Power-Aware Test Generation with Guaranteed Launch Safety for At-Speed Scan Testing

Wen, Xiaoqing; Enokimoto, Kazunari; Miyase, Kohei; Yamato, Yuta; Kochte, Michael A.; Kajihara, Seiji; Girard, Patrick; Tehranipoor, Mohammad

Proceedings of the 29th IEEE VLSI Test Symposium (VTS'11) Dana Point, California, USA, 1-5 May 2011

doi: http://dx.doi.org/10.1109/VTS.2011.5783778

Abstract: At-speed scan testing may suffer from severe yield loss due to the launch safety problem, where test responses are invalidated by excessive launch switching activity (LSA) caused by test stimulus launching in the at-speed test cycle. However, previous low-power test generation techniques can only reduce LSA to some extent but cannot guarantee launch safety. This paper proposes a novel & practical power-aware test generation flow, featuring guaranteed launch safety. The basic idea is to enhance ATPG with a unique two-phase (rescue & mask) scheme by targeting at the real cause of the launch safety problem, i.e., the excessive LSA in the neighboring areas (namely impact areas) around long paths sensitized by a test vector. The rescue phase is to reduce excessive LSA in impact areas in a focused manner, and the mask phase is to exclude from use in fault detection the uncertain test response at the endpoint of any long sensitized path that still has excessive LSA in its impact area even after the rescue phase is executed. This scheme is the first of its kind for achieving guaranteed launch safety with minimal impact on test quality and test costs, which is the ultimate goal of power-aware at-speed scan test generation.

Preprint

General Copyright Notice

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

This is the author's "personal copy" of the final, accepted version of the paper published by IEEE.[1]

[1] **IEEE COPYRIGHT NOTICE**

©2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Power-Aware Test Generation with Guaranteed Launch Safety for At-Speed Scan Testing

X. Wen [1], K. Enokimoto [1], K. Miyase [1], Y. Yamato [1], M. A. Kochte [1,2], S. Kajihara [1], P. Girard [3], and M. Tehranipoor [4]

[1] Kyushu Institute of Technology, Iizuka, Japan
[2] University of Stuttgart, Stuttgart, Germany
[3] LIRMM, Montpellier, France
[4] University of Connecticut, Storrs, USA

Abstract - At-speed scan testing may suffer from severe yield loss due to the ***launch safety problem***, where test responses are invalidated by excessive ***launch switching activity*** (LSA) caused by test stimulus launching in the at-speed test cycle. However, previous low-power test generation techniques can only reduce LSA to some extent but cannot guarantee launch safety. This paper proposes a novel & practical power-aware test generation flow, featuring ***guaranteed launch safety***. The basic idea is to enhance ATPG with a unique two-phase (***rescue*** & ***mask***) scheme by targeting at the real cause of the launch safety problem, i.e., the excessive LSA in the neighboring areas (namely *impact areas*) around long paths sensitized by a test vector. The ***rescue phase*** is to reduce excessive LSA in impact areas in a focused manner, and the ***mask phase*** is to exclude from use in fault detection the uncertain test response at the endpoint of any long sensitized path that still has excessive LSA in its impact area even after the rescue phase is executed. This scheme is the first of its kind for achieving guaranteed launch safety with minimal impact on test quality and test costs, which is the ultimate goal of power-aware at-speed scan test generation.

Keywords – *test generation*; *test power*; *at-speed scan testing*; *power supply noise*; *launch safety*.

I. Introduction

Shrinking feature sizes and increasing clock frequencies have made timing-related defects a major cause for failing integrated logic circuits. At-speed scan testing is thus required so as to achieve sufficient product quality. At-speed scan test vectors are usually generated by *automatic test pattern generation* (ATPG) based on either the transition delay fault model and/or the path delay fault model.

Compared with slow-speed scan testing, at-speed scan testing is confronted with more severe challenges in terms of ***test quality*** and ***test costs***. These challenges have been successfully mitigated to some extent by such approaches as timing-aware ATPG [1] for test quality improvement and test compression [2] for test cost reduction. However, in recent years, at-speed scan testing is starting to suffer more and more from a new challenge, namely ***test power*** [3].

The test power problem is caused by the ever-growing gap between *functional power* (lower) and *test power* (higher), which has widened from about 2X to 5X [4] by ever-shrinking functional power (due to aggressive low-power design practices such as power gating/clock gating) and ever-increasing test-mode power (due to fault / block parallelism and functional circuit/clocking constraints being ignored for test efficiency). This means that function-mode-oriented wafer/package level heat management and *power supply network* (PSN) design are getting relatively weaker with respect to potentially excessive test power [5]. As a result, ***heat-related test safety*** (i.e., over-heat may damage circuits) and ***power-supply-noise-related test safety*** (i.e., power supply noise may invalidate test responses) have become serious problems in at-speed scan testing [5, 6].

Generally, heat-related test safety relies on average shift power, which can be effectively and predictably reduced below a safety level by practical techniques, e.g., *scan chain segmentation* [7]. On the other hand, power-supply-noise-related test safety largely depends on the ***launch switching activity*** (LSA) caused by test stimulus launching at the beginning of the at-speed test cycle. As illustrated in Fig. 1 based on the *launch-on-capture* (LOC) clocking scheme, the first capture C_1 may cause excessive LSA, resulting in IR-drop and Ldi/dt that reduce effective power supplies to cells, leading to increased path delay, and finally timing failures at the second capture C_2. This paper will focus on power-supply-noise-related test safety, referred to as ***launch safety*** hereafter, since *launch switching activity* (LSA) is its determining factor. Clearly, a launch-safe test vector is one that will not cause excessive-LSA-induced timing failures.

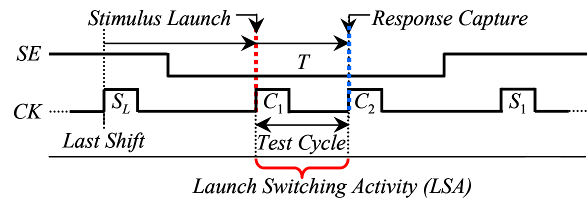

Figure 1. Launch safety in LOC-based at-Speed scan testing.

Achieving launch safety requires sufficient reduction of excessive LSA. However, while previous low-LSA test generation techniques (ATPG, test compaction, and X-filling) [5] can reduce LSA to some extent, they may not guarantee launch safety [8] due to the following problems:

- **Problem-1** (***Unfocused Effect***): Previous techniques only reduce the total LSA for the whole circuit in an unfocused manner. However, the real cause for timing failures (i.e., excessive LSA in neighboring areas around long sensitized paths) often remains [6, 8]. In addition, unfocused LSA reduction constrains too many logic values in a test vector, causing test quality degradation and test data inflation.
- **Problem-2** (***Unguaranteed Sufficiency***): Previous low-LSA techniques can reduce LSA to some extent. However, none of them can guarantee sufficient LSA reduction for all test vectors of any circuit. This is unacceptable in industry since yield loss risk remains even if only one test vector obtained by low-LSA test generation is still launch-unsafe.

In this paper, we propose a novel and practical scheme to achieving guaranteed launch safety with minimal impact on test quality and test costs in power-aware test generation. The basic idea consists of three integral parts as follows:

(**1**) ***Risky Path Identification***: A *risky path P* of a test vector *V* is a path sensitized by *V* and has excessive *launch switching activity* (LSA) in its *impact area* (composed of the cells whose LSA impacts the delay of the path *P*). *V* is said to be *launch-risky* if it has at least one risky path.

(**2**) ***Risky Path Reduction***: Focused LSA reduction is conducted for the impact areas of risky paths in order to effectively reduce risky paths. This may turn a launch-risky test vector into a launch-safe one. Even if it cannot, it usually reduces the number of remaining risky paths.

(**3**) ***Risky Path Masking***: Since the value at the endpoint of any remaining risky path is uncertain due to excessive LSA in its impact area, it is excluded from use for fault detection. This is done by placing an *X* at the expected test-response-vector bit corresponding to that endpoint. As a result, no yield loss will occur. Note that this is data masking, without any performance penalty / additional circuit overhead.

Clearly, *risky path masking* is the core part to achieving guaranteed launch safety. Although being simple and straightforward, this part only becomes feasible under three critical conditions: (**CC1**) risky paths are identified; (**CC2**) the number of risky paths is small; and (**CC3**) an ATPG flow is devised to recover the lost fault detection capability due to masking. Only when **CC1** ~ **CC3** are all satisfied can *risky path masking* achieve guaranteed launch safety with minimal impact on test quality and test costs.

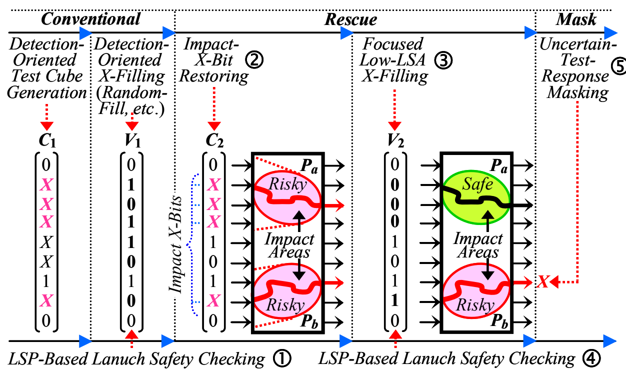

Figure 2. Basic idea for guaranteeing launch safty.

In this paper, we try to satisfy the three critical conditions with a unique two-phase ATPG scheme. As illustrated in Fig. 2, a test cube C_1 (with *X*s) is generated and then turned into a test vector V_1 (without *X*s) by detection-oriented *X*-filling (usually *random-fill* for high test quality and small vector count). Conventional ATPG ends here, but the new scheme continues with two more phases as follows:

• ***Rescue***: ***LSP-based launch safety checking*** (①) identifies all *risky paths* of V_1 by checking the LSA in the impact area of each *long sensitized path* (LSP) under V_1. Suppose that P_a and P_b are found to be risky paths. Then, ***impact-X-bit restoring*** (②) identifies those bits in V_1 that are originally *X*-bits in C_1 (before *X*-filling) and can reach the impact areas of P_a and P_b, and turns them back into *X*-bits (*impact-X-bits*) to create a new test cube C_2. After that, ***focused low-LSA X-filling*** (③) is conducted to turn C_2 into V_2 with reduced LSA in the impact areas of P_a and P_b.

• ***Mask***: *LSP-based launch safety checking* (④) identifies that P_a is now safe but P_b is still risky under V_2. In this case, ***uncertain-test-response masking*** (⑤) is conducted to place an *X* at the endpoint (FF input) of P_b in the test response to V_2. This makes the uncertain value observed by the FF to be ignored in test response comparison, thus avoiding yield loss. Note that this masking needs no additional circuitry.

The advantages of the proposed flow are as follows:

• ***Focused LSA Reduction***: LSA is reduced only for necessary vectors (*launch-risky vectors*) and only in necessary areas (*impact areas*). That is, there is no over-reduction of LSA for launch-safe test vectors or in areas with low or timing-failure-non-causing LSA. This not only greatly improves the effectiveness of risky path reduction but also avoids unnecessary test quality degradation.

• ***Guaranteed Launch Safety***: Masking any uncertain test response guarantees launch safety as the last resort. This is made possible by focused LSA reduction, which makes the number of remaining risky paths small, if any.

• ***Minimal Impact on Test Quality & Test Costs***: Focused LSA reduction only uses necessary resources (i.e., *impact-X-bits*) but keeps original logic values at other bits already optimized by detection-oriented *X*-filling (e.g., *random-fill*). Furthermore, masking-induced loss in fault detection capability is mostly recovered by test vectors generated in subsequent ATPG runs. Therefore, the original test quality is preserved and severe test data inflation is avoided.

The rest of the paper is organized as follows: Sect. II describes the background. Sect. III presents the novel scheme for achieving guaranteed launch safety. Sect. IV shows experimental results, and Sect. V concludes the paper.

II. BACKGROUND

A. Launch Safety Checking

Ideal launch safety checking is time-consuming and memory-intensive due to timing-accurate logic simulation, IR-drop analysis, and delay calculation. This fact makes it necessary to use simplified metrics. As a result, *toggle count* (TC) and *weighted switching activity* (WSA) for FFs, the whole circuit, or regions in a total or instantaneous manner are often used for estimating LSA [5]. However, these metrics are not targeted at long sensitized paths that are most susceptible to the impact of LSA. To address this issue, the *critical capture transition* (CCT) metric assesses LSA around critical paths [9], and the *critical area targeted* (CAT) metric estimates LSA around the longest sensitized path of a test vector [8].

In this paper, we use an improved metric based on the CAT metric [8] for launch safety checking. The CAT metric is extended to check all long sensitized paths for higher accuracy. Details will be presented in Subsection III.B.

B. Low-LSA Test Generation

Three typical approaches to reduce LSA are available [5]:

- **ATPG**: Low-LSA test vectors can be generated with reversible capture-transition-triggered backtracking as well as clock-disabling with pre-calculated (e.g., *default values*, *clock control cubes*, etc.) or dynamically-calculated values.
- **Low-LSA Test Compaction**: This can be done as dynamic compaction by properly selecting secondary faults or as static compaction by properly selecting test cubes to be merged. Both of them try to avoid concentrated LSA.
- **Low-LSA *X*-Filling**: *X*-bits, directly left in a partially-specified test cube or obtained from a fully-specified test vector by test relaxation [10], can be filled with proper logic values so as to reduce LSA [11]. There are three types of such techniques: (**1**) *FF-Oriented*: Transitions at FF outputs are reduced by input-output equalizing (e.g., *preferred-fill* [12], *JP-fill* [13], *iFill* [14], etc.) or clock disabling (e.g., *CTX-fill* [5]); (**2**) *Node-Oriented*: Transitions inside a circuit are directly reduced (e.g., *PWT-fill* [5]); and (**3**) *Critical-Area-Oriented*: Transitions in specific areas inside a circuit are reduced (e.g., *CAT-fill* [8], *CCT-fill* [9], etc.).

However, these previous techniques can only reduce LSA to some extent, but cannot guarantee launch safety. As discussed in Sect. I, the reason comes from two problems: (**i**) ***unfocused effect*** (i.e., most of them only reduce total LSA for the whole circuit but excessive LSA may still remain in neighboring areas around long sensitized paths) and (**ii**) ***unguaranteed sufficiency*** (i.e., they cannot guarantee sufficient LSA reduction for all test vectors). In this paper, we propose a two-phase (***rescue*** & ***mask***) scheme to achieve guaranteed launch safety effectively.

III. New Power-Aware Test Generation Scheme

A. Test Generation Flow

As shown in Fig. 3, conventional test generation (**A**~**E**) starts from initial fault list generation (**A**). A partially-specified test cube C_1 is generated to detect a primary fault and dynamic compaction is then conducted (**B**). Here, any transition, path, or small-delay ATPG can be used. Then, detection-oriented *X*-filling is conducted to turn C_1 into a fully-specified test vector V_1 (**C**). *Random-fill* is often used in industry for this purpose since its fortuitous detection capability greatly improves unmodeled-defect detection (thus *higher test quality*) and reduces test data volume (thus *lower test costs*). After that, fault simulation is conducted to update the fault list (**D**), and the termination condition is checked to decide whether to continue test generation (**E**).

This conventional ATPG flow is enhanced to guarantee launch safety by adding a new two-phase scheme (① ~ ⑤):

- <u>**Phase-I (*Rescue*)**</u>: This phase consists of ① ~ ③. ***LSP-based launch safety checking*** (①) is to identify all *long sensitized paths* (LSP) under V_1 and check the *launch switching activity* (LSA) in the neighboring area (called *impact area* to be defined in III.B) of each LSP. If the impact area of an LSP has excessive LSA, the LSP is called a *risky path*. V_1 is *launch-risky* if it has at least one risky path. In this case, ***impact-X-bit restoring*** (②) is conducted to restore those logic bits in V_1 to *X*-bits (called *impact-X-bits* to be defined in III.C) if they are originally *X*-bits in C_1 and can reach the impact area of at least one risky path. This way, a new test cube C_2 is obtained efficiently without using time-consuming test vector relaxation [10]. After that, ***focused low-LSA X-filling*** (③) is conducted for the impact-*X*-bits to reduce LSA in the impact areas of the risky paths in a focused manner. This results in a new test vector V_2.

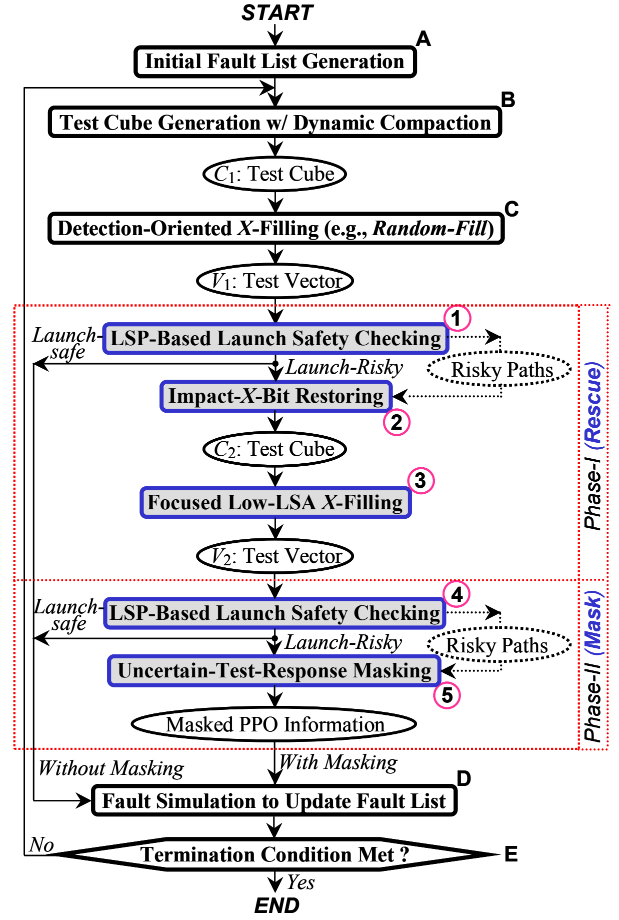

Figure 3. Test generation scheme with guaranteed launch safety.

- <u>**Phase-II (*Mask*)**</u>: This phase consists of ④ and ⑤. First, ***LSP-based launch safety checking*** (④) is conducted on V_2. If V_2 is found to be launch-risky, an erroneous test response may appear at the endpoint (a FF input or a pseudo primary output (PPO) in the circuit model) of a risky path. To avoid this risk of yield loss in production test, ***uncertain-test-response masking*** (⑤) is conducted by placing an *X* as the test response at the endpoint PPO of any remaining risky path for V_2 in production test data. This masking incurs no circuit overhead. In addition, fault simulation with masked PPOs is conducted to update the fault list so that masked-PPO-induced change in fault detection capability is properly reflected in the result of the current ATPG run (**E**).

B. LSP-Based Launch Safety Checking

Since it is a ***long sensitized path*** (***LSP***) that is the most susceptible to the impact of excessive LSA, we conduct ***LSP-based launch safety checking*** as follows:

Definition 1: The ***aggressor region*** of a gate G, denoted by $AR(G)$, is composed of aggressor cells (gates and FFs) whose transitions strongly impact the supply voltage of G.

In an LSI chip, the current flows through C4 pads to cells through a *power grid* composed of alternate metal lines of *VDD* and *GND* in each layer. The metal layers are connected by vias, and cells are connected to lower-level (e.g., M2 through M4) vias. Thus, the aggressor region of a gate G can be identified as follows [15]: First, identify the *powering via* for G, by which G is directly powered. Then, identify all current-sink cells for the powering via of G, and these cells are the aggressor cells for G. A simplified example (*GND* wires ignored) is shown in Fig. 4, where the aggressor region of G_1 consists of G_2, G_3, and G_4. Note that G_2 is a stronger aggressor than G_3 and G_4 that are farther away.

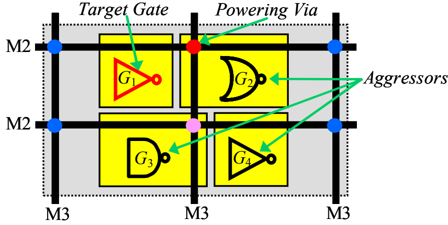

Figure 4. Aggressor region.

Definition 2: The ***impact area*** of P, denoted by $IA(P)$, consists of the aggressor regions of all on-path gates (G_1, G_2, ..., G_n) of P. That is, $IA(P) = AR(G_1) \cup AR(G_2) \cup ... \cup AR(G_n)$, as illustrated in Fig. 5.

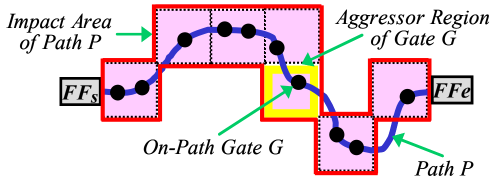

Figure 5. Impact area.

Definition 3: A path P is said to be a ***risky path*** under a test vector V if (1) P is sensitized by V and (2) the LSA in the impact area of P under V is excessive (w.r.t. a threshold).

Definition 4: A test vector V is said to be a ***launch-risky test vector*** if V has at least one risky path.

Based on the above definitions, LSP-based launch safety checking can be conducted by the following procedure:

LSP-Based_Launch_Safety_Checking ()
{ *Input*: test vector V, design data (netlist, layout, power supply network), path length threshold α, WSA level threshold β
Output: status of V, risky paths of V

OP-1: Identify all LSPs, i.e., paths that are sensitized by V and whose lengths are greater than α.

OP-2: Identify the impact area of each LSP.

OP-3: Run logic simulation and calculate the WSA (*weighted switching activity*) for the impact area of each LSP.

Op-4: Identify all risky paths of V by checking if the WSA for the impact area of each LSP is greater than β. Output all risky paths of V.

Op-5: Output the status of V: V is launch-risky if it has at least one risky path; otherwise, V is launch-safe. }

In this procedure, a and β are the *path length threshold* and *WSA threshold*, respectively. In our experiments, they were set to 70% of the length of the longest structural path and 30% of the maximum WSA in the impact area, respectively. In practice, they can be set by test engineers.

The major advantages of LSP-based launch safety checking over previous techniques are as follows:

• ***High Accuracy***: The procedure targets at long sensitized paths (LSPs), whose delay increase is the dominant cause of timing failures in the test cycle as shown in Fig. 1.

• ***High Resolution***: The procedure identifies all risky paths, not just reporting whether a test vector is launch-risky or not. It is this detailed information on risky paths that makes it possible to effectively conduct focused LSA reduction in Phase-1 and guarantee launch safety in Phase-II.

C. Impact-X-Bit Restoring

As shown in Fig. 3, if a test vector is identified by LSP-based launch safety checking (①) as launch-risky, ***rescue*** is then conducted in Phase-I by reducing the excessive LSA in the impact area of each risky path as much as possible. This goal is realized by ***impact-X-bit restoring*** (②) for obtaining necessary X-bits, and ***focused low-LSA X-filling*** (③) for filling those X-bits with proper logic values so as to reduce LSA. This subsection describes impact-X-bit restoring.

Two previous approaches are available for obtaining X-bits needed for low-LSA X-filling. One is *test cube preservation* [3], in which ATPG is forced to leave X-bits in a deterministically-generated test cube by disabling *random-fill* or other detection-oriented X-filling processes. Another is *test vector relaxation* [10], in which a fully-specified test vector set is turned into a partially-specified test cube set while preserving its original fault coverage.

However, *test cube preservation* suffers from significant test quality degradation and test vector count inflation since the fortuitous-detection-capability of detection-oriented X-filling (e. g, *random-fill*) is not used. For example, our experiments on a 600K-gate industrial circuit block showed a 51% increase in test vector count when *random-fill* was disabled. On the other hand, *test vector relaxation* can be conducted on a compact test set and maintains its size. However, this approach is a static post-ATPG process for a complete test set, which is hard to apply in a dynamic ATPG flow for a single test vector.

To preserve the test quality benefit of detection-oriented X-filling as much as possible while obtaining X-bits in a dynamic manner for individual test vectors, we propose a new technique, namely ***impact-X-bit restoring***, as follows:

Definition 5: Let V_1 be a launch-risky test vector, obtained by detection-oriented X-filling from a test cube C_1. An X-bit in C_1 that can reach the impact area of at least one risky path of V_1 is called an ***impact-X-bits*** for V_1.

Impact-X-bit restoring turns every logic bit b in a fully-specified test vector V_1 to X if b corresponds to an impact-X-bit for V_1. The result is a new partially-specified test cube C_2.

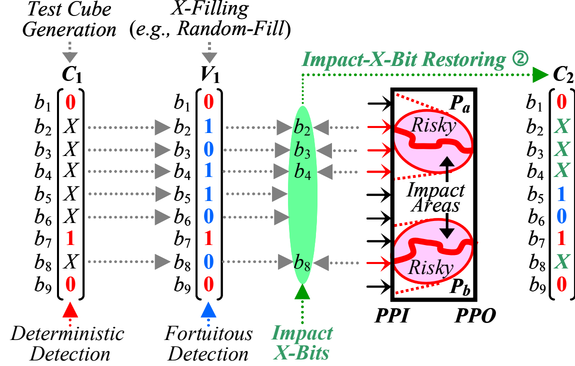

Figure 6. Impact-*X*-bit restoring.

Fig. 6 illustrates *Impact-X-bit restoring*. Here, C_1 is a test cube generated deterministically, with logic values at b_1, b_7, and b_9 for detecting targeted faults. All other bits in C_1 are filled by *random-fill* for fortuitous detection, resulting in V_1. If V_1 has two risky paths (P_a, P_b), the impact-X-bit set for V_1 is $S = \{b_2, b_3, b_4, b_8\}$. Turning the bits in V_1 corresponding to S back into X-bits results in C_2 that detects all faults detected by C_1. Note that *random-fill*-assigned logic values remain unchanged at b_5 and b_6, helping preserve some fortuitous-detection-capability of *random-fill*. Thus, *impact-X-bit restoring* not only obtains X-bits directly related to the LSA in the impact areas of risky paths but also helps preserve test quality and avoid severe test vector count inflation.

D. Focused Low-LSA X-Filling

In Fig. 3, after X-bits are obtained by *impact-X-bit restoring* (②), ***focused low-LSA X-filling*** (③) is conducted on those X-bits. The term "*focused*" means that all of the X-bits are directly related to the impact areas of risky paths.

In this paper, we apply an improved form of *JP-fill* [13] that uses assignment / justification / multi-pass probability calculation to fill X-bits for equalizing PPI and PPO values at ***candidate PPI-PPO bit-pairs*** (i.e., bit-pairs in the form of $<0/1, X>$, $<X, 0/1>$, or $<X, X>$). Since the effectiveness of X-filling depends on the filling order, we improve *JP-fill* with a new weight to order candidate PPI-PPO bit-pairs.

Definition 6: Let $bp = <x, y>$ be a candidate PPI-PPO bit-pair, whose PPI-bit x can reach risky paths: $P_1, P_2, \ldots, P_m$. The ***weight*** of bp, denoted by *weight* (bp), is defined as

$$weight(bp) = \sum_{i=1}^{m} reached(x, P_i) / all(P_i)$$

where $reached(x, P_i)$ is the number of nodes reachable from the PPI-bit x in the impact area of P_i, and $all(P_i)$ is the number of all nodes in the impact area of P_i.

Clearly, $weight(bp)$ indicates the impact of reducing a transition at the candidate PPI-PPO bit-pair bp on reducing LSA in the impact areas of risky paths. We use this weight to determine the order of processing candidate PPI-PPO bit-pairs so as to achieve more effective LSA reduction.

E. Uncertain-Test-Response Masking

As shown in Fig. 3, *focused low-LSA X-filling* (③) in the ***rescue phase*** (Phase-I) results in a test vector V_2. Generally, V_2 may still be launch-risky although it often has fewer risky paths. In this case, the ***mask phase*** (Phase-II) is executed as the last resort for guaranteeing launch safety.

In Phase-II, if *LSP-based launch safety checking* (④) finds V_2 to be launch-risky, ***uncertain-test-response masking*** (⑤) is then conducted as illustrated in Fig. 7.

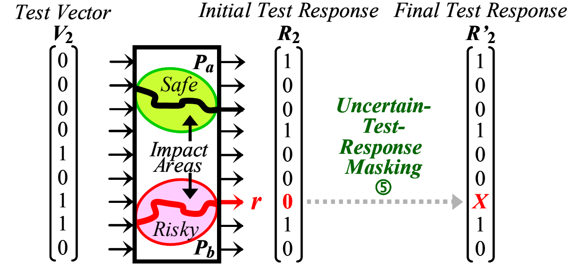

Figure 7. Uncertain-test-response masking.

In Fig.7, test vector V_2 has two long sensitized paths: a safe path P_a (due to focused low-LSA X-filling) and a risky path P_b (due to insufficient LSA reduction) whose endpoint corresponds to the output bit r. Here, although R_2 (the initial test response to V_2 obtained by logic simulation) has 0 at r, excessive LSA in the impact area of P_b may cause a timing error at r, falsely making 1 to appear at r in test mode. To avoid possible yield loss, ***uncertain-test-response masking*** (⑤) is conducted by replacing 0 with X at r in the final test response R'_2. This instructs the tester not to compare at r in production test, thus avoiding any possible power-supply-noise-induced yield loss in at-speeding scan testing.

Note that masking incurs no area/performance overhead as it just puts an X (unknown) at r in the final test response. Although faults detected by V_2 only at r become undetected by V_2 due to masking, r is only masked for V_2 but available for fault detection by other test vectors. That is, the fault list is updated by fault simulation with masked PPO information (**D** in Fig. 3), and ATPG continues in which initially r is not masked. This way, the lost fault detection capability at r for V_2 can be recovered by subsequent ATPG runs, only at the cost of slightly more test vectors.

F. Extension to Compressed Scan Testing

The proposed rescue-&-mask scheme for achieving guaranteed launch safety can be readily extended to any test compression environment. For example, in broadcast-scan-based test compression [2], the constraints imposed by the combinational decompressor on inputs can be embedded into an integrated combinational circuit model [16], on which *LSP-based launch safety checking*, *impact-X-bit restoring*, and *focused low-LSA X-filling* can be directly applied. The major concern is with test response compaction, where Xs introduced by *uncertain-test-response masking* may disturb fault detection. However, such Xs are sparse and their number is very small for a test vector. This makes their impact on test quality and test costs minimal.

IV. EXPERIMENTAL RESULTS

The power-aware test generation flow of Fig. 3 was implemented with TetraMAX® as the base ATPG, and the rescue-&-mask scheme for guaranteeing launch safety was coded in C. Six large ITC'99 benchmark circuits were synthesized and physically designed with a power supply network. Experiments were conducted on a workstation (Intel Xeon® 3.33GHz with 64GB main memory). Table I shows circuit statistics and experimental results.

In our experiments, test costs were evaluated by test vector count (*# of Vectors*), while test quality was evaluated not only by transition fault coverage (*FC*) but also by *bridging coverage estimate* (*BCE*) [17] and *statistical delay quality level* (*SDQL*) [18] for more comprehensive evaluation. *BCE* is used in industry to assess the capability of detecting unmodeled structural defects (especially, bridging defects), and *SDQL* is used in industry to assess the capability of detecting unmodeled small-delay defects.

First, the conventional ATPG flow was executed, and its results are shown under "*Conventional Flow*" in Table I as baseline values. Next, the proposed ATPG flow was executed, and its results are shown under "*Proposed Flow*" in Table I with four parts: (**1**) Changes in test vector count and three test quality metrics (*FC*, *BCE*, *SDQL*) are shown under "*ATPG Results %Change*"; (**2**) the average number of long sensitized paths per vector (*Ave. # of LSPs / Vec.*), the average number of risky paths per vector (*Ave. # of Risky Paths / Vec.*), and the percentage of risky vectors (*% of Risky Vec.*) are shown under "*Launch Safety Checking*"; (**3**) the percentage of impact-*X*-bits (*% of Impact-X-Bits*) and the ratio of focused low-LSA *X*-filling making risky paths into safe paths (*Resuce Ratio (%)*) are shown under "*Rescue*"; (**4**) the number of masked PPOs (*# of Masked PPOs*) and the average number of masked PPOs per vector (*Ave. # of Masked PPOs / Vec.*) are shown under "*Mask*".

Due to the nature of the proposed ATPG flow, it always guarantees launch safety. From Table I, it can be seen that the impact of this new ATPG flow on test quality and test costs is minimal since there is little change in test vector count, *FC*, *BCE*, and *SDQL*. This is because of **the nature of masked PPOs**, i.e., masked PPOs are sparse and the number of masked PPOs is extremely small, as indicated by "*Ave. # of Masked PPOs / Vec.*" in Table I.

Note that the above nature also holds in any test compression environment since it is only related to the combinational portion and independent of the decompressor and the compactor. This indicates that the proposed rescue-&-mask scheme also works for compressed scan testing.

V. CONCLUSIONS

This paper has addressed the fundamental issue in power-aware test generation for at-speed scan testing, i.e., how to guarantee launch safety instead of merely reducing launch switching activity to some extent. A novel two-phase scheme has been proposed to guarantee launch safety with minimal impact on test quality and test costs. The ***rescue phase*** is to reduce excessive LSA around long sensitized paths, and the ***mask phase*** is to exclude any uncertain test response from being used for fault detection. Experimental results have validated this novel approach to guaranteeing launch safety.

Future work includes conducting evaluation experiments by using a commercial-grade test compression tool.

REFERENCES

[1] X Lin, et al., "Timing-Aware ATPG for High Quality At-Speed Testing of Small Delay Defects," *Proc. ATS*, pp.139-146, 2006.

[2] N.A. Touba, "Survey of Test Vector Compression Techniques," *IEEE Design & Test Magazine,* Vol. 23, Issue 4, pp. 294-303, 2006.

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

[4] S. Sde-Paz and E. Salomon, "Frequency and Power Correlation between At-Speed Scan and Functional Tests," *Proc. ITC*, Paper 13.3, 2008.

[5] P. Girard, et al., *Power-Aware Testing and Test Strategies for Low Power Devices*, Springer, 2009.

[6] C. P. Ravikumar, et al., "Test Strategies for Low-Power Devices," *J. of Low Power Electronics,* Vol. 4, No.2, pp. 127-138, 2008.

[7] L.Whetsel, "Adapting Scan Architectures for Low Power Operation," *Proc. ITC*, pp. 863-872, 2000.

[8] K. Enokimoto, et al., "CAT: A Critical-Area-Targeted Test Set Modification Scheme for Reducing Launch Switching Activity in At-Speed Scan Testing," *Proc. ATS,* pp. 99-104, 2009.

[9] X. Wen, et al., "Critical-Path-Aware X-Filling for Effective IR-Drop Reduction in At-Speed Scan Testing," *Proc. DAC,* pp.527-532, 2007.

[10] K. Miyase, et al., "XID: Don't Care Identification of Test Patterns for Combinational Circuits," *IEEE TCAD*, 23-2, pp. 321-326, 2004.

[11] X. Wen, et al., "On Low-Capture-Power Test Generation for Scan Testing," *Proc. VTS*, pp. 265-270, 2005.

[12] S. Remersaro, et al., "Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs," *Proc. ITC*, Paper 32.2, 2006.

[13] X. Wen, et al., "A Novel Scheme to Reduce Power Supply Noise for High-Quality At-Speed Scan Testing," *Proc. ITC,* Paper 25.1, 2007.

[14] J. Li, et al., "On Capture Power-Aware Test Data Compression for Scan-Based Testing," *Proc. ICCAD*, pp. 67-72, 2008.

[15] J. Lee, et al., "Layout-Aware, IR-Drop Tolerant Transition Fault Pattern Generation," *Proc. DATE*, pp. 1172-1177, 2008.

[16] K. Miyase, et al., "A Novel Post-ATPG IR-Drop Reduction Scheme for At-Speed Scan Testing in Broadcast-Scan-Based Test Compression Environment," *Proc. ICCAD*, pp. 97-104, 2009.

[17] B. Benware, et al., "Impact of Multiple-Detect Test Patterns on Product Quality," *Proc. ITC*, pp. 1031-1040, 2003.

[18] Y. Sato, et al., "Invisible Delay Quality - SDQM Model Lights Up What Could Not Be Seen," *Proc. ITC*, Paper 47.1, 2005.

TABLE I. EVALUATION RESULTS

Circuit	# of Gates	# of FFs	*Conventional Flow* (with possible launch risk)				*Proposed Flow* (with guaranteed launch safety)											
			ATPG Result (Baseline)				*ATPG Result (% Change)*				*Launch Safety Checking*			*Rescue*		*Mask*		
			# of Vectors	*FC*	*BCE*	*SDQL*	*Δ # of Vectors*	*Δ FC*	*Δ BCE*	*Δ SDQL*	*Ave. # of LSPs / Vec.*	*Ave. # of Risky Paths / Vec.*	*% of Risky Vec.*	*% of Impact-X-Bits*	*Resuce Ratio (%)*	*# of Masked PPOs*	*Ave. # of Masked PPOs / Vec.*	*CPU (Sec.)*
b17	21,235	1,317	970	73.0	66.1	16.9	+0.05	+0.02	–0.01	+0.01	0.13	0.01	0.19	29.18	25.00	2	0.01	471
b18	64,009	3,064	1,933	65.4	64.0	204.6	+0.01	+0.04	–0.01	+0.03	0.45	0.21	0.90	27.86	8.90	28	0.01	1,777
b19	128,576	6,130	2,806	66.5	63.9	303.8	–0.06	+0.02	–0.01	+0.01	0.25	0.01	0.11	45.29	96.00	1	0.01	4,607
b20	20,271	430	1,496	93.8	63.3	116.5	+0.01	+0.01	–0.04	+0.01	0.25	0.15	0.93	14.70	0.02	8	0.01	343
b21	20,148	430	1,503	94.1	57.1	137.6	+0.04	+0.01	–0.01	+0.01	0.33	0.30	1.22	12.79	0.02	23	0.02	342
b22	29,926	645	1,919	94.4	62.3	140.5	+0.08	+0.01	–0.01	+0.03	0.05	0.01	0.15	11.37	48.00	3	0.01	597