Test Slice Difference Technique for Low-Transition Test Data Compression

Jiann-Chyi Rau, Po-Han Wu and Wei-Lin Li

Department of Electrical Engineering, Tamkang University, Tamsui, Taiwan 251, R.O.C

Abstract

This paper presents a low power strategy for test data compression and a new decompression scheme for test vectors. In our method, we propose an efficient algorithm for scan chain reordering to deal with the power dissipation problem. Further, we also propose a test slice difference (TSD) technique to improve test data compression. It is an efficient method and only needs one extra scan cell. In experimental results, the scheme that we presented achieve high compression ratio. The power consumption is also better compared with other well-known compression techniques.

***Key Words*:** Test Data Compression, Low Power Testing, VLSI, Design for Testability (DFT)

1. Introducion

Generally, the methods can be classified into two categories, test vector compaction [1,2] and test data compression. The latter category is extensively used today. According to [3], test data compression can roughly be divided into three kinds, namely code-based schemes [4,5], linear-decompression-based schemes [2,6], and broadcast-scan-based schemes [7,8]. The code-based schemes only depended on the test cube and don't loss any fault coverage.

Huffman code is typically used for the fixed-to-variable coding scheme. Because Huffman code is an efficient code for encoding test data, it has been used in many compression strategies [9−13]. Selective Huffman code [10] adds symbols with higher appearance frequency to the Huffman tree and encodes them by Huffman codes. In [11], a compression technique for encoding three different kinds of data using the same Huffman codeword is proposed. Selective Huffman coding strategy is also used there to reduce the hardware overhead of Huffman FSM. Although this method can yield a good result, because it needs to deal with three kinds of different information, the hardware cost is still very high.

*Corresponding author. E-mail: jcrau@ee.tku.edu.tw

Block merging technique [4] is an efficient scheme for reducing test data volume. The encoded data contains merged blocks data and the merged blocks count. Nine-Coded [14] is a well-known method based on fixed-to-variable coding scheme and uses nine-types of codeword to encode the test data. Unfortunately, with the increased length of symbol, the method becomes less attractive. Other common methods are based on run-length coding such as Golomb code [15], frequency-directed run-length (FDR) code [16], variable-input Huffman compression (VIHC) code [9], extended FDR (EFDR) code [17], alternating run-length (Alternating RL) code [5], and mixed run-length and Huffman (RL-Huffman) code [13]. These methods will be briefly reviewed in section 2.

Another issue to be explored, in test mode, power consumption is very high, and excessive heat dissipation can cause damage to the circuit. Hence, many ways have been proposed to solve this problem. These techniques include scan chain modification [18], scan chain reordering [19], input blocking [20], low power test pattern generation algorithm [21], vector reordering [22], and X-filling technique [23]. The authors in [18] proposed a useful technique that modifies scan chain to reduce shifting-in power. The approach in [19] indicates that re-ordering scan cells can reduce shifting-in power, and the routing cost will be reduced by commercial tools. The

technique in [20] calculates the signal probabilities at the gate outputs to find the frozen value. Therefore, the shift power reduction can be reduced by blocking the above scan cells with the frozen value.

In this paper, we propose a test slice difference (TSD) technique for power-aware test data set. The proposed technique only needs one bit buffer to hold preceding data. In addition, we shall also propose a scan-chain reordering algorithm. Compared with the forward reordering algorithm, like SkewedSC [24] and Run-BasedReordering [25], our method achieves better results in both power dissipation and compression ratio.

2. Previous Works

2.1 Compression Issues

(a) Filling unspecified bits

How to fill the unspecified bits is the direction to improve the compression ratio. Most methods fill unspecified bits as the same value as their neighbors. All the unspecified bits in test data are filled into zeroes [26]. Even though this process can reduce the peak power, the total power consumption is also very high during test mode.

(b) Change entropy of the original test data

In [27], it is indicated that the maximum compression ratio can be predicted by calculation of the test data entropy. First, we can use a different ATPG procedure to generate needed test data. Second, we can find better ways to change entropy of the original test data to change the maximum compression ratio limit.

(c) Scan chain reordering

Scan chain reordering was generally used to minimize the length of interconnect wire. Which can reduce both test power and test data volume, but it also increases the routing overhead during physical design. Now, it is applied to reduce test data volume [24,25].

3. Proposed Decompression Architecture

Test power reduction can be solved by low transition filling. For this reason, we add a de-TSD (test slice difference) to hold status (0 or 1). The de-TSD is only made of one D-FF and one MUX. If transition occurs, we use the data from ATE to transform the status. We call this transition position as changed point.

Clearly, we only need to store changed point in ATE. Changed point can be obtained easily by using test slice difference. On the other hand, in order to reduce data size and avoid redundant bits in each vector, we use "Similar Filter" to filter test slices which are the same with others. Figure 1 shows our proposed decompression architecture. In the figure, FIFO is used for transition point storage (TP). The counter counts the test sequence for each test vector. When the counter is equal to the values in TP, the status of "Similar Filter" will reverse. Initial status of "Similar Filter" is 1, and de-TSD can be changed by ATE. If the status of "Similar Filter" reverses to 0, de-TSD is fixed.

4. Calculation Procedures

The encoding procedure is shown in Figure 2. First of all, we reorder the test slices smoothly. We can regard the test set as sequence. Therefore we reorder vectors to connect longer runs. At last, test slice difference is used to reduce the number of 1's. We take advantage of the TSD data to complete encoding procedure.

4.1 Scan-Chain Reordering

Here, we illustrate some definitions used in this paper with the help of Figure 3. Let test slice (TS) be the vector of inputs applied to the same scan cell in each test

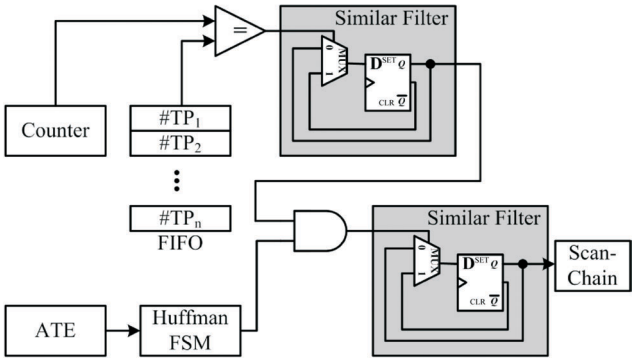


Figure 1. Our test decompression architecture.

Procedure()

Step 1. Scan-Chain Reordering.
Step 2. X-filling with Longer Runs.
Step 3. Test Vector Reordering.
Step 4. Test Slice Difference (TSD).
Step 5. TSD Data Encode.

Figure 2. Main steps of the proposed procedure.

cube. Thus, there are six test slices in this test set, and each test slice has 4 bits. Moreover, if two test slices have no conflicting value in corresponding position, we call them p-compatible. Contrarily, they are n-compatible if two test slices have opposite value in each corresponding position. Where *p* is positive and n is negative. For example, test slice 2 and 6 is p-compatible, and test slice 1 and 3 is n-compatible.

We present two different reordering procedures. The procedures are shown in Figure 4.

(a) Method 1: We consider both p-compatible and n-compatible, then we perform scan-chain reordering algorithm firsthand. The detailed description of the scan-chain reordering algorithm is given in Figure 5.

(b) Method 2: We also consider both p-compatible and n-compatible. First, we establish the compatible table of the test slice set, and then we check conflict between slices before grouping them. The concept of the conflict checking operation is illustration in section 4.1. A test slices group (TSG) is a set of test slices that are pair-wise compatible. Thus, test slices in the TSG are the same, that is, any one of the slice in TSG can represent the other slices data. Finally, we use the scan-chain reordering algorithm to reorder TS. So we can get TS order set.

The proposed scan-chain reordering algorithm is shown in Figure 5. First, we calculate the number of 0's and 1's in each TS. If the number of 0's and 1's are the same, we choose the minimum unspecified bits. Then we set *Ss*. The variable "*Ss*" (shadow slice) corresponds to last specified bits for new ordering. For instance, we

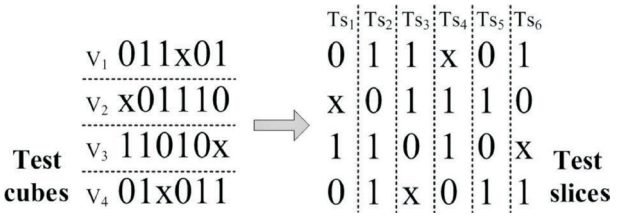


Figure 3. Example of fundamental definitions.

Pseudo-code: Scan-Chain Reordering

Input: Test Set T
Output: new scan-chain order (SCo)

```
01. SC_Reordering (T)
02. S: TS set;
03. SCo: Scan-chain order set;
04. Ss: Shadow Slice;
05. first TS is the minimum difference between '0' and '1';
06. SCo ← first TS;
07. S = S - chosen TS;
08. Ss = first TS;
09. while S not empty
10.     choose the minimum distance between Ss and TS (∈ S);
11.     (if the same, we choose the TS with most specified bits)
12.     SCo ← chosen TS;
13.     if it is n-compatible
14.         mark inversed;
15.         Ss = inversion of specified bits in chosen TS;
16.     else Ss = specified bits in chosen TS;
17.     S = S - chosen TS;
18. return SCo;
```

Figure 5. Pseudo-code of the proposed scan-chain reordering.

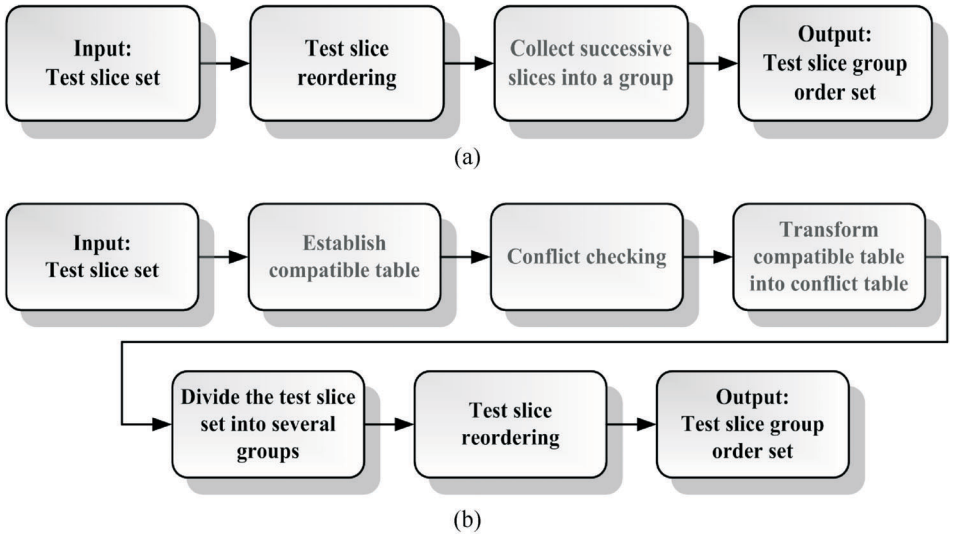


Figure 4. Scan-chain reordering procedure (a) method 1, (b) method 2.

choose TS_{23} where there exists no difference between 0's and 1's in Figure 6. Then we set Ss as $[0\ 1\ 0\ 1]^T$.

Second, we choose the TSG. If the number of 0's and 1's are the same, we choose the one which has the least unspecified bits. Because TS with less unspecified bits may cause more transitions. It is worth noting that besides general difference, we also consider the inversed difference. If the chosen TS needs be inverted, the encoded data must be inverted. And we add inverters between corresponding scan cells. In Figure 6(b), there are two test slices (inversed TS_3 and TS_4) having no difference between Ss. We choose the TS_3 because of fewer unspecified bits in TS_3. And the encoded data for TS_3 is changed to $[0\ 1\ 0\ 1]^T$. At last, we will get reordered test slices.

Comparing with general grouping methods, we allow the n-compatible case. But it may also cause new conflicts between test slices in the TSG. In Figure 7(a), the TS_1 can't be merged with TS_0 and TS_3. Because of TS_1 has a conflict bit between TS_0 and TS_3.

Therefore we will find the conflicts and choose the best compatible relationship, then check compatible type between each pair of test slices. If we find a difference when compared with the main compatible type, we will eliminate it. Figure 8 shows the pseudo code of conflict checking. Figure 7(c) shows the result of after conflict checking. There are three conflicts in the figure and the first conflict is TS_0 and TS_1 in row 3. After conflict checking, we transform compatible table into conflict table, then use the general graph coloring algorithm (Brelaz, 1979) to divide the test slice set into several groups.

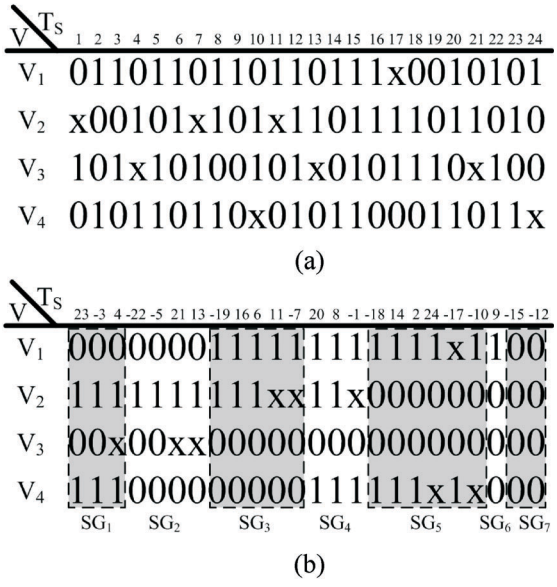


Figure 6. Example of scan-chain reordering (a) initial test vector set, (b) test slice groups.

4.2 X-Filling with Longer Runs

Here, we aim at the unspecified bits in test sequence, and fill them with appropriate values to lengthen runs. We calculate the length of specified bits near unspecified bits, and then fill them with longer specified bits. Figure 9 shows the example of scan-chain reordering and X-filling in our methods.

4.3 Test Vector Reordering

First, we take down the start bit and the end bit of each

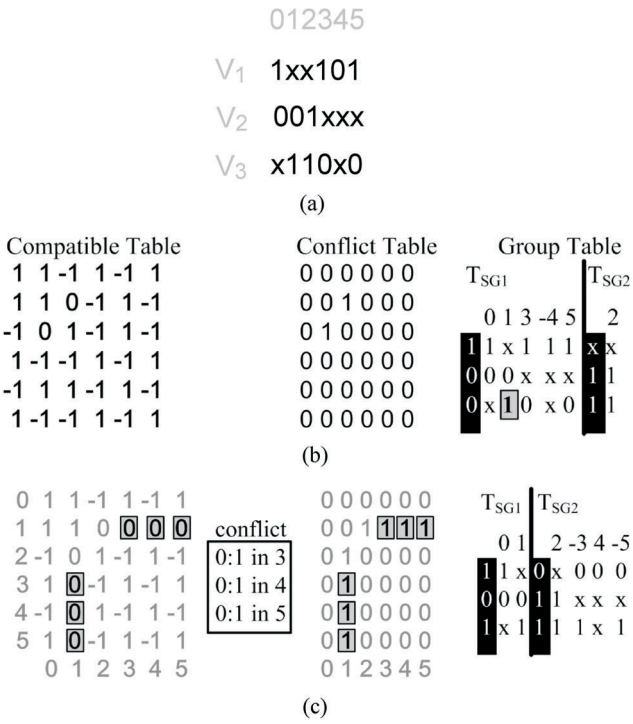


Figure 7. Example of conflict in group (a) initial test cubes, (b) conflict in grouping, (c) conflict checking.

```
Pseudo-code: Conflict Checking
Input: Compatible table CT
01. Conflict_Checking (CT);
02. comType: main compatible type;
03. for each slice i
04.   for each slice j not include i
05.     comType = CT(i, j);
06.     if(comType = 0)  comType = the most compatible type;
07.     for each slice k not include i
08.   type =  CT(k, i)*CT(k, j);
09.   if( type != comType & type != 0) CT(k, j) = CT(j, k) = 0;
```

Figure 8. Pseudo-code of conflict checking.

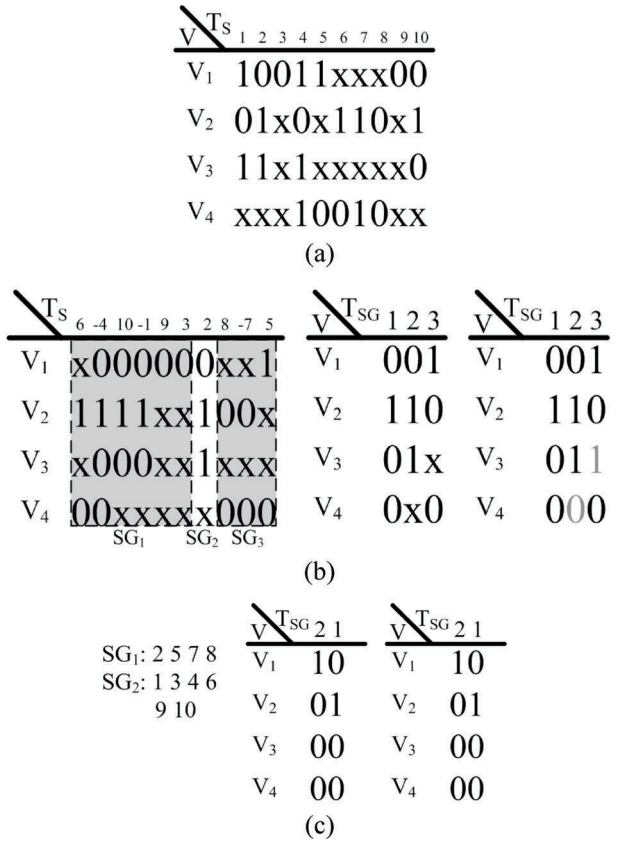


Figure 9. Example of SC reordering and X-filling (a) initial test data, (b) method 1, (c) method 2.

test vector, and compute the amount of consecutive bits with values at the start bit-stream and end bit-stream of each test vector. In order to reduce amount of runs and make length of runs longer, we connect the longest start bit-stream with longest end bit-stream of the same type as the start bit-stream. In Figure 10, we use an example to introduce test vector reordering.

4.4 Test Slice Difference (TSD)

The TSD is used to improve the compression ratio. Figure 11(b) shows the proposed technique. Jas and Touba [28] present the similar architecture of Cyclical Scan-chain (CSR) as illustrated in Figure 11(a). But the architecture of test vector decompression via CSR is not practical to use in the data compression strategy [16] because the hardware overhead grows by the length of scan chain. In contrast to [28], no matter how long is the scan chain, our method only needs one D-FF to store back data and one XOR gate to invert the data when needed. And after simplification, we only need one element as the de-TSD shown in Figure 1. The de-TSD hardware overhead is similar to the scan cell.

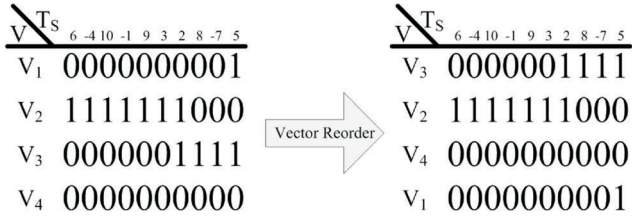


Figure 10. Example of test vector reordering.

"Test slice difference" is the difference between test slice and the one in front of it:

$$TS_{diff} = \{b_0, b_0 \oplus b_1, b_1 \oplus b_2, b_2 \oplus b_3, \ldots, b_{n-1} \oplus b_n\} \quad (1)$$

where n is the length of scan chain multiplied by the number of vectors. After test slice difference, both data volume and entropy of data are reduced. In this paper, we implement the test vector difference (Tdiff) [28]. In Table 1, the proportion of 1's is lower than the CSR. The table shows the average of 50%/48% improvement ratio in MinTest/TetraMAX. It is worth noting that the proposed de-TSD only needs one scan cell, instead of a copied scan chain.

4.5 TSD Data Encode

Finally, we regard TSD data as a bit sequence and we count the distances between 1's. Then we use Huffman code to encode distance according to the frequencies of distance. Note that the first bit (0 or 1) is built in circuit.

Figure 12 shows our entire example. We use method 2 to complete Figure 12(b). Figure 12(f) shows the encoding data size as 5 bits, thus the compression ratio is 89.6%. In Figure 12(h), Peak transition time at V1 is 1 and weighted transitions count metric (WTC) is 14.

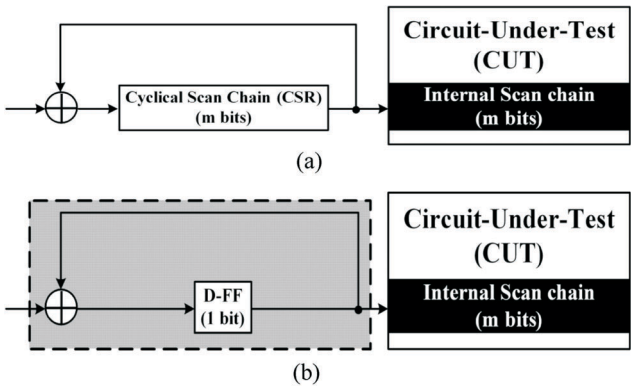
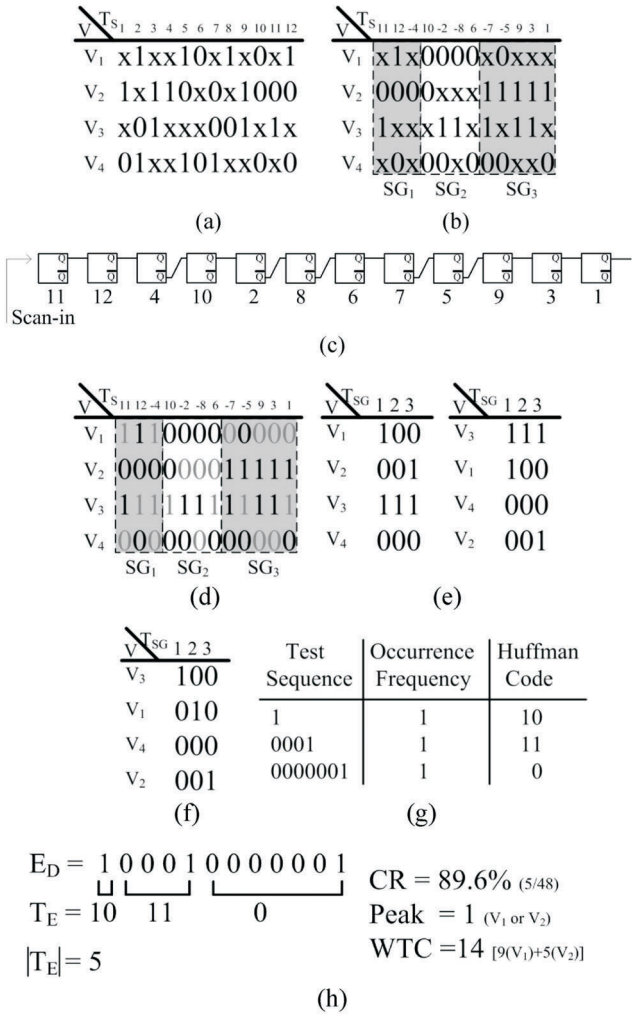


Figure 11. The architecture of test vector decompression (a) cyclical scan chain for test vector difference, (b) our proposed test slice difference.



Test Sequence	Occurrence Frequency	Huffman Code
1	1	10
0001	1	11
0000001	1	0

Figure 12. Example of all procedure (a) initial test vector set, (b) test slice groups, (c) scan chain with scan cell inverted, (d) after X-filling, (e) TSG set and test vector reordering, (f) test slice difference, (g) Huffman table, (h) encoded data information.

5. Experimental Results

We conduct experiments for several large ISCAS'89 benchmark circuits at both test cubes that generated by MinTest [1] and TetraMAX [29].

Table 2 shows the basic information of test cubes. |Td| denotes total data bits in test cubes. Then the percentage of unspecified bits (#X%) and ones (#1%) in test cubes are respectively shown in the table.

5.1 Proposed TSD and Reordering Improvement

Tables 3 and 4 presents our encoding data size (Our) and the percentage of improvement ratio (Imp. (%)). To calculate improvement ratio, the formula is utilized:

$$\text{Imp.}(\%) = [(\text{Method} - \text{Our}) / \text{Method} * 100\% \quad (2)$$

where method denotes other compression methods and the encoded data size is listed in Tables 5 and 6. The first three approaches use reordering and TSD to replace original test cubes. As a result of the latter fourmethods encode both of run 0 and run 1, TSD techniqueis not suitable for them. Thus, we only use proposed reordering algorithm for these methods. There are still good results on these methods.

5.2 Proposed Encoding Procedure

In Tables 5 and 6, we report two results for Golomb [15,26]. It should be noted that, in columns 2~4, test cubes are replaced by "difference vector sequences (T_{diff})" which is different method from ours. The column 5 set the unspecified bits to zeroes.

We also compare with well-known compression schemes including Block Merging [4], VIHC [9], selective Huffman [10], Multilevel-Huffman [11], Ninecoded [14], FDR [16], EFDR [17], RunBasedReordering [25], and MTC [30], etc. However, the V9C technique does not regroup test vectors action due to the author does not show the algorithm in [14]. The data in Table 5 is obtained

Table 1. Percentage of 1's in test slice difference and test vector difference (%)

Circuit	MinTest			TetraMAX		
	T_{diff}	Our	Imp. (%)	T_{diff}	Our	Imp. (%)
s5378	9.21	7.29	20.84	7.78	5.89	24.24
s9234	7.33	6.41	12.54	7.03	5.49	21.88
s13207	2.17	1.21	44.12	3.78	1.57	58.42
s15850	5.67	2.77	51.20	6.32	3.62	42.69
s35932	25.65	0.28	98.89	17.91	0.80	95.55
s38417	8.19	2.44	70.16	2.51	1.38	45.04
s38584	7.71	3.68	52.25	7.64	3.87	49.34
Avg. (%)			50.00			48.17

Table 2. Characteristics of ISCAS'89 test sets

Circuit	MinTest			TetraMAX		
	$\|T_d\|$	#X (%)	#1 (%)	$\|T_d\|$	#X (%)	#1 (%)
s5378	23,754	72.62	14.89	23,326	76.78	11.43
s9234	39,273	73.01	12.26	35,568	74.73	11.50
s13207	165,200	93.15	3.04	100,100	89.74	4.27
s15850	76,986	83.56	6.92	75,764	81.53	7.71
s35932	28,208	35.30	35.51	37,023	63.61	16.73
s38417	164,736	68.08	17.89	647,296	92.69	4.10
s38584	199,104	82.28	8.44	190,320	82.11	8.62

Table 3. Test data volume improvement ratio in MinTest

Circuit	Golomb [15]		FDR [13]		VIHC [11]		Alternating-RL [20]		RL-Huffman [10]		EFDR [6]		Hybrid-ARL [40]	
	Our	Imp. (%)	Our	Imp. (%)	Our	Imp. (%)	Our	Imp. (%)	Our	Imp. (%)	Our	Imp. (%)	Our	Imp. (%)
s5378	9,262	15.72	7,301	20.54	6,969	25.29	7,315	37.45	6,969	36.56	7,452	34.74	8,469	30.12
s9234	14,023	11.06	11,663	24.56	10,994	28.29	11,667	46.02	10,995	46.58	11,816	44.40	13,111	38.91
s13207	16,210	37.35	11,697	42.57	9,538	56.16	11,699	64.17	9,538	66.99	11,885	60.37	12,969	58.29
s15850	14,721	42.83	11,202	48.11	10,146	52.35	11,163	57.56	10,147	59.64	11,381	53.82	13,389	49.96
s35932	798	97.02	922	94.79	472	94.04	1,081	80.35	472	84.64	1,148	79.33	1,517	60.32
s38417	28,212	59.14	23,462	58.89	21,644	60.92	23,465	63.89	21,647	63.33	24,687	62.00	27,341	56.60
s38584	46,754	41.85	37,648	46.47	35,734	46.76	37,651	51.34	35,735	52.27	39,047	47.13	41,839	44.45
Avg. (%)		43.57		47.99		51.97		57.25		58.57		54.54		48.38

Table 4. Test data volume improvement ratio in TetraMAX

Circuit	Golomb [15]		FDR [13]		VIHC [11]		Alternating-RL [20]		RL-Huffman [10]		EFDR [6]		Hybrid-ARL [40]	
	Our	Imp. (%)	Our	Imp. (%)	Our	Imp. (%)	Our	Imp. (%)	Our	Imp. (%)	Our	Imp. (%)	Our	Imp. (%)
s5378	7,866	17.35	6,334	19.41	5,906	21.38	6,361	40.41	5,911	40.87	6,518	39.97	7,439	31.30
s9234	11,400	14.67	9,406	24.97	8,918	25.29	9,409	48.83	8,919	49.58	9,660	50.54	11,085	40.01
s13207	12,154	49.23	9,218	54.07	8,177	56.67	9,289	61.51	8,177	63.97	9,666	61.94	11,743	52.24
s15850	17,632	33.73	13,756	40.83	12,848	42.32	13,777	53.69	12,850	54.84	13,954	55.60	16,249	45.64
s35932	2,538	90.31	2,767	86.02	1,968	89.01	2,751	82.47	1,973	86.54	2,963	80.40	4,087	75.16
s38417	71,362	37.86	44,017	41.29	41,989	40.42	44,077	42.94	41,964	41.91	44,928	43.74	48,347	35.99
s38584	46,419	38.92	37,142	44.96	35,429	45.81	37,143	52.35	35,429	53.15	38,537	53.00	41,437	45.20
Avg. (%)		40.30		44.51		45.84		54.60		55.84		55.03		46.51

Table 5. Compression results in MinTest

Circuit	[15]	[13]	[11]	[35]	[6]	[20]	[10]	[19]	[9]	[12]	[8]	[41]	[5]	[7]	[34]	M1	M2
s5378	10,989	9,188	9,328	14,937	11,419	11,694	10,986	11,487	10,511	10,666	9,597	12,824	10,695	9,358	7,469	6,786	6,489
s9234	15,767	15,460	15,332	21,499	21,250	21,612	20,582	19,279	17,763	17,987	15,711	20,736	19,171	15,511	11,727	10,739	10,018
s13207	25,873	20,368	21,758	33,467	29,992	32,648	28,893	29,224	24,450	37,996	25,283	31,272	24,969	18,384	11,847	8,964	8,654
s15850	25,748	21,590	21,291	28,618	24,643	26,306	25,143	25,883	22,126	26,175	21,405	27,261	23,492	18,926	11,477	9,562	9,435
s35932	26,757	17,706	7,924	34,414	5,554	5,501	3,072	7,149	6,602	8,860	5,198	1,199	6,108	N/A	1,250	306	297
s38417	69,047	57,066	55,387	117,987	64,962	64,976	59,024	64,857	61,134	67,542	58,489	68,267	66,907	58,785	24,746	18,638	17,813
s38584	80,404	70,328	67,114	85,275	73,853	77,372	74,863	68,631	62,897	71,478	60,736	72,852	65,993	55,200	39,099	34,250	31,792

Table 6. Compression results in TetraMAX

Circuit	[15]	[13]	[11]	[35]	[6]	[20]	[10]	[19]	[9]	[12]	[8]	[40]	[5]	[33]	[34]	M1	M2
s5378	9,517	7,860	7,512	12,541	10,858	10,675	9,997	10,981	9,548	9,559	8,329	10,829	9,306	8,062	6,629	5,666	5,325
s9234	13,360	12,536	11,937	18,930	19,530	18,387	17,691	17,310	15,356	16,186	14,165	18,477	16,451	13,073	10,003	8,675	8,526
s13207	23,940	20,070	18,870	26,165	25,396	24,131	22,694	25,266	19,580	26,964	19,861	24,589	20,497	17,507	10,116	7,498	7,473
s15850	26,608	23,247	22,275	30,283	31,429	29,747	28,452	29,778	24,937	28,436	23,709	29,891	26,717	20,299	14,460	12,142	11,517
s35932	26,194	19,799	17,905	25,139	15,116	15,689	14,660	18,299	15,611	17,619	16,415	16,455	15,834	16,132	3,668	1,343	1,341
s38417	114,834	74,971	70,472	167,907	79,854	77,241	72,241	118,740	79,486	154,062	106,178	75,529	85,604	64,022	50,651	40,628	38,629
s38584	75,996	67,476	65,381	82,582	81,987	77,945	75,618	73,214	62,301	69,963	59,952	75,621	66,394	61,858	40,376	34,007	31,317

from recent paper. Gray frames in the table mean that the data is not executed in recent paper. Therefore we use our implemented program to obtain these data here. Clearly, the proposed approach achieves better encoding data volume than other methods. This means our method can greatly reduce the encoding data volume. The entropy of the original test data is also improved.

5.3 Power Dissipation

Another important issue for scan-based testing is the power consumption which is very high during test mode. The number of transitions occurs the switching activate during the scan in and scan out operations. In order to manifest the effect of test power reduction in our encoding methods, we use the weighted transitions count metric (WTC) [23] to estimate the transition.

The formula of WTC is shown as follow, where n is length of scan chain.

$$WTC_j = \sum_{i=1}^{n-1} (n-i) \times (t_{j,i} \oplus t_{j,i+1}) \tag{3}$$

Figure 13 shows the peak power under different X-filling methods. We have applied the test cubes in random-filling (Ran.) and minimum transition X-filling (MT) [31]. We have also given the experimental results using the T_{diff} [15] and mapping all unspecified bits to zeroes (0) [26]. The "S-D" and "RBR" in the figure are referred to as skewed-distribution scan chain partition [24] and RunBasedReordering [25]. As shown in figure, we shall denote our proposed method 1 as "Our_M1". The peak power (Ppeak) and average power (Pavg) are computed as follows, where m is the number of vectors, and TCi is transition count in vector i.

$$P_{peak} = \max_{1 \le j \le m} TC_j \tag{4}$$

$$P_{avg} = \frac{\sum_{j=1}^{m} WTC_j}{m} \tag{5}$$

Similarly, Figure 14 shows the average power under different X-filling methods. Note that we divided the amount of average power by 4 for circuit s35932, s38417, and s38584 here. Because of the quantities in these circuits are greatly over the remainder circuits. Observe that our scheme can reduce the scan-in power dissipation significantly.

6. Conclusion

In this paper, we propose new decompression architecture to filter redundant data in test slice groups. The test slice difference (TSD) technique is an efficient and requires low hardware overhead to reduce test data volume. In the part of power dissipation, we present an ex-

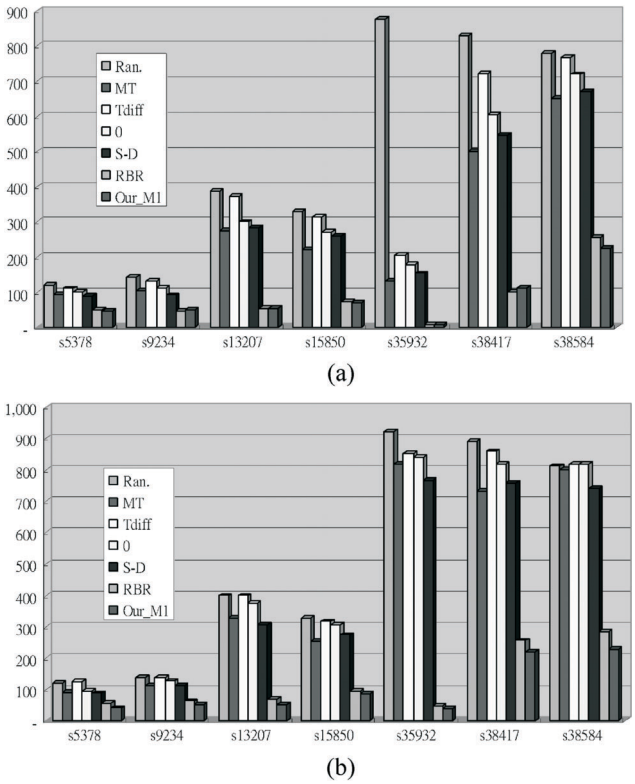


Figure 13. Peak power results (a) MinTest, (b) TetraMAX.

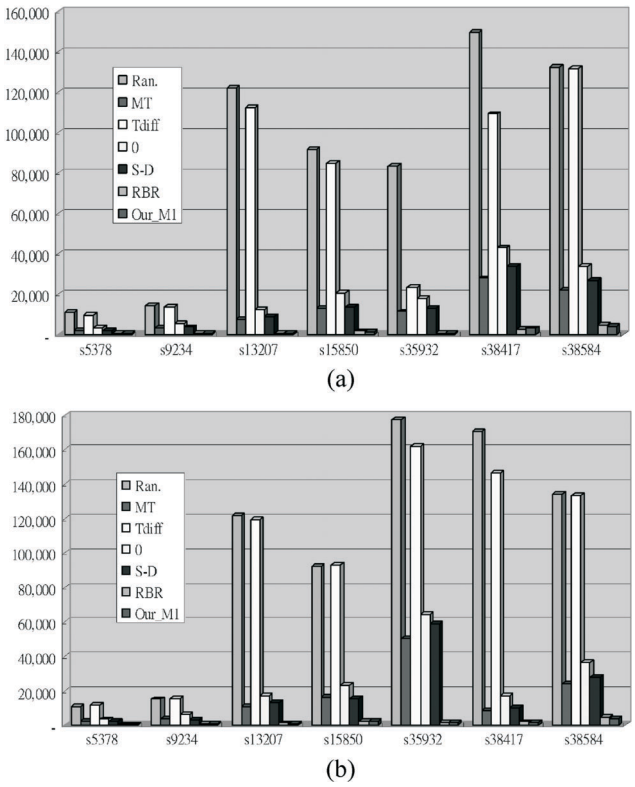


Figure 14. Average power results (a) MinTest, (b) TetraMAX.

tremely efficient algorithm for scan chain reordering. Experimental results show that the proposed scheme achieved higher compression ratio than previous approach, and the power consumption is also significantly reduction simultaneously.

Acknowledgments

This work was supported by the National Science Council of Taiwan under grant number NSC 99-2221-E-032-071. The authors would like to thank Prof. K. Chakrabarty for providing the MinTest test cubes that were used for the experimental results in this paper.

References

[1] Hamzaoglu, I. and Patel, J. H., "Test Set Compaction Algorithms for Combinational Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 957–963 (2000).

[2] Lee, J. and Touba, N. A., "Combining Linear and Non-Linear Test Vector Compression Using Correlation-Based Rectangular Encoding," *Proceedings of the IEEE VLSI Test Symposium*, Berkeley, CA, USA, pp. 252–257 (2006).

[3] Wang, L. T., Wu, C. W. and Wen, X., *VLSI Test Principles and Architectures: Design for Testability*, San Francisco, CA, USA: Morgan Kaufmann (2006).

[4] El-Maleh, A., "Efficient Test Compression Technique Based on Block Merging," *The Institution of Engineering and Technology Computers & Digital Techniques*, Vol. 2, pp. 327–335 (2008a).

[5] Hellebrand, S. and Würtenberger, A., "Alternating Run-Length Coding - A Technique for Improved Test Data Compression," *IEEE International Workshop on Test Resource Partitioning*, Baltimore, MD, USA (2002).

[6] Wu, P. H., Chen, T. T., Li, W. L. and Rau, J. C., "An Efficient Test-Data Compaction for Low Power VLSI Testing," *IEEE International Conference on Electro/Information Technology*, Ames, IA, USA, pp. 237–241 (2008a).

[7] Hamzaoglu, I. and Patel, J. H., "Reducing Test Application Time for Full Scan Embedded Cores," *International Symposium on Fault-Tolerant Computing*, Madison, Wisconsin, USA, pp. 260–267 (1999).

[8] Lee, K. J., Chen, J. J. and Huang, C. H., "Broadcasting Test Patterns to Multiple Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1793–1802 (1999).

[9] Gonciari, P. T., Al-Hashimi, B. M. and Nicolici, N., "Variable-Length Input Huffman Coding for System-on-a-Chip Test," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 22, pp. 783–796 (2003).

[10] Jas, A., Ghosh-Dastidar, J., Ng, M. and Touba, N. A., "An Efficient Test Vector Compression Scheme Using Selective Huffman Coding," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 22, pp. 797–806 (2003).

[11] Kavousianos, X., Kalligeros, E. and Nikolos, D., "Multilevel Huffman Coding: An Efficient Test-Data Compression Method for IP Cores," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Sonoma, CA, USA, Vol. 26, pp. 1070–1083 (2007a).

[12] Kavousianos, X., Kalligeros, E. and Nikolos, D., "Optimal Selective Huffman Coding for Test-Data Compression," *IEEE Transactions on Computers*, Vol. 56, pp. 1146–1152 (2007b).

[13] Nourani, M. and Tehranipour, M. H., "RL-Huffman Encoding for Test Compression and Power Reduction

in Scan Applications," *ACM Transactions on Design Automtion of Electronic Systems*, New York, NY, USA Vol. 10, pp. 91–115 (2005).

[14] Tehranipoor, M., Nourani, M. and Chakrabarty, K., "Nine-Coded Compression Technique for Testing Embedded Cores in SoCs," *IEEE Transactions on VLSI Systems*, Vol. 13, pp. 719–731 (2005).

[15] Chandra, A. and Chakrabarty, K., "System-on-a-Chip Test-Data Compression and Decompression Architectures Based on Golomb Codes," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 20, pp. 355–368 (2001a).

[16] Chandra, A. and Chakrabarty, K., "Test Data Compression and Test Resource Partitioning for System-on-a-Chip Using Frequency-Directed Run-Length (FDR) Codes," *IEEE Transactions on Computers*, Vol. 52, pp. 1076–1088 (2003).

[17] El-Maleh, A., "Test Data Compression for System-on-a-Chip Using Extended Frequency-Directed Run-Length Code," *The Institution of Engineering and Technology Computers & Digital Techniques*, Vol. 2, pp. 155–163 (2008b).

[18] Sinaoglu, O., Bayraktaroglu, I. and Orailoglu, A., "Scan Power Reduction through Test Data Transition Frequency Analysis," *Proceedings of the IEEE International Test Conference*, Washington, DC, USA, pp. 844–850 (2002).

[19] Lai, N. C., Wang, S. J. and Fu, Y. H., "Low-Power BIST with a Smoother and Scan-Chain Reorder under Optimal Cluster Size," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Sonoma, CA, USA, Vol. 25, pp. 2586–2594 (2006).

[20] Alpaslan, E., Huang, Y., Lin, X., Cheng, W. T. and Dworak, J., "Reducing Scan Shift Power at RTL," *Proceedings of the IEEE VLSI Test Symposium*, San Diego, CA, USA, pp. 139–146 (2008).

[21] Wang, S. and Gupta, S. K., "ATPG for Heat Dissipation Minimization during Test Application," *IEEE Transactions on Computers*, Vol. 47, pp. 256–262 (1998).

[22] Tseng, W. D. and Lee, L. J., "Reduction of Power Dissipation during Scan Testing by Test Vector Ordering," *IEEE International Workshop on Microprocessor Test and Verification*, Austin, TX, USA, pp. 15–21 (2007).

[23] Badereddine, N., Girard, P., Pravossoudovitch, S., Landrault, C., Virazel, A. and Wunderlich, H. J., "Minimizing Peak Power Consumption during Scan Testing: Test Pattern Modification with X Filling Heuristics," *Proceedings of the IEEE Design and Test of Integrated Systems in Nanoscale Technology*, Tunis, Beijing, pp. 359–364 (2006).

[24] Wang, S. J., Li, K. S. M., Chen, S. C., Shiu, H. Y. and Chu, Y. L., "Scan-Chain Partition for High Test-Data Compressibility and Low Shift Power under Routing Constraint," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 716–727 (2009).

[25] Fang, H., Tong, C. and Cheng, X., "RunBasedReordering: A Novel Approach for Test Data Compression and Scan Power," *Proceedings of the Asia and South Pacific Design Automation Conference*, Yokohama, Japan, pp. 732–737 (2007).

[26] Chandra, A. and Chakrabarty, K., "Combining Low-Power Scan Testing and Test Data Compression for System-on-a-Chip," *Proceedings of the IEEE Design Automation Conference*, Las Vegas, NV, USA, pp. 166–169 (2001b).

[27] Balakrishnan, K. J. and Touba, N. A., "Relationship between Entropy and Test Data Compression," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Sonoma, CA, USA, Vol. 26, pp. 386–395 (2007).

[28] Jas, A. and Touba, N. A., "Test Vector Decompression via Cyclical Scan Chains and Its Application to Testing Core-Based Designs," *Proceedings of the IEEE International Test Conference*, Yokohama, Japan, pp. 458–464 (1998).

[29] Synopsys TetraMAX ATPG user guide version x-2005.09.

[30] Rosinger, P., Gonciari, P., Al-Hashimi, B. and Nicolici, N., "Simultaneous Reduction in Volume of Test Data and Power Dissipation for System-on-a-Chip," *Electronics Letters*, Vol. 37, pp. 1434–1436 (2001).

[31] Sankaralingam, R., Oruganti, R. R. and Touba, N. A., "Static Compaction Techniques to Control Scan Vector Power Dissipation," *Proceedings of the IEEE VLSI Test Symposium*, Montreal, Que, Canada, pp. 35–40 (2000).

Manuscript Received: Mar. 11, 2011
Accepted: Nov. 22, 2011