压缩效果方面相对于传统的算法无论是使用原始测试集还是使用差分集都具有明显的优势.

表 4 混合定变长码的压缩方法与其它算法比较(Mintest,差分)

电路名称	原始数据 位数/bit	本方案 $k=3$	FDR <sup>[2]</sup>	VIHC <sup>[18]</sup>	$v9C^{[19]}$	HRC <sup>[14]</sup>	SVIC <sup>[15]</sup>	Variable-Tail <sup>[16]</sup>
		$\alpha/\%$	$\alpha/\%$	$\alpha/\%$	$\alpha/\%$	$\alpha/\%$	$\alpha/\%$	$\alpha/\%$
s5378	23754	75.52	61.32	60.73	55.75	57.96	55.03	60.34
s9234	39273	80.89	60.63	60.96	54.77	60.85	60.83	62.73
s13207	165200	97.18	87.67	86.83	85.20	86.78	82.21	87.24
s15850	76986	90.75	71.95	72.34	71.26	72.46	65.84	72.76
s38417	164736	73.18	65.35	66.38	62.89	64.75	57.82	65.71
s38584	199104	77.24	64.67	66.29	69.11	64.02	59.92	64.86
平均值		82.46	68.60	68.92	66.50	67.80	63.60	68.94

为了进一步证实本方案的性能,需要验证本方案的硬件开销.不失一般性,以 ISCAS 89 标准中 5378 电路为例来综合其解压电路,假设测试数据的最大游程长度为 1000,将 Golomb 码、FDR 码和本方案电路使用 Synopsys 公司综合工具 Design Compiler,在 TSMC35 库下用 2 输入与非门映射后所得单元数如表 5 所示.从表 5 可以看出本方案在面积开销方面有很大优势.

表 5 硬件开销比较

压缩方案	面积开销(映射到 2 输入与非门)			
Golomb <sup>[10]</sup>	74	125	227	307
FDR <sup>[2]</sup>		320		
本方案	64	73	82	82
相应组数	2	4	8	16

为了进一步验证本方案的性能,需要比较本方案与其它方案的应用时间.显然,随着  $m=f_{\text{SYS}}/f_{\text{ATE}}$  (其中  $f_{\text{SYS}}$  为解压器工作频率,  $f_{\text{ATE}}$  为 ATE 工作频率)的增大,应用时间就越少.不失一般性,设  $f_{\text{ATE}}$  频率为 50MHz,  $f_{\text{SYS}}$  频率为 500MHz.对 Golomb 码、FDR 码和本方案,分别计算其应用时间,如表 6 所示.由表 6 可知,本方案可以取得比 Golomb 码和 FDR 码更少的应用时间.

表 6 不同方案应用时间比较

电路名称	测试应用时间/ms		
	本方案	Golomb	FDR
s5378	0.498	0.503	0.500
s9234	0.828	0.830	0.830
s13207	3.364	3.387	3.366
s15850	1.589	1.621	1.592
s38417	3.481	3.479	3.482
s38584	4.092	4.190	4.138
平均值	2.308	2.335	2.318

综合分析以上实验发现,本方案可以取得较高的压缩效果、较少的硬件开销和较少的应用时间.因此本方案不失为一种优秀的测试数据编码压缩方案.

5 结束语

为了进一步降低测试成本,本文提出了一种混合定变长码的测试数据压缩方案.该方案引入了首部采用固定位宽的方式,位宽的选择只需根据测试数据中“0”或“1”出现的概率,对于确定的设计,首部的位宽是固定的,另外引入将计数器最低位置数值“1”、计数器向下计数到数值“2”的方法,可以有效地减小硬件解码电路的规模和设计复杂度.理论分析和多组实验结果表明:本方案具高压缩率、解压结构简单、硬件开销小等特点,并且具有很好的压缩稳定性.

本方案独立于被测电路,因此特别适合于系统芯片中没有结构信息 IP 的测试.与同类型的编码方案相比较,本方案特别适用于测试集中游程长度相对集中的情况.

参 考 文 献

[1] Touba N A. Survey of test vector compression techniques. IEEE Design & Test of Computers, 2006, 23(4): 294-303

[2] Chandra A, Chakrabarty K. Test data compression and test resource partitioning for system-on-a-chip using frequency-directed run-length(FDR) codes. IEEE Transactions on Computers, 2003, 52(8): 1076-1088

[3] Chandra A, Chakrabarty K. Test resource partitioning for SOCs. IEEE Design & Test of Computers, 2001, 18(5): 80-91

[4] El-Maleh A H, Khursheed S S, Sait S M. Efficient static compaction techniques for sequential circuits based on reverse-order restoration and test relaxation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2006, 25(11): 2556-2564

[5] Liang Hua-Guo, Liu Jun, Jiang Cui-Yun. Constraint input reduction BIST scheme for mulitple scan chains. Journal of



- Computer-Aided Design & Computer Graphics, 2007, 19(3): 371-375(in Chinese)  
(梁华国,刘军,蒋翠云.约束输入精简的多扫描链 BIST 方案.计算机辅助设计与图形学学报,2007,19(3):371-375)
- [6] Xiang D, Zhao Y, Chakrabarty K et al. A reconfigurable scan architecture with weighted scan-enable signals for deterministic bist. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27(6): 999-1012
- [7] Zhang Lei, Liang Hua-Guo, Zhan Wen-Fa et al. Block marking and updating coding in test data compression for SoC//Proceedings of the 16th Asian Test Symposium. Beijing, 2007: 467-472
- [8] Li Lei, Chakrabarty K. Test data compression using dictionaries with fixed-length indices//Proceedings of the 21st VLSI Test Symposium. Napa Valley, 2003: 219-224
- [9] Udar S, Kagaris D. LFSR reseeding with irreducible polynomials//Proceedings of the 13th IEEE International On-Line Testing Symposium. Heraklion, 2007: 293-298
- [10] Chandra A, Chakrabarty K. Test data compression and decompression based on internal scan chains and Golomb coding. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2002, 21(6): 715-722
- [11] Hellebrand S, Wurternberger A. Alternating run-length coding—A technique for improved test data compression//Proceedings of the IEEE International Workshop on Test Resource Partitioning. Baltimore, MD, USA, 2002: 202-207
- [12] El-Maleh A H. Test data compression for system-on-a-chip using extended frequency-directed run-length code. IET Computers & Digital Techniques, 2008, 2(3): 155-163
- [13] Liang Hua-Guo, Jiang Cui-Yun. Efficient test data compression and decompression based on alternation and run length codes. Chinese Journal of Computers, 2004, 27(4): 548-553(in Chinese)  
(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.计算机学报,2004,27(4):548-554)
- [14] Fang Jian-Ping, Hao Yue, Liu Hong-Xia. A hybrid run-length coding for SOC test data compression. Acta Eelectronica Sinica, 2005, 33(11): 1973-1977(in Chinese)  
(方建平,郝跃,刘红侠.应用混合游程编码的 SOC 测试数据压缩方法.电子学报,2005,33(11):1973-1977)
- [15] Hu Bing, Chen Guang-Ju, Xie Yong-Le. System on chip test data compression based on svc coding. Journal of Electronic Measurement and Instrument, 2006, 20(1): 73-78(in Chinese)  
(胡兵,陈光祚,谢永乐.基于 SVIC 编码的 SOC 测试数据压缩.电子测量与仪器学报,2006,20(1):73-78)
- [16] Han Yin-He, Li Xiao-Wei, Xu Yong-Jun et al. Test resource partitioning using Variable-Tail code. Acta Eelectronica Sinica, 2004, 32(8): 1346 - 1350(in Chinese)  
(韩银和,李晓维,徐勇军等.应用 Variable-Tail 编码压缩的测试资源划分方法.电子学报,2004,32(8):1346-1350)
- [17] Peng Xi-Yuan, Yu Yang. A test set compression algorithm based on variable-run-length code. Acta Eelectronica Sinica, 2007, 35(2): 197-201(in Chinese)  
(彭喜元,俞洋.基于变游程编码的测试数据压缩算法.电子学报,2007,35(2):197-201)
- [18] Gonciari P T, Al-Hashimi B M, Nicolici N. Variable-length input Huffman coding for system-on-a-chip test. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2003, 22(6): 783-796
- [19] Tehranipoor M, Nourani M, Chakrabarty K. Nine-coded compression technique for testing embedded cores in SoCs. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2005, 13(6): 719-731
- [20] Brglez F, Bryan D, Kozminski K. Combinational profiles of sequential benchmark circuits//Proceedings of the IEEE International Symposium on Circuits and Systems. Portland, 1989: 1929-1934



**ZHAN Wen-Fa**, born in 1978, Ph.D. candidate, associate professor. His research interests include test data compression, ATPG algorithms and so on.

**LIANG Hua-Guo**, born in 1959, Ph.D., professor,

Ph.D. supervisor. His research interests include synthesis and testing of embedded systems, built-in self test, ATPG algorithms, and distributed control and so on.

**SHI Feng**, born in 1984, M.S. candidate. His research interests include built-in self test and test data compression.

**HUANG Zheng-Feng**, born in 1978, Ph.D. candidate. His research interests include synthesis and testing of embedded systems, design automation of digital systems.

## Background

As the size and the complexity of systems on a chip continue to grow, test data volume has increased dramatically. In order to apply the large volume of test data to a chip under test, the automatic test equipment (ATE) requires large

memory storage and high bandwidth, so resulting in test cost increases.

To reduce test cost, test data compression schemes were proposed. One of the promising category of test data com-



pression schemes uses different coding techniques for compression. Coding, which compresses the data according to the length of runs in the test data, is one of the effective test data compression methods. It involves partitioning the original data into symbols, and then replacing each symbol with a code word to form the compressed data. According to the volume of the symbol in original data and code word in compressed data, the coding schemes can be divided into four types: fixed-to-fixed-length, fixed-to-variable-length, variable-to-fixed-length, variable-to-variable-length.

In this paper, a new test data compression scheme based on fixed and variable length coding (FAVLC) is presented, whose codewords are divided into fixed length heads, which eases the control of decompression, and variable length tails, which adds the feasibility of encoding. Furthermore, the highest bit of the tail is reduced from the code words. In addition, a special shift counter is also used, which further ease the control circuit. Compared with the previous schemes, the proposed scheme can increase the encoding efficiency with

less hardware overhead.

This work is supported in part by the National Natural Science Foundation of China (No. 90407008, No. 60633060) and Natural Science Foundation of Education Agency of Anhui Province of China under grant No. KJ2008B031. The research topic is to develop test data compression techniques so as to reduce test cost.

In the research group, lots of researches have been achieved to solve the problems and many efficient test data compression schemes have been presented. For example:

- (1) Efficient test data compression and decompression based on alternation and run length codes.
- (2) Test data compression scheme based on variable-to-fixed-plus-variable-length coding.
- (3) A deterministic BIST scheme based on reseeding of folding counters.
- (4) Block Marking and Updating Coding in Test Data Compression for SoC.