1. Introduction

Full-scan is a widely adopted technique for design and testing of LSI chips. In a full-scan sequential circuit, the memory elements such as D flip-flops (DFFs) are replaced with scan cells, e.g., multiplexed DFFs; thus the memory elements become pseudo primary inputs (PPIs) and pseudo primary outputs (PPOs). This scheme dramatically reduces the difficult problem of sequential automatic test pattern generation (ATPG) to the relatively easy problem of combinational ATPG; therefore, the test generation time and the test set quality are greatly improved.

Although the full-scan technique is commonly supported by commercial tools, the biggest disadvantage is that it suffers from large test data volume, excessive test power dissipation, and prolonged test application time due to scan operations. Here, these three problems are combined into the concept of the "VPT issues". As the integration scale doubles every eighteen months, the VPT issues of scan design are becoming more and more serious.

Large test data volume requires large channel capacity of the automatic test equipment (ATE). High power dissipation during scanning test data can cause circuit reliability hazards, instant circuit damage [1], or faulted test results [2]. Long test application time increases the chip test cost.

Although many methods have been proposed to tackle the VPT issues in recent years, most of them focused only on one or two of the VPT issues, instead of addressing all of them simultaneously.

Compression methods such as run-length coding [3], statistical coding (SC) [4], Golomb coding [5], frequency-directed-run-length coding (FDR) [6], alternating run-length coding using FDR (ARL) [7], extended-frequency-directed-run-length coding (EFDR) [8], variable-length input Huffman coding (VIHC) [9], and variable-tail coding [10] reduce test data volume by encoding the original test set \( T_D \) provided by the circuit vendor to a smaller test set \( T_E \). \( T_E \) is stored in the memory of ATE channels. During scan testing, \( T_E \) is transferred to an on-chip decoder. Then \( T_E \) is reverted to \( T_D \) by the on-chip decoder. [11]–[13] compress the test data to seeds, then a linear mapping network, which can be either an LFSR or an XOR network, decompresses the seeds to \( T_D \). Since the data volume of either \( T_E \) or seeds is less than that of \( T_D \), these compression methods reduce the requirement for the ATE channel capacity. Optimally mapping unspecified bits in test data to 0's or 1's [7] can reduce scan-in and scan-out power dissipation as well.

Scan design methods [14]–[17] reduce test data volume and test application time by partitioning and broadcasting. Scan chains are partitioned into shorter segments, and the same test data is broadcasted to multiple segments. When the grain of segments is a scan cell, the scan structure is named a scan tree in [15]–[17] or a scan forest in [18]. VirtualScan [19] is also a broadcasting design with a broadcast network to reduce the correlation in test data. In [20], scan-in power dissipation is saved by freezing scan chain segments, and test data volume is reduced by using the test response to generate the next test stimuli. In [21] and [22], the test response is also reused to generate the next stimuli with dynamic reconﬁgurable scan chains and circular scan. [23] reordered the test data to reduce the Hamming distance between the response vector and the next stimuli vector, and utilizes the random access scan (RAS) [24] to reduce VPT simultaneously. Since [23] is based on the test vector reordering technique and the RAS structure, we denote the technique proposed in [23] as RRAS.

Each issue in the VPT problem is so critical that all three must be solved. Our goal is to reduce all the VPT parts simultaneously. This is achieved by reducing test data volume with variable-to-fixed run-length coding and reducing scan-in power dissipation as well as test application time.

**SUMMARY** LSI testing is critical to guarantee chips are fault-free before they are integrated in a system, so as to increase the reliability of the system. Although full-scan is a widely adopted design-for-testability technique for LSI design and testing, there is a strong need to reduce the test data volume, scan-in power dissipation, and test application time (VPT) of full-scan testing. Based on the analysis of the characteristics of the variable-to-fixed run-length coding technique and the random access scan architecture, this paper presents a novel design scheme to tackle all VPT issues simultaneously. Experimental results on ISCAS' 89 benchmarks have shown on average 51.2%, 99.5%, 99.3%, and 85.5% reduction effects in test data volume, average scan-in power dissipation, peak scan-in power dissipation, and test application time, respectively.

**key words**: compression, run-length coding, random access scan, power dissipation, test application time
with RAS structure. A heuristic algorithm is proposed to arrange the scan cells in RAS to obtain a high compression ratio.

The rest of this paper is organized as follows: Section 2 first describes the characteristics of the variable-to-fixed run-length coding and the RAS structure, respectively, then presents a compression/scan co-design (CSCD) scheme. Section 3 analyzes the VPT reduction effects. Experimental results on larger ISCAS’89 benchmark circuits are shown in Sect. 4, and Sect. 5 concludes this paper.

2. CSCD Scheme

2.1 Variable-to-Fixed Run-Length Coding

In variable-to-fixed run-length coding, the number of consecutive 0’s is encoded as a fixed-length codeword. The fixed-length is denoted as the block length $b$. Since run-length is the number of consecutive 0’s, it is suitable for compressing skewed test data whose 0-probability exceeds 1-probability. In fact, for the stuck-at fault model, an important property of the test data set $T_D$ generated by ATPG is that $T_D$ usually has very high ratio of unspecified bits. These unspecified bits can be randomly mapped to either logic value 0 or 1 without any fault coverage loss. By mapping the unspecified bits to 0’s, the test data can be well skewed. Table 1 shows an example of variable-to-fixed run-length coding with block length $b=3$.

Obviously, the last few bits unable to be encoded are all 0’s because the run-length coding always requires that the last bit be 1 except that the run-length equals $2^b - 1$. In [3], when the last few bits at the end of $T_D$ can not be encoded, extra bits are padded to solve the problem. In the case of the 3-bit run-length coding, for example, if the last two bits are 00, then the codeword 111 will be used to encode the bits. Unlike the padding method of [3], our CSCD run-length coding technique encodes the last few bits as the number of 0’s. For the above example, 00 will be encoded as 010. The CSCD decoder is specially designed to correctly decode such codeword, e.g. 010 will be reverted to 00.

It is clear from Table 1 that, the characteristic of variable-to-fixed run-length coding is that the relative address (RA) of 1 in test data is the run-length plus one. The absolute address (AA) is defined as the number of bits from the designated bit to the first bit. For each 1 in test data, we get its AA by adding its RA and AA of the last bit of the mostly recently decoded test data. For example, let $T_D = 00001001$.

On the other hand, scan cells in the conventional serial scan chain have to be accessed serially. The characteristic of the RAS structure is that any of its scan cells can be selected directly by the address decoders. This characteristic of RAS can be naturally incorporated with the characteristic of the run-length coding. Whenever the absolute address of 1 in test data is provided to the address decoder, the state of the corresponding scan cell is flipped. In the following sections, such incorporation will be described in more details.

2.2 CSCD Architecture

As shown in Fig. 1, the CSCD architecture consists of three parts: Decoder, Address Decoder, and RAS. The Decoder part has an Address Register Bank (ARB), an Adder, a Unit Adder, two MUXs (MUXa and MUXp), two AND gates (ANDa and ANDp), an OR gate, an OR gate and a NOR gate. The Address Decoder part has a Counter, an X Address Decoder, and a Y Address Decoder. The RAS part is constructed with scan cells of the circuit-under-test. The unique technique used here is that the address $y_0$ is left unused. This is specially designed to handle the case that the last few bits in $T_D$ are all 0’s.

Scan cells are classified into two blocks according to the 0-probability and 1-probability of test data. When the 0-probability of a scan cell is greater than its 1-probability, the scan cell is classified to RAS0; otherwise, it is classified to RAS1. The purpose of this classification is to reduce the state transitions during setting test data. The states of the scan cells that are most likely to have the test bit 0 are initialized to 0, and the states of the scan cells that are most likely to have the test bit 1 are initialized to 1. For the sake of clarity, the scan cells are placed regularly in Fig. 1; in practice, however, they can be distributed in the circuit-under-test.

Let the size of RAS be $(r + 1) \times c$, where $r + 1$ is the number of rows of RAS and $c$ is the number of columns. $c$ is set to an integral power of 2 minus 1. It is clear that any scan cell can be accessed by using a $[\log_2 (r + 1)]$-bit X Address Decoder and a $[\log_2 (c + 1)]$-bit Y Address Decoder. An advantage of RAS is that the clock skew becomes less critical now, because there is no need to shift the test data along the connected scan cells one by one. Primary inputs (PIs) are multiplexed to apply the encoded test data. To avoid transitions of Decoder and Address Decoder in the function mode, two MUXs (MUXC and MUXD) are inserted after PIs. Two extra pins are needed in CSCD architecture, the one is TC, and the other one is RST. TC is the signal that selectively sets the circuit-under-test into function mode or test mode. RST resets RAS0 to 0 and RAS1 to 1 whenever it
The RAS\(_0\) and RAS\(_1\) scan cells are different from conventional multiplexed DFFs. An AND gate and an OR gate are inserted in front of the MUX as illustrated in Fig. 2. Comparing the CSCD RAS scan cell design with the conventional multiplexed DFF design makes it clear that, since the AND gate and the OR gate are not on the signal propagation path, the CSCD RAS scan cell design has the same impact on the delay as the conventional multiplexed DFF.

Table 2 shows how TC, the address select signal \((x_i,y_j)\), and RST together determine the operation mode of a RAS scan cell. When TC=1, scan cells are in the function mode. When TC=0, the scan cells are in the test mode. In the test mode, when RST=1, RAS\(_0\) scan cells are reset to 0's and RAS\(_1\) scan cells are reset to 1's. For a RAS\(_0\) scan cell, once it is selected by the X and Y address decoders, its state is set to 1; for a RAS\(_1\) scan cell, once it is selected by the X and Y address decoders, its state is set to 0. The unselected scan cells hold their present states. The hold mode avoids the transition at the scan cell from rippling to the combinational logic. Obviously, this helps reducing the extra power dissipation of the combinational logic.

The stuck-at-faults in Decoder, Address Decoder, RAS and the response analyzer (e.g. XOR net and MISR) can be tested by steps as follows: 1) RAS\(_0\) scan cells are initialized to 0's and RAS\(_1\) scan cells are initialized to 1's by RST=1 and TC=0. 2) The response analyzer (e.g. XOR net and MISR) compacts the values of RAS to a signature, and outputs the signature to ATE for comparison. 3) Set RST=0 and TC=0, select scan cells one by one with ascending order, then the states of RAS\(_0\) scan cells are 1's, and the states of RAS\(_1\) scan cells are 0's. 4) The response analyzer again compacts the values of RAS and outputs the signature to ATE for comparison. After the DFT-circuitry is tested, conventional test data generated by ATPG tools can be encoded by run-length coding, then transferred to PIs to test the circuit-under-test.
2.3 Scan Cell Arrangement

In the CSCD scheme, the original test data $T_D$ provided by the circuit vendor must be partitioned and reordered to $T_{D0}$ and $T_{D1}$ according to the addresses of scan cells, since scan cells have been classified to two blocks: RAS0 and RAS1. $T_{D0}$ and $T_{D1}$ are two non-overlapping sets with $T_{D0}$ for RAS0 and TDI for RAS1. All unspecified bits in $T_{D0}$ are mapped to 0's and all unspecified bits in $T_{D1}$ are mapped to 1's. Therefore, the test data $T_{D0}$ and $T_{D1}$ are skewed more effectively than simply randomly mapping unspecified bits in $T_D$ to 0's or 1's. Obviously, this way of unspecified bits mapping is good for gaining better encoding effect. Then $T_{D0}$ is encoded to the variable-to-fixed run-length code $T_{EO}$. To simplify the encoding and the decoding processes, $T_{D1}$ will be inverted to $T_{E1}$ first, then encoded to $T_{E1}$ in the same way as $T_{D0}$ been encoded.

Since the original test data $T_D$ will be reordered according to the address of RAS scan cells before being encoded. The arrangement of scan cells determines the encoding effect. Obviously, enumerating all possible scan cell permutations to get the minimum size of the encoded test set is infeasible since for $k$ scan cells there are $k!$ such permutations. As a result, a heuristic algorithm is proposed to near-optimally arrange scan cells. In the case of RAS0 scan cell arrangement, for example, suppose that there are $k_0$ scan cells. We have a matrix of $p_0 \times q$ ($k_0 \leq p_0 \times q$) empty boxes where we can put scan cells. One box can have at most one scan cell. From Table 1, we can see that for the variable-to-fixed run-length coding, the best encoding effect can be attained when a codeword presents a $2^b - 1$ bits original data. Moreover, the bigger the block size $b$, the more significant the encoding effect.

The steps of the scan cell arrangement algorithm are as follows:

1. Sort the 1-probability of the scan cells in descending order.
2. Choose the scan cells, whose 1-probability values are no less than a threshold value, as “anchor” scan cells.
3. Distribute the anchors uniformly in the box matrix, and make sure that the distance between two neighbor anchors in a row is the block size of the run-length coding. If the result of the bitwise-AND operation is 1, the output of UnitAdder is provided to the arbiter through MUXa. No scan cell is selected because the content of ARB is 0. Therefore, the output of Adder is directly provided to ARB through MUXb. If the result of the bitwise-AND operation on the codeword with 1’s is 0, the output of NOT is 1. Then the output of UnitAdder is provided to the Y Address Decoder through MUXa, as well as to ARB through MUXb.

The test data decompression procedure of CSCD scheme is as follows: in Fig. 1, when a run-length codeword is transferred to PIs, Adder adds the codeword with the content of ARB. Then UnitAdder increases the output of Adder by one. At the same time the ANDa gate conducts the bitwise-AND operation on the codeword with 1’s. If the result of the bitwise-AND operation at ANDa is 0, the output of NOT is 1. Then the output of UnitAdder is provided to the Y Address Decoder through MUXa, as well as to ARB through MUXb. If the result of the bitwise-AND operation is 1, meaning that the codeword contains only 1’s. Therefore, the output of Adder is directly provided to ARB through MUXb while 0’s are provided to the Y Address Decoder through MUXa. No scan cell is selected because the first address $y_0$ is left unused. It can be seen that ARB stores the absolute Y address of the last bit of the test data mostly recently decoded.

When the last codeword of a row is transferred to PIs,
for the corresponding test data in $T_{DO}/T_{DI}$, there are two possible cases: (1) the last bit is 1 and (2) the last bit is 0.

For the first case, the output of UnitAdder is the column address of the scan cell at the end of the row. Since this address is an integral power of 2 minus 1, it means that the output of AND$_b$ is 1. Therefore INC is asserted. At the next rising edge of the test clock, ARB is reset to 0 by the output of the NOR gate, and the asserted INC let Counter increase by 1. The output of the X Address Decoder points to the next row of RAS. For the second case, UnitAddress is overflow and the carry-out bit (CO) is set to 1. In this case, ARB is reset to 0 and Counter increases by 1.

After the last row of a $T_{E1}$ vector is decompressed, TC is asserted to make RAS operate in function mode, and the test data is applied to the circuit-under-test. Then the test response is captured and compacted by an XOR network and MISR. After that, RAS$_0$, ARB and Counter are reset to 0 and RAS$_1$ is reset to 1.

Figure 4 is an example to illustrate how the CSCD scheme decompresses the encoded test data. Assume that scan cells have been classified into a 2-row-by-15-column RAS$_0$ and a 1-row-by-15-column RAS$_1$. Since the number of rows and columns is 3 and 15, respectively, the widths of Counter and ARB are 2-bits and 4-bits, respectively. RAS$_0$, ARB, and Counter are initialized to 0, and RAS$_1$ is initialized to 1. In Fig. 4, a test vector of $T_{DO}/T_{DI}$ is shown in RAS$_0$/RAS$_1$, which is encoded to codeword$_1$ through codeword$_9$. The column of "Pls" indicates the codeword transferred to Pls, the column of "(x, y)" indicates the X-address and the Y-address of the selected scan cell. "ARB" and "Counter" indicates the values of ARB and Counter, respectively, when the next codeword is transferred to Pls. The straight lines under the values of (x, y), ARB, and Counter emphasize that ARB is reset to 0, and Counter increases by 1.

### 3. Effect Analysis

#### 3.1 Test Data Volume Reduction

In this work, the test data volume is reduced by the run-length coding. Since the encoding and decoding of $T_{DI}$ is conducted in the same manner as that of $T_{DO}$, we only analyze the test data volume reduction with $T_{DO}$ in the following.

According to the theorem of the variable-to-fixed run-length coding [25], the size of the encoded test data $T_{E0}$ is given by

$$RL_{min} = \frac{bn}{2^b - 1}$$

where, $0 \leq \delta_i \leq \frac{2^b - 1}{n}$, $n$ is the total number of bits in $T_{DO}$, $h$ is the total number of 1's in $T_{DO}$, $b$ is the block size, and $\delta_i$ is the run-length in the $i$-th block. From this formula, for a given test data set, it is clear that a bigger block size $b$ and a smaller $\delta_i$ result in shorter $RL$. If each $\delta_i + 1$ is integral times of $2^b - 1$, then $\delta_i = 0$.

This is the lower bound of the variable-to-fixed run-length coding.

The encoding effect is indicated by the compression ratio. The compression ratio is given by

$$CR = \frac{\text{No. of bits in } T_D - \text{No. of bits in } T_E}{\text{No. of bits in } T_D} \times 100\%$$

From the discussions above, its is clear that the upper bound of compression ratio is given by

$$CR_{max} = \frac{n - \text{Rel}_{min}}{n} \times 100\%$$

#### 3.2 Scan-in Power Dissipation Reduction

The scan power dissipation includes both scan-in and the scan-out power dissipation. For the traditional scan chain architecture, the scan-in procedure and the scan-out procedure are symmetric. However, many low power compression techniques (e.g. ARL in [7]) usually only operate against the test stimuli data but not the test response data, which results in that the scan-in power dissipation is usually less than the scan-out power dissipation.

On the other hand, for the CSCD architecture, the scan-in procedure and the scan-out procedure is asymmetric. The test responses are directly transferred to MISR through the XOR network, hence there are no transitions on the scan cells and the combinational circuit-under-test during the scan-out procedure. Therefore, only the XOR network and the MISR contribute to the scan-out power dissipation.

Since the XOR network is a small combinational circuit and MISR is relatively shorter than the rows of RAS,
the scan-out power dissipation is less than the scan-in power dissipation. In this work, we analyze the scan-in power dissipation, but it is reasonable to infer that if the scan-in power dissipation of CSCD is less than that of the techniques based on the scan chain architecture, the CSCD architecture will also outperform those techniques in the scan-out power dissipation.

The weighted transition metric (WTM) has been shown to be strongly correlated with the power dissipation in the circuit during scan operation [26]. Therefore, we estimate the scan-in power dissipation with WTM. The weight assigned to a transition is the difference between the size of the scan chain and the position in the vector in which the transition occurs. For a test vector, the number of weighted transitions is given by

$$\text{Weighted Transitions} = \sum (\text{Size of Scan Chain} - \text{Position of Transition})$$

And the total scan-in power dissipation (PD) of the conventional serial single scan design is the sum of WTM of the test data $T_d$.

$$PD = \sum_{i=1}^{[T_d]} \text{Weighted Transitions}_i$$

where $[T_d]$ is the number of test vectors in $T_d$. The average scan-in power dissipation (APD) and the peak scan-in power dissipation (PPD) are given by

$$\text{APD} = PD/[T_d]$$

$$\text{PPD} = \max \{ \text{Weighted Transitions}_i \}_{i=1,\ldots,[T_d]}$$

For the CSCD architecture, the total scan-in power dissipation has no relation with the position of transitions and is equal to the sum of the number of 1’s in $T_{DO}$ and the number of 0’s in $T_{DI}$. Therefore, PD, APD, and PPD are given by

$$PD = \sum_{i=1}^{[T_d]} (T_{DO}^i(1) + T_{DI}^i(0))$$

$$\text{APD} = PD/[T_d]$$

$$\text{PPD} = \max (T_{DO}^i(1) + T_{DI}^i(0))_{i=1,\ldots,[T_d]}$$

where $T_{DO}^i(1)$ denotes the number of 1’s in the $i$-th vector of $T_{DO}$, and $T_{DI}^i(0)$ denotes the number of 0’s in the $i$-th vector of $T_{DI}$.

### 3.3 Test Application Time Reduction

For a conventional serial single scan design, the test application time is the product of the length of the scan chain and the number of the test vectors in $T_d$ [27]. However, for the CSCD architecture, one cycle is needed to set the state of the selected RAS scan cell. Therefore, the number of codeword in a $T_{E0}$ vector is the number of test cycles needed to load the $T_{DO}$ vector. For the same reason, the number of codeword in a $T_{E1}$ vector is the number of test cycles needed to load the $T_{DI}$ vector. After both $T_{DO}$ and $T_{DI}$ vectors are loaded into RAS$_0$ and RAS$_1$, one cycle is needed to capture the test response, and one more cycle is needed to reset RAS$_0$ to 0 and RAS$_1$ to 1. Therefore, the total test application time of CSCD is given by

$$\text{TAT} = \sum_{i=1}^{[T_E]} (T_{DO}^i(\text{codeword}) + T_{DI}^i(\text{codeword}) + 2)$$

where $[T_E]$ is the number of test vectors in $T_E$. Since the variable-to-fixed run-length coding does not reduce the number of test vectors, $[T_E]$ is equal to $[T_d]$.

For the instance of Fig. 4, a test vector of $T_d$ is encoded to codeword$_1$ through codeword$_9$. Among these codewords, codeword$_i$ to codeword$_j$ are belong to $T_{E0}$, while codeword$_j$ to codeword$_k$ are belong to $T_{E1}$. Therefore, we need 9 cycles to load 9 codewords, and need 2 more cycles to capture the test response and to reset RAS. Hence TAT is 11.

### 4. Experimental Results

Experiments were performed on the full-scan version of ISCAS’89 benchmark circuits to evaluate the proposed CSCD scheme. The test data generated by MinTest ATPG program [28] with full fault coverage were used in our experiments.

Since the threshold value in our heuristic algorithm impacts the number of anchors and the size of encoded test data, we tried the threshold value from 0.05 to 0.45 with a step of 0.05. The number of RAS columns also impacts the arrangement of scan cells; therefore we tried the number 31, 63, 127, 511, and 1023. The experiments were implemented in Matlab version 6.1, executed on a PC with a Pentium 4 2.6 GHz processor and 256 MB memory. All experiments were completed in several seconds of CPU time.

Table 3 shows the impact of the threshold on the test data volume (V), the scan-in power dissipation (P), and the test application time (T) when the number of RAS columns is 63. The row of “Threshold” is the threshold values that achieved the best compression ratio. Row “No. of bits in $T_d$” and “No. of bits in $T_E$” are the number of total bits in $T_d$ and $T_E$ for each benchmark circuit, respectively. Row “No. of specific bits in $T_d$” is the number of specific bits in $T_d$. APD, PPD, and TAT are the average scan-in power dissipation, the peak scan-in power dissipation, and the test application time. Rows of “CSCD” are experimental results of the proposed co-design scheme. The rows of “SS” are experimental results of the conventional single serial scan design.

From Table 3, it can be seen that the CSCD scheme achieved up to 81.20%, 99.85%, 99.80%, and 96.60% reduction in test data volume, average scan-in power dissipation, peak scan-in power dissipation and test application time, respectively. Among the seven benchmark circuits, the compression ratio of s13207 was the highest because its
Table 3  The impaction of the threshold to VPT.

<table>
<thead>
<tr>
<th>Circuit Name</th>
<th>Threshold</th>
<th>No. of bits in $T_D$</th>
<th>No. of specific bits in $T_D$</th>
<th>No. of bits in $T_E$</th>
<th>No. of specific bits in $T_E$</th>
<th>APD</th>
<th>PPD</th>
<th>TAT</th>
<th>Compression ratio (%)</th>
<th>APD reduction (%)</th>
<th>PPD reduction (%)</th>
<th>TAT reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s3578</td>
<td>0.35</td>
<td>23,754</td>
<td>39,273</td>
<td>165,200</td>
<td>66,257</td>
<td>2139</td>
<td>80</td>
<td>2855</td>
<td>46.62</td>
<td>98.81</td>
<td>94.81</td>
<td>88.14</td>
</tr>
<tr>
<td>s9234</td>
<td>0.25</td>
<td>39,273</td>
<td>43,524</td>
<td>165,200</td>
<td>71,731</td>
<td>2969</td>
<td>96</td>
<td>5648</td>
<td>91.20</td>
<td>94.81</td>
<td>94.81</td>
<td>91.20</td>
</tr>
<tr>
<td>s15850</td>
<td>0.25</td>
<td>71,731</td>
<td>43,524</td>
<td>81,85</td>
<td>81,85</td>
<td>3113</td>
<td>31</td>
<td>4668</td>
<td>85.29</td>
<td>94.81</td>
<td>94.81</td>
<td>85.29</td>
</tr>
<tr>
<td>s35932</td>
<td>0.25</td>
<td>81,85</td>
<td>67,99</td>
<td>101,48</td>
<td>67,99</td>
<td>3492</td>
<td>10</td>
<td>15,183</td>
<td>51.52</td>
<td>94.81</td>
<td>94.81</td>
<td>51.52</td>
</tr>
<tr>
<td>s38417</td>
<td>0.30</td>
<td>28208</td>
<td>6572</td>
<td>164,736</td>
<td>6572</td>
<td>98.51</td>
<td>98.51</td>
<td>261496</td>
<td>51.52</td>
<td>94.81</td>
<td>94.81</td>
<td>51.52</td>
</tr>
<tr>
<td>s38584</td>
<td>0.20</td>
<td>164,736</td>
<td>6572</td>
<td>81,207</td>
<td>6572</td>
<td>98.51</td>
<td>98.51</td>
<td>164,736</td>
<td>51.52</td>
<td>94.81</td>
<td>94.81</td>
<td>51.52</td>
</tr>
</tbody>
</table>

Table 4  Comparison of compression ratio with other approaches.

<table>
<thead>
<tr>
<th>Circuit Name</th>
<th>No. of bits in $T_D$</th>
<th>SC</th>
<th>Golomb</th>
<th>FDR</th>
<th>EFDR</th>
<th>VIHC</th>
<th>ARL</th>
<th>RRAS</th>
<th>CSCD (Best)</th>
<th>CSCD (63)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s3578</td>
<td>23,754</td>
<td>34.79</td>
<td>40.70</td>
<td>48.19</td>
<td>51.93</td>
<td>51.52</td>
<td>50.77</td>
<td>57.92</td>
<td>48.15</td>
<td>46.64</td>
</tr>
<tr>
<td>s9234</td>
<td>39,273</td>
<td>35.52</td>
<td>43.34</td>
<td>44.88</td>
<td>48.89</td>
<td>54.84</td>
<td>44.96</td>
<td>56.32</td>
<td>51.33</td>
<td>50.54</td>
</tr>
<tr>
<td>s15850</td>
<td>165,200</td>
<td>77.73</td>
<td>74.78</td>
<td>78.67</td>
<td>81.85</td>
<td>83.17</td>
<td>80.31</td>
<td>82.66</td>
<td>83.11</td>
<td>81.20</td>
</tr>
<tr>
<td>s35932</td>
<td>70,906</td>
<td>40.16</td>
<td>47.11</td>
<td>52.87</td>
<td>67.99</td>
<td>60.68</td>
<td>65.83</td>
<td>60.00</td>
<td>65.80</td>
<td>65.58</td>
</tr>
<tr>
<td>s38417</td>
<td>28,208</td>
<td>65.72</td>
<td>98.51</td>
<td>10.19</td>
<td>80.31</td>
<td>66.47</td>
<td>N/A</td>
<td>25.27</td>
<td>12.44</td>
<td>2.82</td>
</tr>
<tr>
<td>s38584</td>
<td>164,736</td>
<td>37.11</td>
<td>44.12</td>
<td>54.53</td>
<td>60.57</td>
<td>54.51</td>
<td>60.55</td>
<td>58.61</td>
<td>55.49</td>
<td>54.43</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64.49</td>
<td>51.17</td>
</tr>
</tbody>
</table>

Table 5  Comparison of scan-in power dissipation and test application time with other approaches.

<table>
<thead>
<tr>
<th>Circuit Name</th>
<th>APD Reduction(%)</th>
<th>PPD Reduction(%)</th>
<th>TAT Reduction(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ARL</td>
<td>RRAS</td>
<td>CSCD</td>
</tr>
<tr>
<td>s3578</td>
<td>78.02</td>
<td>99.04</td>
<td>98.81</td>
</tr>
<tr>
<td>s9234</td>
<td>76.30</td>
<td>99.14</td>
<td>99.04</td>
</tr>
<tr>
<td>s15850</td>
<td>93.68</td>
<td>99.79</td>
<td>99.63</td>
</tr>
<tr>
<td>s35932</td>
<td>85.27</td>
<td>99.63</td>
<td>99.56</td>
</tr>
<tr>
<td>s38417</td>
<td>93.68</td>
<td>99.99</td>
<td>99.85</td>
</tr>
<tr>
<td>s38584</td>
<td>93.52</td>
<td>99.84</td>
<td>99.81</td>
</tr>
<tr>
<td>Average</td>
<td>83.02</td>
<td>99.56</td>
<td>99.50</td>
</tr>
</tbody>
</table>

The ratio of specific bits is the lowest: 11313/165200 = 0.0685. The lowest compression ratio was for s35932. The reasons were that its ratio of specific bits is the highest: 18251/28208 = 0.6470, and that the number of 0’s and the number of 1’s in $T_D$ were very close. We observed that the standard deviation of 0-probability and 1-probability is 0.1285, which is very small. Table 3 indicates that the variable-to-fixed run-length coding based CSCD is suitable for properly skewed test data.

Table 4 presents the comparison of the proposed CSCD scheme with other compression approaches including SC [4], Golomb [5], FDR [6], EFDR [8], ARL for minimizing WTM [7], and RRAS [23]. The “CSCD (Best)” column is the best compression ratio achieved by the combination of the threshold and the number of RAS columns, and “CSCD (63)” is the optimal compression ratio when the number of RAS columns is 63. The bold entries indicate the best results. “N/A” in row of “s35932” is due to the experimental result of s35932 is not provided in [7].

The row of “Average” indicates that EFDR achieves the smallest encoded test data set. However, it does not address the scan-in power dissipation and test application time issues. Since ARL and RRAS also address the problems of scan-in power dissipation and test application time, we compare the CSCD scheme with these techniques in Table 5. And due to physical design concerns such as place and route, we limited the number of RAS columns to 63. There is no such considerations in RRAS, the experimental results [23] show the number of its RAS columns ranges from 256 to 2048.

In Table 5, the APD, PPD, and TAT Reduction column is the average scan-in power dissipation, peak scan-in power dissipation reduction and test application time reduction, respectively. For the example of APD, the reduction is given by...
The CSCD scheme significantly outperforms ARL in the terms of test power dissipation and test application time, although in the aspect of compression ratio, it did not achieve the best results. ARL needs a test clock that is faster than that of ATE; however, the high test frequency increases the power consumption of the chip-under-test, which explains why the peak power dissipation during scan operation is still high even ARL is optimized to minimize the WTM measure. “N/A” in row of “s35932” is due to the experimental results of s35932 are not provided in [7].

RAS and CSCD have almost the same effect on APD reduction. But in PPD and TAT reductions, the CSCD scheme has much better results than RAS. In CSCD, scan cells with high 0-probability are classified to RAS₀ and scan cells with high 1-probability are classified to RAS₁. Since RAS₀ and RAS₁ are initialized to 0 and 1 before loading in every test stimuli vector, the number of scan cells whose states need to be flipped is small. As a result, the test application time is short. The CSCD scheme can perform good reductions for the VPT issues. At the same time, it ensures full fault coverage without any loss.

In terms of hardware overhead, using Synopsys Design Compiler, we synthesized the serial scan version of the ISCAS’89 benchmark circuits, and also synthesized the CSCD version corresponding to the optimal compression ratio when the number of RAS columns is 63. Then area overhead and routing overhead are given by Synopsys Astro.

In Table 6, the “Area Overhead (µm²)” and “Routing Overhead (µm)” columns show the area overhead and the routing overhead of benchmark circuits designed with serial scan chain and CSCD, respectively. The two “Difference Ratio” columns are the area and routing overhead the CSCD version exceeds the serial scan version. The area overhead of Decoder and Address Decoder parts of the CSCD architecture is indicated in the “D and AD” column. Note Decoder and Address Decoder parts do not include of MUXₓe and MUXₓd. We can see that the area overhead of Decoder and Address Decoder parts increases very slowly with the increasing of benchmark circuit scales. Because the RAS scan cell has an AND gate and an OR gate more than the conventional multiplexed DFF, the area overhead of CSCD scheme is mainly decided by the number of scan cells in the circuit-under-test. And along with the circuits scale increases, the difference between CSCD and SS decreases. As to the routing overhead, the average difference ratio between CSCD and SS is 29.04%. Note for s38584, the routing overhead of CSCD is much smaller than that of SS. It is because the scan cells are close to the address decoders in CSCD for s38584, the wires between the address decoders and scan cells are short.

### Table 6 Hardware overhead of serial scan and CSCD.

<table>
<thead>
<tr>
<th>Circuit Name</th>
<th>Area Overhead (µm²)</th>
<th>Routing Overhead (µm)</th>
<th>Difference Ratio (CSCD-SS)/SS (%)</th>
<th>D and AD (µm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS</td>
<td>CSCD</td>
<td>SS</td>
<td>CSCD</td>
<td></td>
</tr>
<tr>
<td>s5378</td>
<td>71,314.7</td>
<td>92,683.5</td>
<td>29.97</td>
<td>2155.51</td>
</tr>
<tr>
<td>s9234</td>
<td>115,346.2</td>
<td>139,688.8</td>
<td>21.10</td>
<td>2195.42</td>
</tr>
<tr>
<td>s13207</td>
<td>202,614.3</td>
<td>257,154.0</td>
<td>26.92</td>
<td>2295.22</td>
</tr>
<tr>
<td>s15850</td>
<td>218,018.9</td>
<td>269,268.8</td>
<td>23.51</td>
<td>2285.24</td>
</tr>
<tr>
<td>s35932</td>
<td>505,865.6</td>
<td>625,832.2</td>
<td>23.72</td>
<td>2544.70</td>
</tr>
<tr>
<td>s38417</td>
<td>554,204.8</td>
<td>677,178.5</td>
<td>22.19</td>
<td>2514.76</td>
</tr>
<tr>
<td>s38584</td>
<td>513,150.4</td>
<td>615,773.2</td>
<td>20.00</td>
<td>2464.86</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>23.91</td>
<td>-</td>
<td>2355.09</td>
</tr>
</tbody>
</table>

APD Reduction = \(1 - \frac{\text{APD of CSCD}}{\text{APD of Serial Scan}}\) x 100%

### Conclusions

The compression/scan co-design approach provides an efficient solution for testing large integrated circuits and system-on-a-chip. The proposed CSCD scheme achieves the goal of simultaneously reducing test data volume, scan-in power dissipation, and test application time by exploiting the characteristics of both the variable-to-fixed run-length coding and the random access scan architectures. Combined with a heuristic algorithm for an optimal arrangement of scan cells, the proposed scheme on average reduced test data volume, average scan-in power dissipation, peak scan-in power dissipation, and test application time by 51.2%, 99.5%, 99.3%, and 85.50%, respectively, in our experiments on ISCAS’89 benchmark circuits.

As future work, we are planning to make more comparisons with other scan-based compression techniques and reduce the hardware overhead.

### Acknowledgments

This work is supported in part by National Natural Science Foundation of China (NSFC) under the grant No. 90207002, and also by the National Basic Research Program of China (973 Program) under the grant No. 2005CB321604. The authors would like to thank their colleague Ge Zhang for performing place and route experiments.

###References


Huawei Li received her B.S. in computer science from Xiangtan University (China) in 1996, and M.S. and Ph.D. degrees from Institute of Computing Technology, Chinese Academy of Sciences in 1999 and 2001 respectively. She is now an associate professor at the Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include VLSI/Soc design verification and test generation, delay test, and dependable computing. She is a member of IEEE.

Xiaoqing Wen received the B.E. degree from Tsinghua University, Beijing China, in 1986, the M.E. degree from Hiroshima University, Hiroshima, Japan, in 1990, and the Ph.D degree from Osaka University, Osaka, Japan, in 1993. From 1993 to 1997, he was a Lecturer at Akita University. He was a Visiting Researcher at University of Wisconsin, Madison, U.S.A., from Oct. 1995 to March 1996. He joined SynTest Technologies, Inc., U.S.A., in 1998, and served as its CTO until 2003. From 2004, he has been an Associate Professor at Kyushu Institute of Technology, Iizuka, Japan. His research interests include VLSI test, diagnosis, and testable design. He is a member of IEEE.