Compression-Aware Capture Power Reduction for At-Speed Testing

Jia LI*

School of Software
Tsinghua University
Beijing, CHINA 100084
e-mail: jiali@mail.tsinghua.edu.cn

Qiang XU

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
e-mail: qxu@cse.cuhk.edu.hk

Dong XIANG

School of Software
Tsinghua University
Beijing, CHINA 100084
e-mail:dxiang@tsinghua.edu.cn

Abstract— Test compression has become a *de facto* technique in VLSI testing. Meanwhile, excessive capture power of at-speed testing has also become a serious concern. Therefore, it is important to co-optimize test power and compression ratio in at-speed testing. In this paper, a novel X-filling framework is proposed to reduce capture power of both LoC and LoS at-speed testing, which is applicable for different test compression schemes. The proposed technology has been validated by the experimental results on larger ITC'99 benchmark circuits.

I. Introduction

Large test data volume is a serious concern for the semiconductor testing nowadays, because it not only prolongs the testing time of the integrated circuits (ICs), but also raises the tester memory depth requirements [1]. To address this problem, various test data compression techniques have been proposed in the literature, by exploiting the "don't-care" bits (i.e., X-bits) in the given test cubes[1] [2].

At the same time, excessive switching activities during testing has also become a serious problem. Test power can be categorized into shift power and capture power. Shift power can typically be reduced by introducing extra DfT, while capture power can usually be reduced by ATPG methodology. In scan-based at-speed testing, capture power is more important since its clock frequency is higher. Using the two-cycle at-speed testing as an example, there are two widely-used methods to generate the two successive test vectors:$< v_1, v_2 >$, namely, *Launch-off-Capture (LoC)* and *Launch-off-Shift (LoS)* schemes. In LoC scheme, v_2 is produced by capturing the response of the initial vector v_1 in the scan chain; while in LoS scheme [3], v_2 is launched by the last shift cycle, as shown in Fig. 1. Though LoS requires a fast scan enable (SE) signal, which makes it less popular in the industry, several approaches have been proposed to resolve this issue [4]. If circuit transition count in the "test cycle"(Fig. 1) of at-speed testing exceed certain safety limit, there will be power-supply-noise-related effects, thus turning good chips fail the test and leading to unnecessary yield loss [5, 6].

*This work was supported in part by the National Science Foundation of China under grants No. 60425203, 60876029, 60910003 and 61006017, in part by the National 863 Project under grant No. 2009AA01Z129, in part by the Key Laboratory of Computer System and Architecture, ICT, CAS(ICT-ARCH200902), and in part by China Postdoctoral Science Foundation under grants No. 20100470014.

[1]A test cube is a deterministic test vector in which the bits that test generation tool does not assign are left as don't-cares.

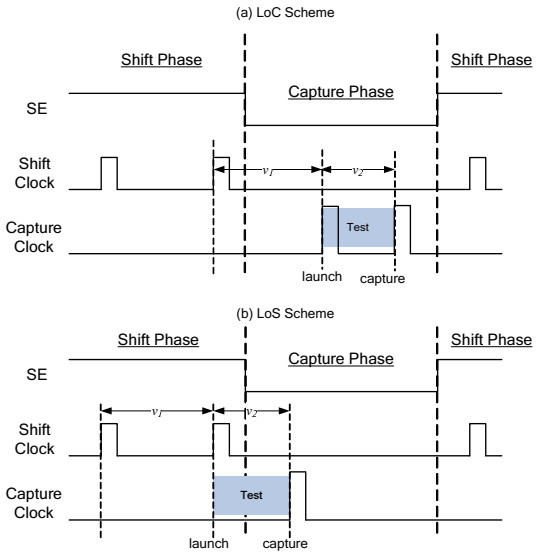

Fig. 1. At-speed scan testing schemes.

X-filling has been proven to be an effective approach on reducing capture power with low overhead [7–10]. However, existing techniques mainly focus on LoC scheme and are not readily applicable for LoS scheme. In addition, as test compression and low power X-filling might take advantage of the very same X-bits for different goals with different approaches, it is essential to develop a solution that targets co-optimizing both together. There were recent works attempted to solve this problem [11–14], but they still target LoC scheme, and can only be applied to specific test compression scheme.

In at-speed testing, the capture power of the circuit under test (CUT) can be reduced by suppressing the scan cell transitions in its launch cycle (the cycle before the first system clock in LoC scheme or the last shift cycle in LoS scheme). As verified in prior works [8–13], it is helpful to reduce the logic value differences between v_1 and its response vector v_2 by filling the X-bits in v_1 intelligently; while in LoS scheme, v_1 is turned into v_2 by a shift clock. Therefore, *Adjacent fill* [7] originally proposed for shift power reduction can be adopted to reduce the capture power of LoS scheme. However, since each X-bit has two adjacent bits, thus it may cause two scan cells to switch in the test cycle. Hence it is required an enhanced *Adjacent fill* such that only scan cells with smaller impacts on capture power switch. More importantly, if all the X-bits are filled for power reduction in the test cycle of at-speed testing, there might be significant test compression loss.

978-1-4244-7516-2/11/$26.00 ©2011 IEEE

Motivated by the above, this paper proposes a *Low Power, High Compression (LPHC)* X-filling framework that can co-optimize capture power and test compression ratio in the context of LoC/LoS at-speed testing for different test compression schemes. The contribution of this paper includes:

1. This work proposes to identify "*X-candidate*" that can be filled with low compression ratio loss by analyzing entropy of the test vector before X-filling for different kinds of test compression schemes.
2. The proposed "*X-propagation*" metric is able to evaluate the impact of the X-bits on capture power of LoC/LoS at-speed testing.
3. In the proposed framework, the "*X-candidates*" selection and the "*X-propagation*" metric are independent parts, which guarantees the general applicability of the proposed framework in different X-filling and test compression strategies.

The remainder of this paper is organized as follows: in Section II, we discuss the background of test compression and low power X-filling techniques; Section III introduces the proposed "*X-candidate*" selection and "*X-propagation*" metric in detail; Section IV demonstrates the proposed *LPHC* X-filling framework; Experimental results on benchmark circuits are given in Section V to show the effectiveness of the proposed work; Finally, Section VI concludes this paper.

II. Background

It has been shown that test cubes may contain as much as 95%-98% X-bits, which can be filled without affecting the CUT's fault coverage[2]. However, as illustrated in the introduction, since the very same X-bits can be used for both test compression and capture power reduction, this section will analyze the characteristics of efficient X-filling strategies for test compression and test power reduction by reviewing related prior works.

A. X-filling for Test Compression

In encoding-based test compression, a test cube can be divided into multiple **codewords** with different or the same length. A group of compatible codewords can be represented by one **symbol**. Since existing encoding-based test compression schemes can be categorized into fixed-symbol-length and variable-symbol-length schemes [15], we will also analyze the impact of X-filling on these two schemes.

A.1 Fixed-Symbol-Length Schemes

In [15, 16], the authors studied to use "entropy" to estimate the test compression ratio limit, which is defined as:

$$H = -\sum_{i=1}^{n} p_i \times \log p_i \quad (1)$$

where p_i is the probability of occurrence of symbol x_i in the test cubes and n is the total number of unique symbols. The entropy indicates the minimum average number of bits required for each codeword. Therefore, the maximum compression ratio can be estimated as:

$$TCR_{max} = (L - H)/L \quad (2)$$

where L is the length of the symbol. Equation (2) shows that the lower the entropy is, the higher test compression ratio can be obtained. Two approaches have been proposed to fill the X-bits in test cubes to achieve lower entropy: *Greedy Fill* and *Alternate Fill* [15], among which *Alternate Fill* is more scalable for larger test cubes and different symbol lengths. The main idea of *Alternate Fill* is to first order the three-valued unspecified symbols from the highest frequency of occurrence to the lowest, and then merge them according to this order to obtain more skewed frequency distribution of the vector to be filled.

After all the merging processes in *Alternate Fill*, there are probably still some X-bits left in the test vector unfilled, these X-bits can be further filled without affecting the maximum compression ratio.

A.2 Variable-Symbol-Length Schemes

The maximum compression ratio calculation of variable-symbol-length compression schemes is slightly different, where the L in Equation (2) now means the average symbol length, which is calculated as:$L = \sum_{i=1}^{n} p_i \times |x_i|$, where $|x_i|$ is the length of symbol x_i [15].

X-filling for minimizing entropy of variable-symbol-length schemes is still an open problem. Existing solutions usually fill the X-bits by simply replacing them with '0's. However, X-filling for capture power reduction usually need to replace some of these '0's with '1's, which will shorten the run-length of the symbol and thus increasing the number of symbols to be encoded. So it is bad for achieving the minimum entropy.

Therefore, the problem of X-filling for power reduction in variable-symbol-length schemes mainly lays on how to fill X-bits intelligently with '0's and '1's for power reduction without or with low impact on the run-length of the codewords, which will be analyzed and solved by this work in the following sections.

B. X-filling for Test Power Reduction

Various X-filling techniques have been proposed in the literature to reduce the test power [7–10]. Since they may contradict with test compression techniques , [11–13] proposed to fill X-bits under the constraint of specific test compression requirement. However, these technique all targeted at capture power of LoC scheme and are only applicable to one specific type of test compression scheme.

Two steps are essential to efficiently fill X-bits for test power reduction: 1) deciding filling order of the X-bits and 2)deciding the logic values to be filled for the X-bits.

Since the objective of capture power reduction is to control it under the safety limit [9–13], we prefer to fill X-bits with higher impact on capture transitions earlier, thus fewer X-bits

[2]X-filling does have implications on the CUT's defect coverage, but it is beyond the scope of this paper.

need to be filled for capture power reduction and more X-bits can be left for test compression.

As illustrated in Fig.1, In LoC scheme, the X-bits in v_1 should be filled as compatible as the corresponding test response bits in v_2, thus reducing the capture power after v_2 is applied. However, to reduce the excessive logic value differences between v_1 and v_2 in LoS scheme, the X-bits should be filled according to their adjacent bits.

III. Proposed Metrics for X-Filling

As analyzed in the previous section, two steps are necessary for efficient X-filling procedure: {**deciding the filling order**} and {**deciding the filling value**}. Therefore, two metrics are proposed accordingly in this paper: "*X-candidates*" and "*X-propagation*". "*X-candidates*" is selected considering the impact of X-bits on test compression, which means the X-bits that may cause less test compression ratio loss are selected to be filled first. "*X-propagation*" is calculated according to the impact of X-bits on the capture power of the CUT, which is used to determine the filling order among the "*X-candidates*" and the proper logic values for them. This section will illustrate the details of the above metrics, and the proposed X-filling Framework will be introduced in the next section based on them.

*A. "***X-candidates***" Selection*

According to different maximum test compression ratio calculations, the selection of "*X-candidates*" in fixed-symbol-length and variable-symbol-length test compression schemes are also different.

A.1 Fixed-Symbol-Length Schemes

As discussed in the previous section, there could be X-bits left after merging the compatible codewords by *Alternate Fill* [17], which can be filled without affecting the entropy of the entire test vector. As the test cube shown in Fig. 2, after merging of the codewords, there will be 7 different symbols in this test cube:{X000, 011X, 110X, 001X, X001, 01X0, 111X}, the X-bits in one symbol mean none of the codewords compatible to this symbol in the test cube have specified logic values in these positions. Therefore, they can be filled without affecting probability distribution of these symbols, thus leading to no entropy loss. So they are selected as "*X-candidates*". After filling one "*X-candidate*" in one codeword, the corresponding position in the symbol will be updated to specified logic value according to the filled X-bit in the codeword. So the "*X-candidates*" should be updated accordingly. If the capture power is still higher than the safety limit after all the X-bits in the symbols are filled, the X-bits in the codewords should be selected as "*X-candidates*" to achieve capture power reduction with some compression ratio loss.

A.2 Variable-Symbol-Length Schemes

It is also important to reduce the symbol number and skew the frequency distribution of the symbols for achieving lower entropy in variable-symbol-length test compression schemes. In these schemes, the codewords are usually divided into symbols with different run-length of '0's. We notice that though the symbols have different run-lengths, there exists a maximum run-length L_{max}, such that if the run-length exceed L_{max}, whether the last bit of this symbol is '1' or '0 will not increase the number of symbols. For example, if $L_{max} = 4$, there will be 5 types of symbols: {1, 01, 001, 0001, 0000}. If we change any '0' in the first 4 symbols into '1', the original symbol will be split into two new symbols. For example, if we change the second '0' into '1' in '0001', it will become two '01's. But if we change the last '0' into '1' in the 5^{th} symbol, there will produce no additional symbol. So if there are X-bits in this position, they can be selected as "*X-candidates*". Since the run-lengths of X-bits in given test cube are usually much longer than L_{max}, there could be many codewords with the maximum symbol length, whose last bits can be selected as "*X-candidates*".

Vector1	111X	XX01	001X	X000	110X	011X
Vector2	X000	001X	X001	011X	01X0	X000
Vector3	X001	011X	011X	110X	110X	X000
Vector4	001X	X000	X000	01X0	110X	011X

Fig. 2. Test cube divided into 4-bit codewords.

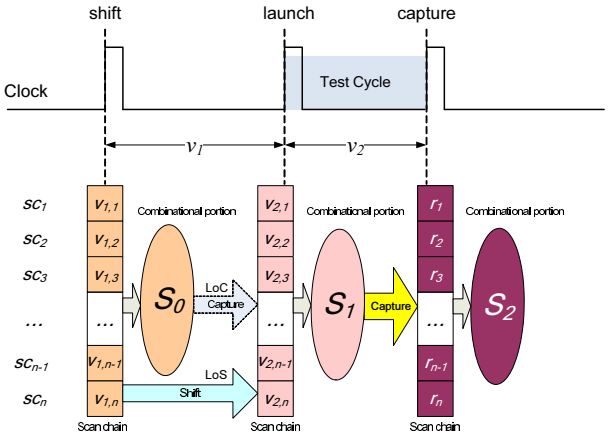

Fig. 3. Capture Transitions in the "test cycle".

*B. "***X-propagation***" Calculation*

Since there would be many "*X-candidates*" that can be filled with little or no compression ratio loss in given test cubes, and filling one "*X-candidate*" may have impact on further "*X-candidates*" selection as explained in the previous subsection, it is important to decide the filling order and values for the "*X-candidates*". In this subsection, the "*X-propagation*" metric is proposed to estimate the impacts of X-bits on the capture power of both LoC and LoS at-speed testing schemes.

In Fig. 3, sc_i stands for the i^{th} scan cell in a scan chain, $v_{1,i}$ and $v_{2,i}$ represents the i^{th} bit in v_1 and v_2 $(1 < i < n)$, S_0 and S_1 are the states of the CUT before and after the launch clock, and S_2 is the state of the responses. From this figure we can see that, the transitions in the "test cycle" of at-speed testing are caused by logic value differences between S_0 and

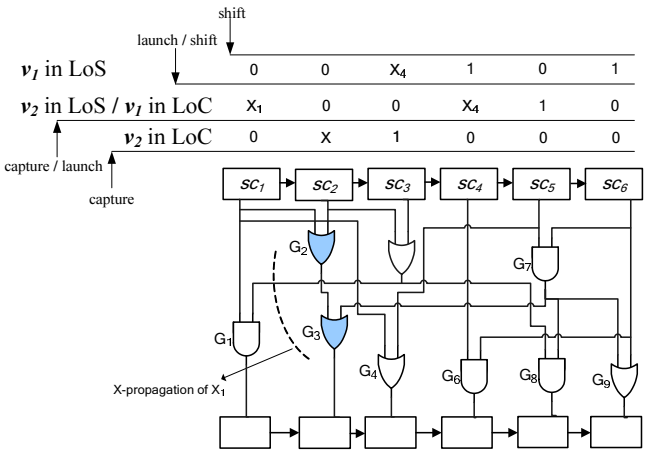

Fig. 4. "*X-propagation*" of X-bits in v_1 of LoC and v_1 & v_2 of LoS.

S_1, which are decided by Hamming distance between the pair of test stimuli vectors (v_1 and v_2). The difference between LoC and LoS schemes in calculating the Hamming distance between v_1 and v_2 is: in LoC scheme, the correlation between v_1 and v_2 is decided by the combinational portion of the CUT; while in LoS scheme, $v_{2,i} = v_{1,i-1}, 1 < i < n$.

For example in Fig. 4, to reduce the Hamming distance between v_1 and v_2 in LoC scheme, X_1 and X_4 can be filled with '0's to eliminate the switching activities in sc_1 and sc_4, but the X-bits in v_2 will also have specific logic values now. If they are different from the corresponding test bits in v_1, there may be switch in sc_2; in LoS scheme, X_1 can be filled with '0' to stop sc_1 from switching in the "test cycle", while for X_4, since it has two adjacent bits, it may cause sc_3 or sc_4 to switch. To reduce capture transitions in the entire CUT, it should be filled to stop the scan cell that may cause more transitions in the combinational portion from toggling. Therefore, we should decide the priorities of the "*X-candidate*s" by the transition count it may cause, which is defined as "*X-propagation*" in this work.

In Fig. 4, In LoC scheme, since X_1 can be propagated to G_2 and G_3 (the output of these gates are also X-bits), and if one X-bit propagated by X_1 is the only X-bit in the input ports of one of these gates, filling X_1 will decide the logic value of this gate. In contrast, if there are other X-bits in the input ports of this gate, its output may stay to be 'X' after filling X_1. The more gates driven by one X-bit in the test stimuli and the fewer X-bits in other input ports of these gates, the more transitions may be caused by filling this X-bit. The X-bits in v_1 and v_2 of LoS scheme can be also analyzed similarly. Therefore, "*X-propagation*" of one X-bit X_i in v_1 of LoC or v_1/v_2 of LoS at-speed testing can be calculated as:

$$Pro_i = \sum_{G_k \ driven \ by \ X_i} \frac{1}{N_{X-input_k}} \tag{3}$$

where $N_{X-input_k}$ means the number of input ports of G_k driven by X_i. The higher Pro_i is, the more circuit node transitions may be caused by the logic value difference between v_1 and v_2 after filling X_i. Therefore, to reduce capture power with fewer "*X-candidates*", it is better to fill the "*X-candidates*" with higher Pro_i to eliminate the transitions in corresponding scan cells. For example, if X_1 in Fig. 4 is filled with '0', since it is compatible with the test bits in v_1 of LoS scheme and v_2 of LoC scheme, there will be no transition in sc_1. Thus G_2 and G_3 will also not switch in the test cycle; otherwise, if it is filled with '1', G_2 and G_3 will both change from '0' to '1' in LoS scheme (or change from '1' to '0' in LoC scheme).

IV. Proposed X-filling Framework

The proposed *LPHC* X-filling Framework is outlined in Fig. 5: The given test vectors containing X-bits are first encoded by the original test compression scheme to check if its capture transition count violates the safety limit T_{th}. If not, no X-bits need to be filled. Otherwise, "*X-candidates*" are first selected as X-bits in the merged symbols which will cause no compression ratio loss. If there is no X-bits in the merged symbols, X-bits in the codewords will be selected. If there are no "*X-candidates*" in the test cubes, this test vector is denoted as a violated vector[3]. Otherwise, the "*X-candidate*" with the highest "*X-propagation*" will be filled with logic value (*D*) of the corresponding scan cell in the other vector, e.g. if "*X-candidate*" is an X-bit in v_2, it will be filled with the logic value of test bit in the same scan cell in v_1. There are possibilities that filling the "*X-candidate*" with *D* will increase the capture transition count because of logic value differences between v_1 and v_2 in other scan cells. In that case, the opposite value of *D* will also be tried to see which value will produce fewer capture transitions. In our experiments, this situation is rather rare, but for efficiency of the X-filling, it is worth to take this heuristic for better capture power reduction. After filling each "*X-candidate*", the capture transitions count will be checked again to see if it has met the safety limit. If it does, the X-filling is done for this test vector. Since the "*X-candidate*" selection and "*X-propagation*" calculation are independent parts in the proposed framework, it is portable for different kinds of test compression techniques and X-filling strategies.

V. Experimental Results

To validate the effectiveness of the proposed X-filling framework on capture power reduction in different test compression schemes for at-speed testing, four sets of experiments are conducted on a PC with 3GHz CPU and 4GB memory by using large ITC'99 benchmark circuits: capture power reduction for LoC & LoS schemes in fixed-symbol-length & variable-symbol-length test compression. In this paper, we use dictionary-based encoding [18] and VIHC[19] as the fixed-symbol-length and the variable-symbol-length test compression schemes, respectively. The test cubes of the experimental circuit for LoC & LoS at-speed testing are generated by a commercial ATPG tool, and their profiles are outlined in Table. I, in this table, the number of test vectors (N_v), the number of scan cells (N_{sc}), fault coverage (*Cov.%*) and percentage of X-bits (X%) are given for 'LoC' and 'LoS' test vectors. From this table, we can find that the percentage of X-bits increases with the size of the circuit. For the three largest circuits in ITC'99 (b17-b19), the percentages of X-bits in their test cubes are all above 90%.

[3]a violated vector should be discarded and the faults it targets should be covered by new ATPG patterns.

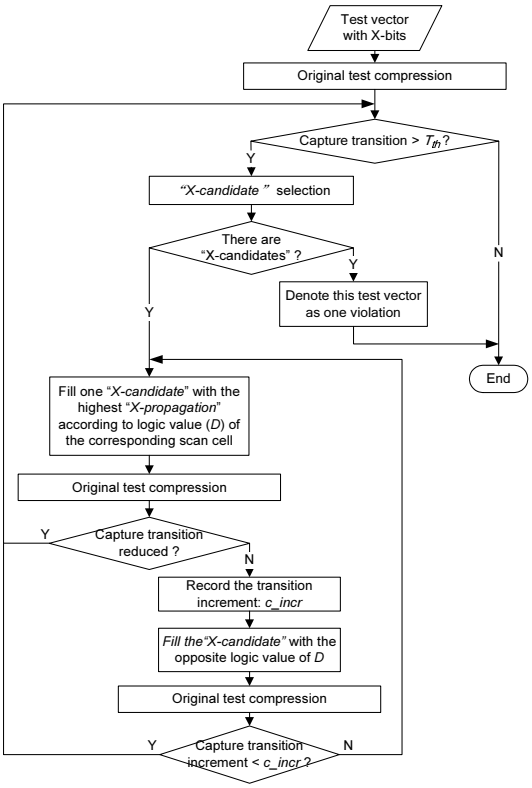

Fig. 5. The proposed LPHC X-filling Framework.

TABLE I
PROFILES OF THE EXPERIMENTAL TEST VECTORS.

circuit	N_{sc}	LoC			LoS		
		N_v	*Cov.%*	X%	N_v	*Cov.%*	X%
b20	544	851	98.74%	68.38%	783	97.80%	66.54%
b21	544	760	98.55%	67.05%	683	97.43%	66.21%
b22	789	844	98.81%	70.01%	811	97.91%	69.31%
b17	1549	2180	99.19%	91.35%	3052	97.59%	92.73%
b18	3027	1875	97.96%	93.09%	2759	93.03%	94.37%
b19	5843	3888	97.71%	95.55%	6336	92.97%	96.86%

A. Capture Power Reduction for LoC at-speed testing

Table II gives the experimental results of LoC test cubes in dictionary-based encoding scheme. The ideal test compression ratio estimated by entropy of the test cubes are given in the column under 'Ideal'. The test compression ratio of original dictionary-based encoding [18] (*Ori.*), *Preferred fill* [8] (*Pref.*), *Adjacent fill* [7] (*Adj.*) and the proposed *LPHC* X-filling framework (*LPHC*) are all given in the columns under "Compression Ratio". The capture transitions of the above approaches are shown in the columns under "Capture Transition Count", and the number in the brackets "(Violations)" represents the number of violated test vectors. The runtime of the proposed *LPHC* fill is shown in seconds in the column under "T(s)".

From Table II we can see that, without X-filling for capture power reduction, the original test compression scheme may produce excessive violated test vectors, which needs to be discarded. Excessive violated vectors require additional ATPG to regenerate more patterns, which could compromise the compression ratio. *Preferred fill* and *Adjacent fill* usually can achieve lower average capture power since all the X-bits in the test cubes are used for test power reduction. However, since they can not identify test vectors violating the safety limit, and the filling decision is arbitrarily made in one pass of filling, the number of violated test vectors of these methods are also very high. In contrast, though the proposed *LPHC* fills only part of the X-bits for capture power reduction, since the X-bits with higher impacts on capture transitions are filled earlier according to the calculation of "*X-propagation*" and the recursive filling procedure facilitates the accuracy of the filling decision, it can significantly reduce the number of violated test vectors.

Moreover, if all the X-bits are filled for capture power reduction as in *Preferred fill* and *Adjacent fill*, there could be much compression ratio loss as shown in Table II. With the proposed *LPHC* fill, the compression ratio is nearly unchanged compared with the original test compression ratio which is very close to the ideal compression ratio. The above results verifies the effectiveness of the selection of "*X-candidates*".

The experimental results of LoC test vectors in VIHC test compression scheme are shown in Table III. From the results we can see that the VIHC test compression can naturally reduce the *average* capture transitions, however, there are still many test vectors with capture transition count violating the safety limit (violations). For these violated vectors, the proposed *LPHC* fill can further reduce the capture transitions in them with low compression ratio loss, while *Preferred fill* and *Adjacent fill* can neither obtain the lowest capture transition count nor higher compression ratio. Therefore, in this set of experiments, the proposed *LPHC* fill can achieve the lowest capture power and violated vector count with the minimum test compression ratio loss compared to other X-filling approaches.

B. Capture Power Reduction for LoS at-speed testing

Similar results for LoS test cubes are given in Table IV and V. These results verifies that though the correlation between the test vector pair $< v_1, v_2 >$ are different in LoC and LoS schemes, the proposed *LPHC* framework can also be efficient for controlling the capture power under the safety limit with low compression ratio loss by identifying "*X-candidates*" and filling them according to their "*X-propagations*".

Though the proposed *LPHC* fill takes both capture power and test compression into account, its computational time is acceptable as shown in all the above four sets of experiments. Therefore, it can be applied to large industrial circuits.

VI. CONCLUSIONS

This paper proposed an X-filling framework that can co-optimize capture power and test compression in both LoC and LoS at-speed testing. "*X-candidates*" that have lower impact on entropy of the test vectors before X-filling are first selected to maintain the maximum test compression ratio. Among the "*X-candidates*", by analyzing the circuit node transitions in the test cycle of LoC/LoS at-speed testing, the "*X-propagation*" metric is proposed to decide the filling order and value of them. The proposed X-filling framework based on the above metrics can efficiently control the capture power under the safety limit while keeping high test compression ratio. Since the proposed framework includes independent "*X-candidates*" selection and "*X-propagation*" calculation, it can be applied to generic test compression techniques and X-filling strategies.

TABLE II
TEST COMPRESSION AND CAPTURE POWER RESULTS OF **LOC** VECTORS ENCODED BY **DICTIONARY**-BASED TEST COMPRESSION.

circuit	Compression Ratio					Capture Transition Count(Violations)				T(s)
	Ideal	*Ori.* [18]	*Pref.* [8]	*Adj.* [7]	*LPHC*	*Ori.* [18]	*Pref.* [8]	*Adj.* [7]	*LPHC*	
b20	97.01%	91.96%	87.59%	89.06%	91.86%	2403(102)	2346(128)	**2340**(141)	2379(**13**)	40.5
b21	96.85%	91.86%	87.86%	88.97%	91.81%	2296(56)	2233(52)	**2194**(69)	2279(**8**)	17.9
b22	97.51%	90.52%	85.93%	87.46%	90.48%	3283(41)	3272(62)	**3148**(69)	3272(**7**)	9.2
b17	99.33%	93.96%	84.22%	89.25%	93.40%	3179(564)	**2774**(412)	2921(662)	3030(**81**)	4253.4
b18	99.28%	92.82%	81.26%	86.78%	92.81%	3631(46)	3465(28)	**3347**(72)	3618(**3**)	634.1
b19	99.63%	92.57%	78.81%	86.16%	92.56%	7408(71)	6269(26)	**5920**(76)	7383(**1**)	7890.2

TABLE III
TEST COMPRESSION AND CAPTURE POWER RESULTS OF **LOC** VECTORS ENCODED BY **VIHC** TEST COMPRESSION.

circuit	Compression Ratio					Capture Transition Count(Violations)				T(s)
	Ideal	*Ori.* [19]	*Pref.* [8]	*Adj.* [7]	*LPHC*	*Ori.* [19]	*Pref.* [8]	*Adj.* [7]	*LPHC*	
b20	63.54%	52.88%	13.20%	21.69%	52.37%	2195(50)	2346(128)	2340(141)	**2185**(**17**)	33.9
b21	60.89%	50.67%	14.67%	20.90%	50.47%	2003(23)	2233(52)	2194(69)	**1999**(**9**)	16.5
b22	63.41%	53.08%	15.77%	20.06%	52.96%	3028(41)	3272(62)	3148(69)	**3025**(**7**)	128.0
b17	85.95%	72.72%	21.63%	32.88%	72.14%	2173(110)	2774(412)	2921(662)	**2170**(**75**)	2355.3
b18	86.33%	72.06%	18.57%	23.14%	72.05%	**1808**(4)	3465(28)	3347(72)	**1808**(**3**)	514.2
b19	90.04%	74.72%	18.67%	23.58%	74.72%	**3270**(2)	6269(26)	5920(76)	**3270**(**1**)	2535.1

TABLE IV
TEST COMPRESSION AND CAPTURE POWER RESULTS OF **LOS** VECTORS ENCODED BY **DICTIONARY**-BASED TEST COMPRESSION.

circuit	Compression Ratio					Capture Transition Count(Violations)				T(s)
	Ideal	*Ori.* [18]	*Pref.* [8]	*Adj.* [7]	*LPHC*	*Ori.* [18]	*Pref.* [8]	*Adj.* [7]	*LPHC*	
b20	96.76%	91.70%	87.60%	88.91%	91.53%	2531(189)	2378(129)	**2062**(47)	2472(**25**)	54.4
b21	96.70%	91.76%	87.86%	89.10%	91.50%	2667(211)	2423(114)	**2122**(57)	2585(**31**)	75.4
b22	97.29%	90.49%	86.01%	87.56%	90.34%	4012(184)	3518(68)	**3159**(37)	3940(**18**)	113.6
b17	99.41%	94.70%	84.49%	90.44%	94.68%	4065(49)	2544(2)	**1719**(**1**)	4047(**1**)	162.1
b18	99.42%	93.38%	81.31%	87.33%	93.38%	7544(22)	4681(**4**)	**2296**(**4**)	7541(**4**)	374.2
b19	99.72%	93.56%	78.90%	87.14%	93.56%	15505(16)	8812(3)	**3334**(3)	15501(**2**)	1553.1

TABLE V
TEST COMPRESSION AND CAPTURE POWER RESULTS OF **LOS** VECTORS ENCODED BY **VIHC** TEST COMPRESSION.

circuit	Compression Ratio					Capture Transition Count(Violations)				T(s)
	Ideal	*Ori.* [19]	*Pref.* [8]	*Adj.* [7]	*LPHC*	*Ori.* [19]	*Pref.* [8]	*Adj.* [7]	*LPHC*	
b20	60.02%	49.98%	13.78%	22.34%	49.44%	**1783**(26)	2378(129)	2062(47)	**1783**(**23**)	19.8
b21	58.43%	48.57%	15.58%	21.67%	47.78%	**1807**(30)	2423(114)	2122(57)	**1807**(**29**)	26.6
b22	62.15%	51.74%	16.64%	21.53%	51.26%	**2764**(22)	3518(68)	3159(37)	**2764**(**21**)	44.3
b17	88.57%	74.67%	24.80%	42.17%	73.64%	**1400**(4)	2544(2)	1719(1)	**1400**(**1**)	179.3
b18	88.63%	74.08%	21.11%	26.59%	73.78%	**1592**(**2**)	4681(4)	2296(4)	**1592**(**2**)	331.3
b19	92.63%	76.56%	21.14%	27.08%	76.44%	**2350**(3)	8812(3)	3334(3)	**2350**(**2**)	1680.5

REFERENCES

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

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

[3] J. Savir and S. Patil. Scan-based transition test. *IEEE Transactions on Computer-Aided Design*, 12(8):1232–1241, August 1993.

[4] G.Xu and A.D.Singh. Scan cell design for launch-on-shift delay tests with slow scan enable. *IET Proceedings, Computers and Digital Techniques*, 1(3):213–219, 2006.

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

[6] J. Saxena, K. M. Butler, V. B. Jayaram, and S. Kundu. A Case Study of IR-Drop in Structured At-Speed Testing. In *Proceedings IEEE International Test Conference (ITC)*, pages 1098–1104, October 2003.

[7] K. M. Butler, J. Saxena, A. Jain, T. Fryars, J. Lewis, and G. Hetherington. Minimizing Power Consumption in Scan Testing: Pattern Generation and DFT Techniques. In *Proceedings IEEE International Test Conference (ITC)*, pages 355–364, October 2004.

[8] S. Remersaro, X Lin, Z Zhang, S.M. Reddy, I. Pomeranz, and J. Rajski. Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs. In *Proceedings IEEE International Test Conference (ITC)*, 2006.

[9] X. Wen, K. Miyase, S. Kajihara, T. Suzuki, Y. Yamato, P. Girard, Y. Ohsumi, and L. T. Wang. A Novel Scheme to Reduce Power Supply Noise for High-Quality At-Speed Scan Testing. In *Proceedings ACM/IEEE Design Automation Conference (DAC)*, page 25.1, 2007.

[10] 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 *Proceedings Design, Automation, and Test in Europe (DATE)*, pages 1184–1189, 2008.

[11] J. Li, X. Liu, Y. Chen, Y. Hu, X. Li, and Q. Xu. On Capture Power-Aware Test Data Compression for Scan-Based Testing. In *Proceedings International Conference on Computer-Aided Design (ICCAD)*, pages 67–72, 2008.

[12] M.-F.Wu, J.-L.Huang, X.Wen, and K.Miyase. Reducing Power Supply Noise in Linear-Decompressor-Based Test Data Compression Environment for At-Speed Scan Testing. In *Proceedings IEEE International Test Conference (ITC)*, paper 13.1, 2008.

[13] X. Liu and Q. Xu. A Generic Framework for Scan Capture Power Reduction in Fixed-Length Symbol-based Test Compression Environment. In *Proceedings Design, Automation, and Test in Europe (DATE)*, pages 1494–1499, 2009.

[14] X. Liu and Q. Xu. On Simultaneous Shift- and Capture-Power Reduction in Linear Decompressor-Based Test Compression Environment. In *Proceedings IEEE International Test Conference (ITC)*, paper 9.3, 2009.

[15] K. J. Balakrishnan and N. A. Touba. Relationship Between Entropy and Test Data Compression. *IEEE Transactions on Computer-Aided Design*, 26(2):386–395, Feb 2007.

[16] A. Chandra and K. Chakrabarty. How effective are compression codes for reducing test data volume. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 91–96, 2002.

[17] N. Badereddine P. Girard, S. Pravossoudovitch, C.Landrault, A.Virazel, and H.-J.Wunderlich. Structural-Based Power-Aware Assignment of Don't Cares for Peak Power Reduction during Scan Testing. In *Proceedings IFIP International Conference on Very Large Scale Integration (VLSI-SOC)*, pages 403–408, 2006.

[18] A. Wurtenberger, C. S. Tautermann, and S. Hellebrand. Data Compression for Multiple Scan Chains using Dictionaries with Corrections. In *Proceedings IEEE International Test Conference (ITC)*, pages 926–935, October 2004.

[19] 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.