

文章编号:1674-2974(2016)02-0130-05

# 一种基于 IFDR 改进的测试激励数据压缩方法<sup>\*</sup>

尤志强<sup>†</sup>, 罗奇钧

(湖南大学 信息科学与工程学院, 湖南 长沙 410082)

**摘要:**通过改进 IFDR 码, 提出一种基于游程相等编码的改进 FDR(ERFDR)方法. 首先, 该方法不仅能同时对原测试集的 0 游程和 1 游程进行编码, 而且, 当相邻游程相等时还可以用较短的码字来代替, 从而进一步提高了压缩率. 其次, 还提出针对该压缩方法的测试集无关位填充算法, 增强提出方法的压缩效果. 实验结果表明, 与 FDR, EFDR, IFDR 和 ERLC 相比较, 本文提出的方法获得了更高的压缩率, 降低了测试费用.

**关键词:**全扫描测试; 测试数据压缩; 无关位; FDR 编码

**中图分类号:**TP302

**文献标识码:**A

## A Test Compression Method Based on IFDR Code

YOU Zhi-qiang<sup>†</sup>, LUO Qi-jun

(College of Information Science and Engineering, Hunan Univ, Changsha, Hunan 410082, China)

**Abstract:** Based on equal runlength code and IFDR, a new coding method (called ERFDR) was proposed. Firstly, the proposed method can not only encode both 0 and 1 runs for a test set simultaneously, but also can use shorter code if the adjacent runlengths are equal. Therefore, the compression ratio can be further improved. This paper also put forward a new filling algorithm for a test set with don't care bits, which can enhance the compression efficiency of the proposed method. Experimental results show that the proposed method can obtain a higher compression rate compared with FDR, EFDR, IFDR and ERLC codes. The test cost can be reduced effectively.

**Key words:** full scan testing; test compression; don't care bits; FDR codes

随着超大规模集成(VLSI)电路制造工艺的不断进步,越来越多的知识产权(IP)核被集成到一个系统芯片(SoC)上,与 SoC 相关的可测试性和测试方法问题被摆到了重要的位置. 近十年来,如何降低测试成本,减少测试应用时间,降低测试功耗成为了研究的热点问题.

数据压缩技术能较好地解决这个问题,而编码

压缩又是众多压缩方法中较好的一种. 当前比较成熟的编码压缩方法有字典编码<sup>[1]</sup>、游程编码<sup>[2]</sup>、Huffman 码<sup>[3]</sup>、Golomb 码<sup>[4-5]</sup>、FDR 码<sup>[6]</sup>、EFDR 码<sup>[7-8]</sup>、IFDR 码<sup>[9]</sup>等. 这些编码压缩方法充分利用了测试集中的无关位(X). FDR 是一种变长 0 游程编码,测试集中的 X 都被填充为 0 以增加 0 游程的长度,当测试集中 1 的个数较少时有较好的压缩效

<sup>\*</sup> 收稿日期:2014-12-08

基金项目:新世纪优秀人才支持计划项目(NCET-12-0165)

作者简介:尤志强(1972-),男,河北阜城人,湖南大学副教授

<sup>†</sup> 通讯联系人, E-mail:469363963@qq.com

果. EFDR 码和 IFDR 码可以同时 对 0, 1 游程进行编码, 当测试集中 1 的个数较多时, 也能取得较好的压缩效果. 然而以上方法均没考虑等游程的情况. 本文在 IFDR 上进行改进, 提出一种基于游程相等的改进 FDR(ERFDR), 一方面能同时对 0, 1 游程编码, 另一方面当相邻游程相等时用较短的码字来代替, 以进一步提高压缩率, 减少测试应用时间.

### 1 IFDR 编码

IFDR 编码是一种改进型 FDR 编码(Improved FDR). 该方法将原测试集看作连续的 0 游程和 1 游程, 0 游程和 1 游程共用同一套码字, 并规定 0 游程后接 1 游程, 1 游程后接 0 游程. 若不是, 即 0 游程后是 0 游程或者 1 游程后是 1 游程, 编码时在两个相同游程中间添加一个“00”作为标识符. 该方法默认从 1 游程开始编码, 若测试集第一位为 0, 则在编码的过程中先必须加个“00”作为标识. 表 1 给出了 IFDR 的编码表, 可以看出游程长度  $l$  和其所在组  $k$  的关系为:  $k = \lceil \log_2(l+3) \rceil - 1$ . 前缀中 1 的个数和其所在组的关系为:  $k$  组的前缀为  $1^{k-1}0$ , 表示有  $k-1$  个 1 再接一个 0. 对于任一组, 前缀和尾部的长度是相等的, 组前缀是用来区分该码字所在的组(通过前缀的长度), 尾部用来确定该码字所在组中的位置. 和 FDR 编码表不同的是 IFDR 编码表的  $A_1$  组只包含一个游程长度, 且没有长度为 0 的游程. 用 IFDR 对 0 游程和 1 游程编码时共用同一套编码.

表 1 IFDR 编码表  
Tab. 1 IFDR coding table

组	游程长度	组前缀	尾部	码字
A <sub>1</sub>	1	0	1	01
	2		00	1 000
	3		01	1 001
A <sub>2</sub>	4	10	10	1 010
	5		11	1 011
A <sub>3</sub>	...	...	...	...

### 2 ERFDR 码

为了进一步提高测试压缩率, 本文在 IFDR 基础上进行改进, 提出一种基于游程相等的改进 FDR(ERFDR), 编码表见表 2. 与 IFDR 码类似, ERFDR 码也能同时对 0 游程和 1 游程编码, 0 游程和 1 游程共用同一套码字, 且默认从 1 游程开始编码. 考虑到相邻游程类型相同的可能性较高, 与 IFDR 码不

同, 本文用一位“0”作为标识, 则可多压缩一位. 代价是游程长度为  $2^{n+1}-3$  的编码增长 2 位, 其中  $n$  为自然数. 当这种游程个数小于相邻游程类型相同的游程个数的一半时, 该编码方式有效. 可以预见, 当测试集中确定位比例越低, 该编码方式越有效. 进一步考虑到, 每个游程的编码都是从 1 开始, 且不会连续出现两个“0”标识符. 我们可以用“00”标识相邻两个游程相等的情况, 从而取得进一步的压缩效果. 在不发生混淆的前提下, 本文用“0000”标识相邻游程类型相同且相等的情况.

总之, 提出的方法有如下 5 个编码原则: 1) 若测试集第一个游程为 0 游程, 须加“0”作为标识; 2) 当相邻游程类型相同但游程不相等时, 在两游程的编码之间加“0”标识; 3) 当相邻游程长度相等且类型不同时, 后一个游程用“00”编码; 4) 当相邻游程长度相等且类型相同时, 后一个游程用“0000”编码; 5) 为了避免解码时发生歧义, 当出现连续 3 个游程长度相等时, 则对第 3 个游程直接用编码表编码, 而不使用原则 3) 和 4) 编码.

如表 2 所示, 全部游程长度分为  $A_1, A_2, \dots, A_{k_{\max}}$  组, 其中  $k_{\max}$  由测试集中最大游程长度  $l_{\max}$  决定, 满足不等式  $2^{k_{\max}+1}-4 < l_{\max} \leq 2^{k_{\max}+2}-4$ . 游程长度  $l$  和其相对应组号  $k$  的关系为:  $k = \lceil \log_2(l+4) \rceil - 2$ .  $A_k$  组包含  $2^{k+1}$  个游程, 如  $A_1$  组包含 1, 2, 3, 4 四个游程. 另外,  $A_k$  组的前缀和尾部的长度相等, 且都等于  $k+1$ , 其编码长度为  $2k+2$ .

表 2 ERFDR 编码表  
Tab. 2 ERFDR coding table

组	游程长度	组前缀	尾部	码字
A <sub>1</sub>	1		00	1 000
	2	10	01	1 001
	3		10	1 010
	4	11	1 011	
A <sub>2</sub>	5		000	110 000
	6		001	110 010
	7		010	110 010
	8	110	011	110 011
	9		100	110 100
	10		101	110 101
	11		110	110 110
	12	111	111	
...	...	...	...	

例 1 下面是 31 位的测试向量, 分别用几种编码方法求 111011111100000000011111111110 的编码.

用 FDR 编码:

00000001101111001100000000000000000001  
共 40 位,比原向量位数还多;

用 EFDR 编码:110001101101100101110010 共  
24 位;

用 IFDR 编码:100100110000110011110011 共  
24 位.

而用本方法编码:1010 0 110001 110100 00 共  
19 位,可见本方法比上面的几种方法压缩效果都有  
改善.

### 3 无关位的填充方法

大规模测试数据中无关位占 95%以上,测试数  
据压缩效果的好坏在一定程度上取决于对 X 的填  
充.本方法对 0,1 同时编码,并充分利用游程长度和  
类型信息进一步提高测试压缩率,在对无关位填充  
的过程中应遵循下列两个基本原则:

- 1) 尽量使用长游程编码;
- 2) 尽可能地让相邻游程相等.

例如:00XX00X00XXXXXXXX11XX11XX10 这  
样一组测试数据,若在填充过程中仅仅遵循原则  
1),则填充后为 00000000000000001111111110,用  
表 2 编码,结果为 11100011 110011,共 14 位.若遵  
循上述两原则:0000000000001 111111111110,用  
本文的方法编码为 110111 00 共 8 位,减少了 6 位,  
压缩效果明显改善.本文所用的填充算法(无关位填  
充算法)如下.

输入:带无关位 X 的测试集  $T_0$ ;

输出:不含无关位 X 的测试集  $T_1$ ;

目标:采用 ERFDR 方法编码长度较短.

STEP1: 从  $T_0$  中按先后顺序选取测试数据  
片段.

首先,标记开始的位置为  $P_0$ ;从  $P_0$  开始寻找第  
一个确定位,标记该位置为  $P_1$ ;接着从  $P_1$  开始找到  
第一个与  $P_1$  所在位不同的确定位,记这个位置为  
 $P_2$ ;再从  $P_2$  开始寻找第一个与  $P_1$  所在位相等的确  
定位,记该位置为  $P_3$ ;接着按相反的顺序从  $P_2$  向  $P_0$   
方向寻找第一个确定位,记为  $P_4$ ;从  $P_3$  向  $P_0$  的方  
向寻找第一个确定位,记为  $P_5$ .从  $P_0$  到  $P_3$  即为一块数  
据片段.此外,再从  $P_3$  往正方向寻找第一个确定位,  
记该位置为  $P_6$ ,从  $P_3$  往正方向寻找第一个与  $P_3$   
不同的确定位,记为  $P_7$ .图 1(a)为数据片段选取实例.

STEP2: 等游程判断.

我们记  $P_0$  到  $P_4$  间的位数(包括  $P_0$  和  $P_4$ )为  $n$ ,

$P_4$  到  $P_2$  间的位数(不包括  $P_4$  和  $P_2$ )为  $h$ , $P_2$  到  $P_5$  间  
的位数(包括  $P_5$  但不包括  $P_2$ )为  $m$ , $P_5$  到  $P_3$  间的位  
数(不包括  $P_5$  和  $P_3$ )为  $r$ .从图 1(a)中选取的测试片  
段可以看出, $P_0$  到  $P_3$  是否可以填充为相等游程取  
决于对  $h$  和  $r$  中 X 的合理填充能否满足要求.若  $P_3$   
所在位的值和  $P_6$  的相等且  $n$  满足关系式  $m - h \leq n$   
 $\leq m$  或  $m < n \leq m + h + r$ ,则  $P_0$  到  $P_3$  可分为两相等  
游程,其中当  $n + h \geq m + r$  时两相等游程长度  $L$  的  
最好情况解为  $\lceil (n + m + h + r) / 2 \rceil$ ,这时将  $P_3$  调整  
为  $P_3 = P_0 + 2 \times (\lceil (n + m + h + r) / 2 \rceil + 1) - 1$ .当  
 $n + h < m + r$  时, $L$  的最好情况解为  $n + h$ ,将  $P_3$  调整  
为  $P_3 = P_0 + 2 \times (n + h + 1) - 1$ .

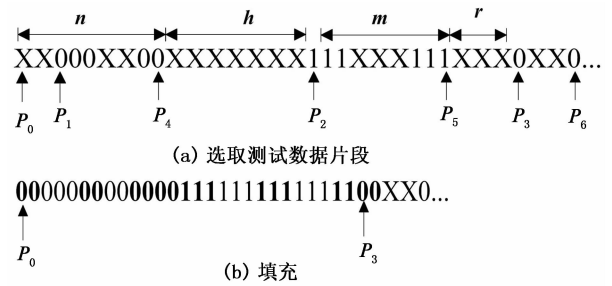


图 1 一测试向量数据片段填充实例  
Fig. 1 Example for X-filling  
of a segment of test data

若  $P_3$  所在位的值和  $P_6$  所在位的值不相等或  $n$   
不满足上述关系,转到 STEP4.

若  $P_2$  与  $P_5$  重合,将  $P_3$  调整到  $P_7$  的位置,并令  
 $P_8 = 2 \times P_2 + 1 - P_0$ .若  $P_3 < P_8 \leq P_7$  且  $P_8$  所在位  
的值和  $P_3$  所在位的值不相同,则将  $P_3$  调整到  $P_8$  所  
在的位置.若  $P_8$  不满足上述关系,转到 STEP4.

STEP3: 等游程填充.

根据游程相等情况填充 X,转到 STEP5.

STEP4: 游程长度不等时的填充.

令从  $P_0$  到  $P_4$  中的所有 X 填充为  $P_1$  所在位  
的值,再令从  $P_2$  到  $P_3$  中的所有 X 填充为  $P_2$  所在位  
的值,这样从  $P_0$  到  $P_3$  即为... aaaXXXbbb... (a  
可以为 0 或 1,相应的 b 为 1 或 0)的形式.若将 X 完  
全填为 a(或 b),且不会使原 a 串(或 b 串)的码字增  
加,则将 X 填充为 a(或 b).否则,填充 X 时尽量让  
长串更长:在不增长 a 串编码长度的前提下,将部分  
X 填为 a,使 a 串饱和;在不增长 b 串编码长度的前  
提下,将剩余部分 X 填为 b,使 b 饱和;若还有 X 未  
被填充,则填为 a,b 串中较长的那个.

STEP5: 直到  $T_0$  中没有 X,算法终止,否则转  
STEP1.

图 1(b)为图 1(a)的填充实例.此时, $n=9, h=7, m=8, r=3, P_3$ 所在位的值和  $P_6$ 的相等(都是 0),且满足  $m < n \leq m+h+r$ ,  $P_0$ 到  $P_3$ 可分为两相等游程,游程长度  $L = \lceil (n+m+h+r)/2 \rceil = 13$ ,此时将  $P_3$ 调整为  $P_3 = P_0 + 2 \times (\lceil (n+m+h+r)/2 \rceil + 1) - 1 = P_0 + 27$ ,  $P_3$ 调整后的位置如图 1(b)中  $P_3$ 所示.按等游程填充后,  $P_0$ 到  $P_3$ 为两段长度都为 13 的游程.

## 4 解码器设计

进一步分析 ERFDR 编码表可以得知游程长度  $l$ 与前缀( $c$ )和尾部( $d$ )的关系: $l = (c)_2 + (d)_2 - 1$ ,例如,游程长度为 3 时, $l = (10)_2 + (10)_2 - 1$ .接下来我们分析得到各码字的前缀都是以 111...1110 的形式出现,可得:

$$\begin{aligned} l &= (c)_2 + (d)_2 - 1 = (c+10+d)_2 - 3 = \\ &= (\underbrace{(111 \dots 110)}_k + \underbrace{(10)}_2 + \underbrace{(d)}_k)_2 - 3 = \\ &= (\underbrace{(100 \dots 00)}_{k+1} + \underbrace{(d)}_k)_2 - 3 = \underbrace{(1d)}_{k+1}_2 - 3. \end{aligned}$$

由该式可以看出在尾部前面添加一个 1 所得二进制数比对应码字所代表的游程长度多 3. 本文利用这一点设计解码器对电路进行解压,用一个  $k+1$  位计数器实现. 该计数器的初始值设为 1,当编码数据输入时,把码字的尾部和 1 一起向高位移动,此外,由于  $l = (\underbrace{1d}_{k+1})_2 - 3$ ,所以  $k+1$  计数器减 1 的终值为  $3(0 \dots 011)$ 而不是 0.

该解码器的结构是基于一个有限状态机(FSM)的设计,类似于 FDR 解码器电路,结构简单,硬件开销小.如图 2 所示,该结构由一个 FSM,  $k+1$  位计数器,  $k+1$  位寄存器,  $\log_2 k$  位计数器和一个 T 触发器组成.其中,bit\_in:输入端口,压缩后的数据  $T_E$ 从此端口输入到有限状态机进行解码;en:使能信号,控制 bit\_in 的输入;clk:时钟信号; $\log_2 k$  位计数器:用来计算前缀的位数,并以此来控制尾部的输入;inc/dec 分别控制  $\log_2 k$  位计数器的加 1 和减 1 操作;dec1 控制  $k+1$  位计数器的减 1,rs 和 rs1 分别指示  $k+1$  位和  $\log_2 k$  位计数器已复位;shift:控制各个数据位移入  $k+1$  位计数器;load:装载信号;reg\_ent1:寄存器控制信号; $v$ :指示 scan input 何时有效.

该解码器的工作原理如下:

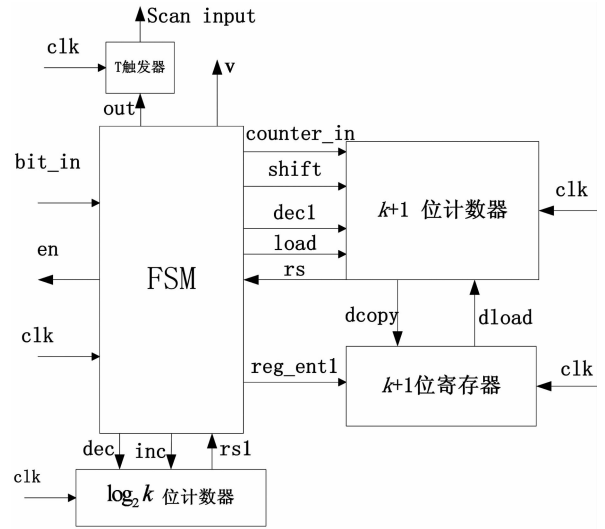


图 2 解码器示意图  
Fig. 2 Decoder sketch map

1)初始化,让使能信号 en 为 1,准备接收数据. T 触发器初始化为 1. 若解压数据第一位是 0,则 FSM 令 out 为 1, T 触发器翻转;若为 1,则令 out 为 0.

2)FSM 让使能信号为 1,shift 和 inc 为 1,FSM 控制  $\log_2 k$  位计数器加 1,直到 bit\_in 为 0 为止.

3)FSM 输出端为低电平,控制 T 触发器输出上一状态值,同时  $v$  为高电平,表示输出有效.

4)FSM 将  $k+1$  位计数器的初始值设为 1,并通过  $\log_2 k$  计数器控制将初始值 1 和尾部一起向高位移动.此时,dec 为高电平控制  $\log_2 k$  计数器的减 1,直到 rs1 为高电平, $\log_2 k$  计数器为 0 时,尾部输入结束.

5)reg\_ent1 为高电平时,通过 dcopy 复制  $k+1$  位计数器的值到  $k+1$  位寄存器中.

6)FSM 控制 dec1 为高电平,控制  $k+1$  位计数器的减 1 操作,直到  $k+1$  位计数器的值为 3(即  $0 \dots 011$ )时,out 输出为高电平,T 触发器的输出翻转.

7)当解码完一个游程后,一直到 bit\_in 为 1 之前,若 bit\_in 共出现 1 个“0”,则令 out 为低电平,且  $v$  也为低电平 0,表示输出无效,同时也为下一个游程编码做准备;出现两个 0 时,即“00”,则置 load 为高电平,通过 dload 把寄存器的值载入  $k+1$  位寄存器中,转到 6);当出现“000”时(共 3 个时钟周期),则在前 2 个时钟周期置 load 为高电平,通过 dload 把寄存器的值载入  $k+1$  位寄存器中,转到 6);第 3 个时钟周期令 out 为低电平, $v$  且也为低电平 0,表示输出无效;出现“0000”时,令 out 为低电平,然后

置 load 为高电平,通过 dload 把寄存器的值载入  $k+1$  位寄存器中,转到 6);当出现“0000”时,重复上述出现 4 个 0 的步骤,之后在最后一个时钟周期令 out 为低电平,且 v 也为低电平。

## 5 实验结果

本文针对 ISCAS89 标准电路中较大的 6 个电路,采用 mintest 测试集在 Visual C++ 平台上实验,得出的结果分别与 Golomb 码、FDR 码、EFDR 码、IFDR 码以及 ERLC<sup>[10]</sup> 码进行比较,实验结果见表 3。可以看到平均压缩率均优于其他方法,平均压缩率比 Golomb 编码方法提高了将近 14%,比 FDR 和 IFDR 分别提高了 6% 和 1.84%,比 EFDR 和 ERLC 也提高了 0.5%。

表 3 本方法与其他几种测试压缩方法的比较  
Tab. 3 The comparisons of compression ratio with other methods

电路名称	平均压缩率/%					
	Golomb	FDR	EFDR	IFDR	ERLC	本方法
S5378	40.70	48.02	53.67	50.64	51.32	52.95
S9234	43.34	43.59	48.66	47.40	49.28	49.41
S13207	74.48	81.30	82.49	82.12	83.05	82.97
S15850	47.11	66.22	68.66	67.54	69.04	68.42
S38417	44.12	43.26	62.02	60.99	62.44	64.12
S38584	47.71	60.91	64.28	63.19	64.72	65.03
AVG	49.58	57.22	63.30	61.98	63.30	63.82

其中压缩率计算公式如式(1)所示。

$$\text{压缩率} = (T_D - T_E) / T_D \times 100\% \quad (1)$$

式中:  $T_D$  为原测试集的大小;  $T_E$  为压缩后的测试集大小。

## 6 结论

本文在 IFDR 编码方法的基础之上进行改进,不仅能同时对 0,1 串编码,而且当出现相邻游程相等时,后一个游程用较短的码字来代替,进一步提高压缩率。实验结果充分验证了本文提出方法的有效性。该法简单可行,解码电路简单,硬件开销不高。

## 参考文献

- [1] TOUBA N. Survey of test vector compression technique [J]. IEEE Design & Test of Computer, 2006, 23(4): 294-303.
- [2] JAS A, TOUBA N. Test vector decompression via cyclical scan chains and its application to testing core-based designs [C]//Proceedings of International Test Conference. New York: IEEE, 1998: 458-464.
- [3] JAS A, GOSH-DASTIDAR J, NG M, *et al.* An efficient test vector compression scheme using selective Huffman coding[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2003, 22(6): 797-806.
- [4] CHANDRA A, CHAKRABARTY K. Test data compression and decompression based on internal scan chains and Golomb coding[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2002, 21(6): 715-722.
- [5] CHANDRA A, CHAKRABARTY K. System-on-a-chip test-data compression and decompression architectures based on Golomb codes[J]. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 2001, 20(3): 355-368.
- [6] CHANDRA A, CHAKRABARTY K. Frequency-directed run-length(FDR) codes with application to system-on-a-chip test data compression[C]//19th IEEE Proceedings on VLSI Test Symposium. New York: IEEE, 2001: 42-47.
- [7] EL-MALEH A, AL-ABAJI R. Extended frequency-directed run-length codes with improved application to system-on-a-chip test data compression[C]//Proceedings of 9th International Conference of Electronics, Circuits and Systems. New York: IEEE, 2002: 449-452.
- [8] EL-MALEH A. Test data compression for system-on-a-chip using extended frequency directed ran-length code[J]. IET Computers & Digital Techniques, 2008, 2(3): 155-163.
- [9] 欧阳一鸣, 郭文鹏, 梁华国. 改进型 FDR 码对 SoC 测试数据的压缩及解压[J]. 计算机应用研究, 2008, 25(1): 174-177. OUYANG Yi-ming, GUO Wen-peng, LIANG Hua-guo. Soc test data compression and decompression with improved FDR code[J]. Application Research of Computers, 2008, 25(1): 174-177. (In Chinese)
- [10] ZHAN W F, EL-MALEH A. A new scheme of test data compression based on equal-run-length coding (ERLC)[J]. Integration the VLSI Journal, 2012, 45(1): 91-98.