Journal of System Simulation

Volume 27 | Issue 11 Article 19

8-5-2020

Adjacent Bit XOR Operation Based Test Data Compression Method

Yifei Cheng
1. School of Computer and Information, Anqing Normal College, Anqing 246001, China; ;

Wenfa Zhan
2. Department of Science Research, Anqing Normal College, Anqing 246001, China;

Adjacent Bit XOR Operation Based Test Data Compression Method

Abstract

Abstract: Data compression is a very effective method to reduce the test cost. *An adjacent bit XOR operation based test data compression method* was proposed based on bitwise XOR operation between itself and its previous bit, which turned continuous series, such as a series of all 0s and all 1s into series of all 0s, and reversal series, such as a series of 01 and 10 into series of all 1s by bitwise XOR operation between adjacent bits. On one hand, the two kinds of series, continuous series and reversal series, were both taken into account, which decreased the number of division. On the other hand, it increases the minimum encoding run length, so the minimum encoding run length increases from a conventional 0 to 2. The compression ratio is further improved without additional hardware decoding circuitry overhead. The experimental results illustrate the method has a high data compression ratio.

Keywords

test data compression, XOR operation, run length, FDR(Frequency Directed Run-Length), EFDR(Extended Frequency Directed Run-Length)

第 27 卷第 11 期 系统仿真学报© Vol. 27 No. 11
2015 年 11 月 Journal of System Simulation Nov., 2015

一种基于相邻位异或运算的测试数据压缩方法

程一飞[1]，詹文法[2]
(1. 安庆师范学院计算机与信息学院，安徽 安庆 246001；2. 安庆师范学院科研处，安徽 安庆 246001)

摘要：压缩测试数据是降低测试成本的一种非常有效的手段。提出了一种*基于相邻位异或运算的测试数据压缩方法*，该方法通过将当前位与其前一位进行异或运算，将测试集中 0 游程和 1 游程转换成 0 游程，01 和 10 交替序列转换成 1 游程，对转换后的游程进行编码。该方法可以减少测试集划分数量；也可以增加最短可编码的划分长度，即可编码的最短划分长度由传统的 0 变成 2，从而达到在不额外增加解码电路硬件开销的前提下进一步提高压缩率。用实验验证了本方法具有极高的压缩效率。

关键词：测试数据压缩；异或运算；游程；FDR(Frequency Directed Run-Length)；EFDR(Extended Frequency Directed Run-Length)
中图分类号：TP391.76 文献标识码：A 文章编号：1004-731X (2015) 11-2756-06

Adjacent Bit XOR Operation Based Test Data Compression Method

Cheng Yifei[1], *Zhan Wenfa*[2]
(1. School of Computer and Information, Anqing Normal College, Anqing 246001, China;
2. Department of Science Research, Anqing Normal College, Anqing 246001, China)

Abstract: Data compression is a very effective method to reduce the test cost. *An adjacent bit XOR operation based test data compression method* was proposed based on bitwise XOR operation between itself and its previous bit, which turned continuous series, such as a series of all 0s and all 1s into series of all 0s, and reversal series, such as a series of 01 and 10 into series of all 1s by bitwise XOR operation between adjacent bits. On one hand, the two kinds of series, continuous series and reversal series, were both taken into account, which decreased the number of division. On the other hand, it increases the minimum encoding run length, so the minimum encoding run length increases from a conventional 0 to 2. The compression ratio is further improved without additional hardware decoding circuitry overhead. The experimental results illustrate the method has a high data compression ratio.

Keywords: test data compression; XOR operation; run length; FDR(Frequency Directed Run-Length); EFDR(Extended Frequency Directed Run-Length)

引言

伴随着集成电路技术蓬勃发展而来的是对 SoC 测试的数据量急剧增加，测试数据急剧增加会导致 ATE(automatic test equipment)的存储容量和传输带宽就会成为测试的瓶颈。测试数据压缩是通过对原始测试集进行压缩，减少了需要存储和传输的数据量，因此它是解决这一问题的非常有效的方法。

基于游程的编码方法是测试数据压缩技术性价比非常高的一类，它以较小的硬件开销换来比较高的压缩率，Golomb[1]编码、FDR(Frequency Directed Run-Length) [2]编码及 AFR(Alternating Frequency

收稿日期：2014-04-25 修回日期：2014-06-20；
基金项目：国家自然科学基金项目(61306046)；
作者简介：程一飞(1976-)，男，安徽怀宁，硕士，副教授，研究方向为测试数据压缩；詹文法(1978-)，男，安徽怀宁，博士，教授，研究方向为测试数据压缩。

Run-length) [3]编码等都是基于游程的编码，国内梁华国提出的交替和连续长度编码[4]、韩银和等提出的 Variable-Tail 编码[5]、彭喜元提出的变游程编码[6]、詹文法提出的混合定变长码[7]等也都是基于游程的编码。其中 Golomb 码和 FDR 码仅对 0 游程进行编码，并未对连续的 1 序列进行编码，因此对未差分的测试向量压缩率不高；AFR 码由于同时对 0 游程和 1 游程编码，因此其压缩效率比 Golomb 码和 FDR 码有所提高；交替和连续长度编码对交替序列和连续序列交替进行编码，若同时出现交替序列或者连续序列则在它们之间插入分隔符，因此编码方法不够灵活；混合定变长码编码前缀长度是由参数 k 确定，一旦 k 确定了，所有组前缀长度即确定，而为了获取更好的压缩率就必须反复实验确定 k 的值。因此基于游程的编码技术中若充分考虑 01 或者 10 序列并给出可靠的解决方案，均能在不明显增加解码硬件开销的前提下进一步提高压缩率。

本文提出编码方法核心思想是采用相邻位异或运算的手段对测试集进行变换。通过将当前位与其前一位进行异或运算，一方面可以减少需要编码的划分数量；另一方面可以增加最短可编码的划分长度，即最短可编码的划分长度由传统的 0 变成 2，从而能够达到在不额外增加解码电路硬件开销的前提下进一步提高压缩效率。后文中实验结果也能验证本方法的数据压缩效率与国内外研究成果相比具有极强的优势。

1 基于相邻位异或的编码方法

分析文献[1-7]中的码表可知，变长－变长编码的压缩增益随着游程长度增加而增加，但对于短游程特别是长度为 0 或 1 的游程压缩增益可能还是负数，为此我们想尽量提高测试集中最短划分长度。

通过分析异或运算真值表可知，相同的两个数异或结果得 0，而不同的两个数异或结果得 1，因此对于 $\underbrace{01...01}_{k}$ 或者 $\underbrace{10...10}_{k}$ 类型的序列通过相邻位异或可以得到 $\underbrace{11...11}_{k}$，而对于 $\underbrace{0...0}_{k}$ 或者 $\underbrace{1...1}_{k}$ 类型的序列通过相邻位异或可以得到 $\underbrace{0...0}_{k}$。其中 $\underbrace{01...01}_{k}$ 或者 $\underbrace{10...10}_{k}$ 类型序列在双游程编码时 01 或者 10 就是一个短游程，通过相邻位异或运算将若干个 01 交替的序列或者 10 交替的序列变成了一个游程，因此能够达到增加最短划分长度、减少总划分数目的目的，最终实现提高压缩效率的目标。

本文中所述 01 序列是指起始位为 0，后续位 0/1 交替出现，并且最后位与其前一位相同，如 011，0100。10 序列定义与 01 序列类似，如 100，1011 均为 10 序列。

具体实现时我们通过在 10 起始的序列和 0 游程前添加一位默认值 0，01 起始的序列和 1 游程前添加一位默认值 1，然后对相邻两位依次异或最终变成一个 0 游程或者 1 游程。如原始数据为 101011，则在其前添加 0 变成 0101011，然后相邻位进行异或运算变成 111110，而原始数据 0100，首位添加 1 变成 10100，相邻位进行异或运算得到 1110。

再分析三位二制数，其中 000 和 111 至少是长度为 3 的游程；001 和 110 是长度为 2 的游程；011 在其前添加 1 变成 1011，然后通过相邻位异或可得到 110，是长度为 2 的游程，100 可通过在其前添加 0 变成 0100，然后通过相邻位异或可得到 110，也是长度为 2 的游程。010 和 101 不论后一位是 0 还是 1 通过相邻位异或运算得到的总是长度超过 3 的游程，因此本方案中最短划分长度为 2。

由于游程类型有两种，并且需要添加的默认位也有两种情况，因此编码时我们首先用两位来分别标记通过相邻位异或运算后得到的游程类型和起始位前添加的默认位。整个代码字由 3 个部分组成：游程类型、默认位、划分长度对应码字。其中划分长度对应码字表按如下规则构建：(1)码字由前缀和后缀组成，前缀表示码字所在的组，后缀表示码字在本组内的序号；(2)码字前缀可变，即前缀有 $\underbrace{0...0}_{k}1$ 和 $1\underbrace{...1}_{k}0$ 两类，其中 k 表示组号；(3)后缀长度等于组号 k。依此方法构建码表，码表所需组数 K 可由表示划分最长长度 L 计算得到，具体计算公

第 27 卷第 11 期 系统仿真学报 Vol. 27 No. 11
2015 年 11 月 Journal of System Simulation Nov., 2015

式：K＝[$\log_2^{L+3}-2$]。划分长度对应码字表见表 1。

表 1 基于相邻位异或编码码字表

组号	划分长度	前缀	后缀	码字
1	2	01	0	010
	3		1	011
	4	10	0	100
	5		1	101
2	6	001	00	00100
	7		01	00101
	8		10	00110
	9		11	00111
	10	110	00	11000
	11		01	11001
	12		10	11010
	13		11	11011
3	14	0001	000	0001000
	…		…	…
	21		111	0001111
	22	1110	000	1110000
	…		…	…
	29		111	1110111
…	…	…	…	…

对于给定测试集 T_D，依据本文编码方法编码过程如下所示。

T_D＝0101000000001010101010100000000011011

划分 010100 0000001 01010101010100 0000001 1011

添加默认位 1010100 00000001 101010101010100 00000001 01011

相邻位异或 111110 0000001 11111111111110 0000001 1110

对应游程长度 5 6 13 6 3

T_E 11101 0000100 1111011 0000100 10011

2 编码算法

上述实例中 T_D 是一个全部由确定位构成的测试集实例，所以压缩率不高。而由于实际测试集中通常包含大量无关位，并且因为对无关位的任意赋值并不影响故障覆盖率，因此可以通过无关位启发式赋值进一步提高压缩率。下面给出本文编码算法。

(1) 将测试向量首尾相连；res[0]–res[3]初始化，即分别将 res[0][0]–res[3][0]设置默认位；

(2) i=0；

(3) 以当前点 i 作为起点，获取最长能取得的 0 游程，将结果存储到 res[0]中，并记录终点位置到 pos[0]中；

(4) 以当前点 i 作为起点，获取最长能取得的 1 游程，将结果存储到 res[1]中，并记录终点位置到 pos[1]中；

(5) 以当前点 i 作为起点，获取最长能取得的 01 序列，将结果存储到 res[2]中，并记录终点位置到 pos[2]中；

(6) 以当前点 i 作为起点，获取最长能取得的 10 序列，将结果存储到 res[3]中，并记录终点位置到 pos[3]中；

(7) 计算 pos[0]到 pos[3]中最大值的序号 index；

(8) 将 res[index]相邻位异或得到 0 或者 1 游程，记录游程类型并记录游程长度；

(9) 获取划分对应码字；

(10) 输出游程类型、添加的默认位和码字；

(11) 将 i 置为 pos[index]+1；

(12) 重复 3~11 直到数据编码完成。

3 理论分析

下面用概率论的方法来分析本文方法的压缩效率。

设 0 出现的概率为 $p(0\leqslant p\leqslant 1)$，则该位为 1 的概率为 1–p，长度为 l 的 0 游程出现的概率 P(0)为 $p^l(1-p)$，长度为 l 的 1 游程出现的概率 P(1)为 $(1-p)^l p$，长度为 l 的 10 序列出现的概率 P(10)为

$$\begin{cases} p^{\frac{l}{2}+1}(1-p)^{\frac{l}{2}}(l\text{为偶数}) \\ p^{\left\lfloor\frac{l}{2}\right\rfloor}(1-p)^{\left\lceil\frac{l}{2}\right\rceil+1}(l\text{为奇数}) \end{cases}$$

长度为 l 的 01 序列出现的概率 $P(01)$为

$$\begin{cases} p^{\frac{l}{2}}(1-p)^{\frac{l}{2}+1}（l为偶数） \\ (1-p)^{\left\lfloor\frac{l}{2}\right\rfloor}p^{\left\lceil\frac{l}{2}\right\rceil+1}(l为奇数) \end{cases}$$

已知第 k 组所表示的划分范围为 $2^{k+1}-3<l_k\leqslant 2^{k+2}-3$，则长度为 l 的划分出现在 k 组的可能 $P(l,k)$可由下式计算得到。

$$P(l,k)=\sum_{l=2^{k+1}-2}^{2^{k+2}-3}[P(0)+P(1)+P(10)+P(01)]=$$
$$\sum_{l=2^k-1}^{2^{k+1}-2}[p^{2l}(1-p)+(1-p)^{2l}p+p^{l+1}(1-p)^l+$$
$$p^l(1-p)^{l+1}]+\sum_{l=2^k-1}^{2^{k+1}-2}[p^{2l+1}(1-p)+$$
$$(1-p)^{2l+1}p+p^l(1-p)^{l+2}+p^{l+2}(1-p)^l]$$

第 k 组码字长度为 $2k+3$，则压缩后编码的平均长度

$$\overline{A}=\sum_{k=1}^{\infty}(2k+3)P(l,k)=\sum_{k=1}^{\infty}(2k+3)$$
$$\{\sum_{l=2^k-1}^{2^{k+1}-2}[p^{2l}(1-p)+(1-p)^{2l}p+$$
$$p^{l+1}(1-p)^l+p^l(1-p)^{l+1}]+$$
$$\sum_{l=2^k-1}^{2^{k+1}-2}[p^{2l+1}(1-p)+(1-p)^{2l+1}p+$$
$$p^l(1-p)^{l+2}+p^{l+2}(1-p)^l]\}$$

而原始数据的平均划分长度为

$$\lambda=\{1+\sum_{l=1}^{\infty}lp^l(1-p)\}+\{1+\sum_{l=1}^{\infty}lp(1-p)^l\}+$$
$$\{1+\sum_{l=1}^{\infty}2lp^{l+1}(1-p)^l+\sum_{l=1}^{\infty}2lp^l(1-p)^{l+1}\}+$$
$$\{1+\sum_{l=1}^{\infty}(2l+1)p^l(1-p)^{l+2}+\sum_{l=1}^{\infty}(2l+1)p^{l+2}(1-p)^l\}$$

因此本文方法编码的压缩增益为

$$\beta_P=\frac{\lambda}{\overline{A}} \quad (1)$$

FDR 码的压缩增益为

$$\beta_F=\frac{1}{2(1-p)\sum_{k=1}^{\infty}(p^{2^k-2})} \quad (2)$$

AFR 码的压缩增益为

$$\beta_A=\frac{1}{2p(1-p)\left[\sum_{k=1}^{\infty}(p^{2^k-2})+\sum_{k=1}^{\infty}((1-p)^{2^k-2})\right]} \quad (3)$$

因上面 3 个式子中的 k 都以指数形式存在于表达式中，则 $k\in[1, 999]$就可以得到足够精度。图 1 是 FDR 码、AFR 码以及本文方法三种编码方法的压缩增益比较。纵轴表示压缩增益 β，横轴表示某一位取 0 的概率 $p(p\in[0, 1])$。从表中可以看出本文方法的压缩增益都好于 FDR 码和 AFR 码的压缩增益。主要原因是采用本文的编码方法，不管 0 游程多还是 1 游程多，都能取得很好的压缩效果。

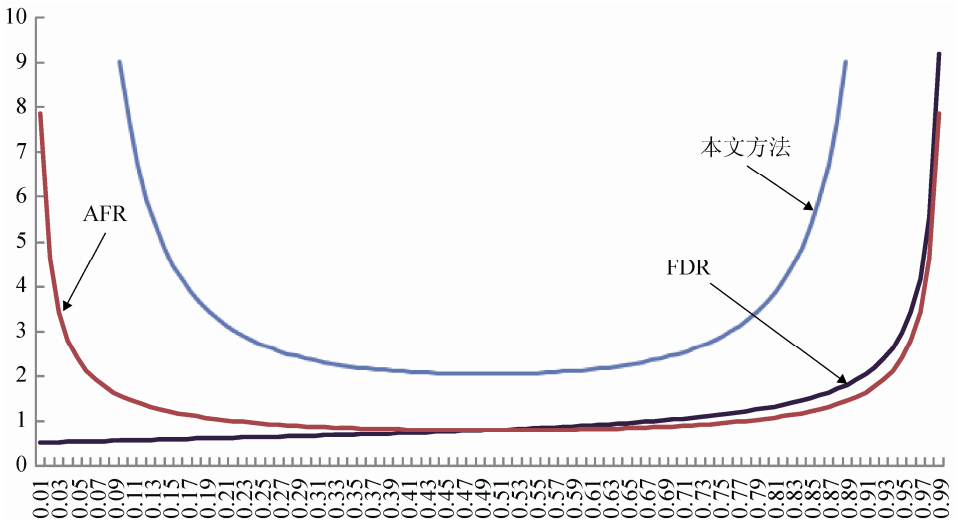

图 1 本文方法、FDR 及 AFR 压缩效率比较

第 27 卷第 11 期 系统仿真学报 Vol. 27 No. 11
2015 年 11 月 Journal of System Simulation Nov., 2015

4 解压分析

分析表 1 可知，划分长度 i=1+前缀起始位+所有后缀组成的二进制数换算为十进制–2，即 i=$(1Xt)_2$–2，其中 X 为对应编码前缀起始位。如游程长度 i=6，对应的码字为 00100，$(1000)_2$–2=8–2=6；游程长度 i=10，对应的码字为 11000，$(1100)_2$–2=12–2=10。因此可以使用一个特殊的 k+2 位计数器控制输出序列的长度。该计数器的初值置为 1，当编码输入时，将 1 和前缀起始位以及后缀依次向高位移位得到一个 k+2 位二进制数。通过该计数器减 1 操作控制输出序列的长度，直到该计数器的值为十进制 2(二进制 10 时)输出结束。而序列的具体形式是连续 0、连续 1、01 交替还是 10 交替则由输入的游程类型以及起始处添加的默认位来确定。

此于上述分析，我们设计一个由一个 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构成的解码器，该解码器结构简单，独立于被测电路且大小可变，解压器结构框图见图 2，其中的信号名称及其对应功能描述见表 2。

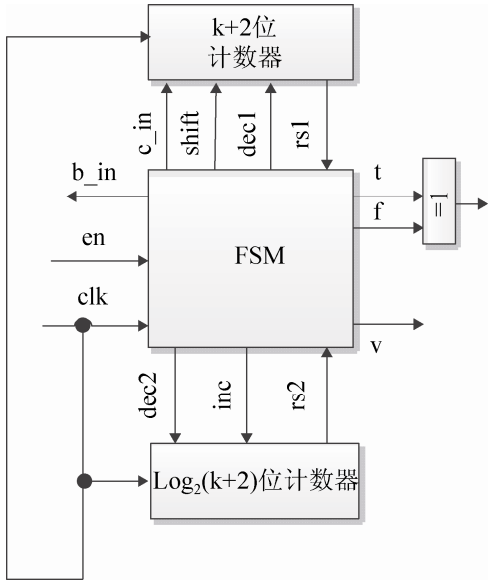

图 2 解压电路框图

表 2 解码器中信号及其对应功能

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
f	游程标志位，0 表示 0 游程，1 表示 1 游程
t	当前期望值标志
out	FSM 输出的解码数据

下面结合框图介绍本方法解码基本过程。

(1) 初始化 en=1；

(2) FSM 读入 1 位，将其输出给 f，表示游程类型；

(3) FSM 读入 1 位，将其输出给 t；

(4) FSM 接收编码的前缀并将前缀首位移入到 k+2 位计数器；每接收 1 位前缀，inc 加 1。

(5) 将尾部移入至 k+2 位计数器，dec2=1，表示每进入一位 log2(k+2)位计数器减 1，直到 log2(k+2)位计数器减为 1；

(6) k+2 位计数器进行减 1 操作。k+2 位计数器每减 1，f 值不变，输出 t 值为 f 与 t 异或结果，v=1，直到 k+2 位计数器减为 2；

(7) 将 t 与 f 异或得到最终输出 out；

(8) t＝0 时输出非 f，t=1 时继续输出上一个 f；

(9) 重复 2-8 直到解码结束。

5 实验结果

为了验证和比较，我们同样将本方法应用到 ISCAS-89 标准电路中六个规模较大的电路，同样采用 Mintest ATPG 预先计算的测试集。

首先将我们的方法与 FDR 以及 EFDR 从划分数目方面进行比较，如表 3。第 1 列是电路名称，第 2~4 列分别是本文方法的划分数、FDR 的划分数及 EFDR 的划分数。

表 3 各种方法划分数比较

电路名称	本文方法	FDR	EFDR
S5378	1654	3537	2297
S9234	2639	4816	3600
S13207	2977	5021	3880
S15850	3004	5329	3819
S38417	8678	29473	10650
S38584	8661	16814	11663
平均	4602	10832	5985

从表 3 可以看出本方法的划分数要远少于其它两种方法，因此它能进一步提高压缩率。具体各种方法压缩率比较见表 4，其中第 1 列是电路名称，第 2 列是该电路原始大小，第 3~8 列分别是 Golomb 码、FDR 码、交替连续长度码、混合定变长码、基于交替-连续长度码[8]的测试数据编码方法以及本文方法压缩效果。

表 4 各种方法压缩效果比较

电路名称	原始长度	Golomb码	FDR码	交替连续长度码	混合定变长码	交替连续长度码	本文方法
S5378	23 754	40.7	48.02	50.77	52.15	45.12	52.91
S9234	39 273	43.34	43.59	44.96	45.82	42.79	51.20
S13207	165 200	74.78	81.3	80.23	81.58	80.43	83.69
S15850	76 986	47.11	66.22	65.83	67.70	65.13	70.11
S38417	164 736	44.12	43.26	60.55	43.06	56.52	60.84
S38584	199 104	47.71	60.91	61.13	72.29	60.57	66.64
平均		49.63	57.22	60.58	60.43	58.43	64.23

分析表 4 可知，本文方法与 Golomb 码压缩效果平均差是 14.60%，比交替连接长度码压缩效果也提高了 3.65%。基于异或的测试数据压缩方法要好于其它方法的原因有：(1)减少了划分数目；(2)提高了最短划分长度。

6 结论

为了降低测试数据量而提出了基于异或运算的测试数据压缩方法。该方法通过相邻位异或将 01 序列(10 序列)变换成一个码字，这样不仅减少了划分数目，而且提高了最短划分长度，从实验结果看它能进一步提高压缩率。通过解码分析，该方法解码结构额外增加的硬件开销不大，并且与被测试电路无关。因此该方法具有极好的压缩率硬件开销比。

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 (S0278-0070), 2002, 21(6): 715-722.

[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. 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. Washington DC, USA: IEEE Computer Society, 2002: 673-678.

[4] 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.

[5] 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 [J]. 电子学报, 2004, 32(8): 1346-1350.

[6] 彭喜元, 俞洋. 基于变游程编码的测试数据压缩算法 [J]. 电子学报, 2007, 35(2): 197-201.

[7] 詹文法, 梁华国, 时峰, 等. 混合定变长码的测试数据压缩方案 [J]. 计算机学报, 2008, 31(10): 1826-1833.

[8] 项莉萍, 梁华国, 刘杰, 等. 基于交替-连续长度码的测试数据编码方法 [J]. 计算机工程, 2010, 36(3): 277-279.