On Capture Power-Aware Test Data Compression for Scan-Based Testing

Jia Li[†§], Xiao Liu[‡], Yubin Zhang[‡], Yu Hu[†], Xiaowei Li[†*] and Qiang Xu[‡*]

[†]Key Laboratory of Computer System and Architecture, ICT, CAS, Beijing, China
[§]Graduate University of Chinese Academy of Sciences, Beijing, China
Email: {gracelee,huyu,lxw}@ict.ac.cn

[‡]CUhk REliable computing laboratory (CURE)
Deptartment of Computer Science & Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
Email: {xliu,ybzhang,qxu}@cse.cuhk.edu.hk

ABSTRACT

Large test data volume and high test power are two of the major concerns for the industry when testing large integrated circuits. With given test cubes in scan-based testing, the "don't-care" bits can be exploited for test data compression and/or test power reduction. Prior work either targets only one of these two issues or considers to reduce test data volume and scan shift power together. In this paper, we propose a novel capture power-aware test compression scheme that is able to keep scan capture power under a safe limit with little loss in test compression ratio. Experimental results on benchmark circuits demonstrate the efficacy of the proposed approach.

1. INTRODUCTION

With the ever increasing integration capability of semiconductor technology, today's large integrated circuits (ICs) require an increasingly amount of data to test them [6]. At the same time, as technology advances, test patterns that target delay faults and many other kinds of subtle faults are essential to improve test quality for deep submicron designs, in addition to the traditional stuck-at test vectors. Large test data volume not only raises memory depth requirements for the automatic test equipment (ATE), but also prolongs ICs' testing time, which significantly increases test cost. Various test compression techniques [14] have been proposed in the literature to address this problem, by taking advantage of the "don't-care" bits (also known as X-bits) in given test cubes.

At the same time, ICs' power dissipation in scan-based testing can be significantly higher than that during normal operation, in both shift mode and capture mode [3, 18]. Shift power violations contribute more to the circuit's accumulated power consumption, which may lead to structural damage to the circuit under test (CUT). Capture power violations, on the other hand, may cause good circuit to fail test (especially in at-speed testing), thus leading to unnecessary yield loss [13]. There is a rich literature on reducing test power in shift mode, in which design-for-testability (DfT) based methods such as scan chain partitioning technique [12, 17] are very effective (when compared to X-filling techniques such as [1]). There are, however, no such effective DfT-based techniques for capture power reduction, and we mainly resort to X-filling techniques (e.g., [11, 16, 19]) to resolve the excessive capture power problem.

*To whom correspondence should be addressed.

Given test cubes that feature a significant percentage of X-bits (typically larger than 95% [14]), prior work either targets test data compression only (e.g., [10, 15]) or tries to reduce shift and/or capture-power only (e.g., [1, 8, 11, 16, 19]). This is unfortunate because both problems are addressed using the very same X-bits and hence they contradict with each other. *As both large test data volume and high capture power are major concerns for the industry today, it is essential to develop a holistic solution that takes both problems into account.*

In this paper, we propose a novel capture power-aware test compression scheme (namely *CPA-compress*) to address the above problem. We select to use the architecture in a code-based test compression framework [15] and we introduce novel X-filling techniques into this architecture. Note, however, while the details of the proposed method is architecture-dependent, the methodology can be generalized and applied to other code-based test data compression schemes. With the proposed *CPA-compress* technique, we are able to eliminate capture power violations with little loss in test compression ratio, as demonstrated in our experimental results on ISCAS'89 and ITC'99 benchmark circuits.

The remainder of this paper is organized as follows. Section 2 gives the preliminaries and motivates this work. The proposed *CPA-compress* technique is detailed in Section 3. Experimental results on benchmark circuits are shown in Section 4. Finally, Section 5 concludes this paper.

2. PRELIMINARIES

2.1 Test Data Compression

Test data compression is widely-used in the industry nowadays to reduce the amount of test data stored on the ATE and to decrease testing time. With test compression, we store the test stimuli in a losslessly-compressed form in the ATE and decompress them to the original test set before shifting them into scan chains. The data decompression is conducted by some on-chip DfT logic. For the test responses, we usually use lossy compaction schemes such as multiple-input signature register (MISR) to generate very small signatures.

A significant amount of research has been conducted in test compression and a wide variety of techniques have been presented in the literature, which can be broadly classified into two categories: (i) nonlinear code-based schemes that use data compression codes to encode test cubes; (ii) linear decompressor-based schemes that decompress the data using linear operations (e.g., XOR network and/or linear feedback shift registers). The two types of techniques

978-1-4244-2820-5/08/$25.00 ©2008 IEEE

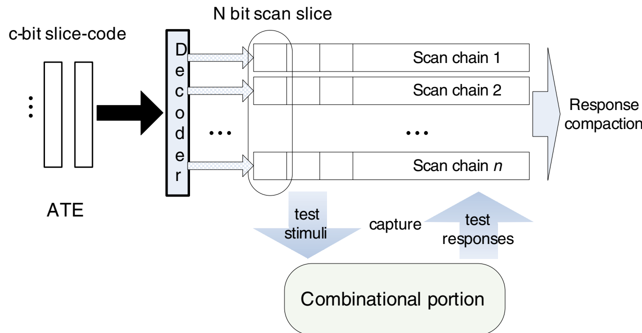

Figure 1: Test compression using selective encoding [13].

have their own pros and cons. As pointed out in [14], code-based schemes (e.g., [2, 4, 15]) can efficiently exploit correlations in the specified bits and do not require ATPG constraints, but linear techniques (e.g., [9, 10]) generally provide greater compression ratio. In this work, we mainly consider code-based schemes and we select to use [15] as our basic decompression architecture (discussed in the following paragraphs). It is important to note, however, while the details of the proposed method depends on the architecture in [15], the basic concept can applied to other code-based test compression schemes.

Fig. 1 shows the test compression architecture with selective encoding [15]. As can be observed from this figure, a series of c-bit slice-codes imported from ATE are decoded into N-bit scan slices[1] before they are fed to scan chains, where $c = \lceil \log_2(N+1) \rceil + 2$. It should be noted that a single scan slice may be encoded into one or more c-bit slice-codes, and each code contains two bits control-code and $\lceil \log_2(N+1) \rceil$ bits data-code. We briefly describe the encoding scheme as follows, please refer to [15] for details.

To start coding a scan slice, the first control-code bit is set to be '0' and the second control-code bit indicates the default value for this scan slice. Here *default value* is determined by comparing the number of 0- and 1-valued bits in the test stimuli. That is, the default value will be '1' if there are more '1's than '0's in one scan slice; otherwise the default value is set to be '0'. The data-code denotes the index of the care-bit that is different from the *default* value. If the scan slice contains only one such care-bit, we simply map the X-bits to be the default value and the encoding for this particular scan slice is finished (see the encoding for scan slice "X01X X0XX" in Table 1). If, however, the scan slice contains more care-bit that is different from the default value, we divide the N-bit slice into $\lceil N/K \rceil$ groups ($K = \lceil \log_2(N+1) \rceil$) and we need to introduce additional codes to encode the scan slice. For those groups wherein all bits are default values, we do not need to encode them. Otherwise, if a group contains one care-bit that is different from the default value, we map the X-bits to be the default value and then encode it with control-code '10' and data-code to be the index of this care-bit (namely single-bit-mode). If a group contains multiple such care-bits, the so-called group-copy-mode is introduced, we need two codes to encode it (with control-code '11' for both of them), wherein the first data-code represents the index of this group's first bit and the second data-code is just a copy of this group (we do not need to map the X-bits in this group to the default value in this case, see the encoding for scan slice "100X 0111" in Table 1). To further improve test compression ratio, adjacent groups can be merged into a *group-subslice* so that they can share the index code in the first group. It should be also noted

[1] A scan slice is the set of test data applied to the scan chain inputs at a scan cycle.

Table 1: A selective encoding example

Slice	Slice code: Control-code	Slice code: Data-code	Description
X01X X0XX	00	0010	Start a new slice, default value: '0', set bit 2 to '1'.
100X 0111	01	0100	Start a new slice, default value: '1', set bit 4 to '0'.
	11	0000	Enter group-copy-mode, starting from bit '0'.
	11	100X	The data is 100X.

that there must be at least one single-bit-mode code between two group-copy-mode codes to identify different group-subslice when describing a slice.

2.2 X-Filling for Test Power Reduction

Scan tests can increase the ICs' switching activities well beyond that of its normal operation. It is possible that the test power consumption exceeds the circuit's power rating in both shift mode and capture mode, and lots of research work has been dedicated to this area, as surveyed in [3]. While there has been some prior work on shift-power reduction with X-filling techniques (e.g., adjacent fill [1]), they are not as effective as those techniques based on scan chain manipulation (e.g, [12, 17]). Therefore, in this work, we assume the shift power issues are handled with the above DfT-based techniques at a small DfT cost. We mainly consider capture power reduction in order to avoid test yield loss.

Wen *et al.* [16] first proposed to use X-filling to achieve low capture-power tests, by filling the X-bits in the test stimuli to be the same as the known test responses as much as possible. One of the main limitations of this work is that the computational time is quite high, because the X-bits are filled incrementally and the time-consuming forward implications and backward justifications are extensively used in their approach. In [11], Remersaro *et al.* developed an efficient probability-based X-filling technique, namely *preferred fill*, which tries to fill all X-bits in the test cube in one step, instead of using incremental fill and logic simulation. Their technique, however, is inherently less effective as the available information for the probability calculation in their single-step filling is quite limited. Recently, Yang and Xu [19] proposed a state-sensitive X-filling scheme that achieves much better scan capture power reduction.

The above works try to reduce capture power consumption as much as possible. This is however unnecessary because the correct operation of the circuits can be guaranteed as long as the peak capture power does not exceed a certain threshold [7, 8].

2.3 Motivation

With given test cubes that feature a large number of X-bits, code-based test compression and X-filling-based capture power reduction both utilize the very same X-bits. Obviously they contradict with each other, and *the relevant question is: can we develop a holistic solution that takes both problems into consideration?*

The answer is positive, as can be observed in the example shown in Fig. 2. In Case 1, with the given initial scan slice and the response probabilities calculated as in [11], we can calculate the transition probability for each scan cell as $P_1(s) \times P_0(r) + P_0(s) \times P_1(r)$, where $P_{1/0}(s/r)$ is the probability to have '1/0' as the logic value of test stimulus/response in this scan cell. With the original selective encoding scheme in [15], we will fill the scan slice as "0000 0000 0000" with one slice-code and the expected number of transitions (i.e., the sum of all scan cells' transition probabilities) is 7.2. If, however, we fill this slice as "0111 1111 1111", we can also en-

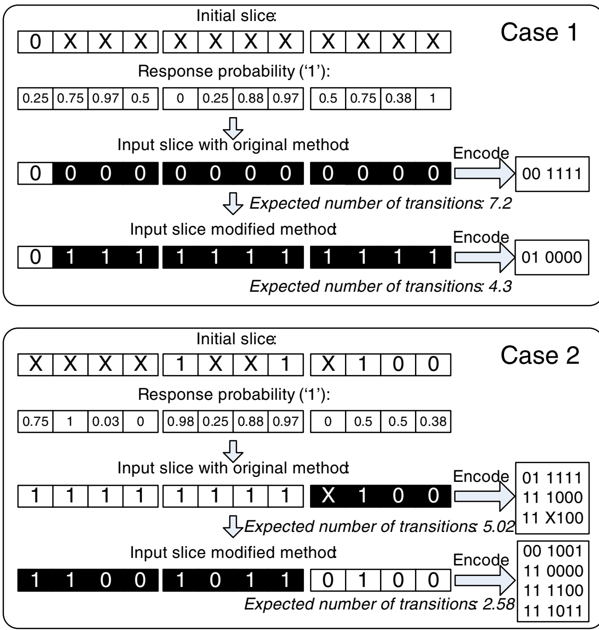

Figure 2: Motivational example.

code it with a single code, but the expected number of transitions is reduced to 4.3. Consider another example scan slice with more care bits in Case 2, it will be filled as "1111 1111 X100"in [15] , which results in 5.02 expected transitions. Suppose we fill this slice to be "1100 1011 0100", the transitions can be dropped to 2.58, but we need one more code to encode this scan slice.

From the above example, we can see that X-fillings for scan slices have a large impact on both test compression ratio and capture power consumption. With an effective X-filling strategy, we can reduce the capture power significantly without much loss in compression ratio. At the same time, because it is not necessary to reduce the capture power as much as possible, we can put more emphasis on test compressing when utilizing X-bits. The above motivates the work studied in this paper.

3. PROPOSED ALGORITHM

In the proposed *CPA-compress* algorithm, we need to consider the impact of X-bits on both test compression and capture power. For a particular X-bit in a scan slice, its impact on test compression can be easily obtained by checking whether we need more codes to encode the slice after filling it; while its impact on capture power can be estimated by the expected number of transitions after filling it. Based on this, we try several methods to avoid capture power violations while keeping the compression ratio as large as possible.

The flowchart of the proposed algorithm is shown in Fig. 3, including four stages. In stage 1, we determine the default value for each scan slice. Stage 2 tries to fill X-bits to reduce capture power without test compression ratio loss. If we are unable to meet the capture power constraint after stage 2, we consider to further reduce capture power at the cost of slight compression ratio loss in stage 3, taking advantage of the group-subslice feature in [15]. Finally, if the capture power is still beyond the threshold, in stage 4, we fill extra X-bits in the scan slice and setup more groups to encode, resulting in less test compression ratio. The above is an iterative process, e.g., when we fill some X-bits in stage 4, we will try stages 2 and 3 again to further reduce capture power with less test compression ratio loss. The iteration is also helpful for the

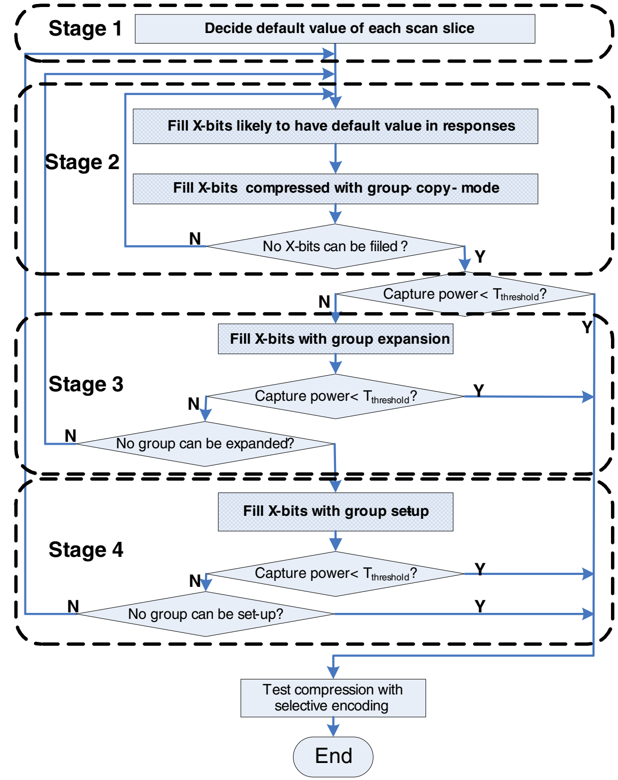

Figure 3: Capture power-aware test data compression flow.

accuracy of the capture power estimation because we have more confidence for the probabilities of the responses once the values of more X-bits are determined. Whenever our capture power transitions are less than the pre-determined threshold, the remaining X-bits in the scan slice will be encoded with the method in [15]. The details of the above four stages are discussed in detail in the following subsections.

3.1 Default Value Decision for Scan Slices

As discussed in Section 2.1, [15] determines the default value merely by comparing the number of 0- and 1-valued bits in test stimuli. Such method, however, may cause large amount of capture transitions. An example is shown in Fig. 4, wherein the scan slice contains eight bits and one bit is '0' while all others are X-bits. [15] will set the the default value to be '0' and fill all the X-bits with '0'. With the probabilities for the test responses (calculated as in [11]) as shown in the figure, the above default value '0' will cause at least five transitions in the capture cycle, which is quite large.

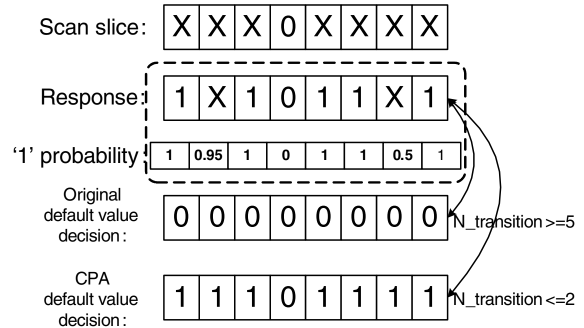

Figure 4: Default value decision example.

Our *CPA-compress* algorithm takes the above issue into consideration and we use the following formula to determine the default value for each scan slice:

$$D = \begin{cases} 1 & , \quad p+m > q+n \\ 0 & , \quad p+m \leq q+n \end{cases} \qquad (1)$$

where p and q are the number of care-bits with value '1' and '0' in the test stimuli, while m (n) denote the number of X-bits that are '1' ('0') or have high probabilities to be '1' ('0') in the test responses. We define a *probability threshold* P_{th} and we say a X-bit in the test response is *"likely to have"* value '1'/'0' when its probability is higher than P_{th} (including the case when it is determined to be '1'/'0'). For the example in Fig. 4, suppose we set $P_{th} = 0.9$, we have $p = 0$, $q = 1$, $m = 6$, and $n = 1$. Since $p+m > q+n$, we will set the default value for this scan slice to be '1' instead and the number of capture transitions will be no more than two.

3.2 X-Filling for Capture Power Reduction with Value Deduction

In this stage, we try to decrease capture power transitions without introducing extra compression codes. The algorithm is shown in Fig. 5, wherein the X-bits are filled iteratively as following.

In each iteration, we first calculate the probabilities for the response bits (line 3). Next, we select those X-bits in the test stimuli whose corresponding response bits that are "likely to have" the same value as the default value x_d and fill them with x_d (lines 4-5). Similarly, we select those X-bits that are kept intact in group-copy-mode during the encoding process and fill them as the same value as the corresponding response bits (lines 6-7). During each iteration, with more X-bits in the stimuli determined, it is expected that more test response bits are deduced to be "likely to have" '1'/'0', which enables us to fill more X-bits in the following iteration. Capture transitions are reduced in every iteration and the procedure ends when capture transitions are reduced to be less than the threshold value or no X-bits conforming to the filling rule in this stage exists.

Stage 2: X-filling with value deduction

INPUT: $S = \{s_i\}$: Scan slices
OUTPUT: $S' = \{s'_i\}$: Scan slices with more filled x-bits

```
1.  while (more X-bit can be filled under the stage rule) {
2.      for each scan slice, s_i, {
3.        Compute the circuit's response, R_i;
4.        Search bits in R_i that are "likely to have" default value x_d;
5.        Fill the corresponding bits in s_i with x_d to get s'_i;
6.        Search the X-bits in R_i in group-copy-mode;
7.        Fill the X-bits in group-subslices to get s'_i;
8.      }
9.  }
10. if (capture power, CP_i ≤ threshold T_th) {
11.     Go to the end;
12. }
13. else Go to Stage 3;
```

Figure 5: Procedure for X-filling with value deduction.

The above X-fillings do not introduce any extra codes during the encoding process, as can be shown in the example in Fig. 6. For a scan slice {XXXX 011X 11X0 001} wherein {011X 11X0} is to be coded in group-copy-mode, with both the encoding scheme in [15] and the proposed method, this scan slice will be encoded with four codes. Because the capture transitions are reduced in each iteration, our method achieves a much lower capture power consumption with the same compression ratio as in [15].

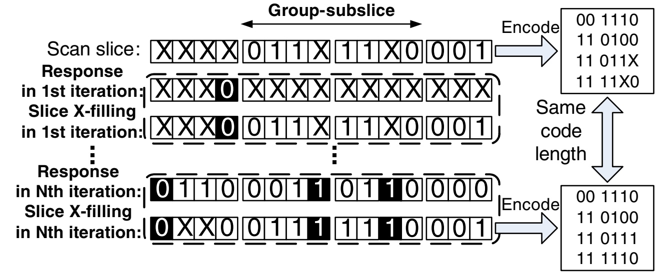

Figure 6: X-filling with value deduction.

3.3 X-Filling for Capture Power Reduction with Group Expansion

Stage 2 tries to reduce capture power without any test compression ratio loss. However, we may not be able to meet the capture power constraint after this step, and we have to trade compression ratio for capture power reduction in such cases.

One observation from the selective encoding scheme in [15] is that, when the *group-subslice* feature is utilized, we are able to expand a group to one of its neighbors (namely *group expansion*) and have at most N_{group} free X-bits (depending on the number of care-bits in the neighboring group) to fill by introducing only one additional code (*group-subslice* index may also need to be modified).

Stage 3: X-filling with group expansion

INPUT: $S = \{s_i\}$: Scan slices
OUTPUT: $S' = \{s'_i\}$: Scan slices with expanded groups

```
1.   Set transition reduction expectation N_m = N_group;
2.   while (N_m > 0) {
3.     for each scan slice, s_i {
4.       Calculate reduced transition in adjacent group N_ra;
5.       while (N_ra>=N_m){
6.           Expand current group-subslice with the group;
7.           Search the bits in R_i "likely to have" '0' or '1';
8.           Fill corresponding X-bits to get s'_i;
9.           Repeat the process of Stage 2;
10.          if (capture power, CP_i ≤ threshold T_th) {
11.            Go to the end;
12.          }
13.          Calculate reduced transition in adjacent group N_ra;
14.      }
15.    }
16.    N_m−−;
17.  }
```

Figure 7: Procedure for X-filling with group expansion.

Fig. 7 shows the procedure of this stage inspired by the above observation. First, a group is selected to be expanded to its neighboring group-subslice if this operation results in the maximum reduction of capture transitions. With N_{group} bits in each group, it is obvious that the maximum capture transition reduction that can be achieved with the group expansion technique is N_{group}. Therefore, to select the most effective group, we initialize the the transition reduction expectation to be $N_m = N_{group}$ (line 1). If a group whose expansion can induce the expected N_m transition reductions can be found, we shall conduct this expansion (lines 6-8). Since more X-bits have been determined and it is possible to be able to further reduce capture power without test compression loss, we will repeat the procedures in stage 2 again. If such a group cannot be found,

the procedure iterates with decreased N_m. Stage 3 terminates when capture transitions are reduced to be less than the threshold value or $N_m = 0$. With the help of the proposed group expansion technique, a large amount of X-bits can be exploited for capture power reduction with little compression ratio loss.

An example is shown in Fig. 8 to illustrate the effectiveness of stage 3. With the initial group-subslice expanded to its left neighboring group, capture transitions can be reduced by three with an additional code. If the new group-subslice is expanded to its left neighboring group again, two more transitions can be reduced at the cost of another additional code.

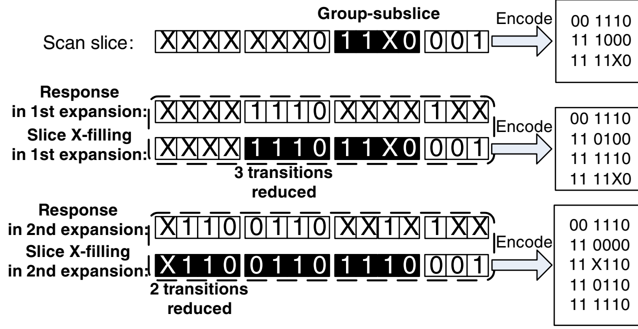

Figure 8: X-filling with group expansion.

3.4 X-Filling for Capture Power Reduction with Group Set-up

If capture power needs to be further reduced after stage 3, we have no other choice but to add new group in group-copy-mode. According to [15], at least two additional codes are necessary to encode a new group. One of them is for the address and the other is for the data. In addition, if there is already a group-subslice in the scan slice, one more code in single-bit-mode is required to differentiate from the two subslices.

The procedure to reduce capture power by group set up is shown in Figure 9. Similar to stage 3, the transition reduction expectation with new group setup is initialized as $N_m = N_{group}$ in the beginning (line 1). If a group whose setup can reduce capture power by N_m can be found, we will setup this group and encode it (lines 6-8). Similarly, we have more X-bits with determined values after a group is setup, and hence we should try to run stage 2 and stage 3 again to further reduce capture power with less compression loss (lines 9-16).

Our *CPA-compress* algorithm will terminate itself after stage 4 no matter whether there are still some capture power violations or not in the tests.

4. EXPERIMENTAL RESULTS

Table 2 shows the experimental results of our proposed *CPA-compress* technique on some larger ISCAS'89 and ITC'99 benchmark circuits. We use MINTEST [5] to generate test cubes for these benchmark circuits. Column "circuit" denotes the name of these circuits. The number of scan cells and test patterns in these circuits are listed in Column "#dff" and "#pattern". The number of scan chains in the circuits are determined according to the sizes of the circuits, and they are shown in Column "#sc". Column "X%" refers to the percentages of X-bits in test patterns. We set the capture transition threshold to be $\#dff/3$ for the ISCAS'89 circuits, $\#dff/5$ for the ITC'99 circuits b20, b21 and b22, and $\#dff/10$ for the ITC'99 circuits b17, b18 and b19.

Stage 4: X-filling with group set-up

INPUT: $S = \{s_i\}$: Scan slices
OUTPUT: $S' = \{s'_i\}$: Scan slices with new groups

1. Set transition reduction expectation $N_{mr} = N_{group}$;
2. **while** ($N_{mr} > 0$) {
3. **for** each scan slice, s_i {
4. Search the group with maximal transition reduction N_{ng};
5. **while** (N_{ng}==N_{mr}) {
6. Set-up the group;
7. Search the bits in R_i "likely to have" '0' or '1';
8. Fill corresponding X-bits in new set-up group to get s'_i;
9. Repeat the process of Stage 2;
10. **if** (capture power, $CP_i \leq$ threshold T_{th}) {
11. Go to the end;
12. }
13. Repeat the process of Stage 3;
14. **if** (capture power, $CP_i \leq$ threshold T_{th}) {
15. Go to the end;
16. }
17. Search the group with maximal transition reduction;
18. }
10. }
20. $N_{mr}--$;
21. }

Figure 9: Procedure for X-filling with group set-up.

In Table 2, "Orig ratio" and "CPA ratio" stand for the compression ratio with the original selective encoding compression scheme in [15] and the proposed $CPA-compress$ scheme. "Orig N_{trans}" and "CPA N_{trans}" represent the average capture transition count of these two schemes. The number of test vectors that violate capture transition threshold of these two schemes are referred to "Orig N_{vios}" and "CPA N_{vios}", respectively. The compression ratio loss (CRL) and the percentage of capture power reduction (CPR) are denoted with "*CRL*" and "*CPR*" . Finally, the execution time of the proposed *CPA-compress* scheme with a 2.99GHz Intel PC with 512MB memory is shown in Column "Exe. time".

The effectiveness of our approach can be observed clearly from Table 2. The capture power transitions are highly reduced (25.1% on average) and the capture transition violations can be totally eliminated, at the cost of little compression ratio loss for most of the circuits. At the same time, it can be noticed that the compression ratio of some circuits (s13207 and s15850) even increases with the proposed technique. This can be explained as follows. The proposed technique and the one in [15] may select different default values for a scan slice. While [15] determines the default value such that less care-bits exist in the scan slice, this strategy does not guarantee to be encoded with less codes, because the to-be-encoded care-bits may spread evenly in the slice and each of them costs one code. If a different default value is selected and the corresponding care-bits are localized, it is possible to encode them with less codes in group-copy-mode event though the number of care-bits is larger. Consider an example slice {X101 1110 X0XX 0XX}. With [15], the default value is determined to be '1' (there are 5 '1' and 4 '0' in the slice), and hence it will be filled as {1101 1110 1011 011} and we need 4 codes to compress this slice: {01 0010;10 0111;10 1001;10 1100}. If, however, the default value is selected as '0' to lower capture power in our method, this slice becomes {0101 1110 0000 000} and will be encoded as {00 0001;11 0011;11 1111} with 3 codes only. Consequently, both higher test compression ratio and less capture power can be obtained for certain circuits with *CPA-compress* technique.

Table 2: Experimental results for capture power and test compression ratio

circuit	#dff	#pattern	#sc	X%	Orig ratio	CPA ratio	*CRL*	Orig N_{trans}	CPA N_{trans}	*CPR*	Orig N_{vios}	CPA N_{vios}	Exe. time(s)
s15850	534	126	32	83.66%	2.31	2.84	-22.9%	103	57	44.7%	4	0	13.344
s13207	638	236	32	93.23%	2.73	3.12	-14.3%	153	64	58.2%	32	0	30.266
s38417	1636	99	32	67.80%	1.45	1.4	3.4%	316	270	14.6%	0	0	61.110
s38584	1426	136	64	82.48%	2.56	2.47	3.5%	330	243	26.4%	4	0	66.953
b20	490	350	32	73.24%	1.47	1.47	0.0%	63	55	12.7%	27	0	30.328
b21	490	369	32	74.02%	1.52	1.51	0.7%	64	56	12.5%	48	0	38.546
b22	735	373	32	73.87%	1.54	1.53	0.6%	83	76	8.4%	6	0	90.735
b17	1415	435	64	89.17%	2.89	2.71	6.2%	109	87	20.2%	98	0	88.797
b18	2988	571	128	90.17%	5.01	4.67	6.8%	236	177	25.0%	87	0	436.469
b19	5736	824	128	92.90%	7.25	6.98	3.7%	450	323	28.2%	60	0	1197.296
Average				82.05%			-1.2%			25.1%			

The computational time of the proposed technique is shown to be acceptable. For the largest circuit b19 with over 5000 flip-flops, the computational time is about 20 minutes. It should be noted that there are several ways to speed up our algorithm if the computational time is a big concern. For example, we can decrease the probability threshold value used in X-filling (e.g., from 0.9 to 0.8), which will effectively increase the number of X-bits to be filled in each pass, thus reducing computational time. In our experiment, the running time for b19 is reduced to be 867.187s, with slightly increase of CPA N_{trans} to be 331. We can also reduce the computational complexity of the proposed algorithm by calculating the response probabilities less times (i.e., not in every pass), again, at the cost of less capture power reduction.

5. CONCLUSION

Large test data volume and high test power consumption are two of the major concerns for the industry when testing large integrated circuits. With given test cubes in scan-based testing, prior work mainly targets only one of these two issues. In this paper, we study the impact of different X-bits on scan capture power and test compression ratio, and fill them intelligently to achieve a capture power-aware test compression scheme, namely *CPA-compress*. The proposed holistic solution is able to keep scan capture power under a safe limit with little loss in test compression ratio, as demonstrated in our experimental results on various benchmark circuits.

6. ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China (NSFC) under grant No. 60633060, 60776031, 90607010, 60606008, in part by National Basic Research Program of China (973) under grant No. 2005CB321604, 2005CB321605, in part by the National High Technology Research and Development Program of China (863 program) under grant no. 2007AA01Z109, 2007AA01Z107 and 2007AA01Z113, and in part by the Hong Kong SAR RGC Earmarked Research Grant 417406 and 417807.

7. REFERENCES

[1] K. M. Butler, *et al*. Minimizing Power Consumption in Scan Testing: Pattern Generation and DFT Techniques. In *Proc. International Test Conference (ITC)*, pp. 355–364, 2004.

[2] A. Chandra and K. Chakrabarty. Reduction of SOC Test Data Volume, Scan Power and Testing Time Using Alternating Run-Length Codes. In *Proc. Design Automation Conference (DAC)*, pp. 673–678, 2002.

[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 Transactions on Computer-Aided Design*, 22(6):783–796, June 2003.

[5] I. Hamzaoglu and J. H. Patel. Test Set Compaction Algorithms for Combinational Circuits. In *Proc. International Conference on Computer-Aided Design (ICCAD)*, pp. 283–289, 1998.

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

[7] J. Li, Q. Xu, Y. Hu, and X. Li. On Reducing Both Shift and Capture Power for Scan-Based Testing. In *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 653-658, 2008.

[8] 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. Design, Automation, and Test in Europe (DATE)*, pp. 1184-1189, 2008.

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

[10] J. Rajski, *et al*. Embedded Deterministic Test. *IEEE Transactions on Computer-Aided Design*, 23(5):776–792, May 2004.

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

[12] 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 Transactions on Computer-Aided Design*, 23(7):1142–1153, October 2004.

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

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

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

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

[17] L. Whetsel. Adapting Scan Architectures for Low Power Operation. In *Proc. International Test Conference (ITC)*, pp. 863–872, 2000.

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

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