A New Hybrid Test Data Compression Technique for Low-Power Scan Test Data

Jaehoon Song, Hyunbean Yi, Dooyoung Kim, Juhee Han, Sungju Park
Dept. of Computer Science & Engineering, Hanyang University
{hjsong, bean, doo0, hanjh, parksj}@mslab.hanyang.ac.kr

Abstract

The huge test data volume, test time and power consumption are major problems in system-on-a-chip testing. To tackle those problems, we propose a new test data compression technique. Initially, don’t-cares in a pre-computed test cube set are assigned to reduce the test power consumption, and then, the fully specified low-power test data is transformed to improve compression efficiency by neighboring bit-wise exclusive-or (NB-XOR) scheme. Finally, the transformed test set is compressed to reduce both the test equipment storage requirements and test application time.

I. Introduction

Power dissipation is one of major challenges in today’s circuit design. Generally, a circuit or system consumes more power caused by excessive switching activities in test mode than in normal mode, and it has been confirmed that the shift power in the scan chains is the dominant contributor to test power [1]. This extra power consumption can incur severe hazards in circuit reliability or, in some cases, can provoke instant circuit damage [1]. Thus, it is extremely important to reduce power consumption for SoC test, and so a number of techniques to reduce power consumption in test mode have been presented in literature [2]–[14].

Thus, to achieve simultaneous reduction in both test data volume and test power consumption, many works have been proposed [1], [6]–[8], [9]–[12]–[14]. In this paper, we also propose a new test data compression/decompression technique for the simultaneous reduction of test data volume and test power consumption.

This paper is organized as follows. Our approach is described in section II and experimental results for ISCAS’ 89 benchmark circuits are presented in section III followed by concluding remarks.

II. Proposed Approach

In this section, we describe our test data compression procedure and decompression architecture. For compression, initially, don’t-cares in a pre-computed test cube set are specified
by using MTC-filling for low-power scan test, and then, the fully specified test set is transformed to improve compression efficiency by using the neighboring bit-wise exclusive-OR (NB-XOR) transform which results in skewed data containing a lot of consecutive 0s. Finally, the transformed test set can be significantly compressed by using 0 run-length encoding followed by Huffman encoding.

A. Don’t-Care Filling for Low-Power Scan Test

It is noted that the test patterns generated by automatic test pattern generator (ATPG) generally contain more don’t-cares than fixed values. This is because, to detect a certain fault, only a small number of bits need to be specified in a scan vector. Therefore, by properly assigning fixed values to don’t-cares, test data set for low-power scan test can be achievable.

Thus, in our technique, we adopted the minimum transition count filling (MTC-filling) technique which minimizes the weighted transition metric (WTM) [15] to reduce scan-in power consumption. The WTM model is proposed to estimate the power by counting transitions during scan shifting.

Consider a scan chain of length \( l \) and a scan vector \( V_j = b_{j,1}, b_{j,2}, \ldots, b_{j,l} \) with \( b_{j,1} \) scanned in first. The WTM for \( j \)-th vector \( V_j \) is given by (1).

\[
W TM_j = \sum_{i=1}^{l} (l-i) \cdot (b_{j,i} \oplus b_{j,i+1})
\]

If \( n \) is the number of scan vectors, the average and peak scan-in power are estimated by (2) and (3), respectively.

\[
P_{avg} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{l} (l-i) \cdot (b_{j,i} \oplus b_{j,i+1})}{n}
\]

\[
P_{peak} = \max_{j=1(2,\ldots, n)} \{ \sum_{i=1}^{l} (l-i) \cdot (b_{j,i} \oplus b_{j,i+1}) \}
\]

B. Neighboring Bit-Wise Exclusive-OR Transform

In this section, we describe our transform technique which is the pre-processing method before compressing low-power scan test data to achieve high compression.

Before describing our technique, we first briefly review the cyclical scan register (CSR) scheme [9] to compare with ours, because, to the best of our knowledge, this is the only state-of-the-art approach to transform the fully specified test data into the skewed data with a lot of 0s for high compression.

In the CSR based technique, compression is performed on the difference vector set achieved by using correlation between scan vectors to generate a lot of 0s in order to enhance compression efficiency, so, as shown in [8], [9], [10]. However, this approach has the obvious drawback requiring a separate cyclical scan register (CSR) for decompressing [7]. Furthermore, optimal scan vector reordering to get the best results is NP-complete problem, and in section III, experimental results comparing between CSR and our techniques show the efficiency of ours.

In our technique, first, don’t-cares in a pre-computed scan vectors are assigned by using MTC-filling to save scan test power, and then this yields the fully specified test set \( T_D = (v_1, v_2, v_3, \ldots, v_n) \), where \( v_i \) is the \( i \)-th scan vector, having a lot of long runs of both 0s and 1s. Next, \( T_D \) is transformed to improve compression efficiency by using the neighboring bit-wise exclusive-OR (NB-XOR) transform which performs XOR between neighboring bits on scan vector \( v_j = (b_{j,1}, b_{j,2}, b_{j,3}, \ldots, b_{j,l}) \), where \( b_{j,i} \) is the \( i \)-th bit in it, and this results in the NB-XORed difference vector set \( T_{NB} \).
which is highly biased to 0s, and compression is carried out on it. \( T_{NB} \) is defined as follows:

\[
T_{NB} = \{ d_{NB1}, d_{NB2}, d_{NB3}, \ldots, d_{NBm} \}
\]

\[
d_{NBi} = \{ b_{NBi_1}, b_{NBi_2}, b_{NBi_3}, \ldots, b_{NBi_n-1}, b_{NBi_n} \}
\]

where \( d_{NBi} \) and \( b_{NBi_j} \) are the \( i \)th NB-XORed difference vector and the \( j \)th NB-XORed difference bit in it, respectively. \( b_{NBi_j} \) is obtained by executing XOR between both neighboring bits \( b_{i,j-1} \) and \( b_{i,j} \) in \( v_i \). It is noted that the first difference bit \( b_{NB1,1} \) of \( d_{NB1} \) is equal to \( b_{1,1} \) in the first scan vector \( v_1 \). Fig. 1 shows an example of NB-XOR transform procedure sequenced by \( v, v_D \) and \( d_{NB} \), where \( v, v_D \) and \( d_{NB} \) are a pre-computed scan vector with don't-cares, a fully specified scan vector after don't-care assignment by MTC-filling and an NB-XORed difference vector we achieved at the end, respectively, and the left most bit for each vector is the first bit scanned into a scan chain with length 30. The total number of 1s in \( v_D \) is 21 and this is greatly reduced to a total of 3 as seen in \( d_{NB} \) by NB-XOR technique.

\[
v = 1xxxx111xxxx0000xx00011111xxx
\]

Don't-care assignment by MTC-filling

\[
v_D = 1111111111111000000000111111111
\]

NB-XOR transformation

\[
d_{NB} = 100000000000100000000000010000000
\]

Fig. 1. Example of NB-XOR transform.

Therefore, as shown in Fig. 1, after MTC-filling of don’t-t-cares, NB-XOR scheme can fully exploit the characteristics of the low-power scan vectors due to a number of long runs of both 0s and 1s which have consecutive bit values. Moreover, this technique can take additional advantage of runs of 1s which are specified already before MTC-filling so that, in some cases, there are more 0s in \( T_{NB} \) than in the test data set applied by zero-filling for don’t-t-cares. Finally, the transformed test set \( T_{NB} \) can be substantially compressed by exploiting the test data highly biased to 0s. Details of the compression technique is described in the next subsection.

C. Compression Technique

This section describes the compression technique for the NB-XOR transformed test data set after "don't cares" are mapped by MTC-filling to reduce test power consumption.

The technique proposed is performed by using 0 run-length encoding followed by Huffman encoding.

**Observation**: 0 run-length encoding decreases considerably the volume of the test data including a lot of consecutive 0s, and then more highly compressed test data can be achievable by chopping the 0 run-length compressed data into fixed length blocks and then applying the Huffman encoding technique.

**Justification**: In 0 run-length encoding, the test data having a large number of consecutive 0s can be significantly compressed, and this condition is satisfied by NB-XOR transform of MTC-filled test data. In Huffman encoding, data blocks that occur most frequently have a smaller number of bits, and those that occur least frequently have a larger number of bits, thus eventually the average length of a codeword is minimized. If certain bit blocks can be managed to occur more frequently, the compression ratio will be increased through the Huffman coding. The 0 run-length encoding may produce a compact and well distributed test data set for Huffman encoding. For example, a scan test data "0001000011XX0", don't care bit X is assigned to "1" value according to MTC-filling, and then "000100001111000" is transformed into "000110001000100". Next, this bit stream is
encoded into "11 00 11 11 10" by 0 run-length codes in Fig. 2(a). If "11 00 11 11 10" is chopped into 2-bit blocks and encoded into Huffman codes in Fig. 2(b), and then the final compressed data would be 7 bit of "0110010". In summary, initial 15 bit data is encoded to 10 bit by 0 run-length and finally encoded to 7 bit Huffman codes while the scan test power is kept low.

<table>
<thead>
<tr>
<th>Run-length</th>
<th>Block</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>00</td>
</tr>
<tr>
<td>1</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>001</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>000</td>
<td>11</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>Block (2-bit)</th>
<th>Frequency</th>
<th>Huffman Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

(b)

Fig. 2. Code word tables. (a) 0 Run-length. (b) Huffman.

D. Decompression Architecture Based on NB-XOR Technique

Fig. 3 shows the generic decompression architecture, where $T_E$, $T_{NB}$, and $T_D$ are the compressed scan data, the decompressed NB-XORed difference scan data, and the original low-power scan data, respectively. The 0 run-length and Huffman decoder in the decoder block of Fig. 3 have very simple and well known logic structure just consisting of counters and state machines, and so we focus the inverse NB-XOR transform technique. In Fig. 3, to test the CUT with a single scan chain, first, the compressed NB-XORed difference vectors ($T_E$) from an ATE are transferred to an on-chip decoder which decompresses them into the original NB-XORed difference vectors ($T_{NB}$). And then the decompressed vectors are fed into the inverse NB-XOR block, which consists of a simple pair of one XOR gate and one flip-flop, to transform them into the original low-power scan vectors ($T_D$) under the assumption that the flip-flop is initialized to 0. Finally, the transformed vectors are shifted into the internal scan chain. The serial output of this flip-flop feeds the serial input of the scan chain, and at the same time it loops back and is XORed with the serial output of the decoder.

It can be seen that our technique requires less area overhead than the CSR, since the CSR-based scheme includes separate cyclical scan chain which has the same length of the internal scan chain of the core under test. In addition, our technique does not require any knowledge or modification of the internal structure of the core. Therefore, with very little hardware overhead, the NB-XOR technique enables us to achieve significant test data compression for the low-power scan test data which have a large number of consecutive same bit values.

![Fig. 3. Generic decompression architecture based on NB-XOR technique.](image)

III. Experimental Results

In this section, the experimental results are described for large ISCAS'89 benchmark circuits to evaluate the efficiency of the proposed technique. For each circuit, we have used the full-scan version with a single scan chain.

Table 1 compares the results on the peak and average power consumption during the scan-in operation for two different don’t-care filling techniques of the Zero-filling and MTC-filling. The equations (4) and (5) indicate the average and peak power reduction of the Zero-Filling and MTC-filling, respectively.

\[
P_{\text{avg}}(\text{Zero-filling}) - P_{\text{avg}}(\text{MTC-filling})
\]
\[
\times 100(\%)
\]

\[
P_{\text{peak}}(\text{Zero-filling}) - P_{\text{peak}}(\text{MTC-filling})
\]
\[
\times 100(\%)
\]

In Table 1, the peak and average power consumption during the scan-in operation for two different don’t-care filling techniques of the Zero-filling and MTC-filling. The equations (4) and (5) indicate the average and peak power reduction of the Zero-Filling and MTC-filling, respectively.
As shown in Table 1, the MTC-filling of the don’t-cares leads to more significant savings in average and peak power consumption than the Zero-filling, in all cases.

Basically, most data compression techniques take advantage of the redundancy of data, and thus the scan test data biased can be more efficiently compressed. Table 2 compares the efficiency of biasing to 0s for different transform techniques for MTC-filled test data. In the CSR-based technique, the test vectors were reordered by a heuristic approach to increase the number of 0s. As shown in Table 2, NB-XOR approach outperforms both the pure MTC-filling and CSR-based techniques, in all cases. Thus, our scheme provides better performance for increasing the ratio of 0s, keeping test power low. To estimate compression efficiency, compression ratio is defined as (6).

\[
\frac{(\text{OriginalBits} - \text{BitsAfterCompression})}{\text{OriginalBits}} \times 100\% \quad (6)
\]

Table 3 shows the compression ratios of our hybrid technique by various Huffman block sizes. It is noted that our scheme provides better performance as increasing the chopping size of compressed data by 0 run-length which uses 4 bit code word.

In Table 4, compression ratios of various compression techniques shown are compared with the best compression results of our hybrid technique.

Since the Huffman decoder dominates the area penalty of the whole decompression block, an analysis is performed to check the actual number of states which Huffman decoder must take into account. Table 5 shows the number of states considered by Huffman decoder upon the chopping size of run-length encoded data for circuit s5378. It is noted that the number of states is increasing very gracefully with regard to the block size, thus we were able to implement the Huffman decoder with very small area penalty. For example, gate counts of decoder for s537 synthesized by Synopsys tools with TSMC0.25 library was 2381, and thus we believe the gate counts of s5378 for the drastically high test data compression can be sufficiently affordable to an SoC with millions of gates.

Table 1. Comparison of power consumption results for two different don’t-care filling techniques.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Test Data (bits)</th>
<th>P_{avg}</th>
<th>P_{peak}</th>
<th>P_{avg}</th>
<th>P_{peak}</th>
<th>Power Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s5378</td>
<td>23754</td>
<td>4300</td>
<td>12085</td>
<td>3523</td>
<td>11732</td>
<td>18.07 2.92</td>
</tr>
<tr>
<td>s9234</td>
<td>39273</td>
<td>6705</td>
<td>15395</td>
<td>4002</td>
<td>14092</td>
<td>40.31 8.46</td>
</tr>
<tr>
<td>s13207</td>
<td>165200</td>
<td>131271</td>
<td>110129</td>
<td>8073</td>
<td>94879</td>
<td>34.46 13.85</td>
</tr>
<tr>
<td>s15850</td>
<td>76986</td>
<td>194484</td>
<td>136114</td>
<td>70875</td>
<td>105912</td>
<td>30.01 15.99</td>
</tr>
<tr>
<td>s38417</td>
<td>164736</td>
<td>194843</td>
<td>118098</td>
<td>437884</td>
<td>145858</td>
<td>39.39 14.93</td>
</tr>
<tr>
<td>s38584</td>
<td>199104</td>
<td>133322</td>
<td>50464</td>
<td>481158</td>
<td>199834</td>
<td>35.39 9.29</td>
</tr>
</tbody>
</table>

Table 2. Comparison of zero ratios for different Don’t-care filling and transformation techniques.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Test Data (bits)</th>
<th>Don’t-Cares (%)</th>
<th>MTC-filling</th>
<th>CSR[9]</th>
<th>NB-XOR</th>
<th>Ratios of ’0’(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s5378</td>
<td>23754</td>
<td>72.62</td>
<td>49.83</td>
<td>79.94</td>
<td>86.87</td>
<td></td>
</tr>
<tr>
<td>s9234</td>
<td>39273</td>
<td>73.01</td>
<td>48.71</td>
<td>78.10</td>
<td>87.54</td>
<td></td>
</tr>
<tr>
<td>s13207</td>
<td>165200</td>
<td>93.15</td>
<td>45.93</td>
<td>85.52</td>
<td>96.72</td>
<td></td>
</tr>
<tr>
<td>s15850</td>
<td>76986</td>
<td>83.56</td>
<td>49.65</td>
<td>77.14</td>
<td>92.70</td>
<td></td>
</tr>
<tr>
<td>s38417</td>
<td>164736</td>
<td>68.08</td>
<td>46.32</td>
<td>75.24</td>
<td>91.66</td>
<td></td>
</tr>
<tr>
<td>s38584</td>
<td>199104</td>
<td>82.28</td>
<td>52.84</td>
<td>64.58</td>
<td>91.83</td>
<td></td>
</tr>
<tr>
<td>average</td>
<td>-</td>
<td>78.78</td>
<td>50.38</td>
<td>76.75</td>
<td>91.22</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Compression ratios by our hybrid technique (%).

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Test Data (bits)</th>
<th>4bit</th>
<th>6bit</th>
<th>8bit</th>
<th>10bit</th>
<th>12bit</th>
<th>14bit</th>
<th>16bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>s5378</td>
<td>48.7</td>
<td>49.9</td>
<td>51.8</td>
<td>52.5</td>
<td>56.2</td>
<td>57.7</td>
<td>62.8</td>
<td></td>
</tr>
<tr>
<td>s9234</td>
<td>45.9</td>
<td>46.2</td>
<td>47.7</td>
<td>50.3</td>
<td>53.0</td>
<td>56.1</td>
<td>61.6</td>
<td></td>
</tr>
<tr>
<td>s13207</td>
<td>79.2</td>
<td>79.7</td>
<td>80.9</td>
<td>81.4</td>
<td>82.5</td>
<td>82.9</td>
<td>84.8</td>
<td></td>
</tr>
<tr>
<td>s15850</td>
<td>62.6</td>
<td>64.3</td>
<td>65.2</td>
<td>68.1</td>
<td>68.6</td>
<td>70.3</td>
<td>74.0</td>
<td></td>
</tr>
<tr>
<td>s38417</td>
<td>59.9</td>
<td>60.1</td>
<td>62.5</td>
<td>62.9</td>
<td>65.4</td>
<td>66.2</td>
<td>69.6</td>
<td></td>
</tr>
<tr>
<td>s38584</td>
<td>59.7</td>
<td>60.7</td>
<td>61.5</td>
<td>62.3</td>
<td>63.5</td>
<td>64.3</td>
<td>68.0</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Compression ratios by different compression strategies (%).

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Existing Compression Schemes</th>
<th>Hybrid(16bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s5378</td>
<td>48.0</td>
<td>55.1</td>
</tr>
<tr>
<td>s9234</td>
<td>43.6</td>
<td>54.2</td>
</tr>
<tr>
<td>s13207</td>
<td>81.3</td>
<td>77.0</td>
</tr>
<tr>
<td>s15850</td>
<td>66.2</td>
<td>66.0</td>
</tr>
<tr>
<td>s38417</td>
<td>43.3</td>
<td>59.0</td>
</tr>
<tr>
<td>s38584</td>
<td>60.9</td>
<td>64.1</td>
</tr>
</tbody>
</table>
Table 5. Number of states for Huffman encoding.

<table>
<thead>
<tr>
<th>Block size</th>
<th>4bit</th>
<th>6bit</th>
<th>8bit</th>
<th>16bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Possible States</td>
<td>16</td>
<td>64</td>
<td>256</td>
<td>65536</td>
</tr>
<tr>
<td>Actual Number of States</td>
<td>16</td>
<td>60</td>
<td>108</td>
<td>294</td>
</tr>
<tr>
<td>State Appearance Ratio (%)</td>
<td>100</td>
<td>93.75</td>
<td>42.19</td>
<td>0.45</td>
</tr>
</tbody>
</table>

### IV. Conclusions

In this paper, a new test compression/decompression technique is introduced to reduce test data volume and scan test power consumption, simultaneously. In the experimental results, considerable reduction of test data volume and scan test power is achieved. Therefore, our approach can provide significant reductions in test data volume and, at the same time, the power consumption.

### Acknowledgments

This work was supported by Brain Korea 21.

### References


