# CTX: A Clock-Gating-Based Test Relaxation and *X*-Filling Scheme for Reducing Yield Loss Risk in At-Speed Scan Testing

H. Furukawa [1], X. Wen [1], K. Miyase [1], Y. Yamato [1], S. Kajihara [1], P. Girard [2], L.-T. Wang [3], and M. Tehranipoor [4]

[1] Kyushu Institute of Technology, Iizuka, Fukuoka 820-8502, Japan
[2] LIRMM, 161 rue Ada, 34392 Montpellier cedex 05, France
[3] SynTest Technologies, Inc., Sunnyvale, CA 94086, USA
[4] University of Connecticut, Storrs, CT 06269, USA

## Abstract

*At-speed scan testing is susceptible to yield loss risk due to power supply noise caused by excessive launch switching activity. This paper proposes a novel two-stage scheme, namely* ***CTX*** *(**C**lock-Gating-Based **T**est Relaxation and **X**-Filling), for reducing switching activity when test stimulus is launched. Test relaxation and X-filling are conducted (1) to make as many FFs inactive as possible by disabling corresponding clock-control signals of clock-gating circuitry in Stage-1 (**Clock-Disabling**), and (2) to make as many remaining active FFs as possible to have equal input and output values in Stage-2 (**FF-Silencing**). CTX effectively reduces launch switching activity, thus yield loss risk, even with a small number of don't care (X) bits as in test compression, without any impact on test data volume, fault coverage, performance, and circuit design.*

***Keywords***: Power Supply Noise, Test Relaxation, *X*-Filling, Clock-Gating, Test Compaction.

## 1. Introduction

***At-speed scan testing*** is mandatory for improving timing-related test quality [1]. It is realized by ***launching*** a transition at the start-point of a path and ***capturing*** its response at the end-point of the path at the system speed. In practice, the ***launch-on-capture*** (***LOC***) clocking scheme is widely used for at-speed scan testing [1].

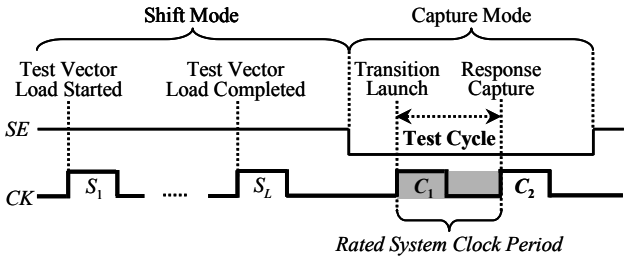



**Fig. 1 LOC-Based At-Speed Scan Testing.**

Fig. 1 shows the essence of the LOC clocking scheme: After a test vector is loaded by a series of shift clock pulses with $S_L$ being the last one ($L$: the length of the longest scan chain), transitions are launched by the first capture clock pulse $C_1$ at the corresponding scan FFs. The transitions are caused by the difference between the values shifted-in by $S_L$ and the values captured by $C_1$. The ***test cycle*** between the transition launch ($C_1$) and the response capture ($C_2$) is the rated system clock period.

Although at-speed scan testing is indispensable for improving timing-related test quality, its applicability is being severely challenged by ***test-induced yield loss***, which occurs when functionally good chips fail only during at-speed scan testing [2-4]. The major cause for this problem is power supply noise, i.e., IR-drop and ground bounce, caused by excessive ***launch switching activity*** at $C_1$, which results in delay increase. It has been shown that a 10% drop in power supply voltage can increase path delay by 30% [5]. Obviously, this may result in capture failures at $C_2$ [4], thus leading to test-induced yield loss [6-8]. This problem is worsening rapidly for deep-submicron and low-power chips [6]. Therefore, there is a strong need to reduce the yield loss risk caused by excessive power supply noise.

Previous techniques for reducing launch switching activity are based on the following three approaches:

(1) ***Partial Capture***: The number of FFs that capture at $C_1$ (Fig. 1) can be reduced by circuit modification [9], one-hot clocking [1], or capture clock staggering [1]. However, this approach may incur significant ATPG change, test data inflation, and even fault coverage loss.

(2) ***Low-Capture-Power ATPG***: 0's and 1's in a test vector can be carefully generated to make it have low launch switching activity, by using such techniques as input-output equalizing at FFs [10], clock-gating, etc. However, this approach may suffer from test data inflation as well as long CPU time.

(3) ***Test Relaxation & X-Filling***: ***Test relaxation*** is to identify don't-care bits (*X*-bits) from a set of fully-specified test vectors without any fault coverage loss [11, 12]. Then, ***X-filling*** is conducted on the resulting partially-specified test cubes to make as many FFs as possible to have equal input and output values [13-15], so as to reduce launch switching activity. The concept of this approach is illustrated in Fig. 2.

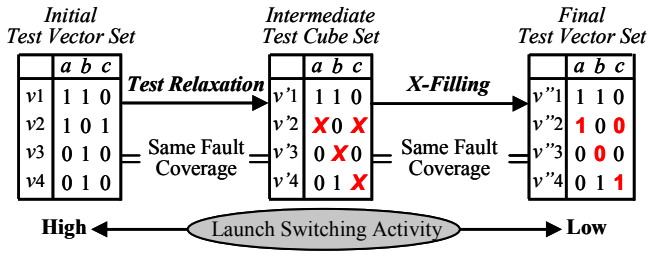



**Fig. 2 Concept of Test Relaxation and *X*-Filling.**

As a simple post-ATPG processing, the approach of test relaxation & *X*-filling is practical, since it reduces launch switching activity without any change in test data volume, fault coverage, performance, and circuit design. However, this approach may suffer from a major limitation: If the number of *X*-bits identified from a test set is small, the effect of launch switching activity reduction may be insufficient. Such *X*-bit shortage may be caused by test compaction or test compression. Fig. 3 shows an example of an industrial circuit (600K gates and 600 transition delay test vectors), for which test vectors obtained at the early stage of test generation have fewer (even less than 60%) *X*-bits.

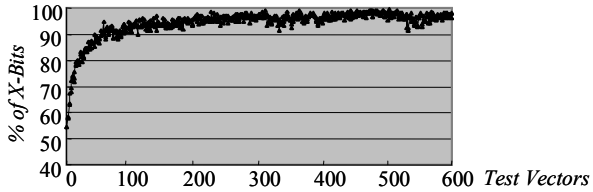
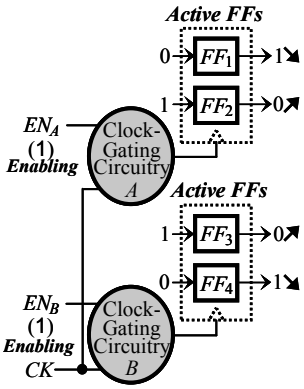
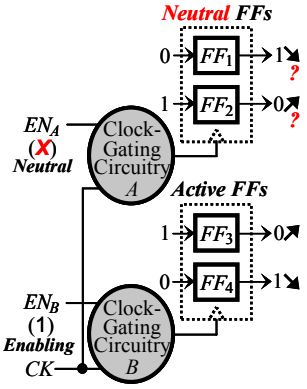



**Fig. 3 *X*-Bit Distribution for an Industrial Circuit.**

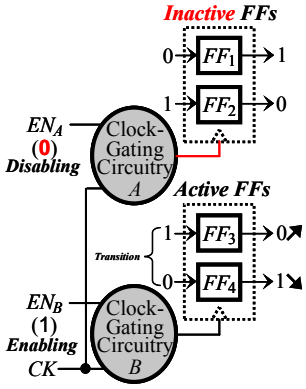
Therefore, there is a strong need to improve the test relaxation & *X*-filling approach so as to effectively reduce launch switching activity even with a small number of available *X*-bits as in test compaction and test compression. This paper achieves this goal by a novel two-stage scheme: ***CTX* (*Clock-Gating-Based Test Relaxation and X-Filling*)**. Its basic idea is to make sophisticated use of clock-gating in test relaxation & *X*-filling, as illustrated in Fig. 4.



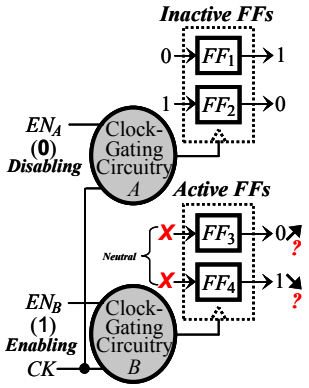

(a) State for Initial Test Vector



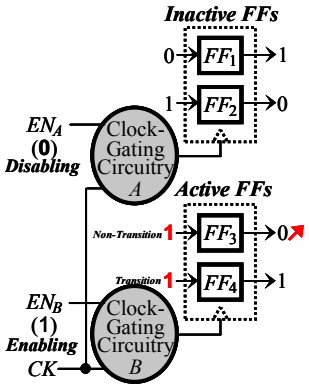

(b) After Test Relaxation in Stage-1




(c) After *X*-Filling in Stage-1




(d) After Test Relaxation in Stage-2




(e) After *X*-Filling in Stage-2

**Fig. 4 Basic Idea of CTX.**

Fig. 4 (a) shows that four FFs are controlled by two clock-control signals, $EN_A$ and $EN_B$, whose values are 1 for the initial test vector. This means that all the FFs are ***active***, i.e. capturing at $C_1$ (Fig. 1), resulting in four initial launch transitions. CTX consists of the following two stages:

- **Stage-1** (***Clock-Disabling***): Test relaxation is conducted to turn as many enabling clock-control signals (with value = 1) as possible into ***neutral*** ones (with value = *X*) as shown in Fig. 4 (b), and *X*-filling is then conducted to turn as many neutral clock-control signals as possible into disabling ones (with value = 0) as shown in Fig. 4 (c). Since all FFs controlled by a disabling clock-control signal (e.g., $FF_1$ and $FF_2$, in Fig. 4 (c)) are ***inactive***, launch transitions are efficiently reduced in a collective manner.
- **Stage-2** (***FF-Silencing***): Test relaxation is conducted to turn as many active transition-FFs (input ≠ output, e.g., $FF_3$ and $FF_4$ in Fig. 4 (c)) as possible into neutral-FFs (input or output = *X*, e.g., $FF_3$ and $FF_4$ in Fig. 4 (d)), and then *X*-filling is conducted to turn as many neutral-FFs as possible into non-transition-FFs (input = output, e.g., $FF_4$ in Fig. 4 (e)) by trying to equalize the input and the output of a neutral-FF. This way, the number of launch transitions at individual FFs is reduced.

The major contributions of the CTX scheme are as follows:

(1) ***Clock-Disabling-Based Test Relaxation & X-Filling***: CTX fully explores the collective power-reducing capability of clock-gating in at-speed scan testing.

(2) ***FF-Silencing-Based Test Relaxation & X-Filling***: CTX tries to equalize the input and output values of the remaining active FFs to further reduce launch transitions at individual FFs.

(3) ***Non-Intrusive Use of Clock-Gating*:** By using clock-gating through test data manipulation, CTX causes no ATPG change, test data inflation, or fault coverage loss.

(4) ***X-Bit-Efficiency***: With clock-disabling and FF-Silencing, CTX significantly reduce launch switching activity even with a small number of *X*-bits.

In the following, Section 2 describes the background, Section 3 presents the CTX scheme, Section 4 shows experimental results, and Section 5 concludes the paper.

## 2. Background

### 2.1 Test Relaxation

As illustrated in Fig. 2, ***test relaxation*** is the process of identifying don't-care bits (*X*-bits) from a fully-specified test vector set *V* to create a partially-specified test cube set *C*, while guaranteeing that some properties of *V* are preserved by *C*. Such properties generally include stuck-at fault coverage [11], transition delay fault coverage [12], and even all sensitized paths for transition delay fault detection [15].

*X*-bits can also be obtained directly from test generation by skipping random-fill. However, this increases ATPG time and test data volume. For example, the test vector count increased by 144.8% when random-fill was disabled to leave *X*-bits for low-power *X*-filling [13]. Thus, it is preferable to apply maximum test compaction by random-fill to generate a compact initial (*fully-specified*) test set first, and then use test relaxation to create a (*partially-specified*) test cube set. This results in a compact final test set with additional benefits after *X*-filling is conducted [13-15].

### 2.2 *X*-Filling

As illustrated in Fig. 2, ***X-filling*** is the process of assigning logic values to the *X*-bits in a test cube for a specific purpose. In this paper, *X*-filling is used for reducing ***launch switching activity*** (***LSA***), which occurs at $C_1$ (Fig. 1). Many ***low-LSA*** *X*-filling techniques have been proposed [13-15], and Fig. 5 illustrates one of them, called ***JP-fill*** [15].

***JP-Fill***: In Fig. 5, the test cube is $c$ = <10*XX*> and the logic function of the combinational portion is *F*. Thus, <*c*: PPI> = <0*XX*> and <*F*(*c*): PPO> = <*XXX*>. First, ***justification*** (①) is conducted to try to set 0 to $p_2$ since $p_1$ is 0. Then, for the $q_1$-$q_2$ and $r_1$-$r_2$ bit-pairs of the form *X*-*X*, the 0 and 1 probabilities of each PPO *X*-bit are calculated by setting 0.50 as the 0 and 1 probabilities for each PI or PPI input *X*-bit and conducting ***probability*** propagation. Since the 0-probability (0.93) of $q_2$ is significantly larger than its 1-probability (0.07), it is reasonable to assign 0 to $q_1$ (②). However, since the 0-probability (0.49) of $r_2$ is close to its 1-probability (0.51), no decision is made for $r_1$ (③). In this case, 3-valued logic simulation is conducted, and another pass of JP-Fill (④) is conducted with justification and/or probability propagation.

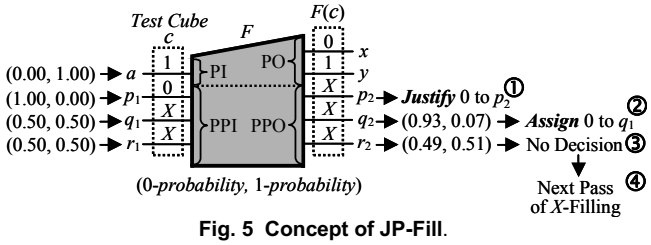



**Fig. 5 Concept of JP-Fill.**

Obviously, JP-fill uses justification and multiple passes to improve its effectiveness, and uses probability propagation to improve its scalability. This way, JP-fill achieves both effectiveness and scalability in a balanced manner.

### 2.3 Clock-Gating

***Clock-gating*** is the most widely used power management mechanism in practice. Fig. 6 shows an example scheme.

A circuit may contain multiple clock-gating blocks. Fig. 6 shows the *i*-th one, which is enhanced for scan testing. In shift mode (*SE* = 1), all FFs are constantly driven by the clock *CK* so that the shift operation is properly conducted. The ***clock-control signal*** ($EN_i$) is generated at the rising edge of *CK* and takes over the control of $GEN_i$ in capture mode (*SE* = 0). $GEN_i$ is then ANDed with *CK* to create the ***gated clock*** $GCK_i$ that is directly connected to the FFs.

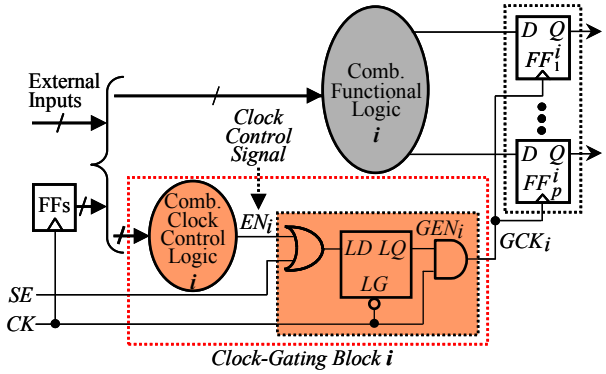



**Fig. 6 Example Scheme of Clock-Gating.**

Clock-gating can be used to reduce launch switching activity in capture mode. This is achieved by setting $EN_i$ to 0 at the last shift pulse $S_L$. This way, the launch capture clock pulse ($C_1$ in Fig. 1) is suppressed as shown in Fig. 7, and all FFs controlled by $GCK_i$, i.e., $FF_1^i$ ~ $FF_p^i$ as shown in Fig. 6, will not capture. As a result, launch transitions at the FFs are reduced in a ***collective*** manner.

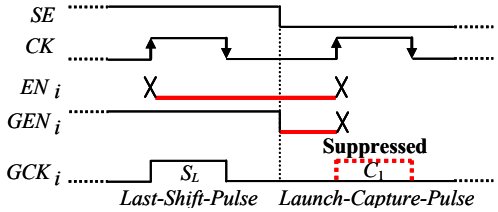



**Fig. 7 Clock-Gating for Reducing Launch Switching Activity.**

**Definition 1**: If the clock of a FF is a gated clock, the FF is called a ***clock-gated FF***, otherwise the FF is called a ***non-clock-gated FF***. The group of FFs controlled by the same gated clock is called a ***clock-gated FF group***.

For example, Fig. 4 shows two clock-gated FF groups: {$FF_1$, $FF_2$} and {$FF_3$, $FF_4$}. Note that all FFs connected through one clock tree are considered as one clock-gated FF group.

The capability of collectively reducing launch switching activity by disabling the clock to a group of FFs makes clock-gating a highly effective technique in reducing the yield loss risk in at-speed scan testing, especially when

the percentage of available *X*-bits in a test cube is low, such as in test compaction and test compression where the effect of conventional *X*-filling on LSA reduction is often limited.

From the ATPG point of view, clock-gating can be utilized by the following two basic approaches:

***Approach*-1** (***Detection-Oriented***): Test generation is conducted in a manner that activates clocks to FFs (i.e., disabling clock-gating) as much as possible, in order to make more FFs available for launch and capture in transition fault detection. Most commercial ATPG systems use this approach explicitly or implicitly. This leads to a smaller test set, higher fault coverage, and shorter CPU time, at the cost of higher launch switching activity.

***Approach*-2** (***Reduction-Oriented***): In test generation, clock-gating can be explicitly used for reducing launch switching activity. However, this reduces the number of FFs available for launch and capture in transition fault detection, leading to test vector count inflation, longer CPU time, and even fault coverage loss under certain conditions.

In practice, it is preferable to use the detection-oriented approach (Approach-1) to generate a compact initial test set first, and then convert it into a final test set with reduced launch switching activity by using CTX, whose basic idea has been illustrated in Fig. 4. The details of CTX is described in the following section.

## 3. The CTX Scheme

### 3.1 Problem Formalization

The problem of clock-gating-based reduction of launch switching activity in at-speed scan testing is as follows:

---

Convert a given set of test vectors, $V_{initial}$, to a new set of test vectors, $V_{final}$, under the following conditions:

(1) The fault coverage of $V_{final}$ is no less than that of $V_{initial}$.
(2) The size of $V_{final}$ is equal to that of $V_{initial}$.
(3) The peak launch switching activity of $V_{final}$ is made lower than that of $V_{initial}$ as much as possible by utilizing clock-disabling and FF-silencing.

---

### 3.2 Basic Concept

This paper proposes a novel two-stage scheme, namely ***CTX*** (***Clock-Gating-Based Test Relaxation and X-Filling***), to solve the above problem. The following terms are useful in describing the procedure of CTX:

**Definition 2**: If the value of a clock-control signal *EN* under an input vector *v* is 1, 0, and X, respectively, *EN* is said to be ***enabling***, ***disabling***, and ***neutral*** under *v*, respectively.

For example, $EN_A$ in Fig. 4 (a), (b), and (c) are enabling, neutral, and disabling clock-control signals, respectively.

**Definition 3**: If the clock to a FF or a clock-gated FF group is enabled (disabled), the FF or the clock-gated FF group is said to be ***active*** (***inactive***).

For example, $FF_1$ and $FF_2$ ($FF_3$ and $FF_4$) in Fig. 4 (c) are inactive (active) FFs. It can also be said that the clock-gated FF group $\{FF_1, FF_2\}$ ($\{FF_3, FF_4\}$) is inactive (active).

**Definition 4**: If the input and the output of a FF have ***different*** (***identical***) logic values, the FF is said to be a ***transition-FF*** (***non-transition-FF***). If the input or the output of a FF is *X*, the FF is said to be a ***neutral-FF***.

For example, both $FF_3$ and $FF_4$ in Fig. 4 (d) are neutral-FFs. In Fig. 4 (e), $FF_3$ is a transition-FF, $FF_4$ is a non-transition-FF. Note that both $FF_3$ and $FF_4$ in Fig. 4 (e) are active.

CTX is based on the following two observations:

***Observation*-1**: Not all enabling clock-control signals really need to be enabling. For example, a clock-control signal may be made enabling to detect a fault in a test vector, but later the fault may be detected by another test vector that does not require the clock-control signal to be enabling.

***Observation*-2**: Even if only one FF needs to be active for fault detection, all other FFs in the same clock-gated FF group also have to be active since they share the same clock-control signal. This results in redundant transitions especially when clock-gating is coarse-grained.

The basic concept of CTX is summarized in Fig. 8.

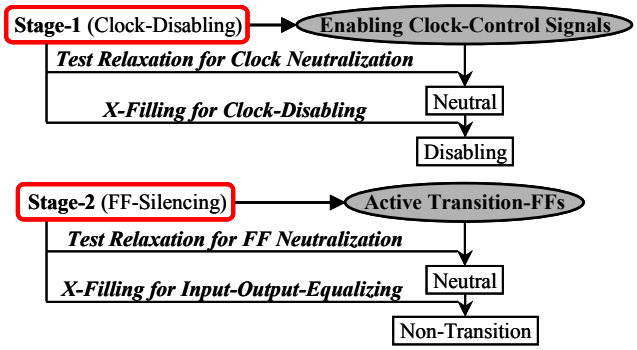



**Fig. 8 Basic Concept of CTX.**

The first observation leads to the Stage-1 (***Clock-Disabling***) of CTX as shown in Fig. 8. Test relaxation is conducted to turn as many enabling clock-control signals as possible into neutral ones, and then *X*-filling is conducted to make as many neutral clock-control signals as possible into disabling ones. This way, launch switching activity can be reduced efficiently at the clock-gated FF group level, due to the collective reduction capability of clock gating.

The second observation leads to the Stage-2 (***FF-Silencing***) of CTX as shown in Fig. 8. Test relaxation is conducted to turn as many active transition-FFs as possible into neutral-FFs, and then *X*-filling is conducted to make as many neutral-FFs as possible into non-

transition-FFs. This way, launch switching activity can be further reduced at the individual FF level.

An example of applying the basic idea of CTX has been shown in Fig. 4. In the following, the details of the CTX procedure are presented.

### 3.3 Circuit Model

Fig. 9 shows the model of a circuit containing $m$ clock-gating blocks, for the purpose of test generation for CTX.

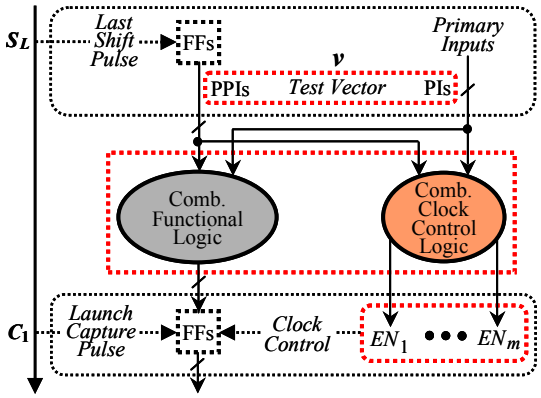



**Fig. 9 Circuit Model for CTX.**

As shown in Fig. 9, a fully-specified test vector $v$ is loaded at the rising edge of the last shift pulse $S_L$. $v$ consists of the PPI part, <v: PPI>, corresponding to the outputs of FFs, and the PI part, <$v$: PI>, corresponding to primary inputs. The combinational clock-control logic circuitry produces $m$ clock-control signals, $EN_1$, $EN_2$, …, $EN_m$, corresponding to the $m$ clock-gating blocks as the one shown in Fig. 6. Each of the clock-control signals may be disabling or enabling, and determines whether the corresponding clock-gated FF group is active or not at the launch capture pulse $C_1$.

### 3.4 CTX Procedure

Based on the basic concept of CTX shown in Fig. 8, the CTX procedure can be described as follows:

---

***Procedure of CTX*:**

**Input**: $V^0 = \{ v_i^0 \mid i = 1, 2, \ldots, n\}$ // initial test set

**Output**: $V^2 = \{ v_i^2 \mid i = 1, 2, \ldots, n\}$ // final test set

**Stage-1 (*Clock-Disabling*)**:

**(1-1)** Obtain $RS(v_i^0) = \{$bits in $v_i^0$ that are reachable from at least one enabling clock-control signal under $v_i^0\}$ for $i = 1, 2, \ldots, n$.

**(1-2)** Obtain $T^1 = RS(v_1^0) \cup RS(v_2^0) \ldots \cup RS(v_n^0)$.

**(1-3)** Conduct constrained test relaxation on $V_0$ to turn as many bits in $T^1$ into $X$-bits as possible, while preserving the fault coverage of $V_0$. Denote the set of resulting partially-specified test cubes by $C^1 = \{ c_i^1 \mid i = 1, 2, \ldots, n\}$.

**(1-4)** Try to justify 0 on each neutral clock-control signal under $c_i^1$ by setting appropriate logic values to some $X$-bits in $c_i^1$ for $i = 1, 2, \ldots, n$.

**(1-5)** Conduct low-LSA $X$-filling for the remaining $X$-bits in $c_i^1$ for $i = 1, 2, \ldots, n$. Denote the set of resulting fully-specified test vectors by $V^1 = \{ v_i^1 \mid i = 1, 2, \ldots, n\}$.

**Stage-2 (*FF-Silencing*)**:

**(2-1)** Obtain $RF(v_i^1) = \{$bits in $v_i^1$ that correspond to active transition-FFs under $v_i^1\}$ for $i = 1, 2, \ldots, n$.

**(2-2)** Obtain $T^2 = RF(v_1^1) \cup RF(v_2^1) \ldots \cup RF(v_n^1)$.

**(2-3)** Conduct constrained test relaxation on $V_1$ to turn as many bits in $T^2$ into $X$-bits as possible, while preserving (1) fault coverage of $V_1$ and (2) the logic values of all clock-control signals. Denote the set of resulting partially-specified test cubes by $C^2 = \{ c_i^2 \mid i = 1, 2, \ldots, n\}$.

**(2-4)** Conduct low-LSA $X$-filling for the $X$-bits in $c_i^2$ for $i = 1, 2, \ldots, n$. Denote the set of resulting fully-specified test vectors by $V^2 = \{ v_i^2 \mid i = 1, 2, \ldots, n\}$.

---

### 3.5 Constrained Test Relaxation

The key operation in CTX described in 3.4 is ***constrained test relaxation*** conducted on a fully-specified test set $V$ to turn as many target bits in $T$ into $X$-bits as possible, while preserving (1) the fault coverage of $V$ and (2) the logic values of target lines in $S$. The result is $C$, a set of partially-specified test cubes. This is illustrated in Fig. 10, where $*$ indicates a target bit position. Note that only fault coverage needs to be preserved in the constrained test relaxation of Step (1-3), for which $S$ can be set $\varnothing$.

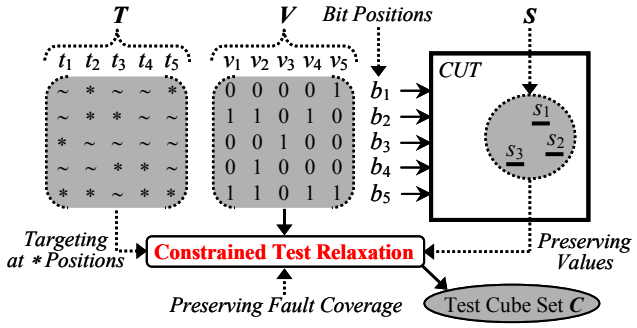



**Fig. 10 Basic Concept of Constrained Test Relaxation.**

Constrained test relaxation is conducted with the following procedure obtained by extending a basic non-constrained test relaxation procedure we previously proposed [12].

---

***Procedure of Constrained Test Relaxation*:**

***Input*** ***V***: *set of fully-specified test vectors*
***T***: *set of target bits*
***S***: *set of target lines* (= $\varnothing$ *in (1-3) of CTX*)

***Output*** ***C***: *set of resulting partially-specified test cubes*

**S-1:** Identify all essential faults of ***V***, each of which can only be detected by one test vector in ***V***, by an efficient procedure based on two passes of fault simulation.

**S-2**: Identify all bits in ***V*** whose logic values are needed for (1) detecting all essential faults and (2) preserving logic values of all target lines in ***S***, in such a manner that bits in ***T*** are avoided as much as possible. Such bits can be readily found by making use of the justification operation also widely applied in ATPG. Then, turn the identified bits in *V* into *X*-bits, and this results in an intermediate test cube set, ***C'***.

**S-3**: Conduct 3-valued fault simulation on ***C'*** for all non-essential faults of ***V***. For all undetected non-essential faults, identify the *X*-bits whose logic values in ***V*** are needed for detecting them, in such a manner that bits in ***T*** are avoided as much as possible. Then, restore the identified *X*-bits with their original logic values in ***V***. This results in the final test cube set, ***C***.

---

The time complexity of the constrained test relaxation procedure is O(*M*×*N*), where *M* and *N* as the number of faults and the number of test vectors, respectively.

## 4. Experimental Results

CTX was implemented in C for evaluation experiments. Since there are no available benchmark circuits with gated clocks, we created two test circuits, TC-1 and TC-2, as shown in Table 1, by synthesizing the open-source microprocessor design *picoJava* [16]. Note that one clock-gating block corresponds to one clock-gated FF group.

**Table 1 Evaluation Circuits**

| *Circuit* | # of *Gates* | # of *FF's* | # of *Clock-Gated FF's* | *% of Clock-Gated FF's* | # of *Clock Gating Blocks* | *Ave. # FF's / Clock Gating Block* |
|---|---|---|---|---|---|---|
| TC-1 | 207.8K | 6,992 | 5,532 | 79.4% | 3,441 | 1.6 |
| TC-2 | 191.5K | 6,992 | 2,176 | 31.1% | 65 | 33.5 |

Evaluation experiments were conducted on TC-1 and TC-2, and the results are summarized in Table 2.

Transition LOC delay test vectors were generated by *TetraMAX*™. Test vector count and fault coverage are shown under "*# of Vec.*" and "*Fault Cov. (%)*", respectively. Three experiments were conducted using (1) XID with Preferred-Fill [13], (2) XID with JP-Fill [15], and (3) the proposed CTX scheme, where XID is a test relaxation system developed based on [12] without using gated clocks. Reduction ratio of launch switching activity w.r.t. the original test set, measured by the WSA metric [13], are shown under "*XID+Preferred*", "*XID+JP*", and "*CTX*", respectively.

**Table 2 Experimental Results**

| *Circuit* | *# of Vec.* | *Fault Cov. (%)* | *Reduction Ratio of Launch Switching Activity* | | |
|---|---|---|---|---|---|
| | | | *XID + Preferred* | *XID + JP* | *CTX* |
| TC-1 | 4025 | 84.7 | 15.2 | 15.9 | 23.9 |
| TC-2 | 3575 | 87.8 | 15.6 | 17.2 | 17.4 |

Table 2 shows that the CTX scheme is more effective than the previous schemes based on conventional test relaxation [12] and *X*-filling [13, 15]. It also shows that CTX is especially effective for low-power devices, in which the clock-gating mechanism is massively used.

### *C. Discussions*

(1) CTX is ***non-intrusive*** in reducing yield loss risk for at-speed scan testing, in that CTX causes no test data inflation, no fault coverage loss, no circuit/clocking modification, and no circuit performance degradation. This makes it easy to incorporate CTX into any test generation flow.

(2) CTX is ***applicable to test compaction and test compression*** where the percentage of *X*-bits in a test cube may be limited. CTX makes up for such *X*-bit shortage by making full use of clock-gating, in a sophisticated manner that the size of the original test set remains changed.

## 5. Conclusions

This paper proposed a novel two-stage scheme, namely **CTX** (***C****lock-Gating-Based* ***T****est Relaxation and* ***X****-Filling*), for effectively reducing launch switching activity that may cause yield loss in at-speed scan testing. The basic idea is to use clock-gating to disable as many as possible FFs that do not contribute to fault detection. CTX is the first of its kind that makes full use of clock-gating in test relaxation and *X*-filling. CTX reduces yield loss risk without any impact on test size, fault coverage, ATPG, circuit or clock design, and functional performance. Furthermore, CTX is applicable to any test compression scheme where *X*-bits are limited.

Future work includes: (1) conducting more evaluation experiments on more industrial circuits with gated clocks, (2) evaluating CTX in a test compression flow, and (3) conducting IR-drop and timing analysis to further demonstrate the effectiveness of the CTX scheme.

## References


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

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

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

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

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

[6] S. Ravi, "Power-Aware Test: Challenges and Solutions," *Proc. Int'l Test Conf.*, Lecture 2.2, 2007.

[7] 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.

[8] 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.

[9] S. Wang, et al., "A Technique to Reduce Peak Current and Average Power Dissipation in Scan Designs by Limited Capture," *Proc. Asian S. 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] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] SCSL: *http://www.sun.com/software/communitysource/processors/download_picojava.xml*