CAT: A Critical-Area-Targeted Test Set Modification Scheme for Reducing Launch Switching Activity in At-Speed Scan Testing

doi: info:doi/10.1109/ATS.2009.22

CAT: A Critical-Area-Targeted Test Set Modification Scheme for Reducing Launch Switching Activity in At-Speed Scan Testing

K. Enokimoto [1], X. Wen [1], Y. Yamato [1], K. Miyase [1], H. Sone [2], S. Kajihara [1], M. Aso [3], and H. Furukawa [3]

[1] Kyushu Institute of Technology, Kawazu 680-4, Iizuka, Fukuoka 820-8502, Japan
[2] Fukuoka Industry, Science & Technology Foundation, Fukuoka, Fukuoka 810-0001, Japan
[3] NEC Micro Systems, Ltd., Kamimashiki, Kumamoto 861-2202, Japan

Abstract: *Reducing excessive launch switching activity (LSA) is now mandatory in at-speed scan testing for avoiding test-induced yield loss, and test set modification is preferable for this purpose. However, previous low-LSA test set modification methods may be ineffective since they are not targeted at reducing launch switching activity in the areas around long sensitized paths, which are spatially and temporally critical for test-induced yield loss. This paper proposes a novel* ***CAT*** *(**C**ritical-**A**rea-**T**argeted) low-LSA test modification scheme, which uses long sensitized paths to guide launch-safety checking, test relaxation, and X-filling. As a result, launch switching activity is reduced in a* ***pinpoint*** *manner, which is more effective for avoiding test-induced yield loss. Experimental results on industrial circuits demonstrate the advantage of the CAT scheme for reducing launch switching activity in at-speed scan testing.*

1. Introduction

At-speed scan testing has become mandatory for deep-submicron (DSM) integrated circuits (ICs), in which timing-related defects are dominant. Two issues, *test quality* and *test safety*, are important for at-speed scan testing.

Test quality depends on test vectors used, which are usually generated by using a delay fault model. Such a test vector sensitizes a set of paths, either implicitly for the transition delay fault model or explicitly for the path delay fault model. It has been shown that the test quality of a test set, measured by its capability of detecting timing-related defects, is largely determined by its ***long sensitized*** paths [1, 2].

Test safety depends on how likely test results are invalidated to result in high risk of circuit damage and/or yield loss. Conventional causes for low test safety include problematic tester/test environment setups. In the DSM era, high-speed/low-power ICs often suffer from signal-integrity-related causes, such as power supply noise and crosstalk [3].

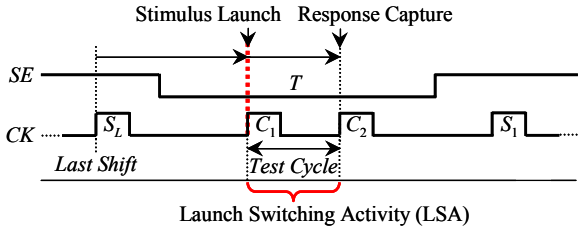

Fig. 1 Launch-safety issue in LOC at-speed scan testing.

In recent years, power supply noise due to excessive ***launch switching activity*** (***LSA***) has rapidly become a major yield-killer in *LOC* (*launch-on-capture*) at-speed scan testing [3-6]. As illustrated in Fig. 1, LSA is caused by the stimulus launch pulse (C_1). If the LSA around a sensitized path is too high, excessive local IR-drop may occur, significantly increasing delay along the path and eventually leading to timing failures at the response capture pulse (C_2). Note that the possibility of such failures is high for long sensitized paths. Generally, ***launch-safety***, defined by whether LSA causes timing failures in the test cycle (T in Fig. 1), is a major test quality issue in LOC at-speed scan testing [3-6].

Launch-safety can be improved by (I) *DFT*, (II) *ATPG*, and (III) *test set modification*. Low-LSA DFT [3] and low-LSA ATPG [17,18] are costly due to circuit/clock change and significant test vector inflation, respectively. Low-LSA test set modification [3] is cost-effective since it has no impact on circuit/clock and test vector count. Therefore, a hybrid approach is preferable that first applies low-LSA test set modification to rescue most of launch-risky test vectors and then uses low-LSA ATPG to rescue the remaining few. This way, launch-safety can be achieved efficiently with no circuit overhead and no significant test vector inflation.

Low-LSA test modification needs three basic operations: (1) *launch-safety checking* to identify launch-risky test vectors, (2) *test relaxation* to create *X*-bits, and (3) *X-filling* to determine proper logic values for the *X*-bits so as to avoid excessive LSA. Many methods have been proposed for these purposes in terms of test power analysis [5-7], *X*-identification [20], and low-capture-power *X*-filling [8-16].

However, previous methods suffer from a serious problem: they are ***gross*** in that they only address whole-circuit LSA, which is not directly related to launch-safety. Such a gross method can be misleading because the LSA around a long sensitized path may still be very high (meaning poor launch-safety) although the whole-chip LSA is greatly lowered.

Therefore, there is a strong need to explore a new direction for low-LSA test set modification, which should be ***pinpoint*** rather than gross. That is, it needs to address the LSA in ***critical areas*** (neighborhoods of long sensitized paths), which is directly related to launch-safety. This paper presents the first scheme of this kind, called ***CAT*** (***Critical-Area-Targeted***) low-LSA test set modification, featuring:

(1) ***CAT launch-safety checking*** targeted at critical areas for identifying launch-risky test vectors.

(2) ***CAT test relaxation*** for finding *X*-bits in launch-risky test vectors, while preserving all long sensitized paths to maintain test quality / fault coverage and maximizing *X*-bits related to critical areas with excessive LSA.

(3) ***CAT X-filling*** for assigning logic values to the *X*-bits to reduce LSA in critical areas with excessive LSA. It uses both clock-disabling (*stopping clock-gators*) and FF-silencing (*equalizing input and output of a FF*) techniques, which are targeted at critical areas.

The advantage of the CAT scheme is that excessive LSA is reduced in a ***pinpoint*** manner, which is more effective for avoiding yield loss induced by power supply noise. This is achieved without any impact on delay test quality, test vector count, circuit overhead, and timing performance.

The rest of the paper is as follows: Section 2 is an overview of the CAT scheme, while the details of launch-safety checking, test relaxation, and *X*-filling in CAT are presented in Sections 3, 4, and 5, respectively. Section 6 shows experimental results, and Section 7 concludes the paper.

2. The CAT Scheme

The overview of the CAT scheme in the launch-safety improvement flow is shown in Fig. 2, which has four steps:

①: ***CAT launch-safety checking*** is conducted on the initial test set V_{init} to identify its launch-safe subset $V_{init\text{-}safe}$ and its launch-risky subset $V_{init\text{-}risky}$. This is achieved by estimating the LSA in critical areas to be defined in Definition 2.

②: ***CAT test relaxation*** is conducted on $V_{init\text{-}risky}$ to turn it into a partially-specified test cube set C, while preserving long sensitized paths & fault coverage and maximizing the number of *X*-bits related to critical areas with excessive LSA.

③: ***CAT X-filling*** is conducted on C to turn it into a fully-specified test set V_{temp}. This is achieved by modifying our *CTX* (*Clock-Gating-Based Test Relaxation and X-Filling*) technique [15] to make both clock-disabling and FF-silencing to target directly at critical areas with excessive LSA.

④: ***CAT launch-safety checking*** is conducted on V_{temp} to find the launch-safe $V_{temp\text{-}safe}$ and the launch-risky $V_{temp\text{-}risky}$.

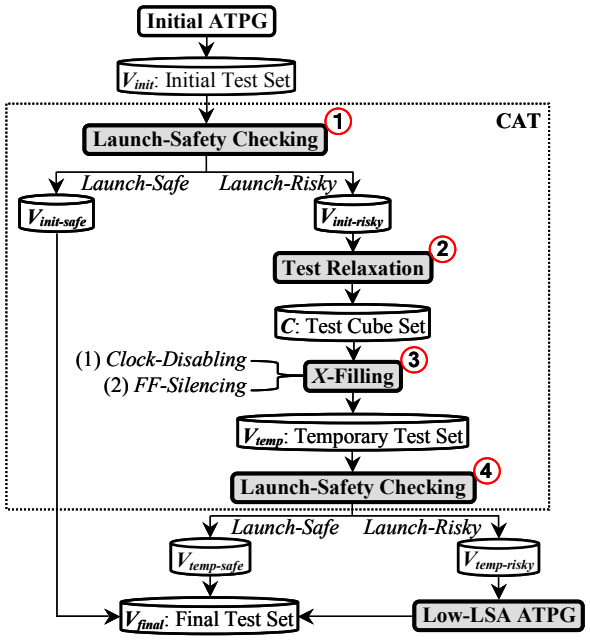

Fig. 2 Launch-safety improvement flow with the CAT scheme.

Note that CAT test relaxation (②) and CAT *X*-filling (③) improve launch-safety without additional test vectors. In addition, since $|V_{temp\text{-}risky}|$ is much smaller than $|V_{init\text{-}risky}|$, low-LSA ATPG [17, 18] only needs to target a small number of faults that are detected only by $V_{temp\text{-}risky}$. As a result, the flow of Fig. 2 can efficiently improve the overall launch-safety without severe test vector count inflation.

3. CAT Launch-Safety Checking

As shown in Fig. 1, a delay test vector v usually sensitizes many paths in the test cycle T, and the IR-drop caused by v at the stimulus launch (C_1) increases the delay along these paths [3]. If the delay increase along at least one such sensitized path breaks the timing requirement of T, yield loss may occur due to possible errors at the response capture (C_2) [4]. Here, the length of a sensitized path plays a critical role. That is, what are susceptible to such IR-drop impact are ***long sensitized paths***. Therefore, launch-safety checking for v should be focused on long paths sensitized by v.

Definition 1: A ***sensitive path*** of a delay test vector v is a path that is sensitized by v and whose length is longer than a limit. The set of all sensitive paths of v is denoted by $SP(v)$.

An example is shown in Fig. 3, where a delay test vector v sensitizes three paths (p_{a2}, p_{a3}, p_{a4}) for transition-delay fault f_a and two paths (p_{b1}, p_{b2}) for transition-delay fault f_b. In this example, only p_{a3} and p_{a4} are sensitive paths of v.

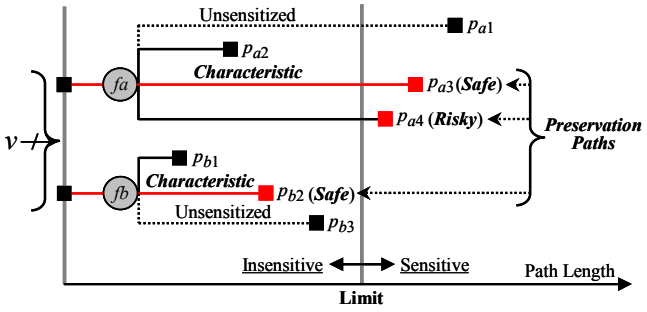

Fig. 3 Sensitive, characteristic, and preservation paths.

Note that whether a path is *long* or *short* should be judged in relation to the test cycle (T in Fig. 1). Conceptually, a path is considered long if its slack is smaller than the maximum delay possibly caused by IR-drop.

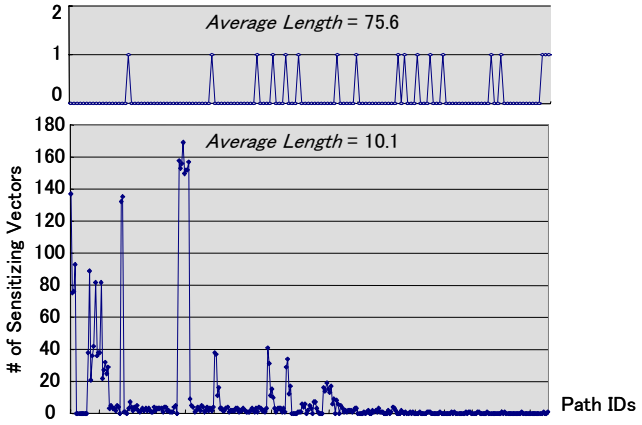

Fig. 4 Information on sensitized paths for an industrial circuit.

Generally, the number of sensitive paths of a delay test vector is small since most of sensitized paths are short even for timing-ware transition delay ATPG [1, 2]. To verify this, we conducted an experiment on an industrial circuit of 50K gates by using 319 transition delay test vectors. The maximum logic level of the circuit is 105. Fig. 4 shows the numbers of vectors that sensitized paths with average logic levels of 75.6 and 10.1. This result indicates that the number of sensitive paths of a test vector is manageable.

Once sensitive paths of a delay test vector v are identified, the launch-safety of v can be checked by estimating how likely a sensitive path $p \in SP(v)$ breaks the timing of the test cycle due to the IR-drop caused by v. High-accuracy estimation requires that the LSA in the close proximity of p be checked. This is because the impact of a switching node on the IR-drop of another node has been shown to be largely determined by the distance between them [6, 7, 19].

Definition 2: The neighborhood of a sensitive path, consisting of all nodes within a given radius R from its on-path nodes, is called the ***critical area*** of the sensitive path.

Note that R is better determined from the size of a feed-region (i.e. a group of nodes sharing the same power via) [7, 19]. Fig. 5 shows the conceptual image of critical area.

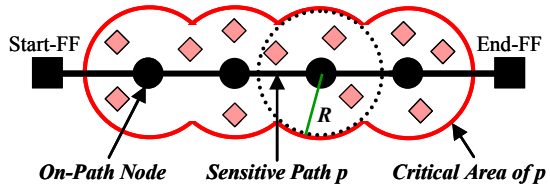

Fig. 5 Critical area of a sensitive path.

Definition 3: A sensitive path p of a delay test vector v is ***risky*** under v if the LSA in the critical area of p is higher than a threshold limit; otherwise, p is ***safe*** under v.

Various metrics have been proposed for estimating the global LSA in a circuit, such as total / instantaneous toggle counts [5], switching cycle average power [6], delay [7], WSA [9], etc. These metrics can be readily tailored for estimating the local LSA in a critical area. The contribution of this paper is that it takes two important factors, ***sensitivity*** and ***proximity***, into CAT launch-safety checking by focusing on the neighborhood of each long sensitized path.

In the example shown in Fig. 3, the test vector v has two sensitive paths: p_{a3} is safe and p_{a4} is risky. Note that any insensitive path, such as p_{a1} (*unsensitized*) or p_{b2} (*sensitized but short*), is considered safe no matter how long it is.

Definition 4: A delay test vector v is ***launch-safe*** if none of its sensitive paths is risky under v; v is ***launch-risky*** if at least one of its sensitive paths is risky under v.

In Fig. 3, the delay test vector v is launch-risky since one of its sensitive path, p_{a4}, is risky. Another example is shown in Fig. 6, in which the delay test vector v has three paths: p_1, p_2, and p_3. p_3 is considered safe since it is not a sensitive path. As for sensitive paths p_1 and p_2, only the critical area of p_1 has excessive LSA. That is, p_1 is risky. Therefore, v is a launch-risky delay test vector.

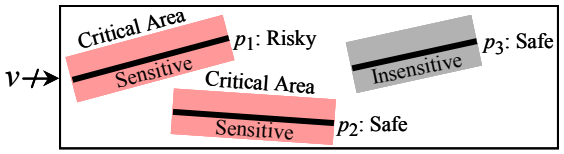

Fig. 6 Critical areas of a launch-risky delay test vector.

As shown in Fig. 2, CAT launch-safety checking is conducted on the initial test set V_{init} to identify a launch-safe subset $V_{init\text{-}safe}$ and a launch-risky subset $V_{init\text{-}risky}$. Launch-risky test vectors need to be rescued by reducing LSA.

4. CAT Test Relaxation

The CAT scheme of test set modification for rescuing launch-risky test vectors consists of two operations: *CAT test relaxation* (②) and *CAT X-filling* (③) as shown in Fig. 2.

This section presents ***CAT test relaxation***, which converts the fully-specified test vector set $V_{init\text{-}risky}$ into a partially-specified test cube set C by turning some logic bits in $V_{init\text{-}risky}$ into X-bits, while achieving the following goals:

G1 (***No Fault Coverage Loss***) All faults detected only by test vectors in $V_{init\text{-}risky}$ are also detected by test cubes in C.

G2 (***No Test Quality Loss***) Any sensitive path (i.e. long sensitized path) that is only sensitized by test vectors in $V_{init\text{-}risky}$ remains sensitized by test cubes in C.

G3 (***X-Bit Optimization***) The number of X-bits related to the critical area with excessive LSA under any launch-risky test vector in $V_{init\text{-}risky}$ is increased as much as possible.

G1 and G2 are for keeping high fault coverage and high test quality, while G3 is for maximizing the effect of LSA reduction by X-filling (i.e. assigning proper logic values to X-bits as shown by ③ in Fig. 2). Previous test relaxation methods [12,20] can only achieve G1 and part of G2 (i.e. only one longest path per fault is kept sensitized [12]).

In the following, we propose a novel test relaxation method that can achieve the three goals (G1, G2, G3) simultaneously, by using two unique concepts: *preservation path* and critical-area-targeted (CAT) *preservation path assignment*.

4.1 Preservation Path

Generally, test set modification (i.e. test relaxation and X-filling) may change the sensitization status of a path. Table 1 summarizes its impact on fault coverage and test quality.

Table 1 Impact of Test Set Modification on Path Sensitization

	Test Set Modification		*Possible Path*	*Impact*
	Before	*After*		*Fault Coverage / Test Quality*
A	Sensitized	Unsensitized	Long	Bad
B	Unsensitized	Sensitized	Short	Negligible
C	Sensitized	Sensitized	Short	No Change
D	Unsensitized	Unsensitized	Long / Short	No Impact

The most severe case is A, where a sensitized path p before modification is unsensitized after modification. The impacts in this case are as follows: (1) If p is the only path that detects a fault, p becoming unsensitized means fault coverage loss; (2) if p is a sensitive (i.e. long & sensitized) path that detects a fault, p becoming unsensitized means lower test quality due to reduced small-delay testing capability.

In order to preserve both fault coverage and test quality, CAT test relaxation explicitly keeps two types of sensitized paths, namely *characteristic paths* (as defined in [12]) and *sensitive paths* (as defined in Definition 1).

Definition 5 [12]: Suppose that V is a transition delay test set. The longest path sensitized by vectors in V for detecting a fault f is called the ***characteristic path*** of f under V.

In Fig. 3, P_{a3} and P_{b2} are characteristic paths of f_a and f_b under $\{v\}$, respectively. Note that a characteristic path, e.g. P_{b2}, may not be long enough to qualify as a sensitive path.

Definition 6: Suppose that V is a transition delay test set. A ***preservation path*** of V is either a characteristic path of a detected fault under V or a sensitive path of a vector in V.

For example, p_{a3}, p_{a4}, and p_{b2} in Fig. 3 are preservation paths of $\{v\}$. Here, the non-characteristic path p_{a4} is a preservation path since it is a sensitive (long and sensitized) path.

As shown in Fig. 2, keeping each preservation path sensitized in test relaxation (②) will keep fault coverage & test quality after *X*-filling (③), thus achieving both goals G1 and G2.

4.2 Preservation Path Classification

Test relaxation and *X*-filling in the CAT scheme as shown in Fig. 2 need to keep all preservation paths of V_{init} sensitized. Since test relaxation (②) is conducted only on $V_{init\text{-}risky}$, the preservation paths sensitized by $V_{init\text{-}safe}$ automatically remain sensitized. Therefore, it is only necessary to explicitly keep a preservation path p sensitized if p is sensitized only by one or more launch-risky test vectors in $V_{init\text{-}risky}$.

An example is shown in Table 2, where $V_{init} = \{v_1, v_2, ..., v_5\}$. Simulation can be conducted to find all preservation paths, as shown under "*Before-Assignment*". Test relaxation will be conducted on $V_{init\text{-}risky} = \{v_1, v_4, v_5\}$, which has 9 preservation paths (p_1, p_2, p_3, p_5, p_6, p_7, p_8, p_{10}, p_{11}). Since p_6 is also sensitized by the launch-safe test vector v_3, there is no need to keep it sensitized in test relaxation conducted on $\{v_1, v_4, v_5\}$. Thus, CAT test relaxation only need to keep 8 preservation paths (p_1, p_2, p_3, p_5, p_7, p_8, p_{10}, p_{11}) sensitized.

Table 2 Preservation Path Information

		Before-Assignment		*After-Assignment*	
	Launch Safety?	Vector-Preservation Paths	Set-Preservation Paths	Vector-Preservation Paths	Set-Preservation Paths
v_1	*Risky*	***p1*** *safe*	***p3*** *risky* ***p8*** *safe* ***p10*** *risky*	***p1*** *safe*	***p3*** *risky* ***p10*** *risky*
v_2	*Safe*	***p4*** *safe*		***p4*** *safe*	
v_3	*Safe*	***p6*** *safe* ***p9*** *safe*		***p6*** *safe* ***p9*** *safe*	
v_4	*Risky*	***p5*** *safe* ***p7*** *risky*	***p8*** *safe* ***p10*** *risky* ***p11*** *safe*	***p5*** *safe* ***p7*** *risky*	***p8*** *safe*
v_5	*Risky*	***p2*** *risky* ***p6*** *safe*	***p3*** *risky* ***p10*** *risky* ***p11*** *safe*	***p2*** *risky*	***p11*** *safe*

Definition 7: Suppose that V is a transition delay test set. A preservation path of V is called a ***vector-preservation path*** if it is only sensitized by one test vector in V. A preservation path of V is called a ***set-preservation path*** if it is only sensitized by multiple test vectors in V.

In the example shown in Table 2, $\{p_1\}$, $\{p_5, p_7\}$, and $\{p_2\}$ are vector-preservation paths for v_1, v_4, and v_5, respectively; p_3, p_8, p_{10}, and p_{11} are set-preservation paths for $\{v_1, v_4, v_5\}$.

4.3 Preservation Path Assignment

From Table 2, it is clear that a vector-preservation path (e.g. p_1) needs to be kept sensitized in test relaxation for the only test vector (e.g. v_1) that sensitizes the path. However, a set-preservation path (e.g. p_{10}) only needs to be kept sensitized in test relaxation for one of the multiple test vectors (e.g. v_1, v_4, v_5) that sensitize the path. Thus, it is necessary to assign each set-preservation path to a launch-risky test vector.

Although any preservation path assignment can achieve goals G1 (no fault coverage loss) and G2 (no test quality loss), different preservation path assignments have different impacts on G3 (maximizing *X*-bits related to the critical area of a launch-risky test vector). In the following, we propose a technique for preservation path assignment by taking critical areas into consideration in order to achieve G3.

Definition 8: All the bits in a test vector that are needed for the sensitization of a path p are called ***sensitization-impact-bits*** of p, denoted by $SIB(p)$.

Definition 9: All the bits in a test vector that are reachable from the critical area of a path p are called ***transition-impact-bits*** of p, denoted by $TIB(p)$.

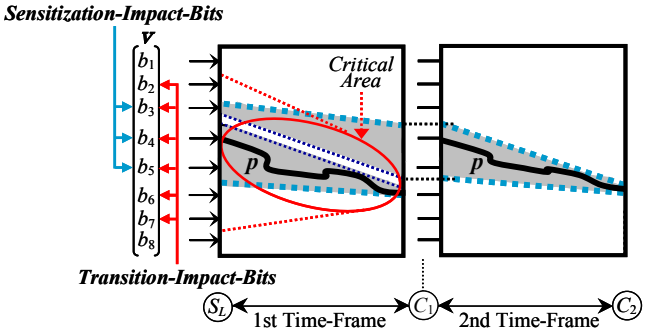

Fig. 7 Sensitization-impact-bit and transition-impact-bit.

Fig. 7 shows an example, where $v = <b_1\ b_2\ \ldots\ b_8>$ is a test vector, p is a path, and the LOC scheme (Fig. 1) is assumed. Cone analysis for p in two time-frames reveals that the set sensitization-impact-bits of p is $SIB(p) = \{b_3, b_4, b_5\}$. Keeping the logic values of $SIB(p)$ unchanged and turning all other bits into *X*-bits will convert the fully-specified test vector v into a partially-specified test cube c. Clearly, p will remain sensitized under c. Furthermore, cone analysis in the 1st time-frame (corresponding to the test cycle in Fig. 1) from the critical area of p reveals that the set of transition-impact-bits of p is $TIB(p) = \{b_2, b_3, \ldots, b_7\}$. Since the values of $SIB(p)$ cannot be changed, only the bits in $TIB(p) - SIB(p)$, i.e. $\{b_2, b_6, b_7\}$, are *X*-bits that have impact on the critical area of p. Therefore, in order to achieve the goal of G3 (increasing the number of *X*-bits related to a critical area), preservation path assignment should be conducted so that that $|TIB(p) - SIB(p)|$ is maximized. This is the basic idea of the following CAT preservation path assignment procedure.

Preservation Path Assignment Procedure

Let p be the next path to be assigned. Let v_1, ..., v_n be the launch-risky test vectors that sensitize p. Let p_{i1}, ..., p_{im} be the current preservation paths assigned to v_i ($i = 1, ..., n$).

(**1**) If p is a vector-preservation path and is only sensitized by v_i, assign p to v_i.

(**2**) If p is a set-preservation path, calculate $SIB_i = SIB(p_{i1}) \cup ... \cup SIB(p_{im}) \cup SIB(p)$ for $i = 1, 2, ..., n$.

(**2-1**) If p is a safe path, assign p to v_i if $|SIB_i|$ is the smallest.

(**2-2**) If p is a risky path, assign p to v_i if $|TIB(p) - SIB_i|$ is the largest.

A sample result of preservation path assignment is shown in Table 2, under "*After-Assignment*".

4.4 Obtaining Test Cubes

After preservation path assignment, the next operation in CAT test relaxation is to keep the values of all sensitization-impact-bits of each assigned preservation path of each launch-risky test vector in $V_{init\text{-}risky}$ (Fig. 2) and turn all other bits into *X*-bits. This way, a partially-specified test cube set C is obtained, while achieving the goals: G1, G2, and G3.

5. CAT *X*-Filling

In the CAT scheme shown in Fig. 2, after test relaxation (②) creates a test cube set C, *X*-filling (③) is conducted on C to create a fully-specified test vector set V_{temp}. Here, ***CAT X-filling*** assigns proper logic values to all *X*-bits in each test cube of C so that the LSA in the critical area of any risky preservation path for the test cube is reduced.

For example, the fully-specified test vector v_5 in Table 2 has two preservation paths: p_2 and p_{11}. The test cube c_5 obtained from v_5 by CAT test relaxation keeps p_2 and p_{11} sensitized. *X*-filling for c_5 will produce a new fully-specified test vector nv_5, which also sensitizes p_2 and p_{11}. Since p_2 is risky under v_5, CAT *X*-filling should reduce the LSA in the critical area of p_2. This concept is illustrated in Fig. 8.

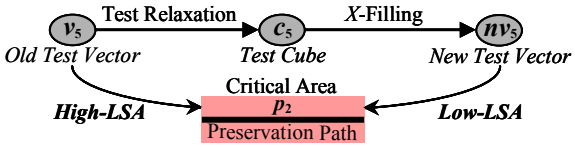

Fig. 8 Basic concept of CAT *X*-filling.

Clearly, a safe preservation path seldom becomes risky after CAT *X*-filling, since LSA is generally reduced. In addition, a long unsensitized path seldom becomes sensitized after CAT *X*-filling, since the conditions for sensitizing a long path are hard to met accidentally. Nonetheless, launch-safety checking (④) is conducted against the two possible cases.

CAT *X*-filling is conducted in two steps by using two unique techniques: *CAT clock-disabling* and *CAT FF-silencing*. Different from [15], these two critical-area-targeted (CAT) techniques are aimed at reducing the LSA in the critical area of a risky preservation path in a pinpoint manner.

5.1 Step-1: CAT Clock-Disabling

Clock-gating is a popular design technique for conditionally disabling clocks for some FFs, resulting in effective power reduction in a collective manner. Fig. 9 shows an example, where *CG* is a clock gator and setting *CG* to 0 disables all FFs controlled by *CG*, called the ***controlled FF*** of *CG*.

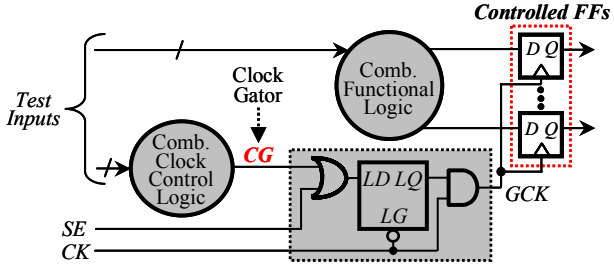

Fig. 9 Controlling FF and controlled FF.

In the following, we present a highly effective clock-disabling technique for reducing the LSA in the critical area of every risky preservation path.

Definition 10: A clock-gator *CG* is called a ***critical clock gator*** for a risky preservation path p under a test cube c if the following two conditions are satisfied: (1) the current value (at S_L in Fig. 1) of each controlled FF of *CG* under c is either *X* or a logic value that is equal to its next value (at C_1 in Fig. 1), and (2) at least one controlled FF of *CG* with the current value of *X* can reach the critical area of p.

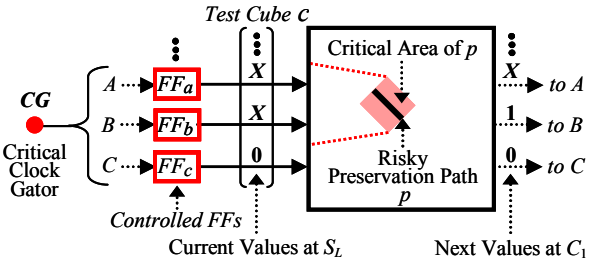

Fig. 10 Critical clock gator.

The usefulness of a critical clock gator is illustrated in Fig. 10. Here, setting 0 to clock gator *CG* will disable all of its controlled FFs: FF_a, FF_b, and FF_c. For FF_a and FF_b whose current values are *X*, disabling them has no impact on fault coverage / test quality. For FF_c whose current value is a logic value, since its current value (loaded by the last shift pulse S_L in Fig. 1) is equal to its next value (loaded by the stimulus launch pulse C_1 in Fig. 1), disabling it also has no impact on fault coverage / test quality. In addition, FF_a and FF_b can reach the critical area of the risky preservation path p. According to Definition 10, *GC* is a critical clock gator. This example demonstrates that disabling a critical clock gator can effectively reduce the LSA in a critical area without any impact on fault coverage and test quality.

The proposed ***CAT clock-disabling*** used in the first step of *X*-filling (③ in Fig. 2) is as follows:

CAT Clock-Disabling Procedure

(**1**) Identify all critical clock gators under a test cube c, and order them by the number of controlled FFs.

(**2**) Use *X*-bits in c to justify 0 to each critical clock gator, following the order determined in (**1**).

5.2 Step-2: CAT FF-Silencing

After justifying 0 to critical clock gators, *X*-bits may still remain in a test cube c. In this case, the second step of *X*-filling (③ in Fig. 2) is conducted on c by using the following ***CAT FF-silencing***. Its basic idea is to focus on the FFs that can reach the critical area of a risky preservation path and try to equalize their input and output values for the stimulus launch pulse (C_1 in Fig. 1) so as to reduce the launch switching activity caused by the FFs.

CAT FF-Silencing Procedure

(**1**) Order FFs by the # of nodes that an FF can reach in the critical areas of risky preservation paths of a test cube c.

(**2**) Process PPI-PPO pairs in c in the above order:

(***Type*-1**) <PPI=*X* / PPO=*val*>: Assign *val* (0 or 1) to the PPI.

(***Type*-2**) <PPI=*val* / PPO=*X*>: Justify *val* (0 or 1) to the PPO.

(***Type*-3**) <PPI=X_1 / PPO=X_2>: Assign 0 (1) to the PPI if $(P_0(X_2) - P_1(X_2)) > \Delta$ ($(P_1(X_2) - P_0(X_2)) > \Delta$), where Δ is the average difference between 0 and 1 probabilities of all *X*-bits in c, and $P_k(X_i)$ is the probability of X_i being k (0 or 1).

(**3**) If *X*-bits remain, run logic simulation and return to (**2**).

Improved from the JP-fill [12], CAT FF-silencing also uses a multi-pass *X*-filling scheme for achieving both effectiveness and scalability. By processing *X*-bits in the order of critical nodes for FF-silencing, more effective LSA reduction is achieved for the critical areas of risky preservation paths.

6. Experimental Results

The proposed CAT scheme was implemented in C and experiments were conducted using a workstation (2.9GHz-CPU/16GB-memory). Two industrial circuits were used for evaluation, and their statistics are shown in Table 3. Layout design was conducted with *SoC Encounter*™ (Cadence).

Table 3 Circuit Statistics

Circuit	# of *Gates*	# of *FF's*	# of *Clock-Gators*	*Max. Logic Level*
clk-S	50K	1,077	47	105
clk-L	600K	35,566	984	226

6.1 Evaluation of Launch-Safety Checking

Initial transition delay test vectors were generated using *TetraMAX*™ (Synopsys), and the results are shown under "*# of Vec.*" and "*Fault Cov.*" in Table 4. CAT launch-safety checking was conducted, in which the path length limit for defining a sensitive path was set as 40-50% of the maximum logic level, the radius for determining a critical area was set from layout information, and the LSA limit for determining a risky path was set as 20% of WSA in its critical area.

The results of CAT launch-safety checking are shown under "*# of Risky Vec.*" (the # of launch-risky vectors), "*Ave. Sen. Paths / Vec.*" (the average # of sensitive paths per test vector), and "*Ave. Risky Paths / Vec.*" (the average # of risky sensitive paths per launch-risky vector).

Table 4 Results of Launch-Safety Checking

Circuit	*# of Vec.*	*Fault Cov.*	*CAT Launch-Safe Checking*				*# of Risky Vec. with low WSA*
			# of Risky Vec.	*Ave. Sen. Paths /Vec.*	*Ave. Risky. Paths /Vec.*	*CPU (sec.)*	
clk-S	319	95.3	8	0.1	1.1	48	3
clk-L	191	85.1	11	0.2	3.8	2,772	5

For compassion, the # of risky vectors with low global WSA (i.e. less than the average WSA) is given as "*# of Risky Vec. with low WSA*". It shows that a launch-risky vector may have low global WSA. This means that conventional launch-safety checking based on global switching activity [3] is less accurate than critical-area-targeted (CAT) launch-safety checking.

6.2 Evaluation of Test Relaxation and *X*-Filling

CAT test relaxation and CAT *X*-filling were conducted and the results are shown in Table 5 under "*Ave. %X / Vec.*" (the average *X*-bit percentage per risky vector), "*Ave. Pres. Paths / Vec.*" (the average # of preservation paths per risky vector), and "*Ave. Risky Pres. Paths / Vec.*" (the average # of risky preservation paths per risky vector). "*Rescue Rate*" is the percentage of launch-risky vectors that became launch-safe after CAT test set modification was conducted. This result clearly demonstrates the effectiveness of the ***pinpoint*** CAT test set modification for LSA reduction.

Table 5 Results of Test Relaxation and *X*-Filling

Circuit	*# of Risky Vec.*	*CAT Test Relaxation & CAT X-Filling*				*Rescue Rate*
		Ave. %X /Vec.	*Ave. Pres. Paths /Vec.*	*Ave. Risky Pres. Paths /Vec.*	*CPU (sec.)*	
clk-S	8	85.6	168.5	1.13	421	37.5
clk-L	11	60.7	3,957.5	1.91	2424	63.6

6.3 Discussions

- The proposed CAT scheme can be directly applied to any combinational-decompressor-based test compression. This is achieved by *circuit remodeling* (i.e. expanding the decompressor as part of the circuit model) and there is no need to revise the CAT scheme itself. The details will appear in a separate paper [21]. In addition, the CAT scheme can be applied to sequential-decompressor-based test compression by using an extension technique [14].
- Launch-safety can be efficiently achieved in practice by a hybrid flow (Fig. 2) with the CAT test set modification (*which does not increase test vectors*) and the follow-up low-LSA ATPG (*which may significantly increase test vectors if used alone*). Since the CAT scheme rescues many of launch-risky test vectors, the test vector count inflation caused by low-LSA ATPG can be significantly reduced.

7. Conclusions

This paper proposed CAT, the first critical-area-based test set modification scheme that uses ***long sensitized paths*** to guide launch-safety checking, test relaxation, and *X*-filling. This ***pinpoint*** scheme improves launch-safety for at-speed scan testing more effectively than previous ***gross*** schemes, as evidenced by experiments on industrial circuits.

Future work includes (1) trying more power analysis metrics in CAT launch-safety checking and (2) evaluating the CAT scheme in compressed-scan circuits remodeled as in [21].

Acknowledgements

The authors thank STARC for providing experimental circuits.

References

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

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

[3] P. Girard, et al., *Low-Power Testing* (Chapter 7) in *Advanced SOC Test Architectures - Towards Nanometer Designs*, Morgan Kaufmann, 2007.

[4] J. Saxena, et al., "A Case Study of IR-Drop in Structured At-Speed Testing," *Proc. ITC.*, pp. 1098-1104, 2003.

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

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

[7] J. Wang, et al., "Modeling Power Supply Noise in Delay Testing," *IEEE Design and Test of Computers*, Vol. 24, No. 3, pp. 226-234, 2007.

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

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

[10] 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.

[11] S.-J. Wang, et al., "Low Capture Power Test Generation for Launch-off-Capture Transition Test Based on Don't-Care Filling," *Proc. ISCAS*, pp. 3683-3686, 2007.

[12] 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.

[13] J. Li, et al, "iFill: An Impact-Oriented *X*-Filling Method for Shift- and Capture-Power Reduction in At-Speed Scan-Based Testing," *DATE,* pp. 1184-1189, 2008.

[14] M.-F. Wu, et al., "Reducing Power Supply Noise in Linear-Decompressor-Based Test Data Compression Environment for At-Speed Scan Testing," *ITC*, Paper 13.1, 2008.

[15] H. Furukawa, et al., "CTX: A Clock-Gating-Based Test Relaxation and X-Filling Scheme for Reducing Yield Loss Risk in At-Speed Scan Testing," *Proc. ATS*, pp. 397-402, 2008.

[16] C.-W. Tzeng and S.-Y. Huang, "QC-Fill: An X-Fill Method for Quick-and-Cool Scan Test," *Proc. DATE*, pp. 1142-1147, 2009.

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

[18] J. Tyszer, et al., "Low-Power Scan Shift and Capture in the EDT Environment," *Proc. ITC*, Paper 13.2, 2008.

[19] J. Ma, et al., "Layout-Aware Pattern Generation for Maximizing Supply Noise Effects on Critical Paths," *Proc. VTS*, pp. 221-226, 2009.

[20] K. Miyase, et al., "XID: Don't Care Identification of Test Patterns for Combinational Circuits," *IEEE TCAD*, Vol.23, No.2, pp. 321-326, 2004.

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