Low-capture-power test generation for scan-based at-speed testing

Low-Capture-Power Test Generation for Scan-Based At-Speed Testing

Xiaoqing Wen [1], Yoshiyuki Yamashita [1], Shohei Morishima [1], Seiji Kajihara [1], Laung-Terng Wang [2], Kewal K. Saluja [3], and Kozo Kinoshita [4]

[1] Dept. of CSE, Kyushu Institute of Technology, Iizuka 820-8502, Japan
[2] SynTest Technologies, Inc., 505 S. Pastoria Ave., Suite 101, Sunnyvale, CA 94086, USA
[3] Dept. of ECE, 1415 Engineering Drive, University of Wisconsin - Madison, Madison, WI 53706, USA
[4] Faculty of Informatics, Osaka Gakuin University, Suita 564-8511, Japan

Abstract

Scan-based at-speed testing is a key technology to guarantee timing-related test quality in the deep submicron era. However, its applicability is being severely challenged since significant yield loss may occur from circuit malfunction due to excessive IR drop caused by high power dissipation when a test response is captured. This paper addresses this critical problem with a novel low-capture-power X-filling method of assigning 0's and 1's to unspecified (X) bits in a test cube obtained during ATPG. This method reduces the circuit switching activity in capture mode and can be easily incorporated into any test generation flow to achieve capture power reduction without any area, timing, or fault coverage impact. Test vectors generated with this practical method greatly improve the applicability of scan-based at-speed testing by reducing the risk of test yield loss.

1. Introduction

Scan-based testing, carried out by a tester on a full-scan circuit with deterministic test vectors obtained through automatic test pattern generation (ATPG), is the most widely adopted test strategy to achieve required test quality for an integrated logic circuit at acceptable costs.

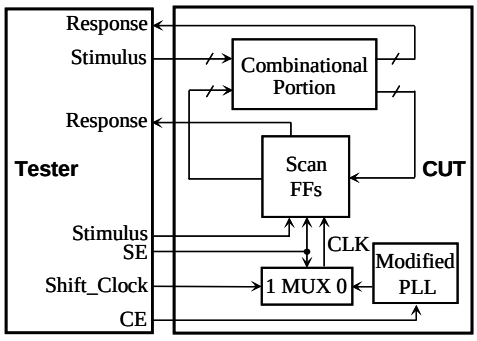
In a full-scan sequential circuit, scan flip-flops (FFs) replace all functional FFs and operate in two modes: *shift* and *capture*. In shift mode, scan FFs form scan chains, through which a test vector is applied during shift-in or a test response is observed during shift-out, for the combinational portion of the circuit. In capture mode, scan FFs operate as functional FFs and load the test response of the combinational portion for a test vector into themselves, getting ready for shift-out later in shift mode. Thus, the problems of testing a full-scan sequential circuit is reduced to that of testing its combinational portion, in that now it is sufficient to generate test vectors only for the combinational portion with combinational ATPG.

In scan-based testing, after a test vector is applied in shift mode, its test response is loaded into FFs in capture mode after a waiting period either *greater than* or *equal to* the rated clock period. The former is called *low-speed testing*, and the latter is called *at-speed testing* [1, 2]. Low-speed testing checks for unexpected logic values based on such fault models as stuck-at and bridging; while at-speed testing checks for unexpected excessive delays based on such fault models as transition delay and path delay.

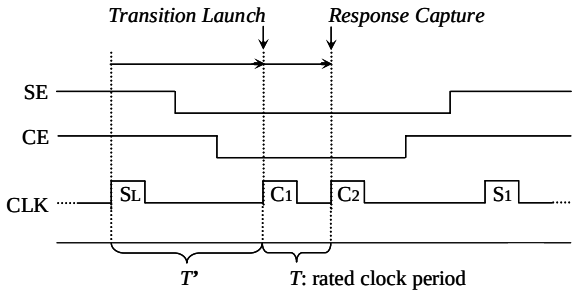
As transistor feature sizes shrink, more chips fail because of timing-related defects [3]. I_{DDQ} testing [4] was widely used for screening out such defective chips, but is now losing its effectiveness due to elevated normal quiescent current. Therefore, at-speed testing through options such as logic built-in self-test (BIST) or external scan-based testing needs to be considered.

Compared to at-speed logic BIST, which is difficult to implement and usually has low fault coverage because of random pattern usage, scan-based at-speed testing with ATPG and an external tester has the advantages of low circuit overhead, low application cost, and high test quality [2]. As a result, scan-based at-speed testing, especially when conducted by using on-chip phase-locked loops (PLLs) [5], has emerged as a key technology for guaranteeing test quality in the deep submicron (DSM) era.

Fig. 1 shows an example scan-based at-speed testing system with on-chip PLL and the broadside clocking scheme.



(a) System Overview



(b) Broadside Clocking Scheme

Fig. 1 Scan-Based At-Speed Testing System.

In Fig. 1, SE and CE are the normal scan enable signal and a newly added capture enable signal, respectively. As shown in Fig. 1 (a), a test vector is applied in shift mode (SE = 1) via a series of shift clock pulses with S_L being the last one. In capture mode (SE = 0), the modified PLL responds to the falling edge of the CE signal to provide two pulses C_1 and C_2 at the rated clock interval of *T* as shown in Fig. 1 (b). C_1 launches a transition with respect to S_L while C_2 captures the circuit response to the transition at-speed. As a result, timing-related defects can be detected. This is the so-called *broadside clocking scheme* or *launch-off-capture clocking scheme* [1, 2].

The scan-based at-speed testing system shown in Fig. 1 has the following advantages: (1) *Low Tester Requirement*: A low-speed tester can be used to provide shift clock pulses at a lower frequency than the rated clock frequency, while only the at-speed capture clock pulse, e.g. C_2 in Fig. 1 (b), needs to be generated by the on-chip PLL that is also used in functional mode. (2) *Easy Physical Implementation*: The broadside clocking scheme only needs a non-timing-critical, thus easy-to-implement, SE signal since *T'* can be much larger than the rated clock period of *T* in Fig. 1 (b). This makes it easier for the broadside clocking scheme to be physically implemented than other clocking schemes, such as *skewed-load* or *launch-off-shift* [2]. (3) *High Test Quality*: The broadside clocking scheme generally activates fewer false paths since logic value transitions are generated by the difference between a shifted-in vector and the functional response to the vector. This generally results in less "over-testing" than the skewed-load scheme.

The above advantages make scan-based at-speed testing with on-chip PLL and the broadside clocking scheme highly preferable for screening out chips with timing-related defects in production testing. However, the adoption of this testing technology is being severely hindered by four problems: (1) *test data volume*, (2) *test application time*, (3) *test heat dissipation*, and (4) *test yield loss*.

The test data volume and test application time problems are caused by larger gate/FF counts, longer scan chains, and the use of complex delay fault models, all inevitable in the DSM era. Several approaches, such as test compaction, multi-capture clocking, decompression-compression, encoding, are available for addressing these problems.

The test heat dissipation and test yield loss problems are both related to test power dissipation during scan testing, which is much higher than during normal operation [6].

Test heat is caused by the *accumulated* effect of test power dissipation, mostly in shift mode for a large number of cycles. Excessive heat may cause permanent damage to the chip-under-test, increasing package costs, or reducing circuit reliability due to accelerated electromigration [7].

Previous methods for test heat reduction are based on four major approaches: *scheduling*, *test vector manipulation*, *circuit modification*, and *scan chain modification*, to reduce the switching activity in shift mode. Test scheduling [8, 9] takes the power budget into consideration when selecting modules to be tested simultaneously. Test vector manipulation includes power-aware ATPG [10, 11], static compaction [12], test vector modification [13], test vector reordering [14], test vector compression [15], and coding [16]. Circuit modification includes transition blocking [17], clock gating [18], and the use of multiple clock duties [19]. Scan chain modification includes scan chain reordering [15, 20], scan chain partitioning [21], and scan chain modification [22]. Methods tailored for BIST applications, such as toggle suppression [23] and low-power test pattern generation [24], have also been proposed.

Test yield loss is caused by excessive *instantaneous* test power dissipation in both shift and capture mode, because FFs and/or PLL may malfunction due to power supply voltage drop and ground bounce [19, 25, 28]. This problem is worsening as feature sizes shrink below 0.18 micron.

Most of the previous methods [8-24] for test heat reduction in shift mode also reduce instantaneous test power dissipation in shift mode, thus lowering the risk of test yield loss in shift mode. Among them, the multi-duty scan method [19] is especially effective, which changes clock duties so that fewer FFs operate simultaneously.

There are a few methods for reducing test yield loss in capture mode. One method [26] uses an interleaving scheme to reduce the number of FFs that are clocked simultaneously in capture mode, at the cost of increased control complexity. Another method [27] uses an *X*-filling technique in static compaction to reduce the number of capture transitions at FFs. Yet another method [28], called *single-capture low-capture-power* (*SC-LCP*) *X*-filling, conducts algorithmic *X*-Filling in dynamic or static compaction so as to reduce circuit switching activity in capture mode. These methods, however, only work for low-speed testing and at-speed testing based on the skewed-load clocking scheme, both featuring a single capture pulse.

The impact of IR-drop for capture mode in scan-based at-speed testing has been analyzed in [25] for the broadside clocking scheme, where two capture pulses are used. Quiet test vectors, which result in low switching activity in capture mode, were shown to be beneficial. However, it was also shown that existing ATPG programs failed to generate such "hot" test vectors when the straightforward approach of placing additional constraints was used. This is a serious problem since "cool" test vectors may result in significant test yield loss, thus severely challenging the applicability of scan-based at-speed testing that is considered indispensable in the DSM era.

This paper proposes a unique and novel ATPG approach to reducing instantaneous test power dissipation in capture mode for scan-based at-speed testing with the broadside clocking scheme. The basic idea is to make use of *test cubes*, i.e., test vectors with unspecified bits (*X-bits*), which

exist during ATPG. We develop a novel technique, called *double-capture low-capture-power* (*DC-LCP*) *X*-filling, for algorithmically assigning 0's and 1's to the *X*-bits in test cubes so as to reduce the circuit switching activity caused by two capture pulses in the broadside clocking scheme.

The DC-LCP *X*-filling method can be easily incorporated into dynamic compaction of any test generation flow, and the resulting "cool" test vectors can achieve capture power reduction without any area, timing, or fault coverage impact. As a result, test yield loss in capture mode can be efficiently lowered, thus greatly improving the applicability of scan-based at-speed testing with the broadside clocking scheme.

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

2. Background

As shown in Fig. 2, an integrated circuit can be seen as a network of interconnected transistors existing between a VDD (power grid) and a VSS (ground grid). These transistors form functional cells, i.e. logic gates and FFs. Cells switch their output values dynamically to perform various required functionality.

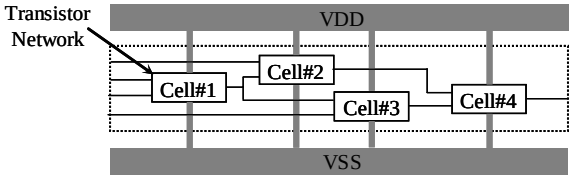


Fig. 2 Example Integrated Circuit with Power/Ground Grids.

Whenever a cell switches its output, a dynamic current path will be established between VDD and VSS. If a large amount of cells switch their outputs simultaneously, i.e. if instantaneous power is too large, a significant power supply voltage drop will occur. The major reason is the IR effect since a dynamic current (I) flows through the resistance (R) of the VDD grid, the VSS grid, and the transistor network. In addition, parasitic or capacitive effects also contribute to power supply voltage drop. Generally, the amount of power supply voltage drop depends on the number of simultaneous switching cells, their types, their locations, etc.

Normally, the power supply/ground pins and distribution system of a circuit is designed only for handling the peak power that occurs during normal operation. Thus, it may not be able to handle the large instantaneous power that could occur during scan testing since test power is much higher than normal power. Unfortunately, IR drop analysis for test mode is almost never conducted in most design flows, making a circuit vulnerable to test yield loss.

For example, as reported in [19], power supply voltage even dropped to 17% of its normal value in a 0.18micron industrial design during scan testing. Such power supply voltage drop in test mode may cause circuit malfunction, resulting in test yield loss [19, 25]. Its mechanism is illustrated in Fig. 3.

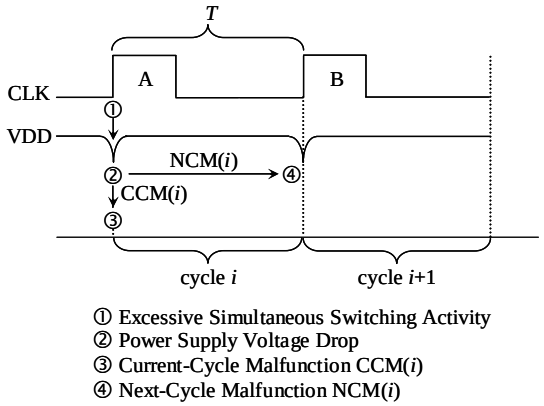


Fig. 3 Malfunctions due to Power Supply Voltage Drop.

In Fig. 3, simultaneous switching activity (①) increases dynamic power dissipation, which in turn causes power supply voltage drop (②). The direct result is nonlinear performance degradation of transistors, especially in a DSM circuit of very fine geometries. For a FF consisting of degraded transistors, the degradation can translate into direct malfunction, i.e. loading of a wrong value into the FF in the same cycle where simultaneous switching activity occurs. This is called the *current-cycle malfunction* (*CCM*) (③). In addition, for a combinational logic gate consisting of degraded transistors, the degradation often translates into increased gate delay, which in turn increases path delays in a circuit. Generally, a 10% drop in power supply voltage can increase path delay by 30%. The increased path delay may violate required timings at some FFs in the next cycle, also resulting in circuit malfunction. This is called the *next-cycle malfunction* (*NCM*) (④). Moreover, supply power drop may also cause PLL malfunction. Obviously, all these factors may result in potential test yield loss.

Note that the clock pulse A in Fig. 3 can be a shift pulse or a capture pulse. This means that test yield loss may occur in shift mode or capture mode or both. Several techniques exist for reducing switching activity in shift mode [8-24] and in capture mode [25-28].

Also note that scan-based at-speed testing is especially vulnerable to the power supply voltage drop problem. The reason is that, in at-speed testing, there must be one clock interval equal to the related clock period for each test vector. Suppose *T* in Fig. 3 is such an interval. Since *T* is very short for a high-speed circuit, the risk of voltage-drop-induced delays causing next-cycle malfunction is high, thus increasing the total risk of test yield loss.

For scan-based at-speed testing with the broadside clocking scheme, which features two capture pulses as shown in Fig. 1 (b), its detailed mechanism for voltage-drop-induced malfunction is illustrated in Fig. 4.

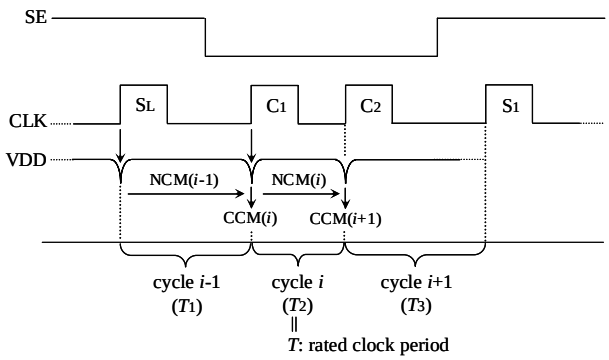


Fig. 4 Possible Malfunctions in Broadside Clocking Scheme.

As shown in Fig. 4, possible malfunctions related to test yield loss in capture mode for the broadside clocking scheme are NCM(*i*-1), CCM(*i*), NCM(*i*), and CCM(*i*+1), as described below:

(1) NCM(*i*-1) is the next-cycle malfunction for cycle *i*-1, which is the last shift cycle. Its risk, however, is low since T_1 can be made as large as necessary, allowing enough timing margin for absorbing any voltage-drop-induced delay. Thus, NCM(*i*-1) can be ignored.

(2) CCM(*i*) and CCM(*i*+1) are current-cycle malfunctions for the two capture pulses, i.e. cycle *i* and cycle *i*+1, respectively. Both of them may cause test yield loss and their risks need to be contained by reducing the circuit switching activity at C_1 and C_2.

(3) NCM(*i*) is the next-cycle malfunction for cycle *i*, and it may cause malfunction at C_2. This is because T_2 is equal to the rated clock period, which is very short for a high-speed circuit, leaving less space in timing margin for absorbing voltage-drop-induced delay. Its risk needs to be contained by reducing the circuit switching activity at C_1.

Thus, to reduce the test yield loss in capture mode for scan-based at-speed testing with the broadside clocking scheme, it is necessary to reduce the risks of CCM(*i*), NCM(*i*), and CCM(*i*+1) by reducing the circuit switching activity at C_1 and C_2. The next section presents an innovative method to achieve this goal.

3. Low-Capture-Power Test Generation

3.1 Test Generation Flow

In ATPG, a primary target fault is selected from undetected faults and a test *v* is generated for it. At this stage, *v* usually contains unspecified bits (*X*-bits), and it is called a *test cube*. Next, a conventional dynamic compaction as shown in Fig. 5 is conducted for *v* to detect more faults.

In Fig. 5, the function *promising*(*v*) decides whether *v* is a good candidate for dynamic compaction. If *v* is promising, *X*-bits in *v* will be explored algorithmically to see whether a secondary target fault can be detected. If *v* is not promising, random *X*-filling is conducted to all remaining *X*-bits in *v*.

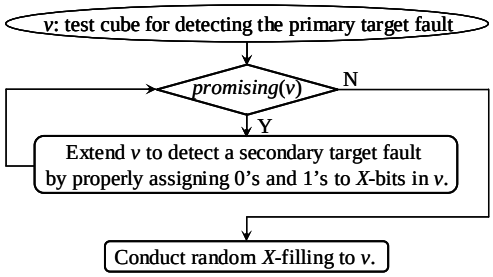


Fig. 5 Conventional Dynamic Compaction Flow.

Random *X*-filling may help in reducing the number of total test vectors since it increases the chances of detecting additional faults. These additionally detected faults can be identified by fault simulation after random *X*-filling. However, random *X*-filling usually adversely affects test power dissipation [12].

The basic idea of low-capture-power test generation is to algorithmically, instead of randomly, assign 0's and 1's to *X*-bits in the *X*-filling stage, so that capture power dissipation is reduced. Fig. 6 shows the proposed dynamic compaction flow for low-capture-power test generation.

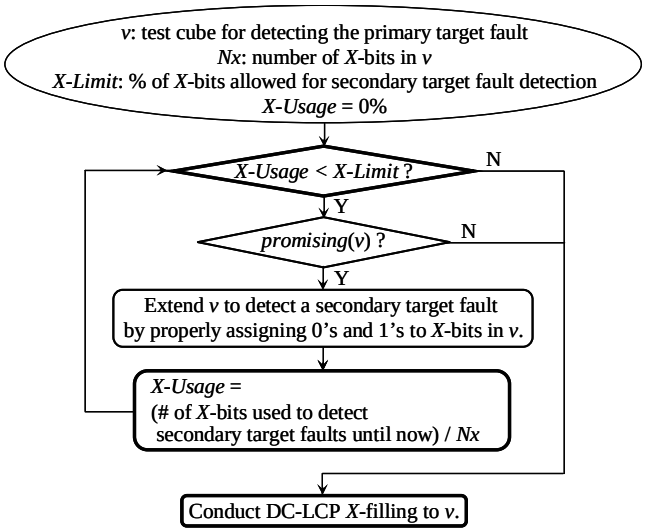


Fig. 6 Proposed Dynamic Compaction Flow.

In Fig. 6, *X*-filling is conduced by a new method called *double-capture low-capture-power* (*DC-LCP*), instead of random *X*-filling. That is, different from the conventional dynamic compaction flow as shown in Fig. 5, *X*-bits in the proposed dynamic compaction are used not only with the objective of reducing the number of total test vectors but also with the objective of reducing capture power dissipation. Obviously, these two objectives can be conflicting.

In order to balance the conflicting objectives, a new concept, called *X-usage control*, is introduced. As shown in Fig. 6, *X-Limit* is a user-specified threshold that defines the percentage of original *X*-bits to be allowed for detecting secondary target faults. A measure, *X-Usage*, is updated

each time a secondary target fault is detected. When *X-Usage* becomes greater than *X-Limit*, test cube extension for the objective of fault-detection is terminated and DC-LCP *X*-filling is invoked for the remaining *X*-bits to achieve the objective of reducing capture power dissipation.

The details of the DC-LCP *X*-filling method are presented in the following subsections 3.2 through 3.4.

3.2 Circuit Model

For the convenience of presentation in the following, a signal scan chain in a single clock domain, as shown in Fig. 7, is assumed. The DC-LCP *X*-filling method, however, can be readily extended for any full-scan circuit with multiple scan chains in multiple clock domains.

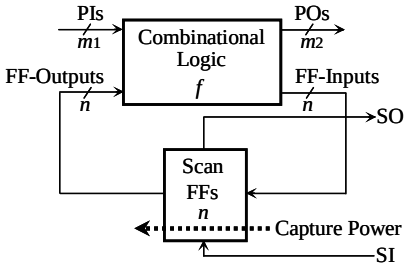


Fig. 7 General Full-Scan Circuit.

Fig. 8 shows the circuit model for low-capture-power test vector generation in the broadside clocking scheme shown in Fig. 1 for the general full-scan circuit shown in Fig. 7. Note that, the combinational logic in Fig. 7 is assumed to implement the logic function *f*.

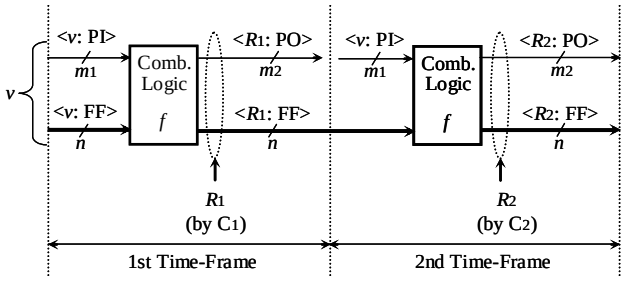


Fig. 8 Circuit Model for the Broadside Clocking Scheme.

In Fig. 8, *v* is the input vector in the first time-frame, which is provided from primary inputs and the scan FFs of a scan chain. That is, *v* consists of two parts: primary input bits denoted by <*v*: PI> and the FF-output bits denoted by <*v*: FF>. The functional response of the combinational logic to *v* is *f*(*v*), denoted by R_1. For R_1, its bits related to primary outputs are denoted by <R_1: PO> and its bits related to FFs are denoted by <R_1: FF>. When the first capture C_1 is conducted, <R_1: FF> is loaded into all FFs to replace <*v*: FF>, and <<*v*: PI>, <R_1: FF>> becomes the input vector in the second time-frame. The functional response of the combinational logic to this input vector is *f*(<<*v*: PI>, <R_1: FF>>), denoted by R_2. For R_2, its bits related to primary outputs are denoted by <R_2: PO> and its bits related to FFs are denoted by <R_2: FF>. When the second capture C_2 is conducted, <R_2: FF> is loaded into all FFs to replace <R_1: FF>. Note that both R_2 and R_2 can be readily obtained through logic simulation.

Also note that the values for the primary inputs remain the same in both time-frames. This assumption is made since it is usually difficult and costly to change primary input values during the rated clock period between the first and second capture pulses in the broadside clocking scheme for a high-speed design.

3.3 DC-LCP *X*-Filling Problem Formalization

As shown in Fig. 8, <*v*: FF> is replaced by <R_1: FF> when the first capture C_1 is conducted, and <R_1: FF> is replaced by <R_2: FF> when the second capture C_2 is conducted. Obviously, if <*v*: FF> is different from <R_1: FF> at some scan FFs, *capture transitions* will occur at the outputs of these scan FFs for C_1 as shown in Fig. 9 (a). Similarly, if <R_1: FF> is different from <R_2: FF> at some scan FFs, *capture transitions* will occur at the outputs of these scan FFs for C_2 as shown in Fig. 9 (b).

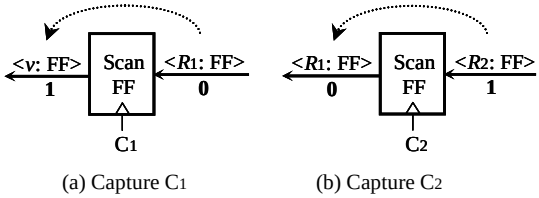


Fig. 9 Capture Transitions at a Scan FF.

Capture transitions at FFs has a strong correlation with circuit switching activity [12], and thus capture test power dissipation. Therefore, capture power reduction can be achieved by reducing the number of capture transitions. Note that not all FFs carry the same weight regarding to power dissipation in practice. That is, capture transitions at some FFs may cause more power dissipation than other FFs. In this paper, it is assumed that all FFs have the same weight. The case where FFs have different weights will be considered in the future.

From Fig. 8 and Fig. 9, it is clear that capture transitions for C_1 and C_2 are caused by the difference between <*v*: FF> and <R_1: FF> and the difference between <R_1: FF> and <R_2: FF>, respectively. Therefore, capture transitions for C_1 and C_2 can be reduced by making <*v*: FF> similar to <R_1: FF> and <R_1: FF> similar to <R_2: FF> as much as possible. DC-LCP *X*-filling is used to achieve this goal by properly assigning 0's and 1's to the *X*-bits in *v*, which is a test cube with at least one *X*-bit.

The obvious requirement for DC-LCP *X*-filling is to reduce capture transitions for C_1 and C_2 as much as possible. In addition, the fact that two captures are involved makes it necessary to conduct capture transition reduction in a balanced manner with respect to C_1 and C_2. Therefore, the DC-LCP *X*-filling problem can be formalized as follows:

DC-LCP X-Filling Problem: Given a test cube v for a full-scan circuit with respect to the broadside clocking scheme as shown in Fig. 8, assign 0's and 1's to all X-bits in v so that $(N_1 + N_2)$ and $|N_1 - N_2|$ are both minimized, where N_1 and N_2 are the numbers of capture transitions for the first capture C_1 and the second capture C_2, respectively.

3.4 DC-LCP *X*-Filling Algorithm

In Fig. 8, suppose that x is one bit in $<v$: FF$>$ with respect to a FF. Then, there must be one bit y in $<R_1$: FF$>$ and one bit z in $<R_2$: FF$>$, both with respect to the same FF as x. $<x, y, z>$ is called a *3-bit-tuple* in this paper. The circuit model in Fig. 8 has n 3-bit-tuples since there are n FFs in the full-scan circuit.

In addition, if v is a test cube with at least one X-bit, there must be some X-bits in 3-bit-tuples for the circuit. Depending on how X-bits appear, 3-bit-tuples can be classified into 8 *X-types* as summarized in Table 1.

Table 1 *X*-Types

Type	# of *X*'s	$<v$: FF$>$	$<R_1$: FF$>$	$<R_2$: FF$>$	Target Capture
1	0	b_1	b_2	b_3	---
2	1	X	b_2	b_3	C_1
3		b_1	X	b_3	C_1, C_2
4		b_1	b_2	X	C_2
5	2	b_1	X	X	C_1, C_2
6		X	b_2	X	C_1, C_2
7		X	X	b_3	C_1, C_2
8	3	X	X	X	C_1, C_2

(b_1, b_2, b_3: 0 or 1)

Obviously, there is no need to consider any 3-bit-tuple of Type-1 in DC-LCP X-filling. A 3-bit-tuple of Type-2 through Type 8 has at least one X-bit and it can be used for capture transition reduction in DC-LCP X-filling.

Note that 3-bit-tuples of different types may reduce capture transitions for different captures. For example, a 3-bit-tuple of Type-2 in the form of $<X, b_2, b_3>$ can only reduce capture transitions for the first capture C_1, and this is achieved if the X-bit can take logic value b_2. On the other hand, consider a 3-bit-tuple $<b_1, X, b_3>$ of Type-3 with $b_1 \neq b_3$. This 3-bit-tuple can be used to reduce capture transitions for C_1 if X-bit takes logic value b_1 or for C_2 if X-bit takes logic value b_3. The type of information on what X-type can reduce capture transitions for what capture is also shown in the column "Target Capture" of Table 1.

In the following, the details of the DC-LCP X-filling algorithm are described, starting with the general procedure and an example in 3.4.1, followed by detailed discussions of the three key operations in 3.4.2 through 3.4.4.

3.4.1 General Procedure

Fig. 10 shows the general DC-LCP X-filling procedure.

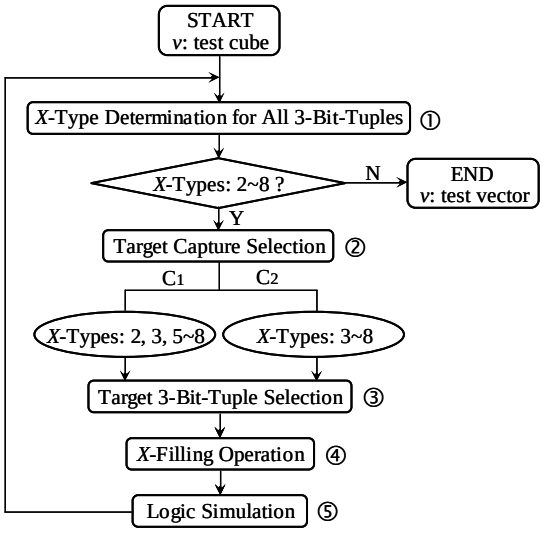


Fig. 10 DC-LCP *X*-Filling Procedure.

The input to the DC-LCP X-filling procedure is a test cube v with at least one X-bit, and the output is a fully-specified test vector. The procedure consists of the following steps:

(1) *X-Type Determination* is conducted to determine the X-types of all 3-bit-tuples. If only 3-bit-tuples of Type-1 exist, v is already a fully-specified test vector and the procedure ends.

(2) *Target Capture Selection* is conducted to determine which capture, C_1 or C_2, should be targeted in the current iteration of capture transition reduction. This is to guarantee that capture transitions for C_1 and C_2 are reduced in a balanced manner.

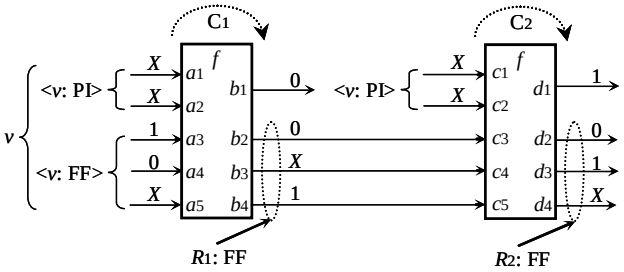
(3) *Target 3-Bit-Tuple Selection* is conducted to pick up one 3-bit-tuple that has at least one X-bit and that has the highest possibility of successfully reducing capture transitions for the capture determined at Step-1.

(4) *X-Filling Operation* uses assignment and justification techniques to find proper logic values for the X-bits in the test cube v so that necessary logic value(s) can appear at the X-bit(s) in the 3-bit-tuple selected at Step-2 in order to reduce capture transitions for the capture selected at Step-1.

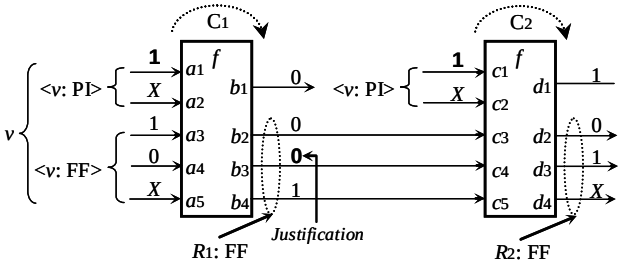
(5) *Logic Simulation* is conducted to spread the effect of the newly determined logic values at X-bits in v to the whole circuit. Obviously, the X-types of some 3-bit-tuples may change because of this.

Clearly, the DC-LCP X-filling procedure shown in Fig. 10 handles one 3-bit-tuple in each iteration, and each iteration consists of the above 5 steps. For a circuit of n FFs as shown in Fig. 8, there are n 3-bit-tuples. That is, at most n iterations are needed in order to complete DC-LCP X-filling. Since each iteration mainly consists of justification and logic simulation operations, the run time of DC-LCP X-filling is strictly under control, making it feasible to be used in the proposed dynamic compaction procedure shown in Fig. 6 for large circuits.

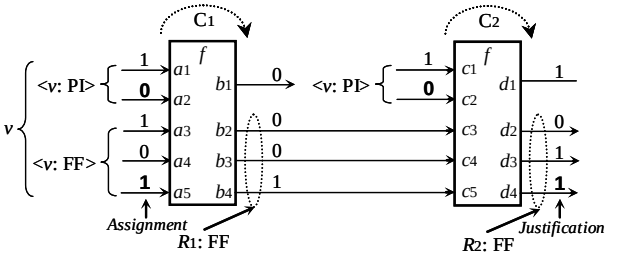
An example of DC-LCP *X*-filling is shown in Fig. 11. The circuit under the original test cube *v* <*X*, *X*, 1, 0, *X*> is shown in Fig. 11 (a). For this test cube, there are three 3-bit-tuples: <1, 0, 0>, <0, *X*, 1>, and <*X*, 1, *X*>.



(a) Circuit under the Original Test Cube



(b) Circuit after Iteration-1



(c) Circuit after Iteration-2

Fig. 11 Example of DC-LCP *X*-Filling.

***Iteration*-1**:

As shown in Fig. 11 (a), there is one capture transition for C1 but no capture transition for C2 with respect to <1, 0, 0>. Capture transition information with respect to <0, *X*, 1> and <*X*, 1, *X*> is unclear since *X*-bits are involved. As a result, in order to achieve balanced capture transition reduction at this stage, it is necessary to reduce capture transitions for C1. Although both <0, *X*, 1> and <*X*, 1, *X*> may serve this purpose, <0, *X*, 1> is selected since it involves only one *X*-bit, making it easier to bring 0 to the *X*-bit to reduce capture transitions for C1.

Proper logic values needed for *X*-bits in *v* in order to bring logic 0 to the *X*-bit in <0, *X*, 1> are found by justifying 0 on *b*3. The result is logic 1 for the *X*-bit on *a*1 and *c*1. As shown in Fig. 11 (b), the 3-bit-tuple <0, *X*, 1> becomes <0, 0, 1>, which causes no capture transition for C1. □

***Iteration*-2**:

As shown in Fig. 11 (b), there is one capture transition for C1 with respect to <1, 0, 0> and one capture transition for C2 with respect to <0, 0, 1>. Capture transition information with respect to <*X*, 1, *X*> is unclear since *X*-bits are involved. As a result, it is necessary to reduce capture transitions for both C1 and C2. Here, <*X*, 1, *X*> is the only 3-bit-tuple to serve this purpose, requiring logic 1 to appear on both *X*-bits in <*X*, 1, *X*>.

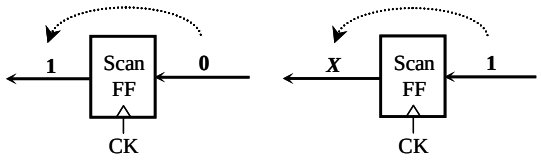
Proper logic values needed for *X*-bits in *v* in order to bring logic 1 to both *X*-bits in <*X*, 1, *X*> are found by assigning 1 to the *X*-bit on *a*5 and justifying 1 on *d*4. The result of justification is logic 0 for the *X*-bit on *a*2 and *c*2. As shown in Fig. 11 (c), the 3-bit-tuple <*X*, 1, *X*> becomes <1, 1, 1>, which causes no capture transition for both C1 and C2. □

As shown in Fig. 11, after two iterations of DC-LCP *X*-filling, the original test cube *v* <*X*, *X*, 1, 0, *X*> becomes a fully-specified test vector <**1**, **0**, 1, 0, **1**>.

In the following, details of three key operations in DC-LCP *X*-filling: *target capture selection*, *target 3-bit-tuple selection*, and *X-filling*, are described in 3.4.2 through 3.4.4.

3.4.2 Target Capture Selection

The DC-LCP *X*-filling method dynamically selects a target capture in order to achieve a balanced reduction of capture transitions for the first capture C1 and for the second capture C2. The target capture selection heuristic is based on the *total estimated capture transition activity* (*TECTA*), which is calculated from *existing capture transitions* (*ECTs*) and *potential capture transitions* (*PCTs*) as illustrated in Fig. 12.



(a) ECT (b) PCT

Fig. 12 Existing and Potential Capture Transitions.

An ECT is a capture transition in the case where a logic value is loaded into a scan FF to replace a different logic value. An example of ECT is shown in Fig. 12 (a). On the other hand, a PCT is a capture transition in the case where a value *V*2 is loaded into a scan FF to replace another value *V*1, where either *V*1 or *V*2 or both are *X*-bits. An example of PCT is shown in Fig. 12 (b).

The probability of an ECT to occur is 100%; while the probability of a PCT to actually cause a real capture transition is 50% if it is simply assumed that all related *X*-bits in the PCT could take any logic value with equal probability. Based on this observation, *TECTA* for capture C*i* (*i* = 1, 2), denoted by *TECTAi*, can be calculated as follows:

$TECTA_i$ = (# of ECTs for Ci) + (0.5 × (# of PCTs for Ci))

Generally, the capture with a higher *TECTA* is selected as the target capture, since the number of capture transitions for this capture is likely to be greater than that for the other capture, and hence it needs to be reduced first. An example is shown in Fig. 13, which has four 3-bit-tuples. In this case, C1 is selected since $TECTA_1 > TECTA_2$.

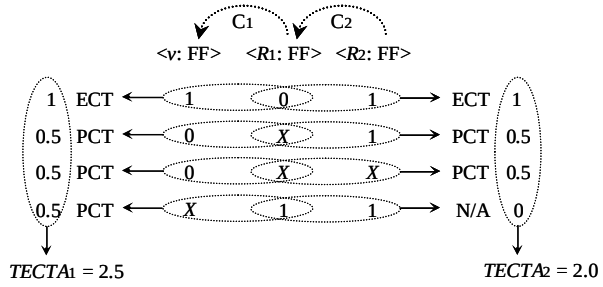


Fig. 13 Example of Target Capture Selection.

3.4.3 Target 3-Bit-Tuple Selection

Once a target capture is selected, it is necessary to further select a target 3-bit-tuple that has at least one *X*-bit and that has the highest possibility of successfully reducing capture transitions for the selected target capture.

As shown in the example of DC-LCP *X*-filling in Fig. 11, assignment and justification are used to determine logic values for *X*-bits in a test cube *v* to make required logic values appear at the *X*-bits in a 3-bit-tuple so that capture transitions are reduced. *Assignment* is to set a logic value to an *X*-bit in <*v*: FF> directly. Since any logic value can be loaded to any scan FF in shift mode for <*v*: FF>, assignment is simple and always successful. On the other hand, *justification* is to identify proper logic values for *X*-bits in *v* to make required logic values appear at the *X*-bits in <*R*1: FF> or <*R*2: FF>. Obviously, justification can be difficult and there is no guarantee that this operation is always successful.

As a result, in target 3-bit-tuple selection, we first select a 3-bit-tuple that only needs assignment in *X*-filling. Only when there is no such 3-bit-tuple, we select from 3-bit-tuples that need justification in *X*-filling, based on a heuristic measure. An example is shown in Fig. 14.

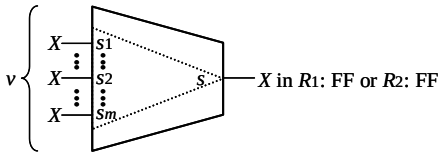


Fig. 14 *X*-Bit Justification.

In Fig. 14, there is one *X*-bit on signal line *s* on which justification is needed. Suppose that the level of *s* is *L*s. Also suppose that *s* can reach *m* *X*-bit signal lines s_1, s_2, …, s_m corresponding to a test cube *v*, and that the levels of these signal lines are L_{s1}, L_{s2}, …, L_{sm}. Here, levels are assigned from the output side to the input side, and the highest level is denoted by *L*.

Conceptually, it is evident that if more *X*-bit signal lines are reachable from *s* and closer they are to *s*, then easier they are to justify a logic value on *s*. Based on this observation, the *justification easiness* (*JE*) of *s*, denoted by *JE*(*s*), is calculated as follows:

$$JE(s) = \sum_{i}^{m} \frac{(L - |Ls - Ls_i|)}{L}$$

Obviously, the larger the value of *JE*(*s*), the easier the justification of a logic value on *s*.

When it is necessary to select a 3-bit-tuple that needs justification, we first select from 3-bit-tuples with one *X*-bit in <*R*1: FF> or <*R*2: FF>. The *JE* value for the signal line with the *X*-bit is calculated, and the 3-bit-tuple of the largest *JE* value is selected. If there are only 3-bit-tuples that have two *X*-bits in <*R*1: FF> and <*R*2: FF>, the sum of the *JE* values for the signal lines with the *X*-bits is calculated, and the 3-bit-tuple with the largest sum of *JE* values is selected.

3.4.4 X-Filling Operation

After a target capture and a target 3-bit-tuple are selected, assignment and/or justification are conducted to determine logic values for *X*-bits in a test cube *v* in order to make required logic values appear at the *X*-bits in a 3-bit-tuple so that capture transitions are reduced.

Note that justification may fail. For example, for 3-bit-tuple <1, *X*, *X*>, the best choice is to make logic 1 appear at both the *X*-bits. This choice is tried first by justification. If it fails, we then try the next-to-best choice of making logic 1 appear at the first *X*-bit and logic 0 at the second *X*-bit. If this justification also fails, we then try to make logic 0 appear at both *X*-bits. If this justification also fails, the last choice is to make logic 0 appear at the first *X*-bit and logic 1 at the second *X*-bit.

3.5 Practical Issues

3.5.1 Handling of X-Sources

In practice, a circuit may contain such *X*-sources as analog blocks, memories, uninitialized FFs, multiple clock domains, floating bus, inaccurate simulation models, etc. These *X*-sources, as well as *X*-bits in a test cube, may result in some *X*-bits in the corresponding test response at the inputs of FFs.

Different from *X*-bits existing in a test cube, above-mentioned *X*-sources are uncontrollable in that it is impossible to set an *X*-source to any required logic value. As a result, in the DC-LCP *X*-filling procedure, if justifying a logic value at an *X*-bit in a test response ends up needing to set a specific value at an *X*-source as the only choice, the justification is considered unsuccessful.

3.5.2 Application to Unconventional Scan Schemes

Conventional scan scheme uses one external scan input pin and one external scan output pin for each internal scan chain. Recently, some unconventional scan schemes, such as OPMISR, VirtualScan, EDT, SoCBIST, etc., have been proposed for reducing test data volume and test application time. These scan schemes can be divided into two groups: *X-independent* (OPMISR and VirtualScan) and *X-dependent* (EDT and SoCBIST) according to if its fault detection capability depends on the use of *X*-bits in a test cube. Obviously, the DC LCP *X*-filling method readily works with any *X*-independent scan scheme.

As for *X*-dependent scan schemes, an interactive approach may be needed. That is, *X*-bits are first utilized to guarantee the minimum fault detection capability. The remaining *X*-bits are then used for detecting more faults or reducing capture test power with the DC LCP *X*-filling method, as long as the resulting test cube can be successfully compressed. Test power analysis may also need to be conducted in order to determine which type of reduction should be targeted with *X*-bits: test size or test power.

4. Experimental Results

X-filling experiments were conducted on ISCAS'89 circuits with an internally developed ATPG program for transition delay faults. Table 2 shows the circuit statistics and *X*-bit information in initial test cubes. An initial test cube is generated for detecting a primary target fault, and its *X*-bits are used in dynamic compaction for detecting secondary target faults and reducing capture power dissipation.

Table 2 Circuit Statistics and *X*-Bit Information

Circuit	# of PIs	# of FFs	# of Faults	*X*-Bits in initial test cubes	
				Ave. *X*-Bits	Max. *X*-Bits
s1423	17	74	2290	63 (69.7%)	87 (95.6%)
s5378	35	179	5980	173 (80.6%)	201 (93.9%)
s9234	19	228	10712	214 (86.4%)	244 (98.8%)
s13207	31	669	15440	658 (93.1%)	699 (99.9%)
s15850	14	597	18324	523 (85.6%)	605 (99.0%)
s35932	35	1728	54044	1485 (84.2%)	1758 (99.7%)
s38417	28	1636	49342	1420 (85.4%)	1648 (99.0%)
s38584	12	1452	55706	1294 (88.4%)	1455 (99.4%)

Table 3 shows the results of random *X*-filling in conventional dynamic compaction as shown in Fig. 8. The number of test vectors and fault coverage are shown under "*# of Vec.*" and "*Fault Cov.*". In addition, the average number of capture transitions per test vector and the maximum number of capture transitions for each circuit for the first and second captures are shown under "*# of Vec.*", "*Fault Cov.*", "*Ave. Trans.*", and "*Max. Trans.*", respectively.

In order to conduct DC-LCP *X*-filling in the proposed dynamic compaction as shown in Fig. 9, it is necessary to set an *X-Limit* for defining the percentage of original *X*-bits in an initial test cube to be allowed for detecting secondary target faults. When this *X-Limit* is reached, dynamic compaction switches immediately to DC-LCP *X*-filling for the remaining *X*-bits to reduce capture power dissipation. Generally, the smaller the *X-Limit*, the more test vectors will be generated since fewer secondary target faults can be detected. However, the smaller the *X-Limit*, the higher the capture transition reduction effect of DC-LCP *X*-filling will be since more *X*-bits are available for this purpose.

Table 3 Results for Random *X*-Filling

Circuit	# of Vec.	Fault Cov. (%)	1st Capture		2nd Capture	
			Ave. Trans.	Max. Trans.	Ave. Trans.	Max. Trans.
s1423	112	86.1	24	50	15	32
s5378	214	88.9	89	106	49	72
s9234	342	81.5	73	107	44	76
s13207	330	79.8	264	324	196	281
s15850	187	69.9	171	251	126	182
s35932	45	84.7	897	991	809	964
s38417	221	98.0	504	668	407	506
s38584	410	82.6	437	813	331	766

Extensive experiments on ISCAS'89 circuits have revealed an interesting fact that the number of test vectors will not grow too much if *X-Limit* is greater than a certain value, which can be as small as 20%. Fig. 17 shows partial experimental results on three largest ISCAS'89. This fact is very useful in achieving a good capture transition reduction effect while keeping a test vector set compact.

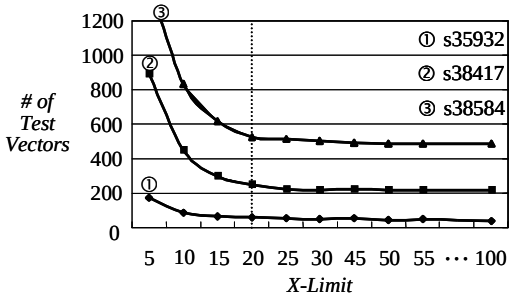


Fig. 17 Impact of *X-Limit*.

Table 4 shows the results of DC-LCP *X*-filling in the proposed dynamic compaction flow (*X*-Limit = 20%) as shown in Fig. 9. The meanings of the columns in Table 4 are the same as Table 3, except that Table 4 also shows CPU time.

Comparing the experimental results for random *X*-filling in Table 3 and for DC-LCP *X*-filling in Table 4, it can be seen that on average, DC-LCP *X*-filling achieved **52.4%** and **41.5%** reduction on average and maximum capture transition for the first capture, respectively, and **39.7%** and **24.6%** reduction on average and maximum capture transition for the second capture, respectively, in a balanced manner and without any fault coverage degradation. The

cost was a slightly larger test vector set. It was possible to keep the number of test vectors unchanged by increasing *X-Limit*, which led to roughly 1/4 lower reduction effect.

Table 4 Results for DC-LCP *X*-Filling

Circuit	# of Vec.	Fault Cov. (%)	1st Capture		2nd Capture		CPU Time (sec.)
			Ave. Trans.	Max. Trans.	Ave. Trans.	Max. Trans.	
s1423	135	86.1	17	32	9	27	1
s5378	248	78.8	42	63	32	57	7
s9234	350	81.5	38	71	37	74	88
s13207	356	79.8	138	202	144	250	66
s15850	220	69.9	56	119	63	116	236
s35932	72	84.7	295	714	297	716	146
s38417	227	98.0	273	421	263	404	149
s38584	444	82.6	165	269	151	274	1427

5. Conclusions

This paper addressed a new test power reduction problem, i.e. reducing capture power dissipation to lower the risk of yield loss caused by faulted test responses in capture mode for scan-based at-speed testing with the broadside clocking scheme. A novel double-capture low-capture-power (DC-LCP) *X*-filling method has been proposed for algorithmically assigning 0's and 1's to *X*-bits in a test cube in order to reduce the switching activity at FFs for the resulting fully-specified test vector. Experimental results have shown the effectiveness of the method, which can be easily incorporated into any test generation flow to achieve capture power reduction without any area, timing, or fault coverage impact in reasonably short CPU time.

Further evaluation is in progress to assess the effect of DC-LCP *X*-filling directly through power consumption instead of switching activity at FFs although it is evident that they have a strong correlation. Research for an algorithmic method to determine a proper value for *X-Limit* in order to balance test set size reduction and capture power reduction in dynamic compaction is also under way.

References

[1] S. Savir and S. Patil, "On Broad-Side Delay Test," *Proc. VLSI Test Symp.*, pp. 284-290, 1994.

[2] X. Lin, R. Press, J. Rajski, P. Reuter, T. Rinderknecht, B. Swanson, and N. Tamarapalli, "High-Frequency, At-Speed Scan Testing," *IEEE Design & Test of Computers*, pp. 17-25, September-October, 2003.

[3] J. Gatej, L. Song, C. Pyron, R. Raina, and T. Munns, "Evaluating ATE Features in Terms of Test Escape Rates and Other Cost of Test Culprits," *Proc. Intl. Test Conf.*, pp. 1040-1048, 2002.

[4] P. Nigh, D. Vallett, A. Patel, J. Wright, F. Motika, D. Forlenza, R. Kurtulik, and W. Chong, "Failure Analysis of Timing and IDDq-only Failures from the SEMATECH Test Methods Experiment ," *Proc. Intl. Test Conf.*, pp. 43-52, 1998.

[5] N. Tendolkar, R. Raina, R. Woltenberg, X. Lin, B. Swanson, and G. Aldrich, "Novel Techniques for Achieving High At-Speed Transition Fault Test Coverage for Motorola's Microprocessors Based on PowerPC™ Instruction Set Architecture," *Proc. VLSI Test Symp.*, pp. 3-8, 2002.

[6] P. Girard, "Survey of Low-Power Testing of VLSI Circuits," *IEEE Design & Test of Computers*, vol. 19, no. 3, pp. 82-92, May/June 2002.

[7] N. Nicolici and B. Al-Hashimi, *Power-Constrained Testing of VLSI Circuits*, Kluwer Academic Publishers, 2003.

[8] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," *Proc. VLSI Test Symp.*, pp. 4-9, 1993.

[9] R. Chou, K. Saluja, and V. Agrawal, "Scheduling Tests for VLSI Systems under Power Constraints," *IEEE Trans. on VLSI Systems*, vol. 5, no. 6, pp. 175-185, 1997.

[10] S. Wang and S. Gupta, "ATPG for Heat Dissipation Minimization during Test Application," *IEEE Trans. on Computers*, vol. 47, no. 2, pp. 256-262, 1998.

[11] F. Corno, P. Prinetto, M. Redaudengo, and M. Reorda, "A Test Pattern Generation Methodology for Low Power Consumption," *Proc. VLSI Test Symp.*, pp. 35-40, 2000.

[12] R. Sankaralingam, R. Oruganti, and N. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation," *Proc. VLSI Test Symp.*, pp. 35-40, 2000.

[13] S. Kajihara, K. Ishida, and K. Miyase, "Test Vector Modification for Power Reduction during Scan Testing," *Proc. VLSI Test Symp.*, pp. 160-165, 2002.

[14] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, "Techniques for Minimizing Power Dissipation in Scan and Combinational Circuits during Test Application," *IEEE Trans. on Computer-Aided Design*, vol. 17, no. 12, pp. 1325-1333, 1998.

[15] 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.

[16] A. Chandra and K. Chakrabarty, "Reduction of SoC Test Data Volume, Scan Power and Testing Time Using Alternating Run-Length Codes," *Proc. Design Automation Conf.*, pp. 673-678, 2002.

[17] A. Hertwig and H. Wunderlich, "Low Power Serial Built-In Self-Test," *Proc. European Test Workshop*, pp. 49-53, 1998.

[18] R. Sankaralingam, R. Oruganti, and N. Touba, "Reducing Power Dissipation during Test Using Scan Chain Disable," *Proc. VLSI Test Symp.*, pp. 319-324, 2001.

[19] T. Yoshida and M. Watari, "MD-Scan Method for Low Power Scan Testing," *Proc. Intl. Test Conf.*, pp. 480-487, 2003.

[20] 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.

[21] J. Saxena, K. Butler, and L. Whetsel, "A Scheme to Reduce Power Consumption during Scan Testing," *Proc. Intl. Test Conf.*, pp. 670-677, 2001.

[22] O. Sinanoglu and A. Orailoglu, "Scan Power Minimization through Stimulus and Response Transformations," *Proc. Design, Automation and Test in Europe*, pp. 404-409, 2004.

[23] S. Gerstendoerfer and H. Wunderlich, "Minimized Power Consumption for Scan-Based BIST," *Proc. Intl. Test Conf.*, pp. 77-84, 1999.

[24] S. Wang, "Generation of Low-Power-Dissipation and High-Fault Coverage Patterns for Scan-Based BIST," *Proc. Intl. Test Conf.*, pp. 834-843, 2002.

[25] 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.

[26] 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.

[27] R. Sankaralingam and N. A. Touba, "Controlling Peak Power During Scan Testing," *Proc. VLSI Test Symp.*, pp. 153-159, 2002.

[28] X. Wen, H. Yamashita, S. Kajihara, L.-T. Wang, K. Saluja, and K. Kinoshita, "On Low-Capture-Power Test Generation for Scan Testing," *Proc. VLSI Test Symp.*, pp. 265-270, 2005.