Test Data Compression Using a New Scheme Based on Extended Variable Length Codes

J. Robert Theivadas and V. Ranganathan

Reserach Scholar, Anna University, Chennai 600025, India
Sree Sastha Institute of Engineering and Technology, Chennai 600123, India

Abstract: The need of testing large amount of data in large ICs has increased the time and memory requirement by many folds. Several test data compression schemes have been proposed for reducing the test data volume. In this paper, we propose a novel, lossless, time and memory minimizing test data compression scheme based on the fixed to variable length coding with limited number of code words. In our scheme, we divide the test vectors into fixed length blocks and then compress them into variable length codes. We use this extended variable length codes algorithm to make changes in the test vectors and to increase the compression ratio. The generation of good compression ratio through our scheme is proved by the experimental results for the ISCAS-89 benchmark circuits and the compressed test data. In comparison to the previous test data techniques based on variable length codes, the outcome of our method has a reasonable effect on compression.

Key words: Test data compression · Variable length codes · Compression ratio

INTRODUCTION

According to Moore’s law, the integration level of the microchips doubles every 18 to 24 months. As a result, the volume of test data increases dramatically; the test costs become higher; the contradiction between test efficiency and test quality sharpens. The effective method to reduce test data volume is test data compression [1].

There are three type of test data compressions viz., the Linear de-compression based schemes [4, 5], where the de-compression of data is done using linear operations; the Broadcast scan based schemes [6, 7], where broadcast scan based scheme broadcasts the same values in multiple scan chain; and the Code based schemes [2, 3], where the code based schemes use data compression codes to encode the test cubes.

Code based scheme is widely used [8] in test data compression as it transfers the specified bit-strings in test data into a segment of code word. A certain encoding scheme is used to encode the original test set. The length of the encoded data is less than the original test set so that the test data can be reduced.

The test set is divided into sequences of specific law, which can be replaced with new code word generated by some kinds of coding method. According to the change rule of the length from the original sequence to the code word, Code based test data compression can be classified into four categories. First category is fixed to fixed, these codes encode fixed size blocks of data using smaller fixed size blocks of encoded data. The [14] dictionary code is used for the fixed to fixed coding scheme. Here the data transfer rate from the tester to decoder is constant.

Second one is Fixed-to-Variable Codes; these codes encode fixed size blocks of data using a variable number of bits in the encoded data. The Huffman code [9-13] is typically used for the fixed-to-variable coding scheme. Huffman code is used in many compression techniques, as it is one of the efficient codes for encoding the test data. Selective Huffman codeword [15] adds symbols to the Huffman tree (i.e.) addition of symbols with higher frequency. One of the advantage of using selective Huffman coding is, it is used to reduce the hardware overhead of Huffman FSM. And this method gives good result, but the hardware cost is high.

Third one is Variable-to-Fixed Codes; these codes encode a variable number of bits using fixed size blocks of encoded data. The run length code [16] is used for variable-to-fixed coding scheme. The Almost Instantaneous VF (AIVF) coding [18] is used to get good compression ratio. The Suffix Tree-based VF (STVF) coding [19] is used to get better compression ratio.
Fourth one is Variable-to-Variable codes; these codes encode a variable number of bits using a variable number of bits in the encoded data. The Golomb code [17] is used for variable-to-variable coding scheme. This Golomb code achieves greater compression ratio and the overhead is small. In variable-to-variable code word one of the difficulties is synchronizing the transfer of data from the tester. The other variable-to-variable coding schemes are alternation and run length code [21], FDR code [20] and EFDR code [22].

In this paper, one of the new techniques is used to compress the original test vectors. Here the original test vectors are divided into fixed length blocks, which are then compressed into variable code words. The technique used in this paper is called the Extended Variable Run Length Coding. Using this compression technique greater compression will be achieved and decompression will be performed. In decompression, the variable length code words are decompressed into fixed length code words using FSM.

The organization of the paper is as follows - Section 2 explains the algorithm, flowchart and an example of the proposed scheme; Section 3 explains the structure of the de-compressor; Section 4 explains the experimental results and analyzes of the compression ratio theoretically; and Section 5 concludes the paper.

**Encoding Rule:** In this paper, the new test vector compression scheme is used, so that the number of test vectors will be reduced. The encoding rule consists of various steps, which are explained below:

**Step 1:** The input given is the original test vectors.
**Step 2:** Divide the original test vectors into four bits.
**Step 3:** If all the bits in the original test vectors are divided into four bits then the compression is achieved using the compression rules.
**Step 4:** The compression rules are given below:

**Rule 1:** If the original test vector is 0000, then the compressed test vector is 1.
**Rule 2:** If the original test vector is 0001, then the compressed test vector is 01.
**Rule 3:** If the original test vector is 0010, then the compressed test vector is 10.
**Rule 4:** If the original test vector is 0011, then the compressed test vector is 11.
**Rule 5:** If the original test vector is 0100, then the compressed test vector is 00.

**Step 5:** Then the process is repeated and all the bits in the test vector is compressed.
**Step 6:** If all the bits are compressed, the process stops otherwise it repeats from step 3.

Here it uses fixed to variable length coding with limited number of code words. This is one of the novel test data compression scheme. The following example shows how the algorithm works. Take C432 circuit, the test vectors in C432 are divided into four bits. Some of the test vectors from C432 are taken and explained below.

110101010101010101010101010101000111
100010001001011101110111011101110100.

The above test vectors are the original test vectors which are divided into four bits. The 1101 is compressed into x01, 0101 is compressed into 001, 0100 is compressed into 00, 0111 is compressed into 011, 1000 is compressed into 100, 1001 is compressed into x1.

X01 001 001 001 001 001 00 011
100 100 x1 011 011 011 011 011 00

Here the 72 bits are compressed into 51 bits.
Table 1: Extended variable run length coding method

<table>
<thead>
<tr>
<th>Run Length</th>
<th>Test Data</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>01</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>00</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>001</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>010</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>011</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>X1</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>X0</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>111</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>000</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>X01</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>X10</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 1: Flow diagram of the compression technique.

Table 1 shows that fixed to extended variable run length code

The above tabular column shows that the fixed length code word and the variable length code words. These code words are used to compress the original test vectors in various benchmark circuits.

Then the flow chart of the compression technique is shown Figure 1.

**Structure of Decompressor:** After the compression process, the test vectors must be de-compressed. Here one of the new techniques is used to de-compress the test vectors. And that technique consists of input which is the compressed test vectors and it is given to the FSM. The output obtained from the FSM is the original test vectors.

The block diagram and the working diagram of FSM is shown Figure 2.

The working diagram of the FSM is shown Figure 3. The working diagram of FSM is explained. Here 0 and 1 is considered when the given input 0 remains as such, the output is 1111 and the given input 1 remains as 1, then the decompressed value is 0000. Then the 0 to 1 transition takes place, the 01 is decompressed to 0001 and when the 1 to 0 transition occurs, the 10 is decompressed to 0010. When the 0 becomes 1, the value becomes 11 and when the 1 becomes 0 the value becomes 00. If 11 is in 0 state then the decompressed output of 11 becomes 0011. And if 00 is in 1 state then the decompressed output of 00 becomes 0100. Then it makes the transition from 11 to 00, the 010 is decompressed in to 0110. And it makes the transition from 00 to 11; the 001 is decompressed in to 0101.
When 11 changes to 1 state, the value becomes 011 and when the 00 changes to 0 state, the value becomes 100. If 011 is in 0 state then the decompressed output of 011 becomes 0111. And if 100 is in 1 state, then the decompressed output of 100 becomes 1000. Then it makes the transition from 011 to 100, the x0 is decompressed into 1010. And it makes the transition from 100 to 011; the x1 is decompressed into 1001.

When 011 changes to 1 state, the value becomes 111 and when the 100 changes to 0 state, the value becomes 000. If 111 is in 0 state then the decompressed output of 111 becomes 1011. And if 000 is in 1 state, then the decompressed output of 000 becomes 1100. Then it makes the transition from 111 to 000, the x10 is decompressed into 1110. And it makes the transition from 000 to 111; the x01 is decompressed into 1101. So by this technique greater compression and decompression has been achieved.

**RESULTS AND DISCUSSION**

Here, the effectiveness of the proposed scheme is verified by using experiment results. For comparison with other schemes, the MinTest test sets provided by Duke University of America are adopted, which are same as the test vectors used in paper [23]. Experiments were performed on the largest ISCAS 89 benchmark circuits [24]. Table 2 shows the analysis of the number of runs on the used test sets. The 1st column shows the circuit name; the 2nd, the number of runs before the extended variable run length code is applied; and the last, the number of runs obtained after the extended variable run length code is applied. It is found that the number of runs is decreased reasonably by using this extended variable run length coding technique. From Table 2, it is found that the decrease of number of runs ranges 75% for different test sets.

Then the extended run length coding scheme is used to compress these new test vectors. The compression ratio of the proposed scheme in comparison to the other coding schemes is shown in Table 3. In these experiments, runs of 0s and 1s are used. The first column of Table 3 is the circuit name. The second column is the test size of original test data. The compression effect of EVRL is shown in the third column of Table 3. The compression result of extended variable run length coding is also given for comparison in Table 3. The other columns are sizes of compressed test data and compression ratio of BM [2], Golomb code [17], VIHC [15], EFDR [22], RL Huffman coding [9] and FDR [23]. The Compression Ratio (CR) is the ratio of the number of bits per sample before compression to the encoded data rate.

\[
\text{Compression Ratio} = \frac{T_D - T_E}{T_D} \times 100
\]

Where

- TD is original length of test data and
- TE is the length of compressed data [23].

From table 3, it is found that the new scheme has effectively generated very good compression ratios when compared to many run length coding schemes. The average compression ratio (60.7583) achieved through our technique is quite higher than that other compression schemes.

The above graphical representation clearly shows that our new scheme generates high compression ratio in comparison with other standard compression algorithms.

### Table 2: Analysis of the number of runs in test sets

<table>
<thead>
<tr>
<th>Name of the circuit</th>
<th>Runs before EVRL</th>
<th>Runs after EVRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>s5378</td>
<td>20758</td>
<td>15570</td>
</tr>
<tr>
<td>S9234</td>
<td>25935</td>
<td>19452</td>
</tr>
<tr>
<td>S13207</td>
<td>163100</td>
<td>122325</td>
</tr>
<tr>
<td>S15850</td>
<td>57434</td>
<td>43077</td>
</tr>
<tr>
<td>S38417</td>
<td>59532</td>
<td>84864</td>
</tr>
<tr>
<td>S38584</td>
<td>161040</td>
<td>120780</td>
</tr>
</tbody>
</table>

### Table 3: Comparison of compressed MINTEST test vector obtained by using different coding technique

<table>
<thead>
<tr>
<th>Circuits</th>
<th>Size of original test vector $T_D$</th>
<th>FDR</th>
<th>Golomb Code</th>
<th>VIHC</th>
<th>EFDR</th>
<th>RL Huffman</th>
<th>Proposed Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>s5378</td>
<td>20578</td>
<td>20.54</td>
<td>15.72</td>
<td>25.29</td>
<td>34.74</td>
<td>36.56</td>
<td>59.9</td>
</tr>
<tr>
<td>S9234</td>
<td>25935</td>
<td>24.56</td>
<td>11.06</td>
<td>28.29</td>
<td>44.40</td>
<td>46.58</td>
<td>58.8</td>
</tr>
<tr>
<td>S13207</td>
<td>163100</td>
<td>42.57</td>
<td>37.35</td>
<td>56.16</td>
<td>60.37</td>
<td>66.59</td>
<td>59.37</td>
</tr>
<tr>
<td>S15850</td>
<td>57434</td>
<td>45.11</td>
<td>42.83</td>
<td>52.35</td>
<td>53.82</td>
<td>54.64</td>
<td>58.84</td>
</tr>
<tr>
<td>S38417</td>
<td>113152</td>
<td>58.8</td>
<td>59.14</td>
<td>60.92</td>
<td>62</td>
<td>63.33</td>
<td>68.34</td>
</tr>
<tr>
<td>S38584</td>
<td>161040</td>
<td>46.47</td>
<td>41.85</td>
<td>46.76</td>
<td>47.13</td>
<td>52.27</td>
<td>59.3</td>
</tr>
</tbody>
</table>

Average 39.675 34.6583 44.9617 50.41 53.3283 60.7583
CONCLUSION

Test data compression is an effective solution to reduce the increasing test data in larger ICs. A new test data compression scheme based on extended variable run length code is proposed here. In this technique, the original test vectors are divided into fixed length blocks; and then, the fixed length blocks are compressed into variable code words. This decreases the number of runs and thereby enabling a better compression effect. It is shows that this technique has the characteristic of high compression ratio, which decreases the ATE memory, the channel capacity requirements and the low hardware overhead. Thus, it is an effective solution to test data compression/decompression for IC design.

REFERENCES


