IEICE TRANS. INF. & SYST., VOL.E97–D, NO.10 OCTOBER 2014

PAPER

# On Achieving Capture Power Safety in At-Speed Scan-Based Logic BIST*

**Akihiro TOMITA**[†], ***Nonmember***, **Xiaoqing WEN**[†a)], **Yasuo SATO**[†], **Seiji KAJIHARA**[†], **Kohei MIYASE**[†], ***Members***, **Stefan HOLST**[†], **Patrick GIRARD**[††], **Mohammad TEHRANIPOOR**[†††], ***and*** **Laung-Terng WANG**[††††], ***Nonmembers***

**SUMMARY** The applicability of at-speed scan-based logic ***built-in self-test*** (***BIST***) is being severely challenged by excessive capture power that may cause erroneous test responses even for good circuits. Different from conventional low-power BIST, this paper is the first to explicitly focus on achieving ***capture power safety*** with a novel and practical scheme, called ***capture-power-safe logic BIST*** (***CPS-LBIST***). The basic idea is to identify all possibly-erroneous test responses caused by excessive capture power and use the well-known approach of masking (***bit-masking, slice-masking***, ***vector-masking***) to block them from reaching the ***multiple-input signature register*** (***MISR***). Experiments with large benchmark circuits and a large industrial circuit demonstrate that CPS-LBIST can achieve capture power safety with negligible impact on test quality and circuit overhead.

***key words:*** *at-speed scan-based logic BIST, capture power safety, masking, IR-drop, transition delay fault, long sensitized path*

## 1. Introduction

### 1.1 Importance of Test Power Safety

***Scan design*** is the foundation for both ***stored pattern testing*** (using tester-applied pre-generated test vectors) and ***built-in self-test*** (***BIST***) (using on-chip-generated pseudo-random test vectors) [1], [2]. Scan design enables ***scan testing***, in which test vectors are shifted-in and test responses are shifted-out simultaneously via scan chains in shift mode and test responses are loaded into individual flip-flops in capture mode. Furthermore, ***at-speed scan testing***, in which the test cycle time is set to match the function clock cycle time, has become indispensable for delay testing [3].

However, power dissipation in scan testing is often excessive [4]–[8] and may result in adverse effects. This is illustrated in Fig. 1 by considering the ***launch-on-capture*** (***LOC***) clocking scheme for at-speed scan testing. ***Shift power*** is caused by shift switching activity in the whole circuit due to the consecutive application of shift clock pulses.

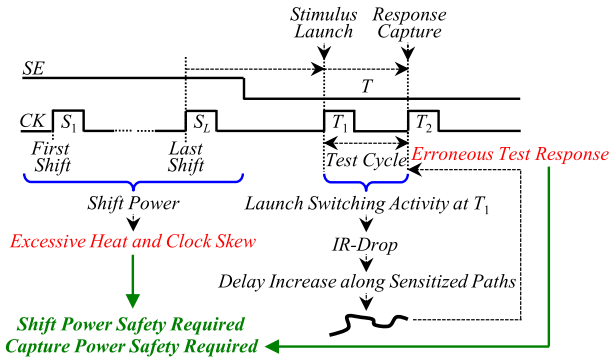



**Fig. 1** Scan test power safety requirements.

The accumulative impact of shift power is excessive heat that may result in chip damage, reliability degradation, and timing-related yield loss [6]–[8]. The instantaneous impact of shift power is excessive clock skew in the clock tree, which may make it impossible to correctly shift-in a new test vector and shift-out the current test response [9]. On the other hand, ***capture power*** is caused by the ***launch switching activity*** (***LSA***) triggered by the stimulus launch clock pulse at $T_1$, causing IR-Drop in the ***power distribution network*** (***PDN***) that results in delay increase along sensitized paths. The instantaneous impact of excessive capture power is significant delay increase, which may cause test responses to be incorrectly captured at the end-points of some sensitized paths at $T_2$, leading to over-kill-induced yield loss (i.e., failing good chips). Therefore, ***test power safety***, including both ***shift power safety*** and ***capture power safety***, need to be guaranteed in order to conduct scan testing [10]–[12] successfully, in the form of both stored pattern testing and logic BIST.

### 1.2 From Low-Power Testing to Power-Safe Testing

The test power issue has been conventionally tackled with ***low-power testing*** [4]–[8], which tries to reduce shift power or capture power or both by *circuit modification*, *test data manipulation*, *test clock adjustment*, etc. However, most of the previous low-power testing techniques can only reduce global power in a coarse-grained manner. As a result, they cannot guarantee that test power, in the whole circuit as well as in all local circuit regions, can always be sufficiently reduced.

Manuscript received February 7, 2014.
Manuscript revised May 30, 2014.

[†]The authors are with Kyushu Institute of Technology, Iizuka-shi, 820–8502 Japan.
[††]The author is with LIRMM, 161 rue Ada, 34095, Montpellier, France.
[†††]The author is with University of Connecticut, Storrs, CT 06296, USA.
[††††]The author is with SynTest Technologies, Inc., Sunnyvale, CA 94086, USA.

*This paper is the extended version of a paper presented at Asian Test Symposium in November 2013.

a) E-mail: wen@cse.kyutech.ac.jp

DOI: 10.1587/transinf.2014EDP7039

• *Toward Shift Power Safety*

There are a few solutions for achieving shift power safety. An example is ***scan segmentation*** [9], [13], in which each original scan chain is split into $N$ shorter segments and only one segment is shifted at a time. It reduces scan shift transitions to $1/N$ without increasing scan shift time. By properly selecting $N$ based on circuit characteristics and package materials, the heat impact of shift power can be effectively and predictably managed. This technique is applicable for both stored pattern testing (including compressed scan testing) and logic BIST. For logic BIST, techniques based on ***suppression*** (toggle and pattern) are also effective in achieving shift power safety [14]–[16].

• *Toward Capture Power Safety*

A typical capture-power-safe solution for stored pattern testing is ***rescue-&-masking*** [11], [12], in which (1) the local switching activity around each ***long sensitized path*** (***LSP***) of a test vector is checked to determine if it is a ***risky path*** (i.e., an LSP whose surrounding switching activity is so high that the test response from the LSP is ***possibly-erroneous*** as an uncertain value); (2) for any risky path, $X$-filling [17] is conducted in a pinpoint manner to directly reduce its surrounding switching activity; (3) if the effect of switching activity reduction is insufficient to turn a risky path into a non-risky path, the uncertain test response from the risky path will be masked to instruct the tester not to use it. This way, any adverse impact of excessive capture power on final test results is avoided, thus realizing capture power safety.

However, *rescue-&-masking* cannot be applied to scan-based logic BIST because (1) the *rescue* process is conducted by changing test vectors with $X$-filling and (2) the *masking* process is conducted by changing test responses. Both processes are impossible in the logic BIST environment. Therefore, this paper will focus on how to achieve capture power safety for logic BIST.

### 1.3 Capture Power Safety Problem in Logic BIST

As shown in Fig. 2, test stimuli in logic BIST are generated by an on-chip ***pseudo-random pattern generator*** (***PRPG***) and test responses are analyzed by an on-chip ***multiple-input signature register*** (***MISR***) [1], [2]. Additionally, logic BIST includes a ***phase shifter*** for reducing inter-pattern bit value correlations, a ***compactor*** for reducing the MISR bit-width, and a ***BIST controller*** for coordinating all BIST operations. These blocks are collectively called ***BIST-specific blocks***. The original circuit is converted into a ***BIST-ready circuit*** by *scan insertion*, *X-bounding*, and *test point insertion* [1], [2].

Suppose that the LOC clocking of Fig. 1 is applied to the logic BIST of Fig. 2. In the design stage, capture power analysis may find that the launch switching activity at $T_1$ (also shown in Fig. 1) causes excessive switching activity around some long sensitized paths in the combinational logic portion of the BIST-ready circuit. These paths

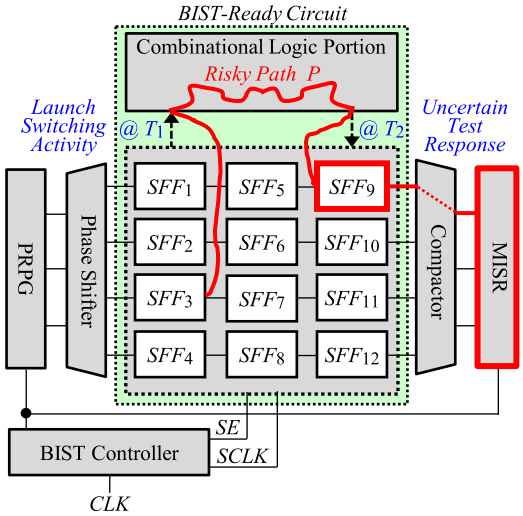



**Fig. 2** Capture power safety problem in logic BIST.

are ***risky paths*** (e.g., $P$ in Fig. 2) since test responses from them are ***possibly-erroneous*** (i.e., test response values become ***uncertain***) [11], [12]. At $T_2$ (also shown in Fig. 1), such ***uncertain test responses*** are loaded into some scan flip-flops (e.g., $SSF_9$ in Fig. 2). Thus, when test responses are shifted-out to the MISR during subsequent shift operations, uncertain test responses will corrupt the MISR content to falsely fail a good chip.

Many low-power logic BIST techniques have been proposed over the years [14]–[16], with most of them focused on shift power reduction. Although some of them also help reduce capture power, the reduction is mostly global and there is no guarantee that all risky paths can be predictably removed. This means that conventional low-power logic BIST techniques cannot always guarantee capture power safety. This is a severe problem due to the fact that the capture power of logic BIST is much higher than that of stored pattern testing since pseudo-random test stimuli are applied [15].

### 1.4 Contributions of This Paper

Hardware-based masking is widely employed in such applications of $X$-bounding in scan-based logic BIST, switching activity blocking for shift power reduction, etc. [2], [15], [18], [19]. It can be easily implemented by ANDing or ORing a signal to be masked with the controlling value of the mask gate (1 for AND and 0 for OR). It is a straightforward idea to apply masking for achieving capture power safety. For example, the uncertain test response at the output of $SSF_9$ (shown in Fig. 2) can be masked, for example, by inserting a 2-input AND gate whose control input can be set to 0, to prevent it from reaching the MISR. However, there has been no report so far on using hardware-based masking to achieve capture power safety. The obvious reason is the perception that such masking may severely impact test quality and circuit overhead in logic BIST.

This paper is the first to explicitly focus on achieving

capture power safety in scan-based logic BIST by demonstrating the feasibility of applying the well-known approach of masking [15], [18], [19]. The major contributions of this paper are as follows:

(1) It reveals an important property that uncertain test responses in logic BIST are actually very few. This indicates that it is feasible to achieve capture power safety by masking uncertain test responses.
(2) Three options, ***bit-masking***, ***slice-masking*** and ***vector-masking***, are used to realize a ***capture-power-safe logic BIST*** (***CPS-LBIST***) scheme. Comprehensive experiments show that the impact of masking on test quality and circuit overhead is negligible.

### 1.5 Paper Organization

Section 2 describes the background, Sect. 3 presents the details of CPS-LBIST, Sect. 4 reports experimental results, and Sect. 5 concludes the paper.

## 2. Background

### 2.1 LSP-Based Capture Power Safety Checking

All gates in a circuit share the ***power distribution network*** (***PDN***) driven by the external power supply, and the effective supply voltage of a gate $G$ provided from the PDN is affected by the amount of current drawn by its neighboring gates (called the *aggressor region* of $G$) from where $G$ is connected to the PDN [3]. The more aggressors make transitions due to the launch switching activity at $T_1$ (shown in Fig. 1), the larger current will be drawn and thus the lower effective voltage of $G$, resulting in increased gate delay of $G$ [6]–[8].

Generally, the root cause of the capture power problem is excessive launch switching activity at $T_1$ (shown in Fig. 1) since it may cause excessive local switching activity around a sensitized path, resulting in excessive delay increase along the path and consequently a timing failure at $T_2$ (shown in Fig. 1). It is obvious that a ***long sensitized path*** (***LSP***) is vulnerable to such timing impact of excessive capture power [11], [12], [20]. Thus, capture power safety checking is preferably conducted with a LSP-based approach based on the following definitions [11], [12]:

**Definition 1**: The ***aggressor region*** of a gate $G$, denoted by $AR(G)$, is composed of aggressor nodes (gates and flip-flops) whose transitions have a strong impact on the supply voltage of $G$.

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

**Definition 3**: A path $P$ is a ***risky path*** under a test vector $V$ if (1) $P$ is long (w.r.t. Threshold-I), (2) $P$ is sensitized by $V$, and (3) the launch switching activity at $T_1$ (shown in Fig. 1) in $IA(P)$ is excessive (w.r.t. Threshold-II).

Details about the thresholds in Definition 3 are as follows: A path is long if its length is greater than Threshold-I (`Path-Limit`), which is usually set as a percentage of the longest path (measured by its delay or logical level) in a circuit. It is noteworthy that it is not necessary to list all paths in order to identify such long paths. One can use an algorithm [21] that directly identifies paths that are longer than a threshold (i.e., `Path-Limit`). Furthermore, whether the launch switching activity in the impact area of a path is excessive can be determined by checking the ***weighted switching activity*** (***WSA***) [22] in the impact area of the path since the local WSA has a strong correlation with the delay increase along the path [8], [20], [23]. Launch switching activity in the impact area of a path is considered excessive if the WSA in the impact area is higher than Threshold-II (`WSA-Limit`), which is usually set as a percentage of the maximum WSA in the impact area.

**Definition 4**: A test response bit is a ***risky response bit*** if it corresponds to the end point of a risky path.

It is clear from the above description that a risky response bit is ***possibly-erroneous***. This means that a risky response bit needs to be treated as an ***uncertain*** value in the test design stage. In this sense, a risky response bit is also a type of unknown value ($X$).

**Definition 5**: A response slice is a ***risky response slice*** if it contains at least one risky response bit.

**Definition 6**: The response vector for a test vector $V$ is a ***risky response vector*** if there is at least one risky path under $V$.

**Definition 7**: Scan-based logic BIST is ***capture-power-safe*** if none of its pseudo-random test vectors is risky.

These definitions are illustrated in Fig. 3. Figure 3 (a) shows how to identify the aggressor region for gate $G_1$ by using layout and PDN data [8], [23]; Fig. 3 (b) shows how to identify the impact area of a path $P$; Fig. 3 (c) shows a risky path $P$, its corresponding risky response bit at the end point of $P$ (i.e., the output of the scan flip-flop $SFF_9$), as well as the risky response slice and the risky response vector.

Such LSP-based capture power safety checking is conducted for all pseudo-random test vectors to be applied in scan-based logic BIST. Position information on risky response bits, risky response slices, and risky response vectors is obtained and stored for later use in the CPS-LBIST design flow to be described in Sect. 3.

### 2.2 Property of Uncertain Test Responses in Logic BIST

As described above, excessive capture power may cause uncertain test responses in logic BIST. Since logic BIST applies a large number of pseudo-random test vectors, the conventional perception is that logic BIST must have a large number of uncertain test responses. However, our

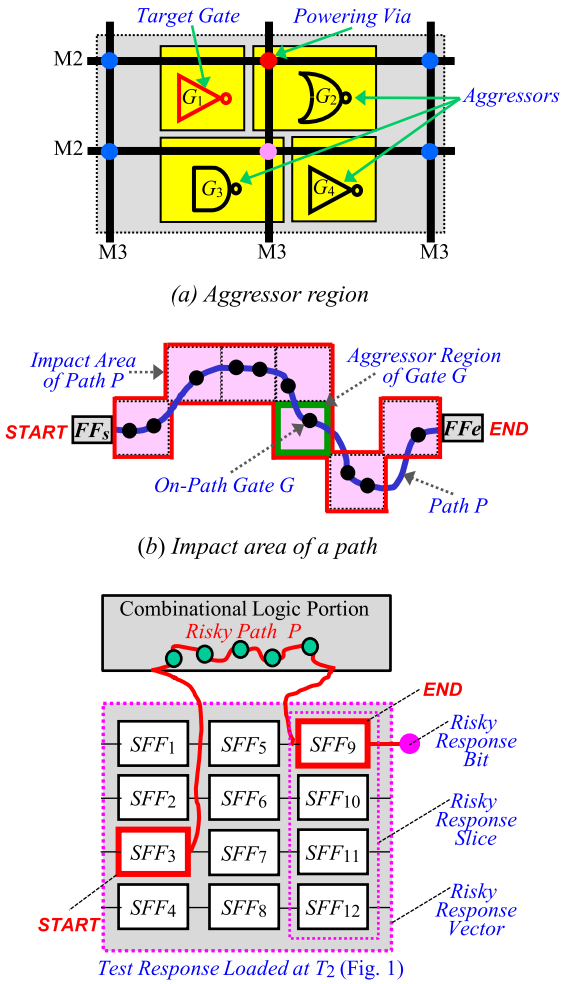



**Fig. 3** LSP-based capture power safety checking.

**Table 1** Circuit statistics.

| | # of Gates | # of FFs | Longest Path Length (logic levels) |
|---|---|---|---|
| *b17* | 29,102 | 1,317 | 44 |
| *b18* | 94,438 | 3,064 | 62 |
| *b19* | 189,225 | 6,130 | 66 |
| *b20* | 29,128 | 430 | 61 |
| *b21* | 29,656 | 430 | 61 |
| *b22* | 42,567 | 645 | 59 |
| *dpro* | 731,487 | 99,759 | 105 |

comprehensive experiments have demonstrated that this perception is false in the case of logic BIST.

Table 1 lists the circuits used in the experiments. Large ITC'99 benchmark circuits and a large industrial circuit (*dpro*) were used. LSP-based capture power safety checking was conducted for 10,000, 30,000, and 50,000 pseudo-random test vectors. Percentages of risky response bits, risky response slices, and risky response vectors for different threshold settings are shown in Table 2 through Table 4, respectively. In all experiments, `Path-Limit` (i.e., the threshold for determining whether a path is long or not) was set to 70% and 80% of the longest structural path in a circuit. `WSA-Limit` (i.e., the threshold for determining whether the launch switching activity in the impact area of a long sensitized path is excessive or not) was set to 20% of the ***maximum WSA*** of the impact area (i.e., the WSA value calculated for the case where all cells in the impact area are assumed to have transitions). 20% is a value commonly used in low-power test solutions and industry tools [6], [8].

**Table 2** Percentage of risky response bits.

| #TV | Path-Limit | *b17* | *b18* | *b19* | *b20* | *b21* | *b22* | *dpro* |
|---|---|---|---|---|---|---|---|---|
| 10,000 | 70% | 0.0003 | 0.0000 | 0.0000 | 0.0020 | 0.0048 | 0.0093 | 0.0000 |
| | 80% | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0002 | 0.0001 | 0.0000 |
| 30,000 | 70% | 0.0004 | 0.0000 | 0.0000 | 0.0050 | 0.0047 | 0.0104 | 0.0000 |
| | 80% | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0003 | 0.0001 | 0.0000 |
| 50,000 | 70% | 0.0003 | 0.0000 | 0.0000 | 0.0048 | 0.0053 | 0.0100 | 0.0000 |
| | 80% | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0003 | 0.0001 | 0.0000 |

*("0.0000" means "<0.00005")*

**Table 3** Percentage of risky response slices.

| #TV | Path-Limit | *b17* | *b18* | *b19* | *b20* | *b21* | *b22* | *dpro* |
|---|---|---|---|---|---|---|---|---|
| 10,000 | 70% | 0.0586 | 0.0006 | 0.0000 | 0.2133 | 0.4700 | 0.8075 | 0.0000 |
| | 80% | 0.0000 | 0.0000 | 0.0000 | 0.0067 | 0.0267 | 0.0150 | 0.0000 |
| 30,000 | 70% | 0.0633 | 0.0004 | 0.0009 | 0.4544 | 0.4656 | 0.8775 | 0.0000 |
| | 80% | 0.0005 | 0.0002 | 0.0000 | 0.0044 | 0.0378 | 0.0200 | 0.0000 |
| 50,000 | 70% | 0.0637 | 0.0001 | 0.0005 | 0.4307 | 0.5013 | 0.8645 | 0.0000 |
| | 80% | 0.0011 | 0.0000 | 0.0000 | 0.0067 | 0.0373 | 0.0130 | 0.0000 |

*("0.0000" means "<0.00005")*

**Table 4** Percentage of risky response vectors.

| #TV | Path-Limit | *b17* | *b18* | *b19* | *b20* | *b21* | *b22* | *dpro* |
|---|---|---|---|---|---|---|---|---|
| 10,000 | 70% | 0.4100 | 0.0100 | 0.0000 | 0.6100 | 1.2000 | 2.8700 | 0.0000 |
| | 80% | 0.0000 | 0.0000 | 0.0000 | 0.0200 | 0.0700 | 0.0600 | 0.0000 |
| 30,000 | 70% | 0.4367 | 0.0333 | 0.0267 | 1.2800 | 1.1467 | 3.0033 | 0.0100 |
| | 80% | 0.0033 | 0.0033 | 0.0000 | 0.0133 | 0.0967 | 0.0800 | 0.0000 |
| 50,000 | 70% | 0.4360 | 0.0020 | 0.0120 | 1.2140 | 1.2240 | 2.9240 | 0.0080 |
| | 80% | 0.0080 | 0.0000 | 0.0020 | 0.0200 | 0.0900 | 0.0500 | 0.0000 |

*("0.0000" means "<0.00005")*

The very small percentages of risky response bits, risky response slices, and risky response vectors in Table 2 through Table 4 clearly demonstrate that uncertain test responses, contrary to the conventional perception, are actually few in logic BIST. This property is especially evident for larger circuits, e.g., the ITC'99 benchmark circuit *b19* and the industrial circuit *dpro*. Possible explanations for this important property are as follows:

(1) ***Strict Sensitization Condition***: It is difficult for long paths to be sensitized even with a large number of pseudo-random test vectors in logic BIST. This is not surprising since, even for transition delay test vectors

generated explicitly by ATPG, the average percentage of risky response vectors for six large ITC'99 circuits (*b17*~*b22*) was 4.17%, and the percentage of risky response vectors for the largest ITC'99 circuit (*b19*) was only 0.2% [11].

(2) ***Uneven Distribution***: Although test vectors applied in logic BIST are pseudo-random in nature, long paths sensitized by them and launch switching activity caused by them in a circuit are usually not evenly distributed across the circuit.

(3) ***Shared Path End-Points***: Many risky paths share the same end-point, which makes a single risky response bit correspond to multiple risky paths.

## 3. CPS-LBIST

The ***capture-power-safe logic BIST*** (***CPS-LBIST***) scheme is aimed to explicitly achieve capture power safety in logic BIST by preventing excessive-capture-power-induced uncertain test responses from reaching the MISR. This goal can be directly achieved by masking all risky response bits (***bit-masking***). As illustrated in Fig. 3 (c), this goal can also be achieved by masking all risky response slices (***slice-masking***) or all risky response vectors (***vector-masking***). Slice-masking and vector-masking may mask non-risky response bits but have much simpler masking control, which is usually implemented with either logic circuitry or ROM.

### 3.1 General CPS-LBIST Design Flow

Figure 4 shows the general CPS-LBIST design flow. It consists of three stages as described below:

- **Stage-1** (***Initial Design***): First, initial logic BIST design (***A***) is conducted. The BIST configuration (***B***) is determined, the circuit-under-test is converted into a BIST-ready circuit (***C***) by scan insertion, *X*-bounding, and test point insertion, and the RTL design of the BIST-specific blocks (***D***) is conducted. Note that some low-shift-power *design-for-test* (*DFT*) techniques [14] can be applied to the BIST-ready circuit to achieve shift power safety. After that, layout design (***E***) is conducted to obtain the layout data for the RTL BIST-ready circuit.
- **Stage-2** (***Capture Power Safety Checking***): LSP-based capture power safety checking (***H***) is conducted for the initial logic BIST by using the BIST-ready circuit design data and the PDN network design data (***G***). The layout data (***F***) and PDN design data (***G***) are used to identify the impact area of each long sensitized path while the netlist (***C***) is used for identifying sensitized paths and conducting logic simulation to calculate WSA values for impact areas. The position information on risky response bits, risky response slices, and risky response vectors (***I***) is also obtained at this stage.
- **Stage-3** (***CPS-LBIST Design***): If the initial logic BIST is capture-power-risky, a masking option (*bit*, *slice*, *vector*) is selected (***J***), and the RTL design for its *mask control unit* (*MCU*) is conducted (***K***). Then, design integration (***M***) is conducted to combine the RTL mask control unit (***L***) with RTL BIST-specific blocks (***D***) to create RTL CPS-LBIST-specific blocks (***N***). After that, logic synthesis and memory compiling (***O***) is conducted to create the netlist and memory blocks of CPS-LBIST-specific blocks (***P***). Finally, layout design (***Q***) is conducted on this synthesis result and the netlist of the BIST-ready circuit (***F***) to create the complete layout data (***R***) of the CPS-LBIST circuit.

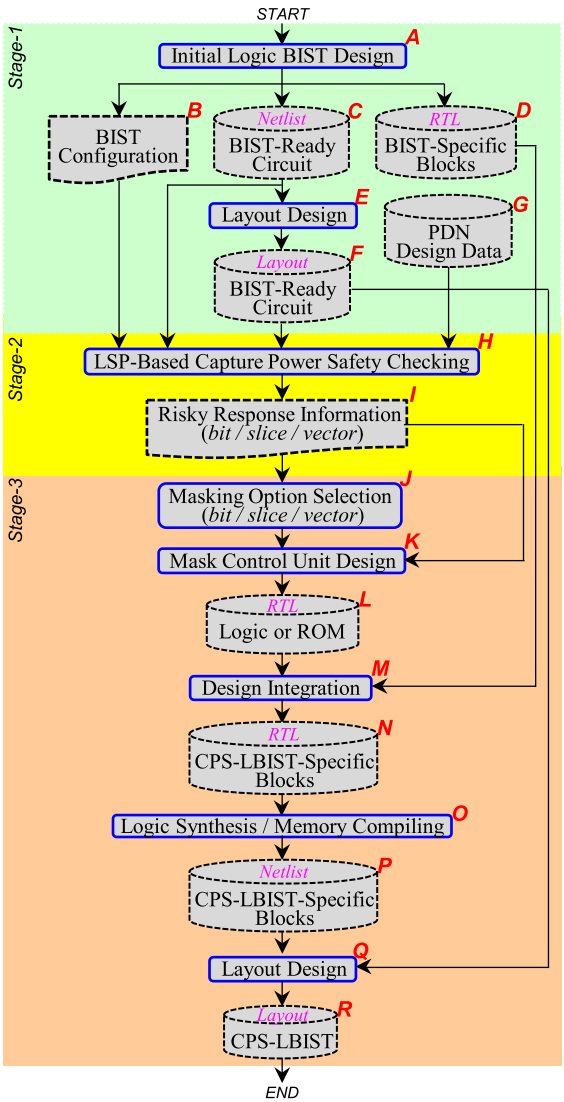



**Fig. 4** General CPS-LBIST design flow.

It is clear that CPS-LBIST is different from conventional low-power BIST in that, instead of reducing capture power, CPS-LBIST allows the existence of excessive capture power but prevents its impact from invalidating the BIST signature by masking all uncertain test responses. Note that excessive capture power has little to do with hot spots, which are mostly related to the accumulative impact

of shift power. *Masking* is similar to *X-bounding* in BIST-ready circuit design [2]. However, CPS-LBIST is the first to make use of masking to explicitly achieve capture power safety in logic BIST.

### 3.2 Masking Options

The key concept of CPS-LBIST is to prevent the impact of excessive capture power from invalidating the BIST signature by masking all uncertain test responses. Three masking options (***bit-masking***, ***slice-masking***, and ***vector-masking***) are available, which differ in impacts on fault coverage and circuit overhead, as described below:

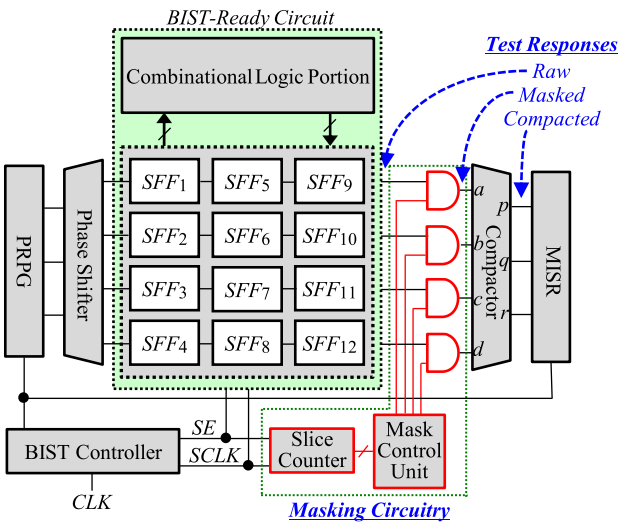
#### 3.2.1 Bit-Masking CPS-LBIST

Figure 5 (a) illustrates the general scheme of bit-masking CPS-LBIST, consisting of a ***BIST-ready circuit***, ***BIST-specific blocks*** (PRPG, phase-shifter, compactor, MISR, BIST controller), and ***masking circuitry***. The BIST-specific blocks and masking circuitry are collectively called ***CPS-LBIST-specific blocks***. The masking circuitry further consists of (1) a ***mask network***, (2) a ***slice counter*** for obtaining the current response slice position, and (3) a ***mask control unit*** for generating required bit-masking signals. In the example of Fig. 5 (a), the mask network consists of four AND gates that are placed before the compactor because every risky response bit needs to be masked.

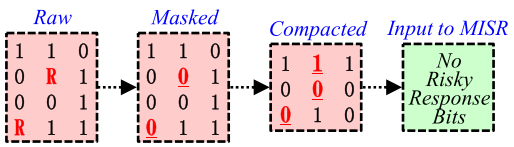
Figure 5 (b) shows an example of the impact of bit-masking CPS-LBIST on test responses, assuming the compactor function to be $<p = a \oplus b,\ q = b \oplus c,\ r = c \oplus d>$. ***Raw*** test responses further become ***masked*** test responses after risky response bits (denoted by R) are masked into 0 (underlined). Masked test responses become ***compacted*** test responses for the MISR. Since no risky response bits can reach the MISR, the signature will not be corrupted by excessive-capture-power-induced uncertain test responses. That is, capture power safety is achieved by bit-masking CPS-LBIST.

#### 3.2.2 Slice-Masking CPS-LBIST

Figure 6 (a) illustrates the general scheme of slice-masking CPS-LBIST. The masking circuitry consists of (1) a ***mask network***, (2) a ***slice counter*** for obtaining the current response slice position, and (3) a ***mask control unit*** for generating required slice-masking signals. In the example of Fig. 6 (a), the mask network consists of three AND gates. However, different from bit-masking CPS-LBIST, these AND gates are placed after the compactor in slice-masking CPS-LBIST. The purpose is to reduce the number of AND gates in the mask network because there are significantly less output lines than input lines for the compactor. This masking option is feasible since it is only necessary to mask all test response bits (both risky ones and non-risky ones) in every risky response slice. Note that instead of using a mask network with multiple AND gates, one can choose to use one AND to gate the clock to the MISR. This implementation has lower circuit overhead but needs clock

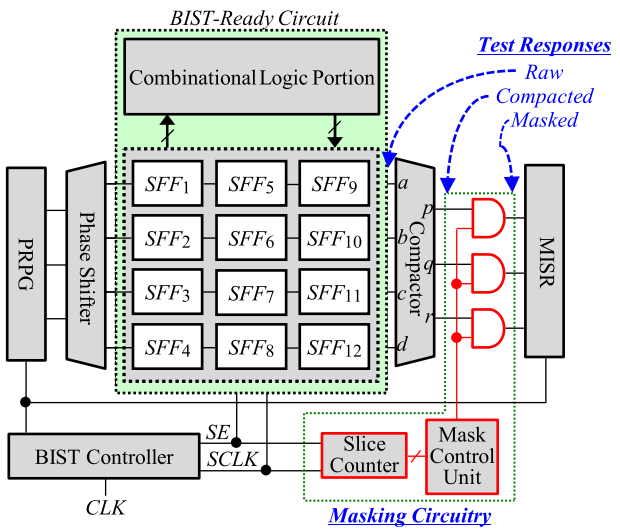



(a) General scheme

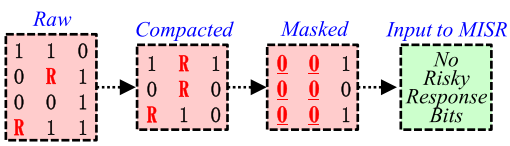



(b) Impact on test responses

**Fig. 5** Bit-masking CPS-LBIST.




(a) General scheme




(b) Impact on test responses

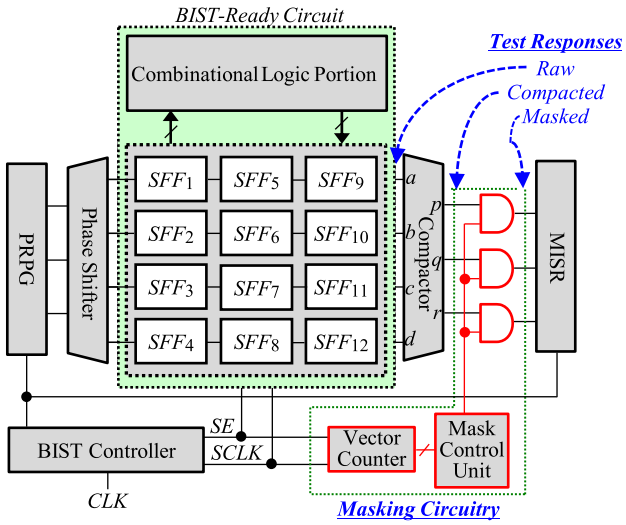
**Fig. 6** Slice-masking CPS-LBIST.

design modification.

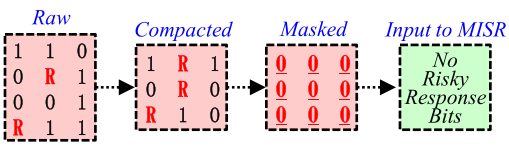
Figure 6 (b) shows an example of the impact of slice-masking CPS-LBIST on test responses, assuming the compactor function to be $<p = a \oplus b,\ q = b \oplus c,\ r = c \oplus d>$. First, ***raw*** test responses become ***compacted*** test responses after going through the compactor. Both raw and compacted test responses contain risky response bits (denoted by R). The compacted test responses further become ***masked*** test responses after all test response bits (both risky ones and non-risky ones) in every risky response slice are masked into 0 (underlined). Since no risky response bits can reach the MISR, the BIST signature will not be corrupted by excessive-capture-power-induced uncertain test responses. That is, capture power safety is achieved by slice-masking CPS-LBIST.

### 3.2.3 Vector-Masking CPS-LBIST

Figure 7 (a) illustrates the general scheme of vector-masking CPS-LBIST, which is similar to slice-masking CPS-LBIST. The major differences are that (1) the ***vector counter*** in vector-masking CPS-LBIST is for obtaining the current response vector position and (2) the ***mask control unit*** is for generating vector-masking signals. As in slice-masking CPS-LBIST, the AND gates in the mask network are also placed after the compactor in vector-masking CPS-LBIST so as to reduce the number of AND gates in the mask network. This masking option is feasible since it is only necessary to mask all test response bits (both risky ones and non-risky ones) in every risky response vector. As in slice-masking CPS-LBIST, instead of using a mask




(a) General scheme




(b) Impact on test responses

**Fig. 7** Vector-masking CPS-LBIST.

network with multiple AND gates, one can choose to use one AND to gate the clock to the MISR for lower circuit overhead but at the cost of clock design modification.

Figure 7 (b) shows an example of the impact of vector-masking CPS-LBIST on test responses, assuming the compactor function to be $<p = a \oplus b,\ q = b \oplus c,\ r = c \oplus d>$. ***Raw*** test responses become ***compacted*** test responses after going through the compactor. Both raw and compacted test responses contain risky response bits (denoted by R). The compacted test responses further become ***masked*** test responses after all test response bits (both risky ones and non-risky ones) are masked into 0 (underlined). Since no risky response bits can reach the MISR, the BIST signature will not be corrupted by excessive-capture-power-induced uncertain test responses. That is, capture power safety is achieved by vector-masking CPS-LBIST.

### 3.2.4 Comparison of Masking Options

As described above, three mask options (*bit-masking*, *slice-masking*, *vector-masking*) are available for CPS-LBIST. Their major differences can be seen from their impacts on fault coverage loss and circuit overhead. Table 5 summarizes the characteristics of the three masking options.

Fault coverage loss is minimized with the bit-masking since only risky response bits are masked. On the other hand, circuit overhead is minimized with the vector-masking since only simple mask control is needed to mask all bits in a risky response vector.

Note that the performance orders shown in Table 5 have only relative meanings. First, performance orders in individual cases may vary depending on the actual number of risky response bits and their distributions. Second, although the fault coverage loss for the vector-masking is conceptually the worst among the three options, its actual quantity is still insignificant. This is because the percentage of risky response vectors is very small in logic BIST as seen from Table 4 in Sect. 2.2. Therefore, vector-masking CPS-LBIST is the preferred choice in practice since it has insignificant fault coverage loss and minimized circuit overhead. Experimental results to be presented in Sect. 4 also support this observation.

Also note that the mask networks in the three options are all composed of AND gates. However, the mask network for the bit-masking option is placed before the compactor since only risky response bits can be masked. On the other hand, the mask network for slice-masking or vector-masking is placed after the compactor since they only need to mask a whole response slice or vector. Placing the mask network after the compactor reduces the number of required

**Table 5** Characteristics of masking options.

| | *Bit-Masking* | *Slice-Masking* | *Vector-Masking* |
|---|---|---|---|
| *Coverage Loss* | Small | Medium | Large |
| *Circuit Overhead* | Large | Medium | Small |

AND gates since there are significantly less outputs than inputs for the compactor.

### 3.3 Mask Control Unit Design

In CPS-LBIST, the role of the ***mask control unit*** (***MCU***) is for generating required masking signals based on necessary position information (*slice position for the bit-masking or slice-masking option and vector position for the vector-masking option*) from the corresponding counter. Generally, an MCU can be implemented by logic circuitry or memory. In the following, more details on logic-based and ROM-based MCU designs for bit-masking CPS-LBIST are provided as examples.

#### 3.3.1 Logic-Based MCU Implementation

• *Design Example*

Figure 8 shows a sample Verilog design file for the mask control unit and the mask network for the bit-masking CPS-LBIST scheme illustrated in Fig. 5 (a). In this example, the number of pseudo-random test vectors to be applied in logic BIST is set to 50,000. Since there are 4 scan chains and 3 scan slices, an 18-bit counter for scan slice counting, a 4-bit mask control unit, and a mask network composed of four AND gates are used. The major portions of this design file are surrounded by the two frames, which is automatically created by using the position information of risky response bits (***I*** in Fig. 4), represented by the counter content and obtained by LSP-based capture power safety checking (***H*** in Fig. 4).

```
module MCU_MASK-NETWORK (COUNTER, RAW, MASKED);
  input [17:0] COUNTER;    // Input from Counter
  input [3:0] RAW;         // Raw Test Response Input
  output [3:0] MASKED;     // Masked Test Response Output
  wire MCU_OUT[3:0];       // Mask Control Unit Output
  // Mask Control Unit
  assign MCU_OUT[0] = func_MCU_0(COUNTER);
  assign MCU_OUT[1] = func_MCU_1(COUNTER);
  assign MCU_OUT[2] = func_MCU_2(COUNTER);
  assign MCU_OUT[3] = func_MCU_3(COUNTER);
  function func_MCU_0;
    input [17:0] COUNTER;
    func_MCU_0 = 1'b0;
  endfunction
  function func_MCU_1;
    input [17:0] COUNTER;
      case (COUNTER)
        18'b010111001100100101: func_MCU_1 = 1'b0;
        18'b100100000010101001: func_MCU_1 = 1'b0;
        default: func_MCU_1 = 1'b1;
      endcase
  endfunction
  function func_MCU_2;
    input [17:0] COUNTER;
    func_MCU_2 = 1'b0;
  endfunction
  function func_MCU_3;
    input [17:0] COUNTER;
      case (COUNTER)
        18'b010100001011001101: func_MCU_3 = 1'b0;
        default: func_MCU_3 = 1'b1;
      endcase
  endfunction
  // Mask Network Composed of AND Gates
  assign MASKED[0] = (MCU_OUT[0] == 1'b0)? 1'b0 : RAW[0];
  assign MASKED[1] = (MCU_OUT[1] == 1'b0)? 1'b0 : RAW[1];
  assign MASKED[2] = (MCU_OUT[2] == 1'b0)? 1'b0 : RAW[2];
  assign MASKED[3] = (MCU_OUT[3] == 1'b0)? 1'b0 : RAW[3];
endmodule
```

Position Information of Risky Response Bits

**Fig. 8** Logic-based MCU implementation for bit-masking.

• *Overhead Analysis*

The overhead of a logic-based MCU implementation depends on the numbers of risky responses bits, risky response slices, and risky response vectors. As shown in Sect. 2.2, the percentages of risky responses bits, risky response slices, and risky response vectors are very small, especially for large circuits. This means that the circuit overhead of a logic-based MCU implementation is usually insignificant.

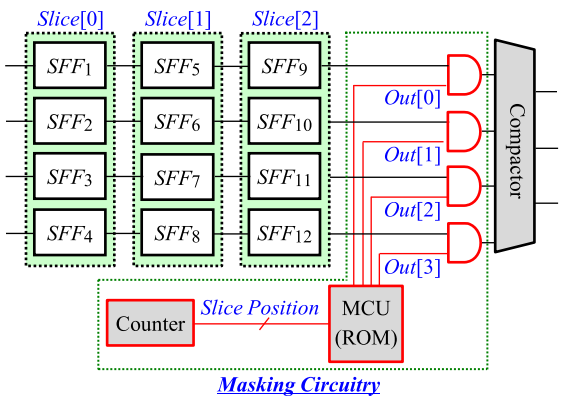
#### 3.3.2 ROM-Based MCU Implementation

• *Design Example*

Figure 9 (a) illustrates the ROM-based MCU implementation for realizing bit-masking CPS-LBIST shown in Fig. 5 (a). The slice counter provides the position information of the current response slice. The information on the risky response bit positions in each response slice is stored in the ROM. That is, the content of the ROM provides mask control to the AND-based mask network so that all risky response bits are masked. The ROM content for the case of Fig. 5 (b) is shown in Fig. 9 (b) as an example.

• *Overhead Analysis*

The size of the ROM for MCU implementation depends on the selected masking option (*bit-masking*, *slice-masking*, *vector-masking*), the number of test vectors (denoted by $V$), the number of scan slices (denoted by $SS$), and the number




(*a*) *General scheme*

| | Out[0] | Out[1] | Out[2] | Out[3] |
|---|---|---|---|---|
| Slice[0] | 1 | 1 | 1 | 0 |
| Slice[1] | 1 | 0 | 1 | 1 |
| Slice[2] | 1 | 1 | 1 | 1 |

(*b*) *ROM content for the case of Fig. 5(b)*

**Fig. 9** ROM-based MCU implementation for bit-masking.

of scan chains (denoted by $SC$). The total number of required bits, denoted by $R$, can be determined as follows:

(a) *Bit-Masking CPS-LBIST:*

$$R = (V + 1) \times SS \times SC$$

(b) *Slice-Masking CPS-LBIST:*

$$R = (V + 1) \times SS$$

(c) *Vector-Masking CPS-LBIST:*

$$R = (V + 1)$$

## 4. Evaluation Results

Comprehensive evaluation of the proposed CPS-LBIST scheme was conducted by using six large ITC'99 circuits (*b17*~*b22*) as well as a large industrial circuit (*dpro*). The statistics of these circuits are as shown in Table 1. Experiments were conducted on a workstation with an Intel Xeon® 3.33GHz CPU and a 24GB main memory.

The BIST configuration in all experiments was <#-Scan-Chains=200, PRPG-Bit-Width=20, Phase-Shifter=20-to-200, Space-Compactor=200-to-20, MISR-Bit-Width=20>. In all experiments, 10,000, 30,000, and 50,000 pseudo-random test vectors were applied. Their capture power safety was determined by LSP-based capture power safety checking (***H*** in Fig. 4), in which the threshold (Path-Limit) for determining whether a path is long or not was set to 70% and 80% of the longest structural path in a circuit, while the threshold (WSA-Limit) for

**Table 6** Evaluation results on test quality.

| | # TV | Ori. FC (%) | Path-Limit (%) | WSA-Limit (%) | # Long Paths | # LSPs | # Risky Paths | # Risky Res. Bits | # Risky Res. Slices | # Risky Res. Vectors | Δ FC (%) | | | CPU (Sec.) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | Bit-Masking | Slice-Masking | Vector-Masking | |
| *b17* | 10000 | 25.1 | 70 | 20 | 732 | 125 | 101 | 42 | 41 | 41 | -0.0074 | -0.0223 | -0.0445 | 1113 |
| | | | 80 | | 66 | 9 | 0 | 0 | 0 | 0 | 0.0000 | 0.0000 | 0.0000 | 2967 |
| | 30000 | 32.7 | 70 | 20 | 732 | 471 | 336 | 142 | 133 | 131 | -0.0099 | -0.2239 | -0.3364 | 4196 |
| | | | 80 | | 66 | 22 | 8 | 1 | 1 | 1 | 0.0000 | 0.0000 | 0.0000 | 5375 |
| | 50000 | 34.2 | 70 | 20 | 732 | 732 | 581 | 229 | 223 | 218 | -0.0037 | -0.0396 | -0.1534 | 4197 |
| | | | 80 | | 66 | 66 | 51 | 4 | 4 | 4 | 0.0000 | 0.0000 | 0.0000 | 5376 |
| *b18* | 10000 | 29.8 | 70 | 20 | 55 | 26 | 26 | 1 | 1 | 1 | -0.0089 | -0.0627 | -0.1564 | 2590 |
| | | | 80 | | 8 | 0 | 0 | 0 | 0 | 0 | 0.0000 | 0.0000 | 0.0000 | 2543 |
| | 30000 | 35.1 | 70 | 20 | 55 | 40 | 32 | 2 | 2 | 1 | -0.0096 | -0.1537 | -0.1537 | 10006 |
| | | | 80 | | 8 | 8 | 1 | 1 | 1 | 1 | -0.0092 | -0.0622 | -0.1537 | 10163 |
| | 50000 | 38.0 | 70 | 20 | 55 | 55 | 52 | 1 | 1 | 1 | 0.0000 | -0.0154 | -0.0476 | 10527 |
| | | | 80 | | 8 | 0 | 0 | 0 | 0 | 0 | 0.0000 | 0.0000 | 0.0000 | 12101 |
| *b19* | 10000 | 29.3 | 70 | 20 | 297 | 0 | 0 | 0 | 0 | 0 | 0.0000 | 0.0000 | 0.0000 | 6966 |
| | | | 80 | | 4 | 0 | 0 | 0 | 0 | 0 | 0.0000 | 0.0000 | 0.0000 | 6960 |
| | 30000 | 35.3 | 70 | 20 | 297 | 75 | 73 | 10 | 8 | 8 | -0.0128 | -0.0826 | -0.2593 | 19544 |
| | | | 80 | | 4 | 0 | 0 | 0 | 0 | 0 | 0.0000 | 0.0000 | 0.0000 | 20591 |
| | 50000 | 37.2 | 70 | 20 | 297 | 297 | 297 | 9 | 7 | 6 | -0.0023 | -0.0205 | -0.0405 | 29594 |
| | | | 80 | | 4 | 4 | 4 | 1 | 1 | 1 | 0.0000 | -0.0021 | -0.0034 | 25886 |
| *b20* | 10000 | 64.4 | 70 | 20 | 10459 | 1260 | 1260 | 87 | 64 | 61 | -0.0838 | -0.7315 | -0.9459 | 755 |
| | | | 80 | | 50 | 5 | 5 | 2 | 2 | 2 | -0.0003 | -0.0304 | -0.0316 | 780 |
| | 30000 | 81.7 | 70 | 20 | 10459 | 5359 | 5359 | 648 | 409 | 384 | -0.0156 | -0.0681 | -0.0996 | 2489 |
| | | | 80 | | 50 | 16 | 16 | 4 | 4 | 4 | 0.0000 | 0.0000 | 0.0000 | 2693 |
| | 50000 | 82.5 | 70 | 20 | 10459 | 10459 | 10456 | 1039 | 646 | 607 | -0.0101 | -0.0436 | -0.0606 | 3814 |
| | | | 80 | | 50 | 50 | 50 | 10 | 10 | 10 | 0.0000 | -0.0011 | -0.0011 | 3728 |
| *b21* | 10000 | 75.1 | 70 | 20 | 22866 | 3786 | 3786 | 208 | 141 | 120 | -0.0322 | -0.1711 | -0.2844 | 814 |
| | | | 80 | | 1044 | 113 | 113 | 10 | 8 | 7 | 0.0000 | -0.01 | -0.0167 | 812 |
| | 30000 | 81.0 | 70 | 20 | 22866 | 9317 | 9317 | 608 | 419 | 344 | -0.0278 | -0.0533 | -0.1211 | 2742 |
| | | | 80 | | 1044 | 237 | 237 | 36 | 34 | 29 | -0.0033 | -0.0066 | -0.0089 | 2928 |
| | 50000 | 81.9 | 70 | 20 | 22866 | 22866 | 22866 | 1134 | 752 | 612 | -0.0067 | -0.0445 | -0.0689 | 3000 |
| | | | 80 | | 1044 | 1044 | 1044 | 67 | 56 | 45 | 0.0000 | -0.0032 | -0.0032 | 3127 |
| *b22* | 10000 | 74.9 | 70 | 20 | 26479 | 5636 | 5633 | 603 | 323 | 287 | -0.0643 | -0.2881 | -0.5043 | 1236 |
| | | | 80 | | 148 | 15 | 15 | 6 | 6 | 6 | 0.0000 | -0.0015 | -0.0076 | 1206 |
| | 30000 | 81.3 | 70 | 20 | 26479 | 16605 | 16590 | 2019 | 1053 | 901 | -0.0318 | -0.1165 | -0.1921 | 4185 |
| | | | 80 | | 148 | 81 | 81 | 27 | 24 | 24 | 0.0000 | -0.0046 | -0.0053 | 4108 |
| | 50000 | 82.5 | 70 | 20 | 26479 | 26479 | 26446 | 3216 | 1729 | 1462 | -0.0197 | -0.0779 | -0.1225 | 5563 |
| | | | 80 | | 148 | 148 | 148 | 29 | 26 | 25 | 0.0000 | -0.0015 | -0.0068 | 4891 |
| *dpro* | 10000 | 65.4 | 70 | 20 | 94 | 0 | 0 | 0 | 0 | 0 | 0.0000 | 0.0000 | 0.0000 | 56234 |
| | | | 80 | | 0 | 0 | 0 | 0 | 0 | 0 | 0.0000 | 0.0000 | 0.0000 | 54543 |
| | 30000 | 69.8 | 70 | 20 | 94 | 73 | 60 | 3 | 3 | 3 | 0.0000 | 0.0000 | 0.0000 | 181880 |
| | | | 80 | | 0 | 0 | 0 | 0 | 0 | 0 | 0.0000 | 0.0000 | 0.0000 | 171385 |
| | 50000 | 71.6 | 70 | 20 | 94 | 94 | 22 | 4 | 4 | 4 | 0.0000 | 0.0000 | 0.0000 | 249057 |
| | | | 80 | | 0 | 0 | 0 | 0 | 0 | 0 | 0.0000 | 0.0000 | 0.0000 | 232618 |

*("0.0000" means "<0.00005")*

**Table 7** Evaluation results on circuit overhead.

| | # TV | Path-Limit (%) | WSA-Limit (%) | Logic-Based MCU | | | ROM-Based MCU | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Area Overhead (%) | | | # Bits | | |
| | | | | Bit-Masking | Slice-Masking | Vector-Masking | Bit-Masking | Slice-Masking | Vector-Masking |
| *B17* | 10000 | 70 | 20 | 0.4676 | 0.5354 | 0.4543 | 14001400 | 70007 | 10001 |
| | | 80 | | 0.0000 | 0.0000 | 0.0000 | | | |
| | 30000 | 70 | 20 | 0.9951 | 1.0566 | 0.8088 | 42001400 | 210007 | 30001 |
| | | 80 | | 0.0462 | 0.1164 | 0.0859 | | | |
| | 50000 | 70 | 20 | 1.4995 | 1.5120 | 1.2401 | 70001400 | 350007 | 50001 |
| | | 80 | | 0.1004 | 0.1722 | 0.1461 | | | |
| *b18* | 10000 | 70 | 20 | 0.0173 | 0.0369 | 0.0271 | 32003200 | 160016 | 10001 |
| | | 80 | | 0.0000 | 0.0000 | 0.0000 | | | |
| | 30000 | 70 | 20 | 0.0200 | 0.1139 | 0.0272 | 96003200 | 480016 | 30001 |
| | | 80 | | 0.0171 | 0.1079 | 0.0272 | | | |
| | 50000 | 70 | 20 | 0.0161 | 0.1764 | 0.0272 | 160003200 | 800016 | 50001 |
| | | 80 | | 0.0000 | 0.0000 | 0.0000 | | | |
| *b19* | 10000 | 70 | 20 | 0.0000 | 0.0000 | 0.0000 | 64006400 | 320032 | 10001 |
| | | 80 | | 0.0000 | 0.0000 | 0.0000 | | | |
| | 30000 | 70 | 20 | 0.0308 | 0.0391 | 0.0310 | 192006400 | 960032 | 30001 |
| | | 80 | | 0.0000 | 0.0000 | 0.0000 | | | |
| | 50000 | 70 | 20 | 0.0270 | 0.0327 | 0.0261 | 320006400 | 1600032 | 50001 |
| | | 80 | | 0.0084 | 0.0189 | 0.0132 | | | |
| *b20* | 10000 | 70 | 20 | 1.2026 | 0.7280 | 0.5804 | 6000600 | 30003 | 10001 |
| | | 80 | | 0.0654 | 0.1332 | 0.1071 | | | |
| | 30000 | 70 | 20 | 6.0246 | 2.3177 | 1.7905 | 18000600 | 90003 | 30001 |
| | | 80 | | 0.1065 | 0.1830 | 0.1447 | | | |
| | 50000 | 70 | 20 | 7.7729 | 3.3409 | 2.8072 | 30000600 | 150003 | 50001 |
| | | 80 | | 0.2249 | 0.3164 | 0.2482 | | | |
| *b21* | 10000 | 70 | 20 | 2.4491 | 1.0428 | 0.7643 | 6000600 | 30003 | 10001 |
| | | 80 | | 0.1900 | 0.2364 | 0.1943 | | | |
| | 30000 | 70 | 20 | 5.5283 | 2.2707 | 1.7097 | 18000600 | 90003 | 30001 |
| | | 80 | | 0.6137 | 0.5945 | 0.4713 | | | |
| | 50000 | 70 | 20 | 7.0312 | 4.0912 | 2.7853 | 30000600 | 150003 | 50001 |
| | | 80 | | 1.0073 | 0.7846 | 0.6210 | | | |
| *b22* | 10000 | 70 | 20 | 3.9880 | 1.1645 | 0.9006 | 8000800 | 40004 | 10001 |
| | | 80 | | 0.0954 | 0.1329 | 0.1085 | | | |
| | 30000 | 70 | 20 | 7.5467 | 3.2547 | 2.0130 | 24000800 | 120004 | 30001 |
| | | 80 | | 0.3030 | 0.3293 | 0.2856 | | | |
| | 50000 | 70 | 20 | 10.2411 | 4.2136 | 2.9158 | 40000800 | 200004 | 50001 |
| | | 80 | | 0.3323 | 0.3320 | 0.2980 | | | |
| *dpro* | 10000 | 70 | 20 | 0.0000 | 0.0000 | 0.0000 | 998099800 | 4990499 | 10001 |
| | | 80 | | 0.0000 | 0.0000 | 0.0000 | | | |
| | 30000 | 70 | 20 | 0.0030 | 0.0040 | 0.0029 | 2994099800 | 14970499 | 30001 |
| | | 80 | | 0.0000 | 0.0000 | 0.0000 | | | |
| | 50000 | 70 | 20 | 0.0027 | 0.0041 | 0.0030 | 4990099800 | 24950499 | 50001 |
| | | 80 | | 0.0000 | 0.0000 | 0.0000 | | | |

*("0.0000" means "<0.00005")*

determining whether the launch switching activity in the impact area of a long sensitized path is excessive or not was set to 20% of the maximum WSA (`max_WSA`) in the impact area of the path. Note that `max_WSA` is calculated by assuming all cells in the impact area have transitions.

Table 6 summarizes the impact of the proposed CPS-LBIST scheme on test quality for three masking options. In this table, `# TV` is the number of pseudo-random test vectors, `Ori. FC` is transition delay fault coverage, `# Long Paths` is the number of long paths, `# LSPs` is the number of long sensitized paths (LSPs), `# Risky Paths` is number of risky paths, `# Risky Res. Bits` is the total number of risky response bits from the circuit, `# Risky Res. Slices` is the total number of risky response slices, and `# Risky Res. Vectors` is the total number of risky response vectors. The impact of the proposed CPS-LBIST scheme with the three masking options (*bit-masking*, *slice-masking*, *vector-masking*) on the final test quality was evaluated by percentage change in fault coverage (`ΔFC`), i.e., fault coverage loss (-) in most cases. In addition, the execution time (`CPU`) for obtaining CPS-LBIST design data (steps *H* through *N* in Fig. 4) is also shown in the table.

Table 7 summarizes the impact of the proposed CPS-LBIST scheme on circuit overhead for two MCU implementation options (*logic-based*, *ROM-based*). In the case of logic-based MCU implementation, the percentage number of area overhead (`Area Overhead (%)`) of the whole masking circuitry, including a mask network, a counter, and

a mask control unit, is shown. In the case of ROM-based MCU implementation, the number of bits (`# Bits`) needed for the ROM is shown.

It is clear that the proposed CPS-LBIST scheme can achieve capture power safety using the well-known approach of masking without explicitly reducing capture power. From the evaluation results, the following three important observations can be made:

(1) The impact of the proposed CPS-LBIST scheme on test quality, measured by fault coverage loss, depends on (a) the selected masking option (*bit-masking*, *slice-masking*, *vector-masking*), (b) the total number of risky response bits, and (c) the distribution of the risky response bits among response vectors and response slices. (b) and (c) are affected by the number of test vectors, the path length threshold (`Path-Limit`) for determining long sensitized paths, and the WSA threshold (`WSA-Limit`) for determining risky test vectors. Generally, fault coverage loss is the smallest for bit-masking CPS-LBIST and the largest for vector-masking CPS-LBIST, with that of slice-masking CPS-LBIST in the middle. However, it can be seen from Table 6 that, even in the case of vector-masking CPS-LBIST, the fault coverage loss is negligibly small. This is because of the property of risky response bits in logic BIST as revealed in Sect. 2.2, which says that the number of risky response bits in logic BIST is very small even when a large number of pseudo-random test vectors are applied. Evidence data shown in Table 2 demonstrate that this property is especially true for large circuits.

(2) The impact of the proposed CPS-LBIST scheme on circuit overhead depends on the selected masking option (*bit-masking*, *slice-masking*, *vector-masking*) and the selected MCU implementation (*logic-based*, *ROM-based*). For logic-based MCU implementation, the circuit overhead also depends on the total number of risky response bits, the distribution of risky response bits in response vectors and response slices. Generally, circuit overhead is the largest for bit-masking CPS-LBIST and the smallest for vector-masking CPS-LBIST, with that of slice-masking CPS-LBIST in the middle. In addition, it can be seen from Table 7 that, no matter what masking option is selected, the circuit overhead is negligibly small if logic-based MCU implementation is applied. However, a sizable ROM, especially for the bit-masking CPS-LBIST scheme, is needed if ROM-based MCU implementation is applied.

(3) Exceptionally good results (negligible fault coverage loss as well as negligible circuit overhead) of the proposed CPS-LBIST scheme with logic-based MCU implementation have been obtained for the largest ITC'99 circuit (*b19*) and the large industrial circuit (*dpro*).

These important observations lead to the following practically important fact about CPS-LBIST: *The vector-masking CPS-LBIST scheme with logic-based MCU implementation is preferable for practical use since its impacts on both test quality and circuit overhead are negligible. In addition, the vector-masking CPS-LBIST scheme is highly scalable in that the larger a circuit, the better its performance*.

## 5. Conclusions

This paper is the first to explicitly address capture power safety (instead of capture power reduction) in scan-based logic BIST by applying the well-known approach of masking. The proposed solution, ***capture-power-safe BIST*** (***CPS-LBIST***), achieves capture power safety in a guaranteed manner by using *bit-masking*, *slice-masking* or *vector-masking* to prevent excessive-capture-power-induced uncertain test responses from reaching the MISR. The feasibility of CPS-LBIST comes from the important property that uncertain test responses in logic BIST are actually very few, as evidenced by data reported in this paper. Comprehensive evaluations with large ITC'99 benchmark circuits and an industrial circuit have demonstrated that CPS-LBIST is a practical and scalable solution for achieving capture power safety in scan-based logic BIST, especially with logic-based mask control unit (MCU) implementation.

Future work includes (1) speeding-up the capture power safety checking procedure and (2) evaluating the effectiveness of CPS-LBIST with a real test chip.

## Acknowledgements

This work was partly supported by JSPS Grant-in-Aid for Scientific Research (B) #25280016, JSPS Grant-in-Aid for Scientific Research on Innovative Areas #24650022, and JST-NSC Grant-in-Aid for Japan-Taiwan Joint Research on the Testing of Nano Devices.

### References

[1] L.-T. Wang, X. Wen, and K.S. Abdel-Hafez, "Design for testability," in VLSI Test Principles and Architectures: Design for Testability, ed. L.-T. Wang, C.W. Wu, and X. Wen, Morgan Kaufmann, San Francisco, 2006.

[2] L.-T. Wang, "Logic built-in self-test," in VLSI Test Principles and Architectures: Design for Testability, ed. L.-T. Wang, C.W. Wu, and X. Wen, Morgan Kaufmann, San Francisco, 2006.

[3] D.M. Walker and M.S. Hsiao, "Delay testing," in System-on-Chip Test Architectures: Nanometer Design for Testability, ed. L.-T. Wang, C.E. Strond, and N.A. Touba, Morgan Kaufmann, San Francisco, 2007.

[4] J. Saxena, K. Butler, and L. Whetsel, "An analysis of power reduction techniques in scan testing," Proc. IEEE Int'l Test Conf., pp.670–677, 2001.

[5] P. Girard, X. Wen, and N.A. Touba, "Low-power testing," in System-on-Chip Test Architectures: Nanometer Design for Testability, ed. L.-T. Wang, C.E. Strond, and N.A. Touba, Morgan Kaufmann, San Francisco, 2007.

[6] S. Ravi, "Power-aware test: Challenges and solutions," Proc. IEEE Int'l Test Conf., Lecture 2.2, 2007.

[7] P. Girard, N. Nicolici, and X. Wen, eds., Power-aware testing and test strategies for low power devices, Springer, New York, Oct. 2009.

[8] M. Tehranipoor and K. Butler, "Power supply noise: A survey on

effects and research," IEEE Des. Test Comput., vol.27, no.2, pp.51–67, March-April 2010.
[9] Y. Yamato, X. Wen, M.A. Kochte, K. Miyase, S. Kajihara, and L.-T. Wang, "A novel scan segmentation design method for avoiding shift timing failure in scan testing," Proc. IEEE Intl. Test Conf., Paper 12.1, 2011.
[10] V.R. Devanathan and S.K. Vooka, "On reducing peak capture power of transition delay fault test for SoCs with unwrapped cores," J. Low Power Electronics, vol.2, no.3, pp.464–476, 2006.
[11] X. Wen, K. Enokimoto, K. Miyase, Y. Yamato, M. Kochte, S. Kajihara, P. Girard, and M. Tehranipoor, "Power-aware test generation with guaranteed launch safety for at-speed scan testing," Proc. IEEE VLSI Test Symp., pp.167–171, 2011.
[12] X. Wen, Y. Nishida, K. Miyase, S. Kajihara, P. Girard, M. Tehranipoor, and L.-T. Wang, "On pinpoint capture power management in at-speed scan test generation," Proc. IEEE Int'l Test Conf., Paper 6.1, 2012.
[13] L. Whetsel, "Adapting scan architectures for low power operation," Proc. IEEE Int'l Test Conf., pp.863–872, 2000.
[14] S. Gerstendorfer and H.-J. Wunderlich, "Minimized power consumption for scan-based BIST," Proc. IEEE Intl. Test Conf., pp.77–84, 1999.
[15] H.-J. Wunderlich and C. Zoellin, "Power-aware design-for-test," in Power-Aware Testing and Test Strategies for Low Power Devices, eds. P. Girard, N. Nicolici, and X. Wen, Springer, New York, 2009.
[16] M.E. Imhof, C.G. Zoellin, H.-J. Wunderlich, N. Maeding, and J. Leenstra, "Scan test planning for power reduction," Proc. Design Automation Conf., pp.521–526, 2007.
[17] X. Wen, Y. Yamashita, S. Kajihara, L.-T. Wang, K.K. Saluja, and K. Kinoshita, "On low-capture-power test generation for scan testing," Proc. IEEE VLSI Test Symp., pp.265–270, 2005.
[18] Y. Tang, H. Wunderlich, P. Engelke, I. Polian, B. Becker, J. Schloffel, F. Hapke, and M. Wittke, "*X*-masking during logic BIST and its impact on defect coverage," IEEE Trans. Very Large Scale Integr. Syst., vol.14, no.2, pp.193–202, Feb. 2006.
[19] R. Garg, R. Putman, and N.A. Touba, "Increasing output compaction in presence of unknowns using an X-canceling MISR with deterministic observation," Proc. IEEE VLSI Test Symp., pp.35–42, 2008.
[20] X. Wen, K. Miyase, T. Suzuki, S. Kajihara, Y. Ohsumi, and K.K. Saluja, "Critical-path-aware *X*-filling for effective IR-drop reduction in at-speed scan testing," Proc. Design Automation Conf., pp.527–532, 2007.
[21] J.-C. Ju and R.A. Saleh, "Incremental techniques for the identification of statically sensitizable critical paths," Proc. Design Automation Conf., pp.541–546, 1991.
[22] S. Wang and S.K. Gupta, "ATPG for heat dissipation minimization during test application," Proc. IEEE Int'l Test Conf., pp.250–258, 1994.
[23] J. Ma, J. Lee, and M. Tehranipoor, "Layout-aware pattern generation for maximizing supply noise effects on critical paths," Proc. IEEE VLSI Test Symp., pp.221–226, 2009.

**Akihiro Tomita** received the B.E. degree in Computer Science and Electronics from Kyushu Institute of Technology, Japan, in 2012. He is currently a graduate student studying for his master degree. His research interest is low power LSI testing.

**Xiaoqing Wen** received the B.E. degree from Tsinghua University, China, in 1986, the M.E. degree from Hiroshima University, Japan, in 1990, and the Ph.D. degree from Osaka University, Japan, in 1993. From 1993 to 1997, he was an Assistant Professor at Akita University, Japan. He was a Visiting Researcher at University of Wisconsin, Madison, USA, from Oct. 1995 to Mar. 1996. He joined SynTest Technologies, Inc., USA, in 1998, and served as its Chief Technology Officer until 2003. In 2004, he joined the Kyushu Institute of Technology, Japan, where he is currently a Professor and the Director of Dependable Integrated Systems Research Center. His research interests include VLSI test, diagnosis, and testable design. He co-authored and co-edited two books: VLSI Test Principles and Architectures: Design for Testability (Morgan Kaufmann, 2006) and Power-Aware Testing and Test Strategies for Low Power Devices (Springer, 2009). He also holds 38 U.S. Patents and 14 Japan Patents on VLSI testing. He received the 2008 IEICE-ISS Best Paper Award for his pioneering work on X-filling-based low-power test generation. He is a Fellow of the IEEE, a member of the IEICE, the IPSJ, and the REAJ.

**Yasuo Sato** received the B.S. and M.S. degrees in mathematics from Tokyo University, Japan, in 1976 and 1978, respectively. He received the Ph.D. in engineering from Tokyo Metropolitan University, Japan, in 2005. He was with Hitachi, Ltd. working on computer-aided design from 1978 to 2008. From 2003 to 2006, he was a senior manager of DFT (Design for Testability) group at the Semiconductor Technology Academic Research Center (STARC). In 2009, he joined the Department of Computer Science and Electronics, Kyushu Institute of Technology, Japan, and is currently a research Professor. His particular interest areas include field test, delay testing, low power BIST, fault diagnosis and FPGA application-independent test. He is a member of the IEEE and the IEICE.

**Seiji Kajihara** received the B.S. and M.S. degrees from Hiroshima University, Japan, and the Ph.D. degree from Osaka University, Japan, in 1987, 1989, and 1992, respectively. From 1992 to 1995, he worked with the Department of Applied Physics, Osaka University, as an Assistant Professor. In 1996, he joined the Department of Computer Science and Electronics of Kyushu Institute of Technology, Japan, where he is currently a Professor. His research interests include test generation, delay testing, and design for testability. He received the Young Engineer Award from IEICE in 1997, the Yamashita SIG Research Award from IPSJ in 2002, and the Best Paper Award from IEICE in 2005. He is a member of the IEEE, the IEICE, and the IPSJ. He serves on the editorial board of the Journal of Electronic Testing: Theory and Applications.

**Kohei Miyase** received the B.E., M.E., and Ph.D. degrees in Computer Science and Systems Engineering from Kyushu Institute of Technology, Japan, in 2000, 2002 and 2005, respectively. From 2005, he was a researcher at Innovation Plaza Fukuoka, Japan Science and technology Agency, Japan. His research interests include test compaction, test compression, design for testability, low power test, and fault diagnosis. He received the Excellent Student Award of the IEEE Fukuoka Section in 2005. He is a member of the IEEE, the IEICE, and the IPSJ.

**Stefan Holst** received the Ph.D. degree from the University of Stuttgart, Germany, in 2012. He now is an Assistant Professor in the Department of Creative Informatics at Kyushu Institute of Technology, Japan. His research interests include GPU-accelerated logic simulation, logic diagnosis, embedded test and its applications to medical devices.

**Patrick Girard** received the M.Sc. degree in Electrical Engineering and the Ph.D. degree in Microelectronics from the University of Montpellier, France, in 1988 and 1992, respectively. He is currently Research Director at CNRS, and Head of the Microelectronics Department of the Laboratory of LIRMM - France. His research interests include all aspects of digital testing and memory testing, with emphasis on critical constraints such as timing and power. He is Technical Activities Chair of the Test Technology Technical Council (TTTC) of the IEEE Computer Society. He is the founder and Editor-in-Chief of the ASP Journal of Low Power Electronics (JOLPE). He is an Associate Editor of the IEEE Transactions on VLSI Systems and the Journal of Electronic Testing – Theory and Applications (JETTA - Springer). He is co-editor of the book "Power-Aware Testing and Test Strategies for Low Power Devices" and co-author of the book "Advanced Test Methods for SRAMs – Effective Solutions for Dynamic Fault Detection in Nanoscale Technologies" in 2009. He has supervised 28 PhD dissertations and has published 6 books or book chapters, 44 journal papers, and more than 160 conference and symposium papers on these fields. He is a Senior Member of IEEE and a Golden Core Member of the IEEE Computer Society.

**Mohammad Tehranipoor** is currently the Charles H. Knapp Associate Professor of Electrical and Computer Engineering Department at the University of Connecticut. His current research projects include computer-aided design and test for CMOS VLSI designs, hardware security and trust, counterfeit IC detection and prevention, and reliable systems design at nanoscale. He has published over 200 journal articles and refereed conference papers and has given more than 125 invited talks and keynote addresses since 2006. In addition, he has published five books and ten book chapters. He co-founded a new symposium called IEEE International Symposium on Hardware-Oriented Security and Trust (HOST) and served as HOST-2008 and HOST-2009 General Chair and continue to serve as Chair of Steering Committee for HOST. He is also a co-founder of TrustHub. He is currently serving as the Associate Editor-in-Chief (EIC) for IEEE Design and Test of Computers, an Associate Editor for JETTA, an Associate Editor for Journal of Low Power Electronics (JOLPE), an Associate Editor for ACM Transactions for Design Automation of Electronic Systems (TODAES), an IEEE Distinguished Speaker, and an ACM Distinguished Speaker. He is a Senior Member of the IEEE, Golden Core Member of IEEE Computer Society, and Member of ACM and ACM SIGDA.

**Laung-Terng (L.-T.) Wang** is chairman and chief executive officer (CEO) of SynTest Technologies (Sunnyvale, CA). He received his BSEE and MSEE degrees from National Taiwan University in 1975 and 1977, respectively, and his MSEE and EE Ph.D. degrees under the Honors Cooperative Program (HCP) from Stanford University in 1982 and 1987, respectively. He worked at Intel (Santa Clara, CA) and Daisy Systems (Mountain View, CA) from 1980 to 1986 and was with the Department of Electrical Engineering of Stanford University as Research Associate and Lecturer from 1987 to 1991. Dr. Wang has filed more than 25 U.S. and European patent applications in the areas of scan synthesis, test generation, at-speed scan testing, test compression, logic built-in self-test (BIST), and design for debug and diagnosis, of which 13 have been granted. He spearheaded efforts to raise endowed funds in memory of his NTU chair professor, Dr. Irving T. Ho, cofounder of the Hsinchu Science Park and vice chair of the National Science Council, Taiwan. Since 2003, he has helped establish a number of chair professorships, graduate fellowships, and undergraduate scholarships at Stanford University, National Taiwan University and National Tsing Hua University in Taiwan, as well as Xiamen University, Tsinghua University, and Shanghai Jiao Tong University in China. Dr. Wang co-authored and co-edited an internationally used DFT textbook - VLSI Test Principles and Architectures: Design for Testability in 2006. He received the Meritorious Service Award from the IEEE Computer Society in 2007 and is a member of Sigma Xi.