

一种用于测试数据压缩的自适应 EFDR 编码方法

邝继顺 周颖波* 蔡烁

(湖南大学信息科学与工程学院 长沙 410082)

摘要: 该文提出一种用于测试数据压缩的自适应 EFDR(Extended Frequency-Directed Run-length)编码方法。该方法以 EFDR 编码为基础,增加了一个用于表示后缀与前缀编码长度差值的参数 N ,对测试集中的每个测试向量,根据其游程分布情况,选择最合适的 N 值进行编码,提高了编码效率。在解码方面,编码后的码字经过简单的数学运算即可恢复得到原测试数据的游程长度,且不同 N 值下的编码码字均可使用相同的解码电路来解码,因此解码电路具有较小的硬件开销。对 ISCAS-89 部分标准电路的实验结果表明,该方法的平均压缩率达到 69.87%,较原 EFDR 编码方法提高了 4.07%。

关键词: 测试数据压缩; EFDR(Extended Frequency-Directed Run-length)编码; 自适应 EFDR 编码; 解码

中图分类号: TP302

文献标识码: A

文章编号: 1009-5896(2015)10-2529-07

DOI: 10.11999/JEIT150177

Adaptive EFDR Coding Method for Test Data Compression

Kuang Ji-shun Zhou Ying-bo Cai Shuo

(College of Information Science & Engineering, Hunan University, Changsha 410082, China)

Abstract: An adaptive Extended Frequency-Directed Run-length (EFDR) code method for test data compression is presented in this paper. The method is based on EFDR code, and adds an additional parameter N , which is used to represent the code length difference between tail and prefix. According to the distribution of the runs in each test vector of the test set, the method selects the most suitable N values to code, and it can improve the compression ratio. For the decompression, according to the size of the codeword, the run length of the original test data can be obtained with a simple mathematical operation. Meanwhile, those codeword under different parameter values can be decoded by the same decompression circuit. Thus, the decompression circuit can keep in a low hardware cost level. The experimental result shows that the average compression rate of the proposed method can achieve to 69.87%, over 4.07% than original EFDR code method.

Key words: Test data compression; Extended Frequency-Directed Run-length (EFDR) code; Adaptive EFDR code; decode

1 引言

测试成本的迅速增长是目前集成电路设计制造中的一个主要问题。一方面,随着特征尺寸的减小,集成电路中所包含的 IP 核与集成的逻辑门数量增多,使得所需测试数据规模增大;而另一方面,自动测试仪(ATE)的工作频率、I/O 通道数目、传输带宽、存储能力却增长有限。两者之间的矛盾是导致测试时间增加,测试成本上升的主要原因^[1]。

对测试数据进行压缩是一种能有效解决该矛盾的方法。目前测试数据压缩方法主要分为 3 类:基于线性解压结构的压缩方法、基于广播扫描的压缩方法和基于编码的压缩方法^[2]。其中编码压缩方法主

要针对给定测试集,将测试集中的测试数据分割成多个符号(字符串),再将每个符号用一个新码字替代,形成压缩后的数据,存储在 ATE 中。编码压缩方法主要分为两类,其中一类是基于游程的编码方法,它的基本原理是对原始测试数据中连续的 0 或 1 的长度(称作游程)进行编码,包括 FDR 编码^[3], EFDR(Extended Frequency-Directed Run-length)编码^[4], 多维游程编码(MD-PRC)^[5], 交替变游程编码(AVR)^[6], 对称游程编码(SVC)^[7]和一位标识混合编码^[8]等;另一类是基于数据块的编码方法,它是将原始数据划分为定长或变长的数据块,根据不同数据块的出现频率进行编码,对高频率的数据块赋予较短的码字。这类编码包括可变长 Huffman 编码(VIHC)^[9], 块合并编码(BM)^[10], 四值稀疏存储编码^[11]等。此外近年来人们也提出了一些适用于多扫描链结构^[12,13]和用于降低测试功耗^[14,15]的编码压缩方法。编码压缩不需要了解被测电路的内部结构信息,也不需要进行故障模拟或测试产生,非常适用

收稿日期: 2015-02-02; 改回日期: 2015-05-21; 网络出版: 2015-07-06

*通信作者: 周颖波 zhouyibo@163.com

基金项目: 国家自然科学基金(61472123, 60673085)

Foundation Items: The National Natural Science Foundation of China (61472123, 60673085)

于嵌入有 IP 核电路的测试数据压缩^[1]。

目前基于游程的编码压缩方法大都是根据整个测试集的游程分布情况提出的, 没有考虑到在测试集的不同测试向量之间, 游程分布存在着巨大差异。本文以此为切入点, 提出一种自适应 EFDR 编码方法, 以 EFDR 编码为基础, 增加了一个用于表示后缀与前缀编码长度差值的参数 N , 对测试集中的不同测试向量, 用不同的 N 值进行编码, 有效提高了测试数据压缩率。

2 EFDR 编码

EFDR 编码是一种重要的基于游程的编码压缩方法。EFDR 编码被称之为 FDR 编码^[3]的扩展码, 在 FDR 编码的基础上增加了一位标识位, 0 表示对 0 游程编码, 1 表示对 1 游程编码。由于可同时对两种游程编码, 因此经过对无关位 X 的适当填充, EFDR 编码的游程总数会远远小于仅基于 0 游程的 FDR 编码, 其压缩率也比 FDR 编码有较大提高。EFDR 的完整编码由标识位加前缀加后缀组成, 其编码表在表 1 中给出。

表 1 EFDR 编码表

组别	游程长度	编码组成		完整编码	
		前缀	后缀	0 游程编码	1 游程编码
A ₁	1	0	0	000	100
	2		1	001	101
A ₂	3	10	00	01000	11000
	4		01	01001	11001
	5		10	01010	11010
	6		11	01011	11011
A ₃	7	110	000	0110000	1110000
	⋮		⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮

然而 EFDR 编码自身也存在一些缺陷, 制约了它编码压缩率的提高:

EFDR 编码对测试集中的所有测试向量均使用相同的编码方法, 而没有考虑到不同测试向量的游程分布存在巨大差异。表 2 为 Mintest 测试集中测试向量的无关位 X 比例表, 列出了各电路测试集的测试向量中包含 X 位比例的最低值、最高值和平均值。由表 2 中可以看出, 最低值与最高值之间差异显著, 这使得各测试向量的游程分布也大为不同^[1]。以电路 s9234 为例, X 位比例的最低值仅为 4.86%, 对应的测试向量中绝大部分位都是确定位, 游程分

表 2 MinTest 测试集中 X 位比例

电路名称	测试向量数	X-min(%)	X-max(%)	X-avg(%)
s5378	111	5.61	92.06	72.62
s9234	159	4.86	95.14	73.01
s13207	236	7.71	98.57	93.15
s15850	126	2.62	98.53	83.56
s35932	16	0	98.01	35.30
s38417	99	3.49	96.57	68.08
s38584	136	1.09	99.66	82.28

布以短游程为主, 若用 EFDR 进行编码, 则编码后的数据量反而大于原始数据, 起不到压缩的效果; 而 X 位比例的最高值则为 95.14%, 此时对应的测试向量中绝大部分位都是无关位, 游程分布以长游程为主(X 填充后, 最短游程长度为 14 位), 而 EFDR 编码中, 却将大量编码码字用于对短游程编码(由表 1 可知, 长度为 1~6 的游程, 占用了前缀为 0 和 10 的编码码字), 这造成了编码资源的极大浪费。因此对这些游程分布差异很大的测试向量单一使用 EFDR 编码并不合适。

3 自适应 EFDR 编码

针对上一节中 EFDR 编码的缺陷, 本文提出了一种用于测试数据压缩的自适应 EFDR(Adaptive EFDR, A-EFDR)编码方法, 在原 EFDR 编码的基础上, 增加了一个用于表示后缀与前缀编码长度差值的参数 N 。对测试集中的每个测试向量, 根据其游程分布情况, 选择最合适的 N 值进行编码。表 3 和表 4 列出了其中 $N = -1$ 和 $N = 1$ 时 A-EFDR 编码的编码表, 其余 N 值下的 A-EFDR 编码表可依类似方法得出(其中 $N = 0$ 的编码表即为表 1)。

表 3 A-EFDR 编码表($N=-1$)

组别	游程长度	编码组成		完整编码	
		前缀	后缀	0 游程编码	1 游程编码
A ₁	1	0	-	00	10
	2	10	0	0100	1100
A ₂	3		1	0101	1101
	4	110	00	01000	11000
A ₃	5		01	01001	11001
	6	10	010100	110100	
7	10	010101	110101		
⋮	⋮	⋮	⋮	⋮	⋮

表 4 A-EFDR 编码表(N=1)

组别	游程长度	编码组成		完整编码	
		前缀	后缀	0 游程编码	1 游程编码
A ₁	1		00	0000	1000
	2	0	01	0001	1001
	3		10	0010	1010
	4		11	0011	1011
A ₂	5		000	010000	110000
	6	10	001	010001	110001
	⋮	⋮	⋮	⋮	⋮
	12		111	010111	110111
	⋮	⋮	⋮	⋮	⋮

与 EFDR 编码类似, A-EFDR 编码也采用由标识位加前缀再加后缀组成。标识位用于区分是对 0 游程编码还是对 1 游程编码; 前缀与 EFDR 编码中前缀相同, 对于不同组别 A_k(k=1,2,3,⋯), 前缀编码由 k-1 个 1 和一个 0 构成; 后缀长度则不再固定与前缀长度相同, 而是相差一个 N 值。因为前缀的起始长度为 1 位, 而后缀起始长度必须大于等于 0, 所以参数 N 的取值范围为: N ≥ -1。A-EFDR 编码表中, 起始的游程长度为 1 位, 对应的起始编码长度为 3 + N 位, 同一组别的游程编码长度不变, 跨组别的相邻游程编码长度增量为 2; 每一个组别 A_k 中, 包含的游程长度个数为 2^{k+N} 个。

给出一个 A-EFDR 编码的实例。

原始测试数据: T_D=000000111110 0000000001 (22 位)

编码后数据:

N = -1 时, T_E=010100 11000 01110001 (19 位)

N=0 时, T_E =01011 11001 0110010 (17 位)

N=1 时, T_E =010001 1011 010100 (16 位)

可以看出, 对于相同的测试数据, 不同 N 值的 A-EFDR 编码对应的编码长度各不相同, 我们选取使编码长度最短的 N 值进行编码, 这里选取 N=1。

对比上一节提到的 EFDR 编码中的缺陷。EFDR 编码中, 前后缀长度相等, N 值固定为 0, 对测试集中所有测试向量都使用同一种编码方法; 而在 A-EFDR 编码中, N 值大小可变, 可根据测试向量中游程的分布选取最合适的 N 进行编码。当测

试向量中 X 位比例较低, 游程分布以短游程为主时, 可选取较小的 N 值, N 值减小, 则编码表中游程的起始编码长度减小, 有利于对短游程的编码; 而当测试向量中 X 位比例较高, 长游程比例增加时, 可选取较大的 N 值, N 值增大, 虽然编码表中游程的起始编码长度增加, 但每一个组别中包含的不同游程长度个数呈指数增长, 仍有利于对长游程的编码。A-EFDR 编码增加了编码方法的灵活性, 根据游程的分布, 优化了编码字的长度, 有效避免了 EFDR 编码中的缺陷。

4 压缩增益分析

我们从概率论的角度来计算压缩增益。根据 A-EFDR 编码规则, 设参数大小为 N, 则组别 A_k 包含的游程范围为 2^{k+N} - 2^{N+1} < i_k ≤ 2^{k+N+1} - 2^{N+1}, 对应的码字长度为 2k + N + 1。

设测试数据中某一位为 0 的概率为 p, 则为 1 的概率为 (1-p), 其中 0 ≤ p ≤ 1。则长度为 i 的 0 游程出现概率为 pⁱ(1-p), 长度为 i 的 1 游程出现概率为 (1-p)ⁱp。由此可得, 对于某一游程 i, 其属于组别 A_k 的概率 P(i, k, N) 为

$$\begin{aligned}
 P(i, k, N) &= \sum_{i=2^{k+N}-2^{N+1}+1}^{2^{k+N+1}-2^{N+1}} [p^i(1-p) + (1-p)^i p] \\
 &= \frac{p^{2^{k+N}} - p^{2^{k+N+1}}}{p^{2^{N+1}-1}} \\
 &\quad + \frac{(1-p)^{2^{k+N}} - (1-p)^{2^{k+N+1}}}{(1-p)^{2^{N+1}-1}}
 \end{aligned}$$

组别 A_k 对应的码字长度为 2k + N + 1。因此 A-EFDR 编码的平均编码长度为

$$\begin{aligned}
 A &= \sum_{k=1}^{\infty} (2k + N + 1) P(i, k, N) \\
 &= \sum_{k=1}^{\infty} (2k + N + 1) \left[\frac{p^{2^{k+N}} - p^{2^{k+N+1}}}{p^{2^{N+1}-1}} \right. \\
 &\quad \left. + \frac{(1-p)^{2^{k+N}} - (1-p)^{2^{k+N+1}}}{(1-p)^{2^{N+1}-1}} \right]
 \end{aligned}$$

而原始数据中的平均游程长度为

$$\begin{aligned}
 \lambda &= \sum_{i=1}^{\infty} (i+1) p^i (1-p) + \sum_{i=1}^{\infty} (i+1) (1-p)^i p \\
 &= \frac{1}{p(1-p)} - 1
 \end{aligned}$$

因此得到 A-EFDR 编码的压缩增益为

$$\beta_{A-EFDR} = \frac{\lambda}{A} = \frac{p^2 - p + 1}{p(1-p) \sum_{k=1}^{\infty} (2k + N + 1) \left[\frac{p^{2^{k+N}} - p^{2^{k+N+1}}}{p^{2^{N+1}-1}} + \frac{(1-p)^{2^{k+N}} - (1-p)^{2^{k+N+1}}}{(1-p)^{2^{N+1}-1}} \right]}$$

取 $N=0$ ，即可得到 EFDR 编码的压缩增益为

$$\beta_{\text{EFDR}} = \frac{p^2 - p + 1}{p(1-p) \sum_{k=1}^{\infty} (2k+1) \left[\frac{p^{2^k} - p^{2^{k+1}}}{p} + \frac{(1-p)^{2^k} - (1-p)^{2^{k+1}}}{1-p} \right]}$$

为了更直观比较不同 N 值下 A-EFDR 编码的压缩增益, 我们通过 Matlab 画出了其中 N 取值为 $-1, 0, 1$ 时 A-EFDR 编码的压缩增益曲线, 如图 1 所示。纵轴表示压缩增益 β , 横轴为每一位取 0 的概率 p 。因增益曲线关于 $p=0.5$ 轴对称, 所以这里只给出了区间 $0.5 \leq p \leq 1$ 中的曲线进行分析。从图中可以看出, 不同 N 值的 A-EFDR 编码有各自优势的 p 值区间, 且 N 值越小, 优势的区间越靠近 $p=0.5$ 。在 $0.5 \leq p \leq 0.78$ 时, $N = -1$ 压缩增益最高; $0.78 < p \leq 0.88$ 时, $N=0$ 压缩增益最高; $p > 0.88$ 时, $N=1$ 压缩增益最高。实际上, 随着 p 的增大, N 值也需要增加才能取得更高的压缩增益, 如 $p > 0.92$ 时, $N=2$ 比 $N=1$ 的压缩增益更高。因此我们可以根据测试向量中 p 值的大小, 选取对应压缩增益最高的 N 值进行编码, 提高编码压缩率。

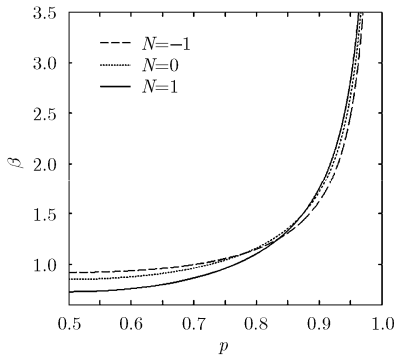


图 1 不同 N 值下 A-EFDR 编码压缩增益图

5 解码器设计

一个优秀的编码压缩方法, 在保持较高数据压缩率的同时, 还应当具有较小的硬件解码开销^[1]。对经过 A-EFDR 编码压缩的码字经过简单的数学运算即可解码, 且不同 N 值的 A-EFDR 编码均可使用相同的解码电路来解码。

设测试数据游程长度为 L , 编码参数为 N , 编码后码字的前缀大小为 M_1 , 后缀大小为 M_2 , 则 L 与 N, M_1, M_2 之间的数学关系表达式为

$$L = M_1 \times 2^N + M_2 + 1$$

以 $N=1$ 时的 A-EFDR 编码为例, 设前缀 $M_1 = (10)_2$, 后缀 $M_2 = (111)_2$, 则游程长度 L 为: $L = (10)_2 \times 2^1 + (111)_2 + 1 = (1100)_2 = 12$ 位, 表达式前缀 M_1 乘以 2^N , 对应于解码器中是将前缀 M_1 左移 N 位的

操作, 不同 N 值的 A-EFDR 编码对应移位的长度不同。在电路中实现移位操作较为简单, 所需的硬件代价较小。

这里需要涉及到对 $N = -1$ 的处理, 与其他 N 值不同, $N = -1$ 时, 前缀 M_1 的移位操作为右移一位。本文对此的处理方法为: 在前缀 M_1 读入计数器时, 最后一位 0 读取但并不存入计数器中, 相当于 M_1 的值已默认右移了一位。若参数 N 为其他值时, 再在 $N = -1$ 的基础上对前缀 M_1 进行左移操作。

图 2 为 A-EFDR 编码方案解码电路框图。其结构由一个有限状态机(FSM)、一个 k 位计数器、一个 $\log_2 k$ 位计数器、一个 m 位计数器、一个 n 位寄存器和一个异或门组成。其中 k 位计数器用于读取编码中的前缀和后缀; $\log_2 k$ 位计数器记录前缀和后缀的长度; n 位寄存器和 m 位计数器分别存储当前编码的参数 N 和对应的测试向量数目; 标识位 flag 通过异或门控制当前解码数据的游程种类。与 EFDR 解码电路^[4]相比, 本解码电路仅增加了一个 m 位计数器和一个 n 位寄存器, 增加的硬件开销相对较小。

该解码电路解码步骤如下:

- (1) 读入信息控制位部分: 信号 en 置为高电平, 接收 bit_in 中的数据, 将当前编码参数 N 和对应的向量数目分别存入 n 位寄存器和 m 位计数器中。
- (2) 读入标识位部分: 接收 bit_in 的一位数据, 赋值给 flag 信号。
- (3) 读入前缀部分: bit_in 中的数据读入 k 位计数器, 直到接收到 0 为止(接收到 0 表示前缀部分读

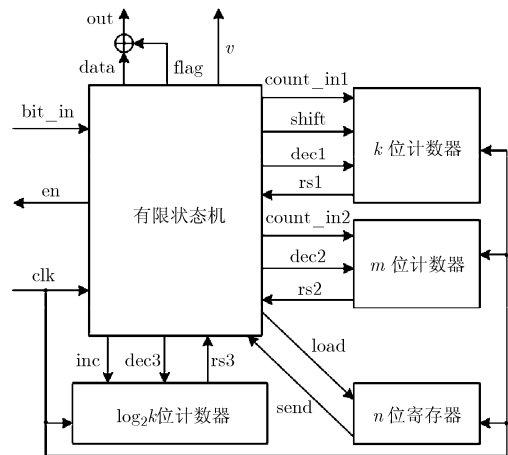


图 2 解码电路框图

入完毕, 这里 0 读取但不存入计数器中); $\log_2 k$ 计数器记录前缀的位数。之后读取 n 位寄存器中的参数 N , 将 k 位计数器中数值左移 N 位, $\log_2 k$ 计数器中数值增加 N 。

(4)前缀部分解压: FSM 输出 0 至 data, v 输出高电平, 表示输出有效。FSM 每输出一个 0, k 位计数器减 1, 直到 k 位计数器为空, rs1 为 1, 前缀部分解压结束。

(5)读入后缀部分: bit_in 中的数据接着读入 k 位计数器, 每读入一位数据 $\log_2 k$ 位计数器减 1, 直到 $\log_2 k$ 位计数器为空, rs3 为 1, 后缀部分读入结束。

(6)后缀部分解压: 与前缀解压部分类似, FSM 输出 0 至 data, 输出 0 的个数为 k 位计数器的数值, 之后再输出一个 1, 游程解压完毕。

(7)若一个测试向量解压完毕, m 位计数器减 1。当计数器数值为 0 时, 转入步骤(1), 重新读入控制信息位; 否则转入步骤(2), 开始新游程的解码。

6 实验结果与分析

为了验证 A-EFDR 编码的压缩效果, 本文以 Mintest 故障测试集为对象, 对 ISCAS-89 标准电路中最大的几个电路进行了实验。

在 A-EFDR 编码中, 需要对测试集中不同测试向量选取各自最合适的 N 值进行编码。本文中对每个测试向量用不同的 N 值进行了尝试, 选取对应最合适的那个, 得到的 N 值分布表如表 5 所示, 表中任意 N 值对应的列, 表示选取相应 N 值的测试向量个数, 最后一列为测试向量数目总和。之后每个测试集中的测试向量按对应 N 值的大小进行了重排序。

从表中可以看出, N 值的大小分布在 $-1 \sim 7$ 之间, 不同测试集的 N 值范围各不相同。其中电路 s5378 和 s38417 的 N 值最高值都比较小, 这是因为这两个电路的高 X 位比例的测试向量中, 确定位并非均匀分布, 而是集中出现的, 这使得测试向量中仍存在相当比例的短游程, 导致对应的 N 值比较小。

表5 N 值分布表

电路名称	不同 N 值下的测试向量个数									向量数总和
	$N = -1$	$N=0$	$N=1$	$N=2$	$N=3$	$N=4$	$N=5$	$N=6$	$N=7$	
s5378	17	44	36	14	-	-	-	-	-	111
s9234	4	13	32	42	39	20	4	5	-	159
s13207	1	4	7	12	14	71	77	39	11	236
s15850	2	6	14	28	28	31	12	5	-	126
s35932	-	-	-	1	5	6	4	-	-	16
s38417	-	14	10	44	31	-	-	-	-	99
s38584	4	7	14	14	34	31	14	18	-	136

由第 5 节提到的解码过程可知, A-EFDR 编码与 EFDR 编码相比, 需要额外存储一些控制信息位, 包括当前编码参数 N , 以及 N 值对应的测试向量数目。表 6 为控制信息位表, 列出了各个测试集中原始测试数据的位数, 和所需额外存储的控制信息位数。信息位计算方法以电路 s9234 为例: 由表 5 可知, s9234 的测试集被分为了 8 个子集, 每个子集需要提供一次对应的 N 值和测试向量数目。 N 值最大为 6, 需要用 3 位表示; 而不同 N 值下, 最大测试向量个数为 42 ($N = 2$ 时), 需要用 6 位表示, 因此 s9234 总共所需存储的控制信息位位数为: $(3 + 6) \times 8 = 72$ 位。

从表 6 中可以看出, 各个测试集中所需存储的控制信息位都很小, 最大不超过 100 位, 对比于

表 6 控制信息位表

电路名称	测试数据位数	控制信息位数	占比(%)
s5378	23754	32	0.13
s9234	39273	72	0.18
s13207	165200	99	0.06
s15850	76986	64	0.08
s35932	28208	24	0.08
s38417	164736	36	0.02
s38584	199104	72	0.04

Mintest 测试集中原始测试数据的位数, 所占比例极小, 因而对测试数据的压缩不会有太大影响。

表 7 为本方法与目前国内外主流的编码压缩方

法在压缩率^[3]方面的比较,其中第 2 列到第 5 列为基于游程的编码压缩方法,分别为 EFDR 编码^[4],多维游程编码(MD-PRC)^[5],交替变游程编码(AVR)^[6]和对称游程编码(SVC)^[7];第 6 第 7 列为基于数据块的编码压缩方法,分别为块合并编码(BM)^[10]和四值稀疏存储编码^[11];最后一列为本文提出的 A-EFDR 编码。以上编码压缩方法都是选用的相同的测试集,且都没有进行向量差分或扫描链重排序等操作。本文 A-EFDR 编码方法中,对无关位 X 的填充,采用文献[16]中提出的动态最优填充算法。

从表 7 中可以看出,与 EFDR 编码相比,本文方法对每个电路测试集的压缩率都有提高,平均提高了 4.07%,这也说明了本方法对 EFDR 编码中缺陷的改进是有效果的。7 个电路中,电路 s35932 在压缩率已经很高(80.80%)的基础上继续提高了 4.27%,这主要是因为 s35932 测试集中确定位 0 和 1 并非均匀分布,而是“抱团”出现,测试集中含有大量的连续 0 和连续 1,形成了许多长游程,使

得每个测试向量对应的 N 值都比较高(参见表 5),因而压缩率较 N 值固定为 0 的 EFDR 编码有较大提高。

对比表中的其他国内外参考文献,除 s13207 电路外,本文方法的压缩效果都是最优的,且本文方法还有解码电路简单的优势,这些数据也充分说明了本编码压缩方法的有效性。

7 结论

针对测试集中不同测试向量的游程分布存在巨大差异的特点,本文提出了一种自适应 EFDR 编码的测试数据压缩方法。本文方法以 EFDR 编码为基础,增加了一个用于表示后缀与前缀编码长度差值的参数 N ,对测试集中的不同测试向量,选用不同的 N 值进行编码,有效提高了测试数据压缩率。在解码方面,本方法较 EFDR 编码仅需增加少量硬件资源,硬件解码开销仍然较低。综上所述,本文方法具有较好的应用前景。

表 7 本方法压缩效果与其他压缩方法的比较(%)

电路名称	EFDR ^[4]	MD-PRC ^[5]	AVR ^[6]	SVC ^[7]	BM ^[10]	稀疏存储 ^[11]	本文方法
s5378	53.67	54.63	52.30	51.80	54.98	53.63	55.70
s9234	48.66	53.20	50.72	50.94	51.19	54.23	54.91
s13207	82.49	86.01	83.38	83.77	84.89	85.91	85.85
s15850	68.66	69.99	70.08	69.98	69.49	71.45	72.32
s35932	80.80	-	81.72	-	78.35	78.07	85.07
s38417	62.02	55.38	62.82	63.30	59.39	62.03	66.02
s38584	64.28	67.73	65.89	66.26	66.86	68.50	69.21
Avg	65.80	64.49	66.70	64.34	66.45	67.69	69.87

参 考 文 献

- [1] Mehta U S, Dasgupta K S, and Evashrayee N J. Un-length-based test data compression techniques: how far from entropy and power bounds a survey[J]. *VLSI Design*, 2010(1): 1-9.
- [2] 刘铁桥, 邝继顺, 蔡烁. 一种将测试集嵌入到 Test-per-Clock 位流中的方法[J]. *计算机研究与发展*, 2014, 51(9): 2022-2029. Liu Tie-qiao, Kuang Ji-shun, and Cai Shuo. A new method of embedding test patterns into test-per-clock bit stream[J]. *Journal of Computer Research and Development*, 2014, 51(9): 2022-2029.
- [3] Anshuman C and Krishnendu C. Frequency-Directed Run-length(FDR) codes with application to system-on-a-chip test data compression[C]. *Proceedings of the 19th IEEE VLSI Test Symposium*, Atlantic, 2001: 42-47.
- [4] EL-Maleh A H. Test data compression for system-on-a-chip using Extended Frequency-Directed Run-Length Code[J]. *IET Computers & Digital Techniques*, 2008(2): 155-163.
- [5] Dauh T W and Jen L L. Test data compression using multi-dimensional pattern run-length codes[J]. *Journal Electron Test*, 2010, 26(3): 393-400.
- [6] Ye B, Zhao Q, Zhou D, et al. Test data compression using alternating variable run-length code[J]. *INTEGRATION, the VLSI Journal*, 2011(44): 103-110.
- [7] 梁华国, 蒋翠云, 罗强. 应用对称编码的测试数据压缩方法[J]. *计算机研究与发展*, 2011, 48(12): 2391-2399. Liang Hua-guo, Jiang Cui-yun, and Luo Qiang. Test data compression and decompression using symmetry-variable codes[J]. *Journal of Computer Research and Development*, 2011, 48(12): 2391-2399.
- [8] 马会, 邝继顺, 马伟. 基于一位标识的测试向量混合编码压缩

- 方法[J]. 电子测量与仪器学报, 2013, 27(4): 312-318.
- Ma Hui, Kuang Ji-shun, and Ma Wei. Hybrid coding compression method of test vector based on an identification [J]. *Journal of Electronic Measurement and Instrument*, 2013, 27(4): 312-318.
- [9] Gonciari P T, Al-Hashimi B M, and Nicolici N. Variable-length input Huffman coding for system-on-a-chip test[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2003, 22(6): 783-789.
- [10] EL-Maleh A H. Efficient test compression technique based on block merging[J]. *IET Computer & Digital Techniques*, 2008, 5(2): 327-335.
- [11] Zhang L and Kuang J S. Test data compression using selective sparse storage[J]. *Journal Electron Test*, 2011, 27(4): 565-577.
- [12] 刘杰, 易茂祥, 朱勇. 采用字典词条衍生模式的测试数据压缩 [J]. 电子与信息学报, 2012, 34(1): 231-235.
- Liu Jie, Yi Mao-xiang, and Zhu Yong. Test data compression using entry derivative mode of dictionary[J]. *Journal of Electronics & Information Technology*, 2012, 34(1): 231-235.
- [13] Sismanoglou P and Nikolos D. Test data compression based on reuse and bit-flipping of parts of dictionary entries[C]. Proceedings of 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, Warsaw, 2014: 110-115.
- [14] Tyszer J, Filipek M, Mrugalski G, *et al.* New test compression scheme based on low power BIST[C]. Proceedings of 18th IEEE European Test Symposium, Avignon, 2013: 1-6.
- [15] Chloupek M, Jenicek J, Novak O, *et al.* Test pattern decompression in parallel scan chain architecture[C]. Proceedings of 16th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, Karlovy, 2013: 219-223.
- [16] 方昊, 姚博, 宋晓笛. 双游程编码的无关位填充算法[J]. 电子学报, 2009, 37(1): 1-7.
- Fang Hao, Yao Bo, and Song Xiao-di. The algorithm of filling X bits in dual-run-length coding[J]. *Acta Electronica Sinica*, 2009, 37(1): 1-7.
- 邝继顺: 男, 1959 年生, 博士, 教授, 研究方向为集成电路测试与设计, 嵌入式系统.
- 周颖波: 男, 1984 年生, 博士生, 研究方向为集成电路测试.
- 蔡 烁: 男, 1982 年生, 博士生, 研究方向为集成电路测试.