A Capture-Safe Test Generation Scheme for At-Speed Scan Testing

著者	Wen X., Miyase K., Kajihara S., Furukawa H., Yamato Y., Takashima A., Noda K., Ito H., Hatayama K., Aikyo T., Saluja K. K.
journal or publication title	2008 13th European Test Symposium
page range	55-60
year	2008-07-02
URL	http://hdl.handle.net/10228/00007592

doi: info:doi/10.1109/ETS.2008.13

A Capture-Safe Test Generation Scheme for At-Speed Scan Testing

X. Wen [1], K. Miyase [1], S. Kajihara [1], H. Furukawa [1], Y. Yamato [1], A. Takashima [1]
K. Noda [2], H. Ito [2], K. Hatayama [2], T. Aikyo [2], and K. K. Saluja [3]

[1] Dept. of CSE, Kyushu Institute of Technology, Iizuka 820-8502, Japan
[2] Semiconductor Technology Academic Research Center, Yokohama 222-0033, Japan
[3] Dept. of ECE, University of Wisconsin - Madison, Madison, WI 53706, USA

Abstract

Capture-safety, defined as the avoidance of any timing error due to unduly high launch switching activity in capture mode during at-speed scan testing, is critical for avoiding test-induced yield loss. Although point techniques are available for reducing capture IR-drop, there is a lack of complete capture-safe test generation flows. The paper addresses this problem by proposing a novel and practical capture-safe test generation scheme, featuring (1) reliable capture-safety checking and (2) effective capture-safety improvement by combining X-bit identification & X-filling with low launch-switching-activity test generation. This scheme is compatible with existing ATPG flows, and achieves capture-safety with no changes in the circuit-under-test or the clocking scheme.

1. Introduction

1.1 Background

Scan testing has been the most widely adopted test strategy, which uses the full-scan methodology for circuit design, automatic test pattern generation (ATPG) for test data creation, and automatic test equipment (ATE) for test execution. Recently, ***at-speed scan testing*** has become mandatory in achieving high test quality for deep submicron (DSM) circuits by detecting timing-related defects [1].

Fig. 1 shows the concept of at-speed scan testing based on the ***launch-off-capture*** (***LOC***) clocking scheme [1]. In ***shift*** mode (SE = 1), a test vector is applied by operating scan chains as shift registers over multiple shift clock pulses, with S_L as the last shift clock pulse. In ***capture*** mode (SE = 0), two capture pulses are applied: C_1 (***launch capture***) for launching a transition at the start-point of a path and C_2 (***response capture***) for capturing the circuit response to the launched transition at the end-point of the path. Note that the ***test cycle*** is equal to the rated clock cycle in at-speed scan testing, which is very short for a high-speed design.

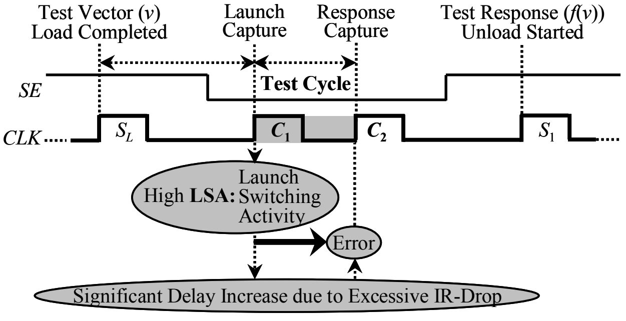

Fig. 1 Importance of Capture-Safety in At-Speed Scan Testing.

Conventionally, ***high quality*** and ***low cost*** have been two basic requirements for scan test vectors. Test quality can be improved by increasing fault coverage, using realistic fault models, and testing for small-delay defects; while test cost can be reduced by test compression. Over the past decade, ***low test power***, especially in the context of heat dissipation, has also become an important requirement [2, 3].

Recently, a new requirement, called ***capture-safety***, has emerged and is rapidly becoming mandatory for at-speed scan test vectors. It is also referred to as supply-voltage-noise-safety [4] or power-safety [5]. As shown in Fig. 1, the launch capture (C_1) may cause high ***launch switching activity*** (***LSA***). This may lead to excessive IR-drop, which significantly increases path delay so that timing errors occur at C_2 only during at-speed scan testing [6]. Such test-induced yield loss is rapidly worsening [7] due to shrinking feature sizes, growing gate counts, increasing clock frequencies, and decreasing supply voltages. Therefore, the capture-safety of an at-speed scan test vector v needs to be guaranteed, i.e., the delay increased in the test cycle by the IR-drop due to the LSA of v at C_1 should not cause any timing error at C_2.

1.2 Related Previous Work

Two basic tasks, ***capture-safety checking*** and ***capture-safety improvement***, are needed to guarantee capture-safety for at-speed scan test vectors. The former is for judging whether a test vector is capture-safe, and the latter is for making a capture-unsafe test vector capture-safe. Previous methods related to the basic tasks are summarized as follows:

1.2.1 Capture-Safety Checking Methods

The ultimate capture-safety checking should be based on the path delay increase in the test cycle during at-speed scan testing [8]. Since the cost of directly analyzing path delay impact is prohibitive, realistic capture-safety checking often uses indirect metrics to estimate IR-drop [4] or launch switching activity [5]. For example, the average capture power can be estimated using ***switching cycle average power*** (***SCAP***) metric [4], which has a good correlation with the actual IR-drop. However, SCAP calculation is computation-expensive since physical design information is needed. On the other hand, launch switching activity can be estimated by toggle constraint metrics, such as ***global toggle constraint*** (***GTC***), ***global instantaneous toggle constraint*** (***GITC***), and ***regional instantaneous toggle constraint*** (***RITC***) [5]. These gate-level metrics are computation-efficient, but correlation with IR-drop and path delay impact may be weak.

1.2.2 Capture-Safety Improvement Methods

There are several techniques for alleviating problems that make a test vector capture-unsafe. Most of them try to reduce launch switching activity (i.e. LSA in Fig. 1), through ***circuit modification*** [9] or ***test data manipulation*** [5,10-12].

• Circuit Modification

The number of simultaneously-switching flip-flops (FF′s) can be reduced through ***partial capture*** by modifying scan chains, clock control, or FF design [9]. However, this approach may suffer from possible fault coverage loss, test data increase, and/or physical design difficulty.

• Test Data Manipulation

Launch switching activity can also be reduced by using the following techniques to manipulate the content of test data. This approach has no impact on design or performance.

① LCP (Low-Capture-Power) ATPG

These techniques reduce launch switching activity through carefully determining logic values (0 or 1) for fault detection during test generation, by adding more constraints to conventional ATPG algorithms [4] or by employing new ATPG algorithms [5, 10]. However, these techniques may result in significant test data increase.

② LCP (Low-Capture-Power) X-Filling

These techniques reduce launch switching activity through properly assigning logic values to don’t-care bits (*X*-bits) [11, 12], which are left out from test generation or identified from a fully-specified test set by *X*-identification [13, 14]. The advantage of LCP *X*-filling is that it improves capture-safety without design change, fault coverage loss, and test data increase if used together with *X*-identification.

③ LSP (Low-Shift-Power) X-Filling

Some LSP *X*-filling techniques may accidentally reduce launch switching activity in capture mode [4, 15], although they are originally intended for reducing switching activity in shift mode. For example, “0-fill” increases 0′s in a test vector, which helps block switching activity in a circuit mostly composed of AND-type gates. However, it is obvious that the effect of LSP *X*-filling for capture-safety improvement may be unpredictable and insignificant.

④ Target Fault Restriction

It is observed that the launch switching activity of a test vector increases with the number of faults it detects. Thus, launch switching activity can be reduced by restricting ATPG to target faults in a limited number of blocks each time [4]. This technique is easily applicable to any ATPG flow, but may increase the number of final test vectors.

1.2.3 Limitations

It is clear that the capture-safety of at-speed scan test vectors can only be guaranteed by properly combining capture-safety checking and capture-safety improvement into a tightly integrated test generation flow. However, previous schemes proposed for achieving capture-safety generally suffer from the following two major limitations:

Limitation-1: *Unsatisfactory Capture-Safety Checking*

Previous metrics for capture-safety checking are either too computation-costly to be practical or too simplistic to have good correlation with actual IR-drop. Clearly, this limitation affects the validity of capture-safe test generation [4, 5].

Limitation-2: *Insufficient Capture-Safety Improvement*

There are very few test generation schemes proposed for achieving capture-safety. All of them first conduct capture-safety checking to identify capture-unsafe vectors and from which further identify faults detected only by these vectors. Then, such techniques as LCP ATPG [5], LSP *X*-filling [4, 15], and target fault restriction [4] are used to improve the chance for a vector to become capture-safe. However, the effectiveness of these techniques may be insufficient, in terms of fault coverage loss or test data inflation.

1.3 Contributions and Paper Organization

Therefore, there is a strong need for ***better capture-safety checking metrics*** and ***better capture-safety improvement flows***. These issues are addressed in this paper by a novel and practical capture-safe test generation scheme, with the following technical contributions as illustrated in Fig. 2:

(1) Metrics for Capture-Safety Checking

Four gate-level easy-to-compute metrics are proposed to estimate the launch switching activity (LSA) of a test vector in both ***temporal*** (total / instantaneous) and ***spatial*** (global / regional) manners. This set of metrics is a super-set of the one used in [5], and provides a more accurate LSA-profile.

(2) Hybrid Capture-Safety Improvement

Capture-safety is first improved by *X*-identification & low-LSA *X*-filling so as to reduce launch-switching activity without test data increase. If capture-unsafe vectors still remain, direct low-LSA test generation is then conducted so as to effectively improve capture-safety deterministically.

(3) Focused *X*-Identification

X-identification is ***focused***, in that it is conducted exactly on capture-unsafe initial test vectors. This greatly improves the effectiveness and efficiency of the follow-up *X*-filling.

(4) LSA-Based Dynamic Compaction

Any ATPG system can be readily extended for low-LSA test generation by introducing ***LSA-guided*** dynamic compaction. This new technique features (***a***) capture-safety checking for each test cube and (***b***) secondary fault selection based on the LSA-profile of the current test cube. This greatly increases the chance for one capture-safe test vector to detect a larger number of faults, thus minimizing test data inflation, if any.

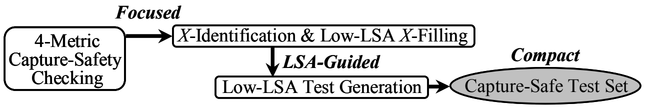
Focused
4-Metric Capture-Safety Checking
X-Identification & Low-LSA X-Filling
LSA-Guided
Compact
Low-LSA Test Generation
Capture-Safe Test Set

Fig. 2 Technical Contributions.

The paper is organized as follows: Section 2 outlines the new scheme. Section 3, 4, and 5 present capture-safety checking, *X*-identification & *X*-filling, and low-LSA test generation for capture-safety improvement, respectively. Section 6 shows experimental results, and Section 7 concludes the paper.

2. Capture-Safe Test Generation Scheme

The new capture-safe test generation scheme is outlined in Fig. 3. Its goal is to achieve capture-safety with no fault coverage loss after excluding capture-undetectable faults, less test data inflation if any, and minimal ATPG change.

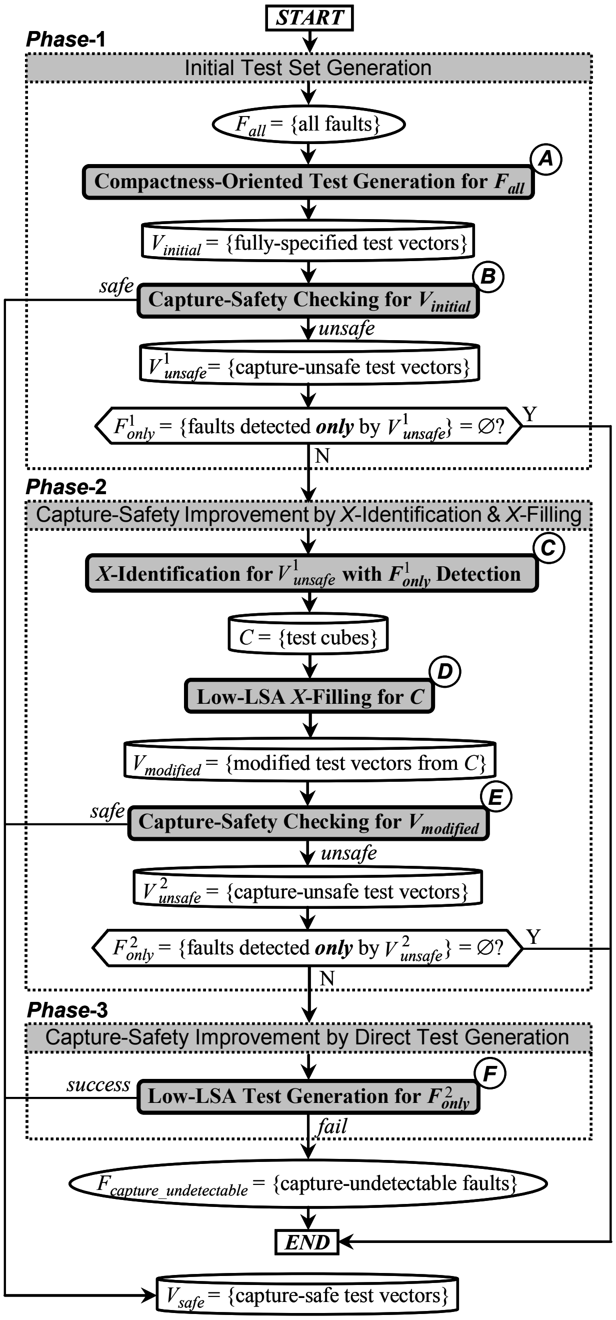
START
Phase-1
Initial Test Set Generation
F_{all} = {all faults}
A
Compactness-Oriented Test Generation for F_{all}
$V_{initial}$ = {fully-specified test vectors}
B
safe
Capture-Safety Checking for $V_{initial}$
unsafe
V^1_{unsafe}= {capture-unsafe test vectors}
F^1_{only} = {faults detected *only* by V^1_{unsafe}} = ∅?
Y
N
Phase-2
Capture-Safety Improvement by X-Identification & X-Filling
C
X-Identification for V^1_{unsafe} with F^1_{only} Detection
C = {test cubes}
D
Low-LSA X-Filling for C
$V_{modified}$ = {modified test vectors from C}
E
safe
Capture-Safety Checking for $V_{modified}$
unsafe
V^2_{unsafe} = {capture-unsafe test vectors}
F^2_{only} = {faults detected *only* by V^2_{unsafe}} = ∅?
Y
N
Phase-3
Capture-Safety Improvement by Direct Test Generation
F
success
Low-LSA Test Generation for F^2_{only}
fail
$F_{capture\_undetectable}$ = {capture-undetectable faults}
END
V_{safe}= {capture-safe test vectors}

Fig. 3 Overview of Capture-Safe Test Generation Scheme.

• **Phase-1**: First, a compact initial test set, $V_{initial}$, is generated by any ATPG with maximum dynamic compaction and random-fill (***A***). Then, ***capture-safety checking*** (***B***) identifies the capture-unsafe vectors (V^1_{unsafe}), and thus the faults (F^1_{only}) detected only by the vectors. Details on new metrics for capture-safety checking are presented in Section 3.

• **Phase-2**: If F^1_{only} is not empty, the flow proceeds to Phase-2, where capture-safety is improved without test data inflation or fault coverage loss. This is achieved as follow: [**Step-1**: ***Focused X-identification*** (***C***) is conducted on the capture-unsafe test set (V^1_{unsafe}) to extract don't-care bits (*X*-bits) while guaranteeing that all faults in F^1_{only} are still detected by specified logic bits.] [**Step-2**: The identified *X*-bits are re-filled with proper logic values so that launch-switching-activity (LSA) is lowered (***D***).] Then, capture-safety checking (***E***) is conducted to identify the set of capture-unsafe vectors (V^2_{unsafe}), and from which the set of faults (F^2_{only}) detected only by the vectors. Details of focused *X*-identification and low-LSA *X*-filling are presented in Section 4.

• **Phase-3**: If F^2_{only} is not empty, the flow proceeds to Phase-3, where capture-safety is improved by direct low-LSA test generation (***F***). ***LSA-guided dynamic compaction***, featuring (1) capture-safety checking on test cubes and (2) LSA-profile-based secondary fault selection, is introduced so as to detect a larger number of faults by one capture-safe vector. Details of this new technique are presented in Section 5.

3. Capture-Safety Checking

As shown in Fig. 1, the capture-safety of an at-speed scan test vector depends on its launch switching activity (LSA). In this section, we first describe the power grid and circuit model, then define new metrics for LSA-profiling, and finally present their application in capture-safety checking.

3.1 Power Grid and Circuit Model

A popular model for expressing the relation between the nodes (FF's and gates) in a circuit with its power grid is shown in Fig. 4. A similar model is used in [5]. In such a model, a ***feed-region*** is a group of nodes that share the same power via, and a circuit is composed of such feed-regions.

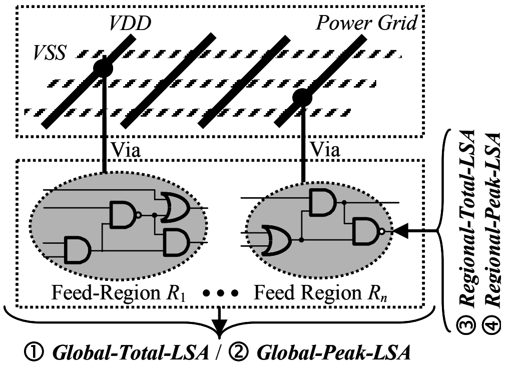
VDD
Power Grid
VSS
Via
Via
Feed-Region R_1 • • • Feed Region R_n
③ Regional-Total-LSA
④ Regional-Peak-LSA
① Global-Total-LSA / ② Global-Peak-LSA

Fig. 4 Power Grid and Circuit Model.

It is evident that ***spatial*** (not only the whole circuit but also each feed-region) and ***temporal*** (not only the whole test cycle but also each time instant in the test cycle) requirements must be considered so as to obtain a good profile of launch switching activity (LSA). This makes it reasonable to use four metrics (①~④ in Fig. 4) for capture-safety checking.

3.2 Metrics for LSA-Profiling

Let v be an input vector for a circuit with n feed-regions (R_1, R_2, ..., R_n), and suppose that a total of m time instants (T_1, T_2, ..., T_m) are used in timing-based logic simulation for v. The ***launch switching activity of v for feed-region R_i at time T_j***, denoted by ***LSA(v, R_i, T_j)***, is defined as follows:

$$LSA(v, R_i, T_j) = \sum_{k=1}^{p}((f_k + 1) \times s_k(T_j))$$

where p is the number of nodes in feed-region R_i, f_k is the number of fanout branches from node k (k = 1, 2, ..., p), and $s_k(T_j)$ is the transition probability of the output of node k at time T_j due to the launch capture (i.e. C_1 in Fig. 1).

Note that $s_k(T_j)$ can be easily obtained by 2-valued logic simulation for a fully-specified test vector. For a partially-specified test cube with X-bits, 50% is assigned as the 0/1 probability of each X-bit and probability propagation [11] is conducted to efficiently obtain transition probability $s_k(T_j)$.

Once $LSA(v, R_i, T_j)$ for each feed-region R_i (i = 1, 2, ..., n) at each time instant T_j (j = 1, 2, ..., m) is obtained, four new metrics for profiling the LSA (Launch Switching Activity) of the at-speed scan test vector v can be defined as follows:

① **Global Total LSA (*$LSA_{GT}(v)$*)**

$$LSA_{GT}(v) = \sum_{i=1}^{n}\sum_{j=1}^{m} LSA(v, R_i, T_j)$$

② **Global Peak LSA (*$LSA_{GP}(v)$*)**

$$LSA_{GP}(v) = max(\sum_{i=1}^{n} LSA(v, R_i, T_1), ..., \sum_{i=1}^{n} LSA(v, R_i, T_m))$$

③ **Regional Total LSA (*$LSA_{RT}(v, R_i)$*)**

$$LSA_{RT}(v, R_i) = \sum_{j=1}^{m} LSA(v, R_i, T_j)$$

④ **Regional Peak LSA (*$LSA_{RP}(v, R_i)$*)**

$$LSA_{RP}(v, R_i) = max(LSA(v, R_i, T_1), ..., LSA(v, R_i, T_m))$$

$LSA_{GT}(v)$ and $LSA_{GP}(v)$ are for profiling the LSA in the whole circuit accumulatively and instantaneously, respectively; while $LSA_{RT}(v, R_i)$ and $LSA_{RP}(v, R_i)$ are for profiling the LSA in each feed-region R_i accumulatively and instantaneously, respectively. They have the following characteristics:

(1) **Efficiency**: Only gate-level information and unit-delay logic simulation are needed.

(2) **Accuracy**: The weight of a node is introduced to better approximate the capacitive load at its output.

(3) **Completeness**: The total LSA in a feed-region is also profiled by a new metric, i.e. $LSA_{RT}(v, R_i)$, in contrast with the metrics proposed in [5].

(4) **Flexibility**: The LSA of a partially-specified test cube can also be profiled. This makes the new metrics useful in guiding secondary fault selection for a test cube during dynamic compaction. Section 5.2 provides more details.

3.3 Capture-Safety Checking

Definition 1: A fully-specified test vector v is said to be ***capture-safe*** if the following criteria are satisfied:

$$LSA_{GT}(v) \le Limit\_LSA_{GT}$$
$$LSA_{GP}(v) \le Limit\_LSA_{GP}$$
$$LSA_{RT}(v, R_i) \le Limit\_LSA_{RT}(R_i) \quad (i = 1, ..., n)$$
$$LSA_{RP}(v, R_i) \le Limit\_LSA_{RP}(R_i) \quad (i = 1, ..., n)$$

where $Limit\_LSA_{GT}$, $Limit\_LSA_{GP}$, $Limit\_LSA_{RT}(R_i)$, and $Limit\_LSA_{RP}(R_i)$ are limits for corresponding metrics (i = 1, ..., n).

Clearly, $Limit\_LSA_{GT}$, $Limit\_LSA_{GP}$, $Limit\_LSA_{RT}(R_i)$, and $Limit\_LSA_{RP}(R_i)$ (i=1, ..., n) need to be properly set in order to make capture-safety checking neither over-optimistic nor over-pessimistic. Limit setting depends on what type of testing is intended. Generally, the manufacturing test power limits are allowed to be 2X functional power limits, while field test power is required to be as low as worst-case functional power [7]. Based on power characteristics, the limits for capture-safety metrics can be readily obtained [4], by assuming a functional toggle rate or simulating functional vectors. Test vectors may also be used for this purpose, but appropriate adjustment needs to be conducted accordingly [5].

4. Capture-Safety Improvement by *X*-Identification & Low-LSA *X*-Filling

As shown in Fig. 3, capture-safety checking conducted on the initial test set identifies a set of capture-unsafe test vectors (V^1_{unsafe}), and thus the set of faults (F^1_{only}) detected only by the vectors. If $F^1_{only} \ne \emptyset$, capture-safety improvement is conducted in Phase-2, featuring (***A***) focused X-identification followed by low-LSA (Launch-Switching-Activity) X-filling and (***B***) multi-round execution, as illustrated in Fig. 5.

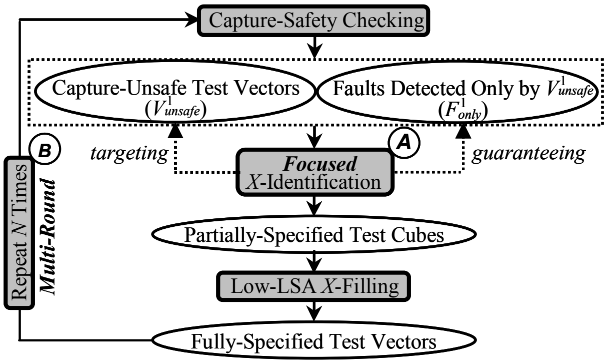
Capture-Safety Checking
Capture-Unsafe Test Vectors (V^1_{unsafe})
Faults Detected Only by V^1_{unsafe} (F^1_{only})
B
targeting
A
guaranteeing
Focused X-Identification
Repeat N Times
Multi-Round
Partially-Specified Test Cubes
Low-LSA X-Filling
Fully-Specified Test Vectors

Fig. 5 Focused *X*-Identification and Multi-Round Execution.

4.1 Focused *X*-Identification and Low-LSA *X*-filling

X-identification is to identify don't-care bits (X-bits) from a set of fully-specified test vectors, while maintaining such properties as fault coverage [13, 14], small-delay-defect detecting capability [12], etc. Phase-2 uses ***focused X-identification***, which (1) targets the capture-unsafe test set (V^1_{unsafe}), instead of the initial test set ($V_{initial}$), and (2) guarantees the detection of the faults (F^1_{only}) detected only by V^1_{unsafe}, instead of all faults (F_{all}). Since $V^1_{unsafe} \subseteq V_{initial}$ and $F^1_{only} \subseteq F_{all}$, focused X-identification results in more X-bits from capture-unsafe test vectors, thus increasing the chance of making them capture-safe by low-LSA X-filling.

Low-LSA X-filling is used to assign proper logic values for *X*-bits in a partially-specified test cube so as to create a fully-specified test vector with low launch switching activity. Many LCP (Low-Capture-Power) *X*-filling methods, such as *preferred-fill* [11] and *JP-fill* [12], can serve this purpose.

4.2 Multi-Round Execution

The major advantage of *X*-identification and low-LSA *X*-filling in Phase-2 is that capture-safety is improved without test data inflation. Thus, it is beneficial to repeat Phase-2 as long as capture-safety can still be significantly improved. This is because the next phase (Phase-3) is based on direct test generation, which is time-consuming and may result in more test vectors. This "*multi-round execution*", as illustrated in Fig. 5, helps improve capture-safety more efficiently and with less test data inflation, if any.

5. Capture-Safety Improvement by Direct Low-LSA Test Generation

If capture-safety can no longer be significantly improved in Phase-2 by repeating *X*-identification & *X*-filling, Phase-3 is then conducted to further improve capture-safety by direct low- LSA (Launch Switching Activity) test generation.

5.1 Overview of Low-LSA Test Generation

Fig. 6 illustrates the overall flow of low-LSA test generation.

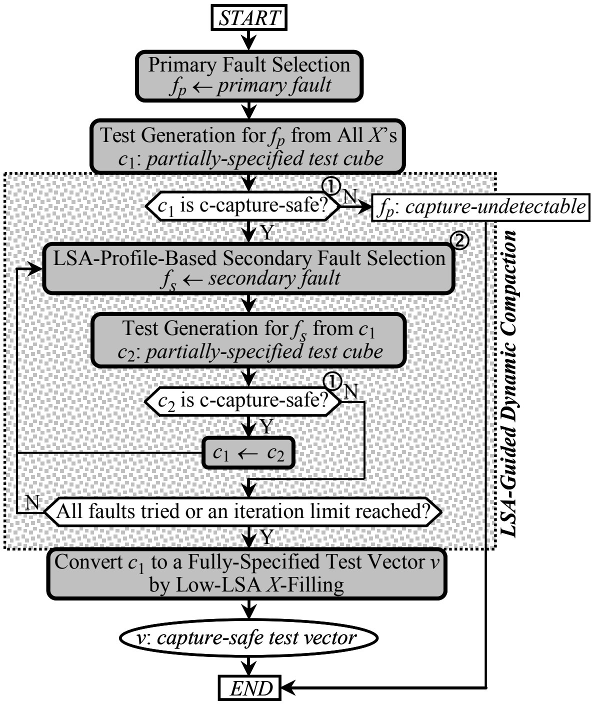
START
Primary Fault Selection
f_p ← primary fault
Test Generation for f_p from All X's
c_1: partially-specified test cube
①
c_1 is c-capture-safe?
N
f_p: capture-undetectable
Y
②
LSA-Profile-Based Secondary Fault Selection
f_s ← secondary fault
Test Generation for f_s from c_1
c_2: partially-specified test cube
①
c_2 is c-capture-safe?
N
Y
c_1 ← c_2
N
All faults tried or an iteration limit reached?
Y
Convert c_1 to a Fully-Specified Test Vector v by Low-LSA X-Filling
v: capture-safe test vector
END
LSA-Guided Dynamic Compaction

Fig. 6 Overview of Low-LSA Test Generation.

Low-LSA test generation is similar to conventional ATPG, except that it uses ***LSA-guided dynamic compaction***. That is, capture-safety checking is conducted for intermediate test cubes (①), and secondary fault selection is based on the LSA-profile of the current test cube (②). This unique dynamic compaction technique increases the chance for a capture-safe test vector to detect a larger number of faults.

5.2 LSA-Guided Dynamic Compaction

5.2.1 Capture-Safety Checking for Test Cubes

In LSA-guided dynamic compaction, capture-safety checking needs to be conducted for partially-specified test cubes with *X*-bits. This is made possible by the following definition:

Definition 2: A partially-specified test cube *c* is said to be ***conditionally capture-safe***, denoted by ***c-capture-safe***, if the fully-specified test vector obtained by conducting low-LSA *X*-filling on *c* is capture-safe.

That is, a c-capture-safe test cube can be converted into a fully-specified capture-safe test vector by low-LSA *X*-filling. For such a test cube, dynamic compaction is continued to check whether it can be extended to detect more faults.

5.2.2 LSA-Profile-Based Secondary Fault Selection

In LSA-guided dynamic compaction, once a test cube c_1 is found to be c-capture-safe, a secondary fault f_s is selected for further test generation trial. Suppose that c_1 is extended to c_2 for detecting f_s. Obviously, this trial fails if c_2 is found to be c-capture-unsafe. In order to increase the chance of c_2 being c-capture-safe so as to detect more faults by extending c_1, the secondary fault f_s is selected by using a unique technique based on the LSA-profile of the current test cube c_1.

The LSA-profile of the test cube c_1, with respected to the four metrics defined in Section 3.2, can be expressed as:

***LSA_Profile*(*c*₁)** = <$LSA_{GT}(c_1)$,
$LSA_{GP}(c_1)$,
<$LSA_{RT}(c_1, R_1)$, ..., $LSA_{RT}(c_1, R_n)$>,
<$LSA_{RP}(c_1, R_1)$, ..., $LSA_{RP}(c_1, R_n)$>>

From the capture-safety limits ($Limit\_LSA_{GT}$, $Limit\_LSA_{GP}$, $Limit\_LSA_{RT}(R_1)$, ..., $Limit\_LSA_{RT}(R_n)$, $Limit\_LSA_{RP}(R_1)$, ..., $Limit\_LSA_{RP}(R_n)$), the ***slack*** of an entry *e* in $LSA\_Profile(c_1)$ can be calculated by $slack(e) = (l - e)*100/l$, where *l* is the capture-safety limit for *e*. Clearly, the LSA-profile entry with the least slack, called the ***risky entry***, should be considered with the highest priority in secondary fault selection.

Generally, there are four types of risky entry: ***global-total***, ***global-peak***, ***regional-total***, and ***regional-peak***, and they necessitate different strategies for secondary fault selection. Due to page limitation, only the basic selection idea for the regional-total type is described in the following:

Suppose that the risky entry is the regional total launch switching activity of the feed-region R_i, i.e. $LSA_{RT}(c_1, R_i)$. As illustrated in Fig. 7, it is preferable to select such a secondary fault f_s that its activation and propagation cones do not overlap with R_i. Clearly, this reduces the possibility of new switching activity in R_i when c_1 is extended to detect f_s.

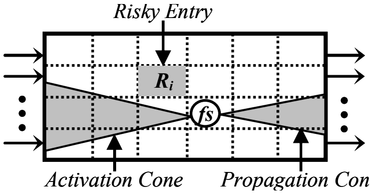

Fig. 7 LSA-Profile-Based Secondary Fault Selection.

6. Experimental Results

6.1 Validation of Capture-Safety Metrics

First, layout was conducted on an industrial circuit (90nm / 1.2V / 50K gates), and V_{DD} IR-drop was analyzed by RedHawk^{TM} for three different transition delay test sets, each with 318 vectors. A typical core V_{DD} IR-drop budget of 1.5%, i.e. 0.018V V_{DD} IR-drop limit [16], was used to determine the capture-safety of each test vector. Fig. 8 shows the results.

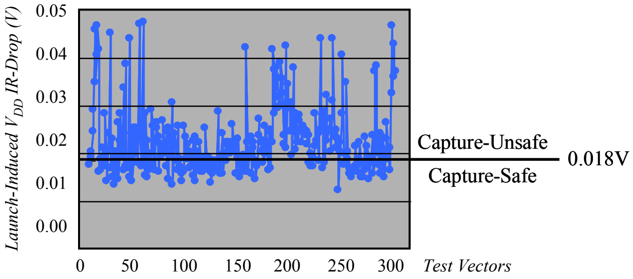

Fig. 8 Launch-Induced IR-Drop and Capture-Safety Checking.

Next, capture-safety was checked by the new metrics. It was verified that no capture-unsafe vector determined by IR-drop analysis was wrongly declared to be capture-safe by metric-based analysis. However, metric-based analysis made 0.3% ~ 16.0% over-pessimistic calls. This indicates that the accuracy of the metrics still needs to be improved.

6.2 Evaluation of Capture-Safe Test Generation Flow

Experiments were conducted using a workstation (2.9GHz CPU / 16GB memory) on 10 largest benchmark circuits (5 from ISCAS'89 and 5 from ITC'99) to evaluate the proposed capture-safe test generation flow. In capture-safety checking, each feed-region was assumed to have 20 gates, LSA-profiling metrics were calculated with unit-delay simulation, and the metric limits were set to 90% of peak values of the initial test sets [5]. The transition delay fault model was used.

Table 1 Results of Capture-Safe Test Generation

Factor / Circuit	*Init. Test Set*		*Init. Fault Cov.*	*XID & X-Fill*		*Test Gen.*		*# of Cap.-Undet. Faults*	*# of Final Test Vec.*	*Final Fault Cov.*
	Safe Vec.	*Unsafe Vec.*		*Unsafe Vec.*	*CPU (sec.)*	*New Vec.*	*CPU (sec.)*			
s13207	312	11	79.5	0	30	0	22	0	323	79.5
s15850	195	26	70.2	0	24	0	28	0	221	70.2
s35932	331	6	82.5	0	97	0	168	0	337	82.5
s38417	201	69	98.0	20	125	25	216	3	270	98.0
s38584	241	171	83.9	131	258	189	1954	218	470	83.4
b15	1115	26	89.8	4	203	9	90	1	1146	89.8
b17	1231	19	92.6	4	682	8	642	0	1254	92.6
b20	966	23	89.8	5	203	8	110	15	992	89.8
b21	885	38	89.9	4	176	9	93	2	928	89.9
b22	1086	37	89.9	15	446	25	308	41	1133	89.8

Table 1 shows the results. The capture-safety breakdown of initial test vectors is shown in "*Safe Vec.*" and "*Unsafe Vec.*" under "*Init. Test Set*". First, focused *X*-identification & low-LSA *X*-filling was repeated 3 times on the initial capture-unsafe vectors, and the number of remaining capture-unsafe vectors is shown in "*Unsafe Vec.*" under "*XID & X-Filling*". Then, low-LSA test generation was conducted for the faults detected only by the remaining capture-unsafe vectors, and the number of newly generated test vectors is shown in "*New Vec.*" under "*Test Gen.*". Finally, the information on capture-undetectable faults, final test vectors, and final fault coverage are shown under "*# of Cap.-Undet. Faults*", "*# of Final Test Vec.*", and "*Final Fault Cov.*", respectively.

A ***capture-undetectable*** (***CU***) fault is a fault that none of its test vectors is capture-safe. In the experiments, the average percentage of CU faults was 0.1%. Note that the average test size increased by 1.7% due to low-LSA test generation, and that there was no fault coverage loss excluding CU faults.

7. Conclusions

Capture-safety is required for at-speed scan test vectors to avoid test-induced yield loss. This paper proposed a novel and practical capture-safe test generation scheme, featuring (1) a set of metrics for reliable capture-safety checking and (2) a hybrid flow for effective capture-safety improvement. Its major advantage is that no circuit modification or clocking change is needed. Experiments have validated the metrics and demonstrated the effectiveness of the flow.

Experiments on more industrial circuits are being conducted to fully quantify the correlation between capture-safety and the new metrics. This also helps in reducing over-pessimistic calls in metric-based capture-safety checking.

References

[1] L.-T. Wang, et al., (Editors), *VLSI Test Principles and Architectures: Design for Testability*, Elsevier, 2006.

[2] P. Girard, "Survey of Low-Power Testing of VLSI Circuits," *IEEE Design & Test of Computers*, Vol. 19, No. 3, pp. 82-92, 2002.

[3] N. Nicolici, et al., *Power-Constrained Testing of VLSI Circuits*, Kluwer Academic Publishers, 2003.

[4] N. Ahmed, et al., "Transition Delay Fault Test Pattern Generation Considering Supply Voltage Noise in a SOC Design," *Proc. Design Automation Conf.*, pp. 533-538, 2007.

[5] V. R. Devanathan, et al., "A Stochastic Pattern Generation and Optimization Framework for Variation-Tolerant, Power-Safe Scan Test," *Proc. Intl. Test Conf.*, Paper 13.1, 2007.

[6] J. Saxena, et al., "A Case Study of IR-Drop in Structured At-Speed Testing," *Proc. Int'l Test Conf.*, pp. 1098-1104, 2003.

[7] S. Ravi, "Power-Aware Test: Challenges and Solutions," *Proc. Intl. Test Conf.*, Lecture 2.2, 2007.

[8] J. Wang, et al., "Power Supply Noise in Delay Testing," *Proc. Int'l Test Conf.*, Paper 17.3, 2006.

[9] S. Wang, et al., "A Technique to Reduce Peak Current and Average Power Dissipation in Scan Designs by Limited Capture," *Proc. Asian and South-Pacific Design Automation Conf.*, pp. 810-816, 2007.

[10] X. Wen, et al, "A New ATPG Method for Efficient Capture Power Reduction During Scan Testing," *Proc. VLSI Test Symp.*, pp.58-63, 2006.

[11] S. Remersaro, et al., "Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs," *Proc. Int'l Test Conf.*, Paper 32.2, 2006.

[12] X. Wen, et. Al., "A Novel Scheme to Reduce Power Supply Noise for High-Quality At-Speed Scan Testing," *Proc. Int'l Test Conf.*, Paper 25.1, 2007.

[13] K. Miyase, et al., "XID: Don't Care Identification of Test Patterns for Combinational Circuits," *IEEE Trans. on Computer-Aided Design*, Vol. 23, No. 2, pp. 321-326, Feb. 2004.

[14] A. H. El-Maleh, et al., "An Efficient Test Relaxation Technique for Synchronous Sequential Circuits," *IEEE Trans. on Computer-Aided Design*, Vol. 23, No. 6, pp. 933-940, June 2004.

[15] J. Saxena, et al., "A Scheme to Reduce Power Consumption during Scan Testing," *Proc. Intl. Test Conf.*, pp. 670-677, 2001.

[16] R. Bhooshan, "Novel and Efficient IR-Drop Models for Designing Power Distribution Network for Sub-100nm Integrated Circuits," *Proc. Intl. Symp. On Quality Electronic Design*, pp. 26-28, 2007.