PAPER

A New Method for Low-Capture-Power Test Generation for Scan Testing

Xiaoqing WEN[†a)], ***Member***, **Yoshiyuki YAMASHITA**[††], ***Nonmember***, **Seiji KAJIHARA**[†], ***Member***, **Laung-Terng WANG**[†††], **Kewal K. SALUJA**[††††], ***Nonmembers***, ***and*** **Kozo KINOSHITA**[†††††], ***Fellow***

SUMMARY Research on low-power scan testing has been focused on the shift mode, with little consideration given to the capture mode power. However, high switching activity when capturing a test response can cause excessive IR-drop, resulting in significant yield loss due to faulty test results. This paper addresses this problem with a novel low-capture-power X-filling method by assigning 0's and 1's to unspecified bits (X-bits) in a test cube to reduce the switching activity in capture mode. This method can be easily incorporated into any test generation flow, where test cubes can be obtained during ATPG or by X-bit identification. Experimental results show the effectiveness of this method in reducing capture power dissipation without any impact on area, timing, and fault coverage.

key words: *scan testing, capture power, X-bit, IR-drop*

1. Introduction

Integrated circuit testing based on the full-scan methodology and automatic test pattern generation (ATPG) is the most widely adopted test strategy, and it is well supported by test engineers, tool vendors, and tester makers. This situation will remain in the foreseeable future.

In a full-scan sequential circuit, scan flip-flops (FFs) replace all functional FFs and operate in two modes: *shift* and *capture*. In shift mode, scan FFs are connected as shift registers or scan chains directly accessible from a tester. This mode is used to load a test vector through shift-in or observe a test response through shift-out, for the combinational portion of the sequential circuit. In capture mode, scan FFs operate as functional FFs and load the test response of the combinational portion to a test vector into themselves preparatory to shift-out later in shift mode. As a result, testing a full-scan sequential circuit is reduced to testing its combinational portion, in that now it is only necessary to generate test vectors for the combinational portion with a combinational ATPG program [1].

Despite its usefulness, the applicability of scan testing is increasingly being challenged due to the following three problems: *test data volume*, *test application time*, and *test power dissipation*. The first two problems are caused by larger gate and FF counts, longer scan chains, and the use of complex fault models, all inevitable in the deep submicron (DSM) era. Several approaches, such as built-in self-test (BIST), test compaction, multi-capture clocking, and decompression-compression, have been proposed to address the problems of test data volume and test application time. In this paper, we focus on the test power dissipation problem.

The power dissipation of a CMOS circuit consists of *static* dissipation due to leakage current and *dynamic* dissipation due to switching activity, with the latter being dominant. Dynamic power dissipation in full-scan testing occurs in both shift mode and capture mode. In shift mode, a test vector is shifted into all scan chains of a full-scan circuit, bit by bit. This results in *shift power dissipation*. In capture mode, the test response of the combinational portion of the full-scan circuit to a test vector is loaded into all FFs, replacing the test vector that the FFs currently contain. This results in capture power dissipation, whenever the test vector and its corresponding test response have opposite logic values at some FFs.

Generally, test power dissipation, consisting of both shift and capture power dissipation, can be 2 to 3 times higher than functional power dissipation [2]. This is especially the case for high-speed and high-density DSM integrated circuits with the system-on-a-chip (SoC) scheme [3]. Excessive test power dissipation may permanently damage a circuit under test, reduce its reliability due to accelerated electromigration, or result in yield loss due to faulty test results caused by IR-drop [4], [23]. Circuit damage and reliability degradation are mostly caused by excessive heat due to shift power dissipation, while significant yield loss can also be caused by excessive capture power dissipation.

Previous techniques for test power reduction have focused on reducing shift power dissipation during test application, based on four major approaches: *scheduling*, *test vector manipulation*, *circuit modification*, and *scan chain modification*. Test scheduling [2], [5] takes the power budget into consideration when selecting modules to be tested simultaneously. Test vector manipulation includes power-aware ATPG [6], [7], static compaction [8], test vector mod-

Manuscript received May 26, 2005.
Manuscript revised September 14, 2005.

[†] The authors are with the Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka-shi, 820–8502 Japan.

[††] The author is with Densotechno Co., Nagoya-shi, 450–0002 Japan.

[†††] The author is with SynTest Technologies, Inc., 505 S. Pastoria, Suite 101, Sunnyvale, CA 94086, U.S.A.

[††††] The author is with the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI 53706, U.S.A.

[††††] The author is with the Faculty of Informatics, Osaka Gakuin University, Suita-shi, 564–8511 Japan.

a) E-mail: wen@cse.kyutech.ac.jp

DOI: 10.1093/ietisy/e89–d.5.1679

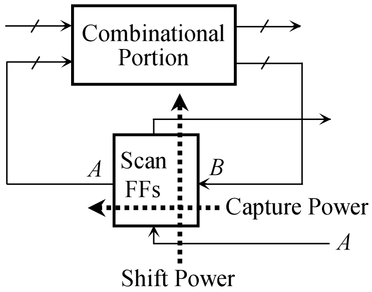


Fig. 1 Two types of test power dissipation.

ification [9], test vector reordering [10], test vector compression [11], and coding [12]. Circuit modification includes transition blocking [13] and clock gating [14]. Scan chain modification includes scan chain reordering [11], [15], scan chain partitioning [16], and scan chain modification [17]. Techniques for BIST applications, such as toggle suppression [18] and low-power test pattern generation [19], have also been proposed.

In addition to shift power dissipation, capture power dissipation is also part of test power dissipation in scan testing, as illustrated in Fig. 1. In shift mode, test vector *A* is shifted through all scan FFs in the scan chain with several hundreds to several thousands of clock cycles, depending on the scan chain length. In capture mode, test response *B* is loaded into scan FFs to replace the current test vector *A*. This is done in one or multiple clock cycles, depending on whether multi-capture clocking is used.

Although capture power dissipation has less impact on the total heat dissipation than shift power dissipation, it may nonetheless cause significant yield loss [4], [23]. This is because high switching activity in capture mode at the scan FFs due to the difference between *A* and *B* may result in instantaneously excessive IR-drop, causing a faulty test response $B' \neq B$ to be loaded into the scan FFs. This results in yield loss, even though the excessive capture power dissipation does not cause too much heat dissipation.

For example, in one recently reported case, a 3M-gate industrial circuit passed all functional tests and all scan chain flush tests but showed un-repeatable behaviors only in capture mode during scan testing. Detailed analysis revealed that the circuit had multiple functional clocks, each driving a portion of the circuit; but only one test clock was used to drive all FFs in the circuit during scan testing. IR-drop caused by high switching activity due to many FFs operating simultaneously was the reason for the yield loss.

The above explanation and example suggest that it is not sufficient to reduce only shift power dissipation. Capture power dissipation should also be reduced, especially in order to avoid yield loss caused by faulty test results. The ultimate solution is to reduce the number of FFs that can operate simultaneously. For a single-clock circuit, this can be achieved by selective clock gating. However, its impact on physical design is high. For a multiple-clock circuit, this can be achieved by either the one-hot or the multi-capture clocking scheme. However, the former suffers from large test data volume and the latter suffers from complicated ATPG with high memory consumption as well as the need of controlling multiple test clocks. The method [24] uses an interleaving scheme to reduce the number of FFs that are clocked simultaneously in capture mode, at the cost of increased control complexity. The method [25] uses an *X*-filling technique to reduce the number of capture transitions at FFs. The technique, however, only works for static compaction. In addition, *X*-filling is conducted in a simple hill-climbing way without taking into consideration various bit-pair relations between a test cube and its test response.

These disadvantages motivated us to propose a new solution for reducing capture power dissipation, which should be simple, effective, and of no impact on physical and test design flows. In addition, this new solution should be able to work in both dynamic and static compaction.

We notice the fact that many *test cubes*, i.e., test vectors with unspecified bits (*X*-bits), are usually generated either during ATPG [1] or obtained by *X*-bit identification [20] from a set of fully-specified test vectors. In this paper, we propose a *low-capture-power* (*LCP*) *X*-filling method for assigning 0's and 1's to the *X*-bits in a test cube so that the number of transitions at the outputs of scan FFs in capture mode for the resulting fully-specified test vector is reduced. Test vectors obtained by the LCP *X*-filling method have low capture power dissipation, resulting in reduced yield loss caused by faulty capture operations. As a totally software-based solution, the LCP *X*-filling method has no physical design impact and can be easily incorporated into any test generation flow.

The rest of the paper is organized as follows. Section 2 describes the research background. Section 3 presents the LCP *X*-filling method. Section 4 shows experimental results, and Sect. 5 concludes the paper.

2. Background

2.1 Test Cube Processing

A general ATPG procedure repeats the operations of selecting an undetected fault and finding necessary input logic assignments to detect the fault. The result is usually a *test cube* with *X*-bits. This test cube can be processed either immediately after its generation in *dynamic compaction* [1] or together with other test cubes in a post-ATPG operation in *static compaction*, for the purpose of reducing the test set size or test power dissipation [21], [25].

Note that test cubes processed in static compaction can also be obtained by *X*-bit identification [20] from a set of fully-specified test vectors. It has been shown that a significant percentage of bits, as high as 90% in some cases, in a fully-specified test vector set can be turned into *X*-bits without affecting its fault coverage.

The key operation in processing a test cube in dynamic or static compaction is to properly determine 0's and 1's for its *X*-bits. This operation is called *X*-filling. Obviously, different X-filling methods have different impact on test data volume, test application time, and test power.

2.2 Previous X-Filling Methods

Generally, there are three approaches to X-filling: *random*, *algorithmic*, and *merge-based*. Random X-filling assigns 0's and 1's randomly to X-bits in a test cube. Algorithmic X-filling determines logic values for the X-bits in a test cube in a more sophisticated way in order to better achieve a specific goal. Merge-based X-filling determines the logic value for an X-bit in a test cube depending on the logic value of the corresponding bit in another test cube to be merged with. For example, merging test cube t_1 $1X0$ with test cube t_2 $11X$ will cause assigning 1 to the X-bit in t_1 and 0 to the X-bit in t_2, resulting in one test vector 110.

Algorithmic X-filling is often used in dynamic compaction for reducing the number of final test vectors [1]. The key issue is how to select a secondary target fault which has higher chances of being detected with the X-bits in a test cube. Selection methods based on fault simulation by critical path tracing, independent faults, etc. have been shown to be effective. Algorithmic X-filling is also used for reducing shift power dissipation by properly re-assigning 0's and 1's to the X-bits found by X-bit identification [9].

Merge-based X-filling is often used in static compaction for reducing the number of test vectors [1], [22] as well as for shift power reduction by carefully selecting the order of test cubes to be merged by using a cost function reflecting shift transition activity [8].

Random X-filling is conducted for remaining X-bits after algorithmic or merge-based X-filling is done. Its purpose is to reduce the number of test vectors since randomly assigning 0' and 1's to the X-bits in a test cube often increases the chances of detecting additional faults [1]. However, random X-filling usually adversely affects test power dissipation [8].

2.3 Motivation

Previous X-filling methods are largely used for reducing the number of test vectors [1], [21], [22], and there are a few X-filling methods available for shift power reduction [8], [9]. There is one X-filling method [25] for capture power reduction but it only works for static compaction. Moreover, this method relies on a simple bit-stripping technique to identify X-bits in a test vector, and X-filling is conducted in a simple hill-climbing way without taking into consideration various bit-pair relations between a test cube and its test response. All these factors limit its generality and effectiveness.

To solve this problem, we propose a novel algorithmic X-filling method, called the *LCP* (*Low-Capture-Power*) *X-filling method*, for determining proper logic values for X-bits in a test cube to reduce capture power dissipation. Test cubes can be generated during ATPG or identified by X-identification, making this method applicable in both dynamic and static compaction. In addition, more effective X-identification [20] is used to turn more test vectors into test cubes for more effective static compaction. Moreover, bit-pair relations between a test cube and its test response are fully analyzed and both assignment and justification operations are conducted in order to achieve greater capture power reduction. The details of this method are described in the following section.

3. LCP X-Filling

3.1 Problem Formalization

A general full-scan circuit is shown in Fig. 2. It consists of a combinational portion with m_1 primary inputs (PIs) and m_2 primary outputs (POs) as well as n scan FFs. The outputs of the scan FFs that feed the combinational portion are pseudo primary inputs (PPIs) and the functional inputs from the combinational portion to the scan FFs are pseudo primary outputs (PPOs). Note that the number of PPIs is the same as that of PPOs, while the number of PIs may or may not be the same as that of POs. Also note that, for the convenience of presentation, all scan FFs are assumed to form one scan chain with SI as the scan input and SO as the scan output. The X-filling method to be presented in the following, however, can be readily extended for a full-scan circuit with multiple scan chains.

In Fig. 2, v is a test cube with at least one X-bit. The PI and PPI bits in v are denoted by an m_1-bit vector <v: PI> and an n-bit vector <v: PPI>, respectively. The combinational portion is assumed to have logic function f, and its functional response to v is $f(v)$. The PO and PPO bits in $f(v)$ are denoted by an m_2-bit vector <$f(v)$: PO> and an n-bit vector <$f(v)$: PPO>, respectively.

If <v: PPI> and <$f(v)$: PPO> are fully-specified, the result of their bit-wise exclusive-OR operation is an n-bit vector, denoted by <v: PPI> $\oplus$ <$f(v)$: PPO>. Obviously, if the corresponding bits in <v: PPI> and <$f(v)$: PPO> are different as shown in Fig. 3, a transition, called *capture transition* in this paper, will occur at the output of the scan FF in capture mode. Obviously, the number of 1's in <v: PPI> $\oplus$ <$f(v)$: PPO>, denoted by |<v: PPI> $\oplus$ <$f(v)$: PPO>|, is the total number of capture transitions for v.

Since the number of capture transitions is closely correlated with the circuit switching activity as demonstrated in [8], the LCP X-filling problem can be formalized as fol-

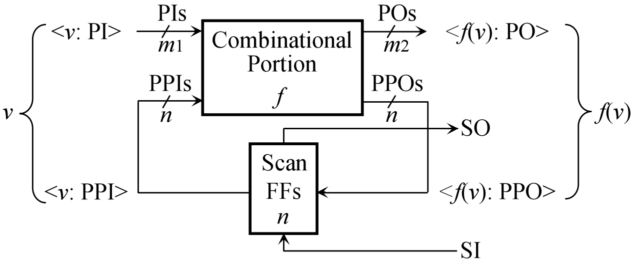


Fig. 2 A general full-scan circuit.

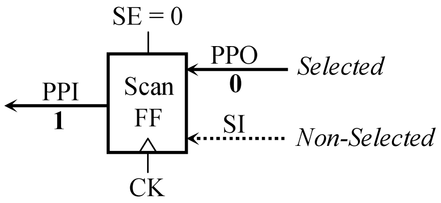


Fig. 3 Capture transition at a scan FF.

Table 1 *X*-cases.

<*v*: PPI>	<*f*(*v*): PPO> without *X*	<*f*(*v*): PPO> with *X*
without *X*	Case-1	Case-3
with *X*	Case-2	Case-4

lows:

LCP X-Filling Problem: Given a test cube *v* for a full-scan circuit with combinational logic function *f*, assign 0's and 1's to the *X*-bits in *v* such that |<*v*: PPI> ⊕ <*f*(*v*): PPO>| is minimized.

3.2 *X*-Filling Algorithm

Suppose that *v* is a test cube with at least one *X*-bit and *f*(*v*) is the simulated response of the combinational portion of a full-scan circuit to *v*. Note that *f*(*v*) may also have *X*-bits due to the *X*-bits in *v*. Depending on how *X*-bits appear in <*v*: PPI> and <*f*(*v*): PPO>, we define four *X*-cases as shown in Table 1:

The algorithm for LCP *X*-filling in each *X*-case is presented next.

3.2.1 Case-1

In Case-1, since <*v*: PPI> and <*f*(*v*): PPO> have no *X*-bits, |<*v*: PPI> ⊕ <*f*(*v*): PPO>|, the total number of capture transitions, is determined and will not change no matter what the logic values are assigned to the *X* bits in <*v*: PI>.

Since *v* is a test cube with at least one *X*-bit and <*v*: PPI> has no *X*-bits, <*v*: PI> must have at least one *X*-bit. Therefore, *X*-filling in Case-1 can be targeted for any other purpose, such as reducing the number of test vectors or reducing shift power dissipation, with *X*-filling methods mentioned in 2.2.

3.2.2 Case-2

In Case-2, since <*v*: PPI> has at least one *X*-bit, *X*-filling is first conducted for <*v*: PPI> to reduce the number of capture transitions. This is achieved by replacing all *X*-bits in <*v*: PPI> with the same logic values at the corresponding bits in <*f*(*v*): PPO>. Note that <*f*(*v*): PPO> has no *X*-bit. After this assignment is done, Case-2 reduces to Case-1 since <*v*: PPI> no longer has any *X*-bit. Then Case-1 *X*-filling can be conducted for all the remaining *X*-bits in <*v*: PI> as described in 3.2.1.

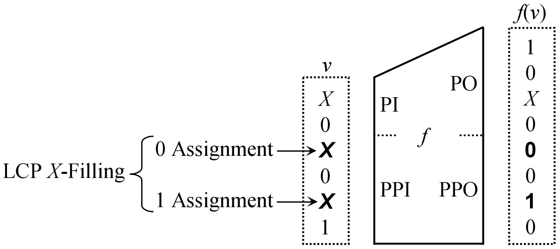


Fig. 4 Assignment-based *X*-filling.

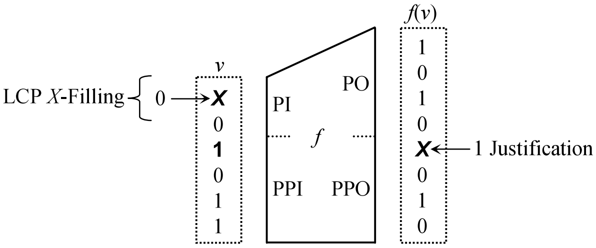


Fig. 5 Justification-based *X*-filling.

An example is shown in Fig. 4, where <*v*: PPI> = <*X*0*X*1> and <*f*(*v*): PPO> = <0010>. First, 0 and 1 are assigned to the first *X*-bit and the second *X*-bit, respectively, in <*v*: PPI>, and then logic simulation is conducted. If only one *X*-bit remains in <*v*: PI>, Case-1 *X*-filling is conducted.

3.2.3 Case-3

In Case-3, <*v*: PI> has at least one *X*-bit since <*v*: PPI> has no *X*-bit. In addition, <*f*(*v*): PPO> has at least one *X*-bit. *X*-filling for the *X*-bits in <*v*: PI> is conducted in such a way that as many *X*-bits as possible in <*f*(*v*): PPO> are made to have the same logic values as the corresponding bits in <*v*: PPI>, in order to reduce the number of capture transitions.

Whether an *X*-bit *a* in <*f*(*v*): PPO> can have the same value as its corresponding bit *b* in <*v*: PPI> is determined by justification. For example, if *b* is 1, then one can try to justify 1 on *a*. If successful, 1 is placed on *a*; otherwise, 0 is placed on *a*. Note that, during justification, the logic values for some *X*-bits in <*v*: PI> will be determined.

An example is shown in Fig. 5, where <*v*: PPI> = <1011> and <*f*(*v*): PPO> = <*X*010>. Obviously, placing 1 to the *X*-bit in <*f*(*v*): PPO> reduces the number of capture transitions. Thus, justification of 1 on the *X*-bit in <*f*(*v*): PPO> is conducted. Suppose that this is successful if 0 is assigned to the *X*-bit in <*v*: PI>. As a result, a fully-specified test vector *v* = <001011> is obtained and its simulated response is *f*(*v*) = <10101010>.

It is possible that <*f*(*v*): PPO> has multiple *X*-bits. In this case, the order of the *X*-bits being justified affects the success ratio of justification, hence the capture transition reduction effect. We propose the following criterion for selecting an *X*-bit, based on the easiness of justification:

Criterion-1: Suppose that a_1 and a_2 are two *X*-bits in <*f*(*v*):

Table 2 Bit-pair types.

	a in <*v*: PPI>	*b* in <*f*(*v*): PPO>
Type-A	0 or 1	0 or 1
Type-B	*X*	0 or 1
Type-C	0 or 1	*X*
Type-D	*X*	*X*

PPO>. We obtain the sets of X-bits in <v: PI>, denoted by $X(a_1)$ and $X(a_2)$, that can be reached from a_1 and a_2, respectively. If $|X(a_1)| > |X(a_2)|$, a_1 is selected for justification since more X-bits are available for justifying a logic value on a_1. If $|X(a_1)| = |X(a_2)|$, we further obtain the average levels of all PIs with X-bits in $X(a_1)$ and $X(a_2)$, denoted by $L(a_1)$ and $L(a_2)$, respectively. Note that levels are assigned to all lines in a circuit from POs and PPOs. If $L(a_1) < L(a_2)$, a_1 is selected for justification since the PIs with X-bits in $X(a_1)$ are closer to the justification target of a_1, increasing the success possibility of the justification.

Once all X-bits in <$f(v)$: PPO> are determined, Case-3 becomes Case-1. Then, Case-1 X-filling can be conducted for all the remaining X-bits in <v: PI>.

3.2.4 Case-4

In Case-4, both <v: PPI> and <$f(v)$: PPO> have X-bits. For a bit-pair <a, b> consisting of a bit a in <v: PPI> and its corresponding bit b in <$f(v)$: PPO>, there are four possible bit-pair types as summarized in Table 2.

Obviously, there is no need to consider any Type-A bit-pair. For other bit-pairs with at least one X, we process Type-B and Type-C bit-pairs first. Only when there are no more such bit-pairs, we then process Type-D bit-pairs.

If both Type-B and Type-C bit-pairs exist, it is necessary to determine which type of bit-pairs to process first. Note that an X-bit in <v: PPI> for a Type-B bit-pair indicates that a capture transition can be avoided if a proper logic value is assigned to the X-bit. Also note that an X-bit in <$f(v)$: PPO> for a Type-C bit-pair indicates that a proper logic value may be successfully justified on the X-bit so that a capture transition can be avoided. Therefore, we propose the following selection criterion for selecting a proper target bit-pair:

Criterion-2: We compare the number of X-bits in <v: PPI> for all Type-B bit-pairs and the number of X-bits in <$f(v)$: PPO> for all Type-C bit-pairs. If the former is larger than the latter, all Type-B bit-pairs are processed first by using the assignment technique described in 3.2.2. If the latter is larger than the former, all Type-C bit-pairs are processed first by using the justification technique described in 3.2.3.

After X-filling for all Type-B and Type-C bit-pairs are conducted, it is possible that Type-D bit-pairs still remain. Suppose that <a, b> is such a bit-pair, where a in <v: PPI> and b in <$f(v)$: PPO> both have X. In this case, we first check if 0 (1) can be assigned to both a and b in order to avoid a capture transition. This can be conducted by placing 0 (1) on a and trying to justify 0 (1) on b. If this is successful, we use 0 (1) for both a and b; otherwise, we must use different values for a and b.

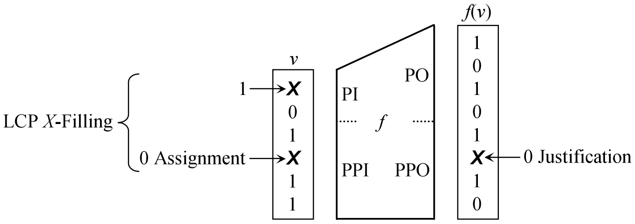


Fig. 6 Assignment-justification-based X-filling.

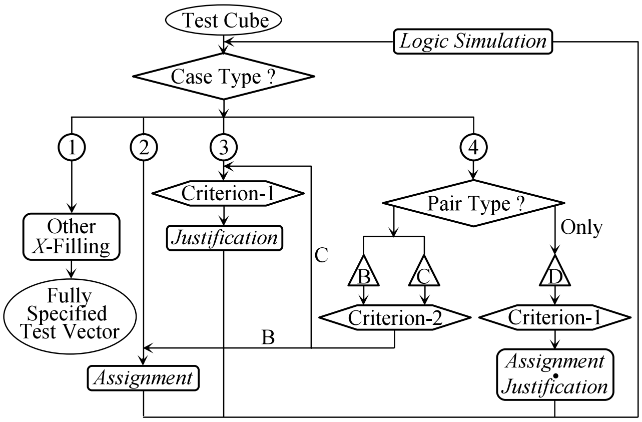


Fig. 7 LCP X-filling procedure.

It is possible that there are multiple Type-D bit-pairs in Case-4. In this case, we take the multiple X-bits in <$f(v)$: PPO> into consideration and use the Criterion-1 proposed for Case-3 X-filling to determine the order of processing Type-D bit-pairs.

An example for Type-D is shown in Fig. 6, where <v: PPI> = <$1X11$> and <$f(v)$: PPO> = <$1X10$>. In this case, we try placing 0 on the X-bit in <v: PPI> and justifying 0 on the X-bit in <$f(v)$: PPO>. Suppose that this is successful if 1 is assigned to the X-bit in <v: PI>. As a result, a fully-specified test vector v = <101011> is obtained and its simulated response is $f(v)$ = <10101010>. Note that one capture transition is successfully avoided in this case.

After Type-B, Type-C, and Type-D bit-pairs are processed, Case-4 becomes Case-1 since both <v: PPI> and <$f(v)$: PPO> no longer have any X-bit. Therefore, Case-1 X-filling can be conducted for all the remaining X-bits in <v: PI>.

3.3 X-Filling Procedure

The general procedure for LCP X-filling is illustrated in Fig. 7. A test cube v is processed based on its case type. For a Case-4 test cube, its bit-pairs for <v: PPI> and <$f(v)$: PPO> will be further checked. If Type-B or Type-C bit-pairs exist, they should be processed as in Case-2 or Case-3. If there are only Type-D bit-pairs, assignment-justification will be conducted for *X*-filling. The final result of this procedure is a fully-specified test vector.

4. Experimental Results

X-filling experiments were conducted on ISCAS'89 circuits [26]. Since the major process was logic simulation, the total run time for these circuits was quite insignificant and thus omitted in this paper.

4.1 Dynamic *X*-Filling Results

Table 3 shows the results obtained by random *X*-filling and LCP *X*-filling for test cubes generated in ATPG. In ATPG, a test cube was generated for a primary fault. Then, the *X*-bits in the test cube were used to detect a secondary fault. This process was repeated until the number of detected secondary faults reached a threshold, denoted by *Limit*. Then, the remaining *X*-bits in the test cube were filled randomly or with the LCP method. In Table 3, the number of test vectors, the average number of capture transitions per test vector, and the maximum number of capture transitions for each case in capture mode are shown under "*# of Vec.*", "*Ave. Trans.*", and "*Max. Trans.*", respectively.

Table 3 Results for dynamic *X*-filling.

Circuit	*Fault Cov.* (%)	Random			LCP					
		Limit = ∞			*Limit* = ∞			*Limit* = 100		
		# of Vec.	*Ave. Trans.*	*Max. Trans.*	*# of Vec.*	*Ave. Trans.*	*Max. Trans.*	*# of Vec.*	*Ave. Trans.*	*Max. Trans.*
s1196	100	130	8.9	14	126	0.8	7	129	0.8	6
s1238	94.91	141	8.7	14	139	0.8	6	146	0.8	7
s1423	99.08	36	22.7	44	37	17.4	32	45	16.8	38
s5378	99.13	113	90.2	106	112	41.8	94	115	40.1	66
s9234	93.48	138	80.8	106	141	33.6	99	145	32.0	76
s13207	98.46	262	244.6	289	263	95.3	223	262	94.1	184
s15850	96.68	132	179.6	261	124	57.1	178	119	57.1	111
s35932	89.91	18	825.6	1063	18	483.0	1063	38	254.2	493
s38417	99.47	102	495.6	840	102	186.3	803	118	170.4	281
s38584	95.85	124	429.4	892	124	214.3	745	135	205.2	302

Table 3 shows that on average, LCP *X*-filling (*Limit* = ∞) achieved 60.0% reduction for the average number of capture transitions and 22.8% reduction for the maximum number of capture transitions, compared with the results of random *X*-filling.

Note that the smaller the value of *Limit*, the more remaining *X*-bits in a test cube, thus the higher capture transition reduction effect achieved by LCP *X*-filling. However, the smaller the value of *Limit*, the larger the number of test vectors. These contradicting trends were verified by experimenting with three largest ISCAS'89 benchmark circuits, as shown in Fig. 8. It is clear that a "good" value exists for *Limit*, which balances the capture transition reduction effect and the number of test vectors. In the case of Fig. 8, for example, 100 is obviously such a value for *Limit*.

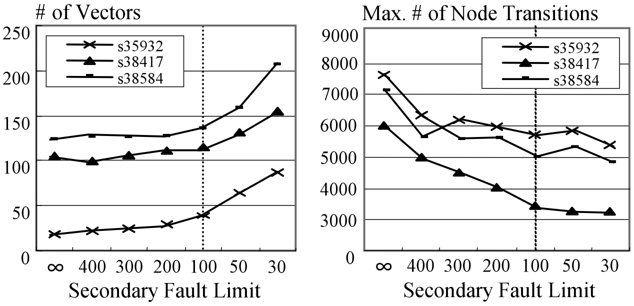


Fig. 8 Impact of secondary fault limit.

The experimental results for LCP *X*-filling (*Limit* = 100) are also shown in Table 3. It can be seen that on average, LCP *X*-filling (*Limit* = 100) can achieve 7.5% more reduction in the average number of capture transitions and 104.8% more reduction in the maximum number of capture transition,, compared with LCP *X*-filling (*Limit* = ∞), at the cost of 16.9% more test vectors.

4.2 Static *X*-Filling Results

Table 4 Results for static *X*-filling.

Circuit	*# of Vec.*	*Fault Cov.* (%)	*X* (%)	Random		LCP	
				Ave. Trans.	*Max. Trans.*	*Ave. Trans.*	*Max. Trans.*
s1196	113	100	55.06	8.70	14	1.62	10
s1238	125	94.91	54.98	9.24	14	1.78	9
s1423	24	99.08	41.11	26.5	43	20.63	34
s5378	100	99.13	71.03	90.6	108	40.82	91
s9234	111	93.48	67.17	80.24	110	34.36	61
s13207	235	98.46	91.61	245.6	333	74.27	244
s15850	97	96.68	76.14	181.0	252	66.78	173
s35932	12	89.91	34.35	817.3	1533	569.0	1517
s38417	87	99.47	73.40	491.2	592	177.1	323
s38584	114	95.85	79.65	424.3	785	193.9	437

Table 4 shows the results obtained by random *X*-filling and LCP *X*-filling for test cubes obtained by an *X*-bit identification procedure [20] from a set of fully-specified test vectors. As shown in Table 4, even with compacted test vectors, an average of 64.5% of all bits in a set of fully-specified test vectors could be turned into *X*-bits without affecting its fault coverage. These *X*-bits were then filled randomly or with the LCP method. In Table 4, "*X* (%)" shows the percentage of *X*-bits identified from a set of fully-specified test vectors, while the meanings of all other items are the same as in Table 3.

Table 4 shows that on average, LCP *X*-filling achieved 57.8% reduction for the average number of capture transitions and 29.4% reduction for the maximum number of capture transitions, compared with random *X*-filling.

4.3 Observations

Generally, test power can be two to three times higher than functional power. Obviously, it is not enough to use only the approach of test power reduction based on test vector manipulation. The conventional approach is to strengthen the power grid. However, this is very costly, and sometimes causes performance degradation. Therefore, the best solution is to use both approaches together to achieve a good result at acceptable costs of power grid enhancement. That is, one first reduces test power as much as possible by test vector manipulation, and only when this is not enough, one then conducts power-grid enhancement.

Another issue is that the risk of high switching activity exists in both shift and capture modes. As a complete solution to this problem, one can use multiple shift clock phases [4] to reduce shift power and test vector manipulation to reduce capture power. The reason is that the result of the shift operation is independent of shift clock phases, while that of the capture operation depends on capture clock phases. A single capture clock phase is often used to lower ATPG complexity and memory usage. That is, it is better to use X-bits for capture power reduction while using multiple shift clock phases in shift power reduction.

5. Conclusions

This paper addressed a critical test power reduction problem, i.e. reducing capture power dissipation to avoid yield loss caused by faulty test responses in capture mode. A novel low-capture-power X-filling method, called *LCP X-filling*, was proposed for assigning 0's and 1's to unspecified bits in a test cube in order to reduce the switching activity at FFs and in the circuit for the resulting fully-specified test vector. This method can be applied into any ATPG system in either dynamic compaction or static compaction. Experimental results showed the effectiveness of this method in reducing capture power dissipation without any impact on area, timing, and fault coverage.

More evaluations are under way to assess the effect of the LCP X-filling method directly through power consumption instead of capture switching activity. Research on algorithmically setting *limit*, investigating the impact of the order of conducting individual steps in the LCP X-filling procedure, and finding the lower bound on the capture power of a circuit are also being planned.

References

[1] M. Abramovici, M. Breuer, and A. Friedman, Digital Systems Testing and Testable Design, Computer Science Press, 1990.

[2] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices," Proc. VLSI Test Symp., pp.4–9, 1993.

[3] P. Girad, "Survey of low-power testing of VLSI circuits," IEEE Des. Test Comput., vol.19, no.3, pp.82–92, 2002.

[4] T. Yoshida and M. Watari, "A new approach for low power scan testing," Proc. Intl. Test Conf., pp.480–487, 2003.

[5] R. Chou, K. Saluja, and V. Agrawal, "Scheduling tests for VLSI systems under power constraints," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.5, no.6, pp.175–185, 1997.

[6] S. Wang and S. Gupta, "ATPG for heat dissipation minimization during test application," IEEE Trans. Comput., vol.47, no.2, pp.256–262, 1998.

[7] F. Corno, P. Prinetto, M. Redaudengo, and M. Reorda, "A test pattern generation methodology for low power consumption," Proc. VLSI Test Symp., pp.453–459, 1998.

[8] R. Sankaralingam, R. Oruganti, and N. Touba, "Static compaction techniques to control scan vector power dissipation," Proc. VLSI Test Symp., pp.35–40, 2000.

[9] S. Kajihara, K. Ishida, and K. Miyase, "Test vector modification for power reduction during scan testing," Proc. VLSI Test Symp., pp.160–165, 2002.

[10] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, "Techniques for minimizing power dissipation in scan and combinational circuits during test application," IEEE Trans. Comput.-Aided Des. Integr. Ciruits Syst., vol.17, no.12, pp.1325–1333, 1998.

[11] A. Chandra and K. Chakrabarty, "Combining low power scan testing and test data compression for system-on-a-chip," Proc. Design Automation Conf., pp.166–169, 2001.

[12] A. Chandra and K. Chakrabarty, "Reduction of SoC test data volume, scan power and testing time using alternating run-length codes," Proc. Intl. Conf. on Computer Aided Design, pp.673–678, 2002.

[13] A. Hertwig and H. Wunderlich, "Low power serial built-in self-test," Proc. European Test Workshop, pp.49–53, 1998.

[14] R. Sankaralingam, R. Oruganti, and N. Touba, "Reducing power dissipation during test using scan chain disable," Proc. VLSI Test Symp., pp.319–324, 2001.

[15] Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossoudovitch, "Power driven chaining of flip-flops in scan architectures," Proc. Intl. Test Conf., pp.796–803, 2002.

[16] J. Saxena, K. Butler, and L. Whetsel, "A scheme to reduce power consumption during scan testing," Proc. Intl. Test Conf., pp.670–677, 2001.

[17] O. Sinanoglu and A. Orailoglu, "Scan power minimization through stimulus and response transformations," Proc. Design, Automation and Test in Europe, pp.404–409, 2004.

[18] S. Gerstendoerfer and H. Wunderlich, "Minimized power consumption for scan-based BIST," Proc. Intl. Test Conf., pp.77–84, 1999.

[19] S. Wang, "Generation of low-power-dissipation and high-fault coverage patterns for scan-based BIST," Proc. Intl. Test Conf., pp.834–843, 2002.

[20] K. Miyase and S. Kajihara, "XID: Don't care identification of test patterns for combinational circuits," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.23, no.2, pp.321–326, Feb. 2004.

[21] S. Kajihara, K. Taniguchi, K. Miyase, I. Pomeranz, and S. Reddy, "Test data compression using don't-care identification and statistical encoding," Proc. Asian Test Symp., pp.67–72, 2002.

[22] X. Lin, J. Rajski, I. Pomeranz, and S.M. Reddy, "On static test compaction and test pattern ordering for scan designs," Proc. Intl. Test Conf., pp.1088–1097, 2001.

[23] J. Saxena, K.M. Butler, V.B. Jayaram, and S. Kundu, "A case study of IR-drop in structured at-speed testing," Proc. Intl. Test Conf., pp.1098–1104, 2003.

[24] K. Lee, T. Huang, and J. Chen, "Peak-power reduction for multiple-scan circuits during test application," Proc. Asian Test Symp., pp.435–440, 2000.

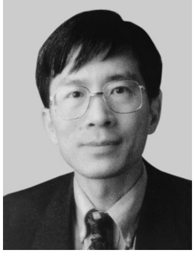
[25] R. Sankaralingam and N.A. Touba, "Controlling peak power during scan testing," Proc. VLSI Test Symp., pp.153–159, 2002.

[26] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," Proc. ISCAS'89, pp.1929–1934, 1989.

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

Yoshiyuki Yamashita received his B.E. and M.E. degrees in Computer Science and Systems Engineering from Kyushu Institute of Technology, Japan, in 2003 and 2005, respectively. From 2005, he has been working at Densotechno Co., Japan. His research interest is low power testing of VLSI circuits.

Seiji Kajihara received the B.S. and M.S. degrees from Hiroshima University, Japan, and the Ph.D. degree from Osaka University, Osaka, Japan, in 1987, 1989, and 1992, respectively. From 1992 to 1995, he worked with the Department of Applied Physics, Osaka University, Japan, as an Assistant Professor. In 1996, he joined the Department of Computer Science and Electronics of Kyushu Institute of Technology, Japan, where he is a Professor currently. His research interest includes test generation, delay testing, and design for testability. He received the Young Engineer Award from IEICE in 1997 and the Yamashita SIG Research Award from IPSJ in 2002. Dr. Kajihara is a member of the IEEE, the IEICE, and the IPSJ. He serves on the editorial board of the Journal of Electronic Testing: Theory and Applications.



Laung-Terng Wang received his B.S. and M.E. degrees in Electrical Engineering from National Taiwan University, Taiwan, in 1975 and 1977, respectively, and his M.S. and Ph.D. degrees in Electrical Engineering from Stanford University in 1982 and 1987, respectively. He was a Lecturer at the Department of Electrical Engineering and Computer Science of Stanford University from 1989 to 1991. In Jan. 1990, he founded SynTest Technologies, Inc. headquartered in Sunnyvale, California. Since then, he has led the company to grow to more than 60 full-time employees and 200 customers worldwide. Prior to founding SynTest, he worked at several technology companies, including Intel and Daisy Systems. He has published more than 30 technical papers and filed more than 20 US patents in the areas of test compression, logic built-in self-test, design for testability, and design for debug/diagnosis. He is a Senior Member of the IEEE.

Kewal K. Saluja obtained his Bachelor of Engineering B.E. degree in Electrical Engineering from the University of Roorkee, India in 1967, M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Iowa, Iowa City in 1972 and 1973 respectively. He is currently with the Department of Electrical and Computer Engineering at the University of Wisconsin-Madison as a Professor, where he teaches courses in logic design, computer architecture, microprocessor based systems, VLSI design and testing, and fault-tolerant computing. Prior to this he was at the University of Newcastle, Australia. Professor Saluja has held visiting and consulting positions at various national and international institutions including University of Southern California, Hiroshima University, Nara Institute of Science and Technology, and the University of Roorkee. He has also served as a consultant to the United Nations Development Program. He served as an Editor of the IEEE Transactions on Computers (1997–2001), and he is an Associate Editor for the letters section of the Journal of Electronic Testing: Theory and Applications (JETTA) published by Kluwer. Professor Saluja is a fellow of the JSPS and a Fellow of the IEEE.

Kozo Kinoshita received B.E., M.E., and Ph.D. in Communication Engineering from Osaka University in 1959, 1961, and 1964, respectively. From 1964 to 1966 he was an Assistant Professor and from 1967 to 1977, an Associate Professor of Electronic Engineering at Osaka University, Osaka, Japan. From 1978 to 1989, he was a Professor in the Department of Information and Behavioral Sciences, Hiroshima University, Hiroshima, Japan. From 1989 to 2000, he again joined Osaka University as a Professor in the Department of Applied Physics, and is enumerates professor of Osaka University. Since April 2000, he has been a professor at Faculty of Informatics, Osaka Gakuin University, and is the Dean of Informatics. His fields of interest are test generation, fault diagnosis, memory testing, current testing, crosstalk testing, compact testing and testable design for logic circuits. He organized a series of Asian Test Symposium and was the Group Chair of Asian and Pacific Activities in Test Technology Technical Council of IEEE Computer Society until 2002. Prof. Kinoshita is IEEE Life Fellow and a member of the Institute of Information Processing of Japan. He was a member of the editorial board of JETTA until 2000.