A Generic Framework for Scan Capture Power Reduction in Fixed-Length Symbol-based Test Compression Environment

Xiao Liu[†] and Qiang Xu[†‡]
[†]CUhk REliable computing laboratory (CURE)
Department of Computer Science & Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
[‡]CAS-CUHK Shenzhen Institute of Advanced Integration Technology
Email: {xliu,qxu}@cse.cuhk.edu.hk

Abstract

Growing test data volume and overtesting caused by excessive scan capture power are two of the major concerns for the industry when testing large integrated circuits. Various test data compression (TDC) schemes and low-power X-filling techniques were proposed to address the above problems. These methods, however, exploit the very same "don't-care" bits in the test cubes to achieve different objectives and hence may contradict to each other. In this work, we propose a generic framework for reducing scan capture power in test compression environment. Using the entropy of the test set to measure the impact of capture power-aware X-filling on the potential test compression ratio, the proposed holistic solution is able to keep capture power under a safe limit with little compression ratio loss for any fixed-length symbol-based TDC method. Experimental results on benchmark circuits demonstrate the efficacy of the proposed approach.

1 Introduction

With the advancement of semiconductor technology, more and more transistors are integrated on a single silicon die, which require an increasingly amount of data to test them [5]. Such large test data volume is a big concern for the industry because it not only prolongs the testing time of the integrated circuits (ICs), but also raises memory depth requirements for the automatic test equipment (ATE). Both significantly increase the ICs' manufacturing test cost. To address this problem, various test data compression (TDC) techniques [16] have been proposed in the literature, which exploit the "don't-care" bits (also known as X-bits) in the given test cubes efficiently to reduce test data volume.

At the same time, it is shown that ICs' power consumption in scan-based testing can be significantly higher than that during normal operation, in both shift mode and capture mode [3]. Elevated shift power may lead to structural damage to the circuit under test (CUT). Capture power violations, on the other hand, may cause good circuits to fail at-speed test, thus leading to unnecessary yield loss [14]. To reduce test power in shift mode, various techniques have been proposed in the literature, in which design-for-testability (DfT) based methods such as scan chain partitioning technique [13, 19] are very effective. For capture power reduction, however, there are no such effective DfT-based methods and we mainly resort to low-power X-filling techniques (e.g., [12, 18, 21]) to reduce switching activities in capture mode.

As test compression and low-power X-filling techniques might take advantage of the very same X-bits in test cubes for different objectives, it is essential to develop a holistic solution that targets both issues together. At the same time, considering the large number of TDC schemes proposed in the literature, instead of introducing customized solutions to reduce scan capture power for specific test compression methods, we present a generic framework for scan capture power reduction that is applicable for any fixed-length symbol-based TDC method.

The basic idea of the proposed technique is to use the change of *entropy* of the test set to evaluate the impact of capture power-aware X-filling on test compression ratio. Generally speaking, if filling X-bits to keep capture power under a safe limit results in a small rise of the entropy, the amount of information contained in the test set does not increase much and hence the test compression ratio loss should also be modest for a good TDC technique. Effective and efficient heuristics are developed based on the above idea. Experimental results on ISCAS'89 and ITC'99 benchmark circuits show that, the proposed technique is able to significantly reduce capture power violations with little loss in test compression ratio, for several types of fixed-length symbol-based TDC schemes.

The remainder of this paper is organized as follows. Section 2 reviews related work and motivates this work. In Sections 3, we illustrate the proposed technique in detail. Experimental results on benchmark circuits are then shown in Section 4. Finally, Section 5 concludes this work.

2 Preliminaries and Motivation

2.1 Prior Work in Test Data Compression

To reduce the large test data volume, various TDC schemes have been presented in the literature, which can be broadly classified into two categories: (i) nonlinear symbol-based schemes that use data compression codes to encode test cubes;

(ii) linear decompressor-based schemes that decompress the free variables utilizing linear operations (e.g., XOR network and/or linear feedback shift registers). As pointed out in [16], symbol-based schemes (e.g., [4, 17]) can efficiently exploit correlations in the specified bits and do not require ATPG constraints, while linear techniques (e.g., [9, 10]) is easy for implementation. In this work, we mainly target symbol-based test compression schemes.

Although different in terms of coding styles, all symbol-based TDC techniques are developed based on the *same* principle. That is, they first divide the test data into several types of symbols, and then utilize shorter codewords to encode symbols occurring more frequently and use longer codewords for less-frequent symbols (See Table 2 for an example). According to the characteristics of the symbol length, symbol-based TDC techniques can be further categorized as variable-length TDC methods (e.g., [2]) and fixed-length TDC ones (e.g., [6, 8, 15]). This work focuses on scan capture power reduction for fixed-length symbol-based TDC schemes[1]. For these methods, the test data after X-filling is first divided into fixed-length symbols, as shown in Table 1, and then apply different coding methods to compress the data.

X-filling needs to be conducted for the given test cubes before the encoding process, which determines the *entropy* of the test data. Entropy is a measure for the disorder of a data set and it presents a fundamental limit on the potential test compression ratio independent of the TDC schemes [1]. Consider the test data after X-filling as shown in Table. 1. The number of occurrences for each type of symbol is shown in Column "frequency" in Table. 2, and the "probability p_i" is calculated by dividing the frequency with the number of symbols (n) in the data set. Based on the above, its entropy can be calculated as $-\sum_{i=1}^{n} p_i \cdot log p_i = 2.564$. Then, the maximum compression ratio can be obtained by $(symbol_length - entropy)/symbol_length = 35.9\%$. It should be noted that, the actual compression ratio varies with different coding schemes, but the above calculation indeed reflects the potential compression ratio of the test set [1]. *Obviously, the higher the entropy, the less the potential test compression ratio.*

Vector1	0001	0011	0001	1100	0000
Vector2	0000	1001	0000	0010	1000
Vector3	0000	0000	0000	0011	0010
Vector4	1000	0001	1001	0010	0000

Table 1. Test Set with 4-bit Symbols

i	Symbol x_i	Frequency	Probability p_i	Huffman Code
1	0000	7	0.35	10
2	0001	3	0.15	110
3	0010	3	0.15	111
4	1000	2	0.10	010
5	0011	2	0.10	000
6	1001	2	0.10	011
7	1100	1	0.05	0010

Table 2. Probability Table

[1]The basic principle presented in this paper can be extended to variable-length symbol-based TDC schemes.

2.2 Prior Work in Test Power Reduction

The ICs' power dissipation in scan tests can exceed the circuit's power rating in both shift mode and capture mode. Various techniques have been presented in the literature to address this problem, as surveyed in [3]. By setting X-bits in the given test cubes carefully, X-filling techniques can be used to reduce both scan shift power and scan capture power. However, X-filling for shift power reduction (e.g, [13, 19]) is not as effective as techniques based on scan chain manipulation (e.g, [13, 19]). Therefore, in this work, we assume the shift power issues are resolved with DfT-based techniques and we mainly consider to use X-filling for capture power reduction.

Excessive switching activities in scan capture mode leads to abnormal IR-drop. This associated extra gate propagation delay may cause good circuits to fail at-speed testing, thus resulting in unnecessary yield loss [11, 14]. To avoid this problem, we mainly resort to X-filling techniques that try to fill X-bits in the test stimuli to be as similar to test responses as possible. Wen *et al.* [18] proposed to fill the X-bits incrementally based on known test response bits. As forward implications and backward justifications are extensively used in their method, the computational complexity of this technique is quite high. To address this problem, Remersaro *et al.* [12] developed an efficient probability-based X-filling technique, namely preferred fill, which tries to fill all X-bits in one step, based on the probability information of corresponding response. The above works try to reduce capture power consumption as much as possible. This is however unnecessary because the yield loss can be avoided as long as the peak capture power does not exceed a certain threshold so that a good circuit would not fail at-speed tests [7, 20].

From the above, TDC schemes tend to fill X-bits to reduce the entropy of the test set, while low capture-power techniques try to fill X-bits so that the test stimuli and the test responses are as similar as possible. These two objectives may contradict with each other and it is therefore essential to develop a holistic solution that targets both issues. This has motivated the work studied in our paper.

3 Proposed Algorithm

We consider the impact of X-filling on both test compression and capture power in our proposed solution. In particular, the entropy of the test set is used to evaluate the X-filling effect for compression, while the scan capture power is evaluated by the Hamming distance between each test pattern and its response (i.e., *capture transitions* in the state elements [18]). By refilling few X-bits for the given test cube with the targeted TDC scheme, our method is able to significantly reduce capture power violations with little loss in test compression ratio. The flowchart for the proposed algorithm is depicted in Fig. 1 and we explain each step in the following.

In the beginning of our algorithm, we fill the given test cubes with the targeted TDC scheme. We can then conduct functional simulation to obtain test response for each test pattern and identify those unsafe patterns that violate the given

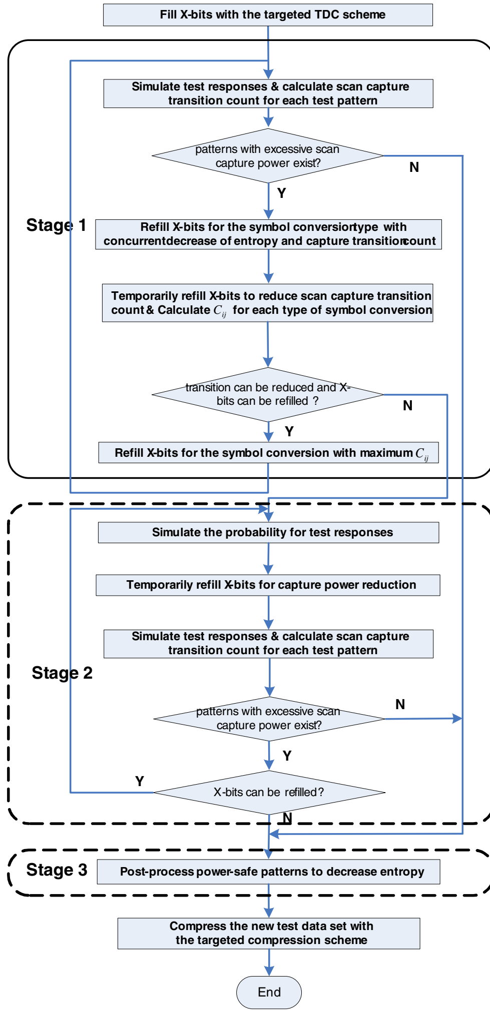

Figure 1. Flow chart of the proposed methodology.

capture transition threshold. The X-bits in these patterns are to be refilled to resolve their capture power violations and we use a 3-stage procedure to achieve this objective. In Stage 1, an entropy-aware X-filling algorithm is conducted, which tries to resolve capture power violations with little entropy increase (detailed in Section 3.1). If there are still some unsafe patterns left, we have to rely on a brute-force X-filling method mainly for capture power reduction in Stage 2, at the cost of higher entropy increment (detailed in Section 3.2). Next, we conduct a post-processing procedure for those safe patterns in Stage 3, trying to refill some of their X-bits to decrease the entropy without introducing capture power violations (see Section 3.3). Finally, the new test set is encoded with the targeted compression scheme again and we can evaluate the actual test compression ratio loss after resolving capture power violations.

3.1 Entropy-Aware X-Filling for Scan Capture Power Reduction

In this stage, we start with the unsafe patterns that exceed capture transition threshold and we try to refill some of the X-bits to reduce capture transitions without incurring high rise of the entropy.

For symbol-based compression scheme, refilling an X-bit can be treated as converting a symbol from one type to another. As introduced in Section 2.1, entropy is a measure of *disorder* of the test data set, and it increases when the symbol distribution becomes less biased, as can be seen clearly in Table 3 for Case 1 and Case 2 (Some symbols are converted from x_1 to other types). It is obvious that the increase of entropy in Case 2 is higher because many x_1 symbols are converted to 3 types of other symbols and this makes the data set more disordered. On the other hand, the entropy can be decreased when symbol distribution becomes more biased, e.g., for Case 3 in Table 3, two x_2 symbols with low occurrence rate are converted to x_1 type with high frequency. Based on the above observation, our entropy-aware capture power reduction method is conducted with two steps, as shown in Fig. 2.

Stage 2: Entropy-Aware X-Filling

INPUT: : test cube, cur_pattern: current filled test patterns
OUTPUT: refi_pattern: refilled test patterns

1. **while** (more X-bit can be refilled and transition count can be reduced) {
2. Simulate responses & calculate capture transitions for each pattern;
3. for each pattern above capture transition limit {
4. Temporarily refill X-bits for the symbol conversion type that decreases entropy;
5. Simulate responses & calculate capture transition count;
6. Refill X-bits that will reduce capture transition count;
7. }
8. for each type of symbol conversion {
9. for each pattern above capture transition limit {
10. Temporarily refill X-bits for this symbol conversion type;
11. Simulate responses & calculate capture transition count;
12. Update Δp_{ij} with *actual* capture transition reduction count;
13. }
14. Calculate Δe_{ij} for the test data set;
15. Calculate evaluation metric $C_{ij} = w_p \times \Delta p_{ij} - w_e \times \Delta e_{ij}$;
16. }
17. Refill X-bits for the symbol conversion type with maximum C_{ij};
18. }
19. **return** *refi_pattern*;

Figure 2. Procedure for Entropy Aware X-Filling for Capture Power Reduction.

In Step 1, we try to refill those X-bits that could lead to concurrent decrease of entropy and capture transitions. This process is conducted pattern by pattern (Lines 4-7). For each test pattern (e.g. Vector 1 in Table 1), its response is simulated first, and then we try to temporarily refill X-bits with the same value as their corresponding response bits, as shown in Fig. 3, under the condition that the corresponding symbol conversion will decrease entropy (refer to Case 3 in Table 3). After temporary refill is done, the pattern is simulated again, and the refilled X-bits is accepted only when it actually results in capture power reduction. In this example, the transition count is reduced from 7 to 5 and the entropy is decreased from 2.564 to 2.358, and hence we accept this refilling result.

		Original		Case 1		Case 2		Case 3	
i	Symbol x_i	Frequency	Probability p_i	Frequency	Probability p_i	Frequency	Probability p_i	Frequency	Probability p_i
1	0000	7	0.35	**4**	0.20	**4**	0.20	**9**	0.45
2	0001	3	0.15	**6**	0.30	**4**	0.20	**1**	0.05
3	0010	3	0.15	3	0.15	**4**	0.20	3	0.15
4	1000	2	0.10	2	0.10	**3**	0.15	2	0.10
5	0011	2	0.10	2	0.10	2	0.10	2	0.10
6	1001	2	0.10	2	0.10	2	0.10	2	0.10
7	1100	1	0.05	1	0.05	1	0.05	1	0.05
		$Entropy = 2.564$		$Entropy = 2.609$		$Entropy = 2.684$		$Entropy = 2.358$	

Table 3. Probability Table for Symbol Conversion

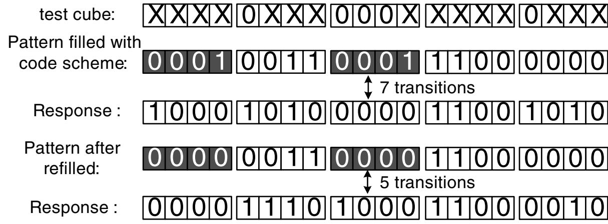

Figure 3. Example for Case 3 in Table 3.

Next, in Step 2, we continue to refill X-bits to further reduce capture transitions, but at the cost of entropy increase. Here we conduct only one type of symbol conversion every time. As shown in Table 3, single type of conversion will not make symbol distribution disorder significantly for the entire test set, when compared to Case 2 in Table 3. For each type of conversion, we search the whole test set and temporarily refill their X-bits together to reduce capture transitions. As in Step 1, for each pattern, the actual reduction is obtained by simulating it with temporary refilled X-bits. Only the refilling result that brings capture power reduction is accepted. Its *actual* reduced transition count is recorded accordingly.

We can then calculate the impact of each type of such trial symbol transition (x_i to x_j) in terms of the overall capture power reduction Δp_{ij} and the entropy increase Δe_{ij}. Δp_{ij} is obtained by the sum of *actual* reduced transition count, while Δe_{ij} is the entropy difference acquired by re-calculating the probability of each symbol and compared to the original value. Let us define an evaluation metric $C_{ij} = w_p \times \Delta p_{ij} - w_e \times \Delta e_{ij}$, where w_p and w_e are the optimization weights designated by users ($w_p + w_e = 1$). From all the possible types of symbol conversions, we choose the one with the maximum C_{ij}. The X-bits in those symbols are then refilled accordingly and we go back to the beginning of Stage 1 to check scan capture power violations. Again, patterns that violate capture transition limit are picked out to be refilled. The above procedure terminates when the scan capture transitions of all test patterns are under safe limit. Otherwise, it iterates until no more capture transitions can be reduced or no more X-bits can be refilled, and goes to Stage 2 as described in the following.

3.2 Brute-Force X-filling for Capture Power Reduction

For those unsafe patterns that cannot be resolved by the previous entropy-aware refilling strategy, wherein only one type of symbol is converted to another, they are refilled in Stage 2 in a kind of brute-force manner. The focus in this stage is capture power reduction, but at the same time, we try to keep the original test set as intact as possible, so that the entropy is not adversely increased.

This procedure is conducted pattern by pattern, and for each pattern, we simulate the response probability with the original test cube first (i.e., not the test pattern filled with the targeted TDC scheme), as in [12]. When the response probability for an X-bit is above a threshold (e.g., 0.8 for the example shown in Fig. 4), we treat this response bit as a specified value '1/0' and its corresponding stimuli bit in the pattern is refilled to be the same value. Also, a copy of the test cube is utilized to record the same X-filling result (depicted as "Test Cube" in Fig. 4). Next, we simulate the refilled test pattern to check its power safety. If the pattern still violates our limit, the updated copy of test cube is used for another round of response probability simulation. We conduct the above simulation and X-filling process iteratively until the pattern is power-safe or no more X-bits can be filled with highly-biased response probability.

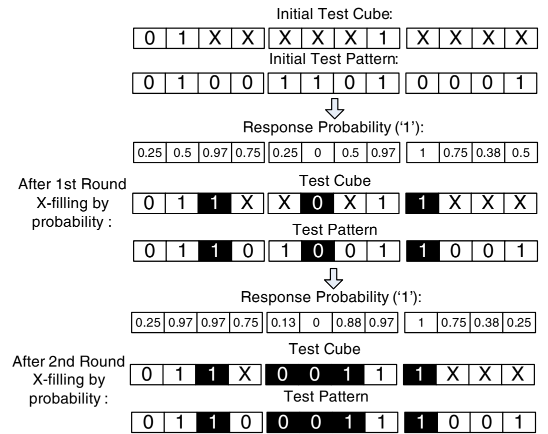

Figure 4. Example for X-filling with Probability.

3.3 Post-Processing for Entropy Reduction

The above two stages try to refill X-bits in those power-unsafe patterns to reduce capture power violations, which are typically achieved at the cost of entropy increase. To compensate such compression ratio loss, this stage refills X-bits for those safe patterns to decrease the entropy of the entire test set, under the constraint that their capture power transition count is lower than the limit.

Table 4. Experimental results for capture transitions and compression ratio (Huffman Code)

circuit	#dff	#pattern	X%	#unsolv pattern	Orig ratio(E)	Pro ratio(E)	$CRL(E)$	Orig ratio(C)	Pro ratio(C)	$CRL(C)$	Orig N_{vio}	Pro N_{vio}	VPR	Exe. time(s)
s9234	211	347	76.37%	1	55.5%	41.4%	25.4%	54.1%	40.7%	24.8%	226	7	96.9%	41.78
s15850	534	226	84.96%	1	73.9%	63.3%	14.3%	65.9%	58.2%	11.7%	88	1	98.9%	48.65
s13207	638	303	91.42%	0	80.2%	38.7%	51.7%	69.3%	38.3%	44.7%	303	0	100%	105.88
s38417	1636	267	73.78%	0	60.7%	54.8%	9.71%	56.3%	52.9%	6.03%	485	0	100%	159.12
s38584	1426	651	90.63%	2	78.4%	60.7%	22.6%	68.6%	57.6%	16.0%	635	2	99.7%	573.56
b20	490	1156	83.56%	0	72.4%	69.6%	3.87%	63.3%	61.7%	2.53%	292	9	96.9%	107.78
b21	490	1126	83.84%	1	71.9%	69.1%	3.89%	63.0%	61.4%	2.54%	320	5	98.4%	84.97
b22	735	1927	86.72%	0	76.7%	74.7%	2.60%	65.9%	65.0%	1.37%	512	2	99.6%	204.31
b17	1415	3041	95.86%	0	91.0%	90.8%	0.22%	72.7%	72.6%	0.14%	109	0	100%	169.23
Average			85.24%		73.4%	62.6%	14.9%	64.3%	56.5%	12.2%			98.9%	

Table 5. Experimental results for capture transitions and compression ratio (Nine Code)

circuit	#dff	#pattern	X%	#unsolv pattern	Orig ratio(E)	Pro ratio(E)	$CRL(E)$	Orig ratio(C)	Pro ratio(C)	$CRL(C)$	Orig N_{vio}	Pro N_{vio}	VPR	Exe. time(s)
s9234	211	347	76.37%	0	53.2%	45.2%	15.0%	36.7%	28.0%	23.7%	186	7	96.2%	30.81
s15850	534	226	84.96%	1	71.9%	71.3%	0.83%	55.1%	54.5%	1.01%	13	1	92.3%	5.546
s13207	638	303	91.42%	0	77.2%	74.4%	3.63%	59.8%	57.6%	3.68%	31	0	100%	38.34
s38417	1636	267	73.78%	0	55.5%	55.5%	0%	46.6%	46.9%	-0.64%	2	0	100%	10.98
s38584	1426	651	90.63%	2	75.8%	75.5%	0.40%	58.1%	57.9%	0.34%	4	2	50%	95.75
b20	490	1156	83.56%	0	70.4%	68.8%	2.27%	57.9%	57.0%	1.55%	183	5	97.3%	62.19
b21	490	1126	83.84%	1	70.0%	68.3%	2.43%	57.6%	56.7%	1.56%	224	3	98.7%	59.66
b22	735	1927	86.72%	0	74.9%	74.0%	1.20%	61.5%	61.1%	0.65%	201	0	100%	131.63
b17	1415	3041	95.86%	0	90.5%	90.4%	0.11%	70.7%	70.7%	0%	24	0	100%	143.63
Average			85.24%		71.7%	69.3%	2.87%	56.0%	54.5%	3.54%			92.7%	

Stage 3: Post-Processing for Entropy Reduction

INPUT: : test cube, cur_pattern: current filled test patterns
OUTPUT: refi_pattern: refilled test patterns

```
1. for each power-safe pattern {
2.   Temporarily refill X-bits for the symbol conversion type
       that decreases entropy;
3.   Simulate responses & calculate capture transition count;
4.   if (The capture transitions is under safe limit)
5.     Refill X-bits to decrease entropy;
6. }
7. return refi_pattern;
```

Figure 5. Procedure for Post-Processing.

We implement a fast post-processing procedure for the above duty, as depicted in Fig. 5. For each power-safe pattern, the conversion from a symbol occurring less often to another one with higher frequency is temporarily accepted, as this conversion leads to the reduction of entropy. After that, the refilled pattern is simulated to see wether its capture transitions is still below the safe limit. If so, we accept this refilling decision.

Our algorithm will terminate itself after Stage 3 regardless of whether there are any capture power-violated patterns left.

4 Experimental Results

We conduct experiments on several ISCAS'89 and ITC'99 benchmark circuits using four symbol-based TDC schemes: the Huffman code [1], the nine code [15], the selective Huffman code [8], and the selective dictionary code [6], and we set the symbol length to be four in our experiments. The test cubes are broadside transition patterns generated using a commercial ATPG tool, and the optimization weights w_p and w_e in our algorithm are set to be 0.003 and 0.997, respectively.

Experimental results are presented from Table 4 to Table 6. In these tables, Column "circuit" denotes the name of these circuits. The number of scan cells and the number of test patterns in these circuits are listed in Column "#dff" and "#pattern". Column "X%" refers to the percentages of X-bits in the initial test data set. For the capture transition threshold, we set it to be $\#dff/4$ for all the circuits. With the above configuration, there are some unfilled patterns whose capture transition count already exceeds the threshold. The number of such kind of patterns that are certain to violate our capture transition limit is shown in Column "#unsolv pattern".

"Orig ratio (E)" and "Pro ratio (E)" in the tables stand for the theoretical compression ratio limit for the test cube filled with the original TDC scheme and the proposed one, calculated by entropies according to [16]. "Orig ratio (C)" and "Pro ratio (C)" present the *actual* compression ratio encoded with the targeted TDC scheme for the original test cubes and the one refilled with our method for capture power reduction, respectively. The percentage of compression ratio loss (CRL) are denoted with "$CRL(E)$" for the ratio calculated with entropy and that with the targeted TDC scheme ($CRL(C)$). The number of test patterns that violate capture transition threshold of these two schemes are represented with "Orig N_{vio}" and "Pro N_{vio}", respectively. "VPR" denotes the reduction ratio for violated patterns. Finally, the execution times of the proposed method are shown in Column "Exe. time", obtained with a 2.13GHz PC with 1GB memory.

When the targeted TDC scheme is Huffman code with *greedy* X-filling [1], our experimental results are shown in Table 4. As can be observed from this table, with our proposed approach, most of the original unsafe patterns that violate capture transition limit (98.9% on average) are resolved, at the cost of 12.2% compression ratio loss in average. With the specific X-filling method used in nine code [15], as can be seen

Table 6. Experimental results for capture transitions and compression ratio (Selective Huffman Code and Selective Dictionary Code)

circuit	#unsolv pattern	Orig ratio(E)	Pro ratio(E)	$CRL(E)$	Selective Huffman code Orig ratio(H)	Pro ratio(H)	$CRL(H)$	Selective dictionary code Orig ratio(D)	Pro ratio(D)	$CRL(D)$	Orig N_{vio}	Pro N_{vio}	VPR	Exe. time(s)
s9234	0	55.8%	42.1%	24.6%	32.0%	23.2%	27.5%	20.8%	17.2%	17.3%	271	8	97.0%	26.33
s15850	1	73.9%	69.7%	5.68%	41.6%	39.8%	4.32%	23.7%	22.6%	4.64%	40	1	97.5%	5.546
s13207	0	80.2%	38.4%	52.1%	44.5%	17.4%	60.9%	24.3%	16.0%	34.2%	303	0	100%	38.34
s38417	0	52.8%	40.8%	22.7%	28.7%	19.9%	30.7%	18.1%	17.7%	2.21%	261	0	100%	210.67
s38584	2	78.2%	48.7%	37.7%	43.7%	26.2%	40.0%	24.3%	19.6%	19.3%	651	2	99.7%	1088.2
b20	0	71.1%	68.5%	3.65%	39.5%	38.4%	2.78%	22.1%	21.6%	2.26%	237	8	96.6%	93.58
b21	1	70.3%	67.5%	3.98%	39.2%	38.1%	2.81%	21.6%	21.2%	1.85%	278	5	98.2%	85.47
b22	0	76.4%	74.3%	2.75%	42.8%	42.0%	1.87%	22.6%	22.4%	0.88%	555	1	99.8%	131.63
b17	0	91.0%	90.8%	0.22%	48.3%	48.3%	0%	24.3%	24.3%	0%	37	0	100%	168.58
Average		72.2%	60.1%	17.0%	40.0%	32.6%	19.0%	22.4%	20.3%	9.18%			98.8%	

in Table 5, only a small number of initial test patterns violate the capture transition constraint (when compared to Huffman bode). Therefore, the compression ration loss in this coding scheme is only 4% in average after resolving most of the violated patterns. One special case to note is that for s38417, we avoid capture power violations (only two patterns) with the increase of compression ratio. We attribute this to the post-processing stage that decreases entropy in our algorithm.

The selective Huffman code [8] and the selective dictionary code [6] utilize the *alternate fill* [1] to fill X-bits. As our method for capture power reduction is independent to the different coding schemes (i.e., dependent on the fully-specified test patterns only), their experimental results are put in Table 6 together. Similarly, most of the initial power-unsafe patterns are resolved, at the cost of 9.18% compression ratio loss for selective dictionary code and 19.0% compression ratio loss for selective Huffman code in average, respectively.

Finally, as can be seen from the above tables, the execution time is in the range of tens to hundreds of seconds for all the benchmark circuits, which is acceptable.

5 Conclusion and Future Work

In this paper, we propose a generic framework for reducing scan capture power in fixed-length symbol-based test compression environment. Using the entropy of the test set to measure the impact of capture power-aware X-filling on the potential test compression ratio, the proposed method is able to keep test patterns' capture power under a safe limit with little loss in test compression ratio. Experimental results on benchmark circuits demonstrate the effectiveness of the proposed approach. In the future we plan to extend the proposed method on those variable-length symbol-based coding schemes.

6 Acknowledgements

This work was supported in part by the General Research Fund 417406, 417807, and 418708 from Hong Kong SAR Research Grants Council (RGC), in part by National Science Foundation of China (NSFC) under grant No. 60876029, in part by a grant N_CUHK417/08 from the NSFC/RGC Joint Research Scheme, and in part by the National High Technology Research and Development Program of China (863 program) under grant no. 2007AA01Z109.

References

[1] K. J. Balakrishman and A. Touba. Relationship Between Entropy and Test Data Compression. In *IEEE TCAD*, pp. 386–395, 2007.

[2] A. Chandra and K. Chakrabarty. Test Data Compression for System-On-Chip Using Golomb Codes. In *Proc. VTS* , pp. 113-120, 2000.

[3] P. Girard. Survey of Low-Power Testing of VLSI Circuits. *IEEE Design & Test of Computers*, 19(3):80–90, May-June 2002.

[4] P. T. Gonciari, B. M. Al-Hashimi, and N. Nicolici. Variable-length input Huffman coding for system-on-a-chip test. *IEEE TCAD*, 22(6):783–796, June 2003.

[5] *The International Technology Roadmap for Semiconductors (ITRS): 2001 Edition*. http://public.itrs.net/Files/2001ITRS/Home.htm, 2001.

[6] L. Li, K. Chakrabarty, and N.A. Touba. Test Data Compression Using Dictionaries with Selective Entries and Fixed-Length Indices. In *ACM TODAES*, pp. 470–490, 2003.

[7] J. Li, Q. Xu, Y. Hu, and X. Li. iFill: An Impact-Oriented X-Filling Method for Shift- and Capture-Power Reduction in At-Speed Scan-Based Testing. In *Proc. DATE*, pp. 1184-1189, 2008.

[8] A. Jas et al., An Efficient Test Vector Compression Scheme Using Selective Huffman Coding. In *In IEEE TCAD*, pp. 797–806, June 2003.

[9] C. Krishna, A. Jas, and N. Touba. Test Vector Encoding Using Partial LFSR Reseeding. In *Proc. ITC*, pp. 885–893, 2001.

[10] J. Rajski, *et al.* Embedded Deterministic Test. *IEEE TCAD*, 23(5):776–792, May 2004.

[11] S. Ravi et al., Power-aware Test: Challenges and Solutions. In *Proc. ITC*, 2007.

[12] S. Remersaro, *et al.* Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs. In *Proc. ITC*, paper 32.2, 2006.

[13] P. M. Rosinger, B. M. Al-Hashimi, and N. Nicolici. Scan Architecture with Mutually Exclusive Scan Segment Activation for Shift- and Capture-Power Reduction. *IEEE TCAD*, 23(7):1142–1153, October 2004.

[14] J. Saxena, *et al.* A Case Study of IR-Drop in Structured At-Speed Testing. In *Proc. ITC*, pp. 1098–1104, 2003.

[15] M. Tehranipour, M. Nourani and K. Chakrabarty. Nine-coded compression technique for testing embedded cores in SoCs. In *IEEE TVLSI*, pp. 719–731, June 2005.

[16] N. A. Touba. Survey of Test Vector Compression Techniques. *IEEE Design & Test of Computers*, 23(4):294–303, Jul.-Aug. 2006.

[17] Z. Wang and K. Chakrabarty. Test data compression for IP embedded cores using selective encoding of scan slices. In *Proc. ITC*, pp. 581–590, 2005.

[18] X. Wen, *et al.* Low-Capture-Power Test Generation for Scan-Based At-Speed Testing. In *Proc. ITC*, pp. 1019–1028, 2005.

[19] L. Whetsel. Adapting Scan Architectures for Low Power Operation. In *Proc. ITC*, pp. 863–872, 2000.

[20] Q. Xu, D. Hu, and D. Xiang. Pattern-Directed Circuit Virtual Partitioning for Test Power Reduction. *Proc. ITC*, paper 25.2, 2007.

[21] J.-L. Yang and Q. Xu. State-Sensitive X-Filling Scheme for Scan Capture Power Reduction. *IEEE TCAD*, 27(7):1338–1343, July 2008.