PAPER

Improving Small-Delay Fault Coverage of On-Chip Delay Measurement by Segmented Scan and Test Point Insertion*

Wenpo ZHANG[†a)], ***Student Member***, **Kazuteru NAMBA**[†], ***Member***, ***and*** **Hideo ITO**[†], ***Fellow***

SUMMARY With IC design entering the nanometer scale integration, the reliability of VLSI has declined due to small-delay defects, which are hard to detect by traditional delay fault testing. To detect small-delay defects, on-chip delay measurement, which measures the delay time of paths in the *circuit under test* (CUT), was proposed. However, our pre-simulation results show that when using on-chip delay measurement method to detect small-delay defects, test generation under the single-path sensitization is required. This constraint makes the fault coverage very low. To improve fault coverage, this paper introduces techniques which use segmented scan and *test point insertion* (TPI). Evaluation results indicate that we can get an acceptable fault coverage, by combining these techniques for *launch off shift* (LOS) testing under the single-path sensitization condition. Specifically, fault coverage is improved 27.02~47.74% with 6.33~12.35% of hardware overhead.

key words: *small-delay defects, fault coverage, segmented scan, control point, observation point*

1. Introduction

As technology scales to 45nm and below, semiconductor device scaling has significantly improved performance and circuit integration density. With the increasing speed of integrated circuits, violations of the performance specifications are becoming a major factor affecting the product-quality level [1]. Delay defects that degrade performance and cause timing related failures are emerging as a major problem in nanometer technologies. Several delay fault models and delay test methods have been proposed. Transition fault and path delay fault are two prevalent fault models [2]. With the growing complexity of designs, scan-based techniques of testing are becoming very popular. However, it is inherently limited by the accuracy of the provided test clock frequency with the external *automatic test equipment* (ATE), which can be affected by factors such as parasitic capacitance, resistance of probe and tester skew [3].

Small-delay defects are known to degrade in operation and cause early life failure. A small-delay defect has a defect size that is not large enough to cause a timing failure under the system clock cycle. Small-delay defects represent a significant reliability concern when resistive defects are present in a technology. For example, a resistive open caused by a minimally connecting via can become a complete open in operation due to metal migration. Recently, increasing random process variations contribute to significant timing variability, which is indistinguishable from the effects of small-delay defects. Such variations can be beyond the performance variations caused by resistive small-delay defects. Therefore, these process variations need to be detected to improve the reliability of chips [4]. Since they might escape detection during traditional Pass-Fail delay fault testing with functional clock, small-delay defects have become a significant problem and it is essential to detect such defects during manufacturing tests [5]. Such manufacturing flaws have traditionally been eliminated through burn-in stress testing. Burn-in fallout can be as high as 0.5-1% (5,000-10,000 *defect parts per million* (DPPM) for large complex die, making stress testing essential for high end parts such as microprocessors. Unfortunately, traditional burn-in is becoming extremely expensive for delicate nanometer technologies; it also appears to be losing effectiveness in accelerating certain types of early life failures [6]. As a result, industry is looking for alternate low-cost methods for eliminating latent defects [7], [8].

To screen small-delay fault, small-delay defect screening with criteria based on statistical analysis is proposed [8]. In this technique, small-delay defects are detected as outliers. Delay distributions for each path can be obtained by measuring many chips of the same design. If a path delay time is beyond a specified time such as the three-sigma limit (users can set the specified time freely by taking into consideration the trade-off between yield and dependability), even if it is not beyond the system clock cycle, we regard it as a faulty path. Some previous works presented on-chip path delay time measurements based on this strategy [9]–[14]. By measuring delay time of the *path under measurement* (PUM), not only the gross and small-delay faults can be detected but also the amount of timing violation in the failing paths can be obtained under certain environment conditions [15], [16].

This paper points out a considerable issue of testing using on-chip path delay measurement: in the measurement, PUMs are sensitized by delay fault test patterns. However, this paper reveals that the robust test patterns are not suitable for on-chip delay measurement. Specifically, they require test generation under the single-path sensitization condition, which causes its small-delay fault coverage to be very low. Thus, a method improving fault coverage is strongly required. This paper uses the transition fault coverage as the

Manuscript received April 7, 2014.
Manuscript revised June 27, 2014.

[†]The authors are with the Graduate School of Advanced Integration Science, Chiba University, Chiba-shi, 263–8522 Japan.

*The earlier version of this paper was presented at the 2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI & Nanotechnology Systems (DFT'2012).

a) E-mail: wenpo.zhang@ieee.org

DOI: 10.1587/transinf.2014EDP7110

small-delay fault coverage because the aim of this paper is to detect increases of gate and line delays caused by resistive faults and other reasons (for example process variation) to reduce early life-failure.

This paper gives evidence that, for improving small-delay fault coverage of on-chip delay measurement, the use of segmented scan and *test point insertion* (TPI) is efficient.

The rest of the paper is organized as follows. Section 2 introduces some terminology and related works, including on-chip delay measurement. Section 3 analyzes the constraint of single-path sensitization and the reasons for low small-delay fault coverage. Section 4 explains methods for improving fault coverage. Section 5 evaluates the introduced methods. Finally, Sect. 6 concludes the paper.

2. Preliminaries

This section explains some terminology related to scan-based delay testing. We also introduce explain some related works for small-delay testing and the on-chip delay measurement method.

2.1 Transition Fault and Path Delay Fault

Transition fault and *path delay fault* are two prevalent fault models. The *transition fault* model targets each gate output in the design for a slow-to-rise and slow-to-fall delay fault while the *path delay fault* model targets the cumulative delay through the entire list of gates in a pre-defined path. *Transition fault* model is more widely used than *path delay fault* because it tests for at-speed failures at all nets in the design and the total fault list is equal to twice the number of nets. On the other hand, there are billions of paths in a modern design to be tested for path delay fault leading to high analysis effort [7]. This paper uses the transition fault coverage as the small-delay fault coverage because the aim of this paper is to detect increases of gate and line delays caused by resistive faults to reduce early life-failure. The *transition fault* is detected if a transition occurs at the fault site and if a sensitized path extends from the fault site to a primary output.

Transition faults are detected if a transition occurs at the fault site and if a sensitized path extends from the fault site to any primary output. From this reason, a path through the fault should be sensitized for testing the transition delay fault. In path delay fault testing, a signal is an *on-input* of path p if it is on path p. If a gate g is on path p and an input line of the gate g is not on p, the line is called an *off-input* of p [1], [17]. A logic value is the *controlling value* to a gate if it determines the output value of the gate regardless of the values on the other inputs to the gate. If the value on some input is the complement of the *controlling value*, the input is said to have a *non-controlling value*. Typically, there are three classes of path delay faults according to the sensitization criteria: single-path sensitization, robust sensitization and non-robust sensitization. Consider an AND gate with two inputs, this gate is a part of a target path p with one input as the on-input and the other input as the off-input for p. Table 1 shows the sensitization versus pairs of initial and final values in on-input and off-input of the AND gate. Symbol S0 (S1) represents a stable 0 (1) value on some signal under the initial value and final value. Symbol T0 (T1) represents a 1 to 0 (0 to 1) transition. H0 (H1) represents a 0 (1) value that might have hazard.

Table 1 sensitization versus pairs of initial and final values in on and off-input of an AND gate.

Sensitization	On-input	Off-input
Single path sensitization	T1	S1
	T0	S1
Robust	T1	S1, T1 or H1
	T0	S1
Non-robust	T1 or H1	S1, T1 or H1
	T0 or H0	S1, T1 or H1

2.2 Related Works

Recently, various methods for small-delay defect detection have been proposed. Methods using faster-than-at-speed have been proposed to detect delay faults [18], [19]. These methods use multiple clock frequencies that are higher than the system clock. These methods have a drawback that the test time is very long and it causes the test cost high. To detect small-delay faults, methods with delay fault testing using a ring oscillator have been proposed [20]–[22]. In these, the PUM is made a part of ring oscillator, delay of the target path can be translated into the oscillation period. However, the timing resolution is not very good. Some *time-to-voltage converter* (TVC) based schemes have been proposed [23]–[25]. The delay of the PUM is converted to a certain voltage, and by comparing the converted voltage with the reference voltage, the delay of the PUM can be obtained. These techniques give good timing resolution. However, the calibration is difficult.

Some on-chip path delay time measurement methods using embedded delay measurement were proposed [9]–[14]. In these, delay times of paths are measured. Datta et al. proposed a modified *vernier delay line* (VDL) technique for path delay measurement [9]. High-resolution delay measurement capability can be achieved by using this method. The paper [10] presented modified boundary scan cells in which a *time-to-digital converter* (TDC) is embedded. Tsai et al. proposed a *built-in delay measurement* (BIDM) circuit consisting of coarse and fine blocks, which is an extension of the modified VDL technique. A built-in-self delay testing methodology based on BIDM and self-calibration methods can be developed [11]. Pei et al. also proposed an area-efficient version of the modified VDL [12]. The feature of this method is delay range of each stage. The delay ranges increase by a factor of two gradually, which reduces the required stages. Thus, without decreasing delay measurement resolution, this method expands delay measurement range much more easily with significantly less hardware overhead. The authors' group proposed a measurement system, which

is different from the VDL method, to improve the accuracy of the measured value [13]. This method measures delay times of two paths: a path which includes the PUM, and the extra path whose length is almost equal to the redundant line of the path, is measured previously. The difference between the delay times gives the delay of the PUM. This method is able to give a precise measurement. In addition, a method with smaller execution time and circuit area has been proposed [14].

2.3 On-Chip Delay Measurement Method

Figure 1 shows the architecture of the on-chip path delay measurement method of [14]. The on-chip delay measurement system measures the delay of each path including a PUM; the input (output) of the PUM is the output (input) of a flip-flop (FF) in the left (right) scan chain. The delay measurement system consists of *delay value measurement circuit* (DVMC), *stop signal generator* (SSG, which is an N-to-1 multiplexer), and *circuit under test* (CUT). The embedded delay measurement circuit DVMC has two input lines, *start* line and *stop* line. The DVMC, which is a class of TDC, consists of a delay chain and an n-bit up counter. DVMC measures the time difference between the transition signals sent to *start* line and *stop* line. The clock line of the CUT *clk* is directly connected to *start* of the DVMC; the transition of *start* triggers the measurement. The DVMC starts the measurement when a positive transition of *clk* is sent to *start*. The input line of the FF_i in the right scan chain, ssg_{in_i}, is connected to the SSG; the SSG detects the transition at ssg_{in_i} and sends the transition to stop of the DVMC, by setting the corresponding control data of SSG. The input line *clk* is the clock signal of CUT. The line clk_i is the clock line of FF_i. The input of FF_i is connected to ssg_{out} through ssg_{in_i} and the SSG. The system measures a path including one clock line clk_j, a PUM p_i, and some redundant lines ssg_{in_i} and ssg_{out}. For example, after the measurement of the path $p' = clk_j\text{-}p_i\text{-}ssg_{in_i}\text{-}ssg_{out}$, by comparing the measured delay time with the expected delay time, small-delay defects on clk_j and p_i can be detected [14]. In this paper, we insert one DVMC circuit in one CUT. Thus, only one path is selected to be measured for each test.

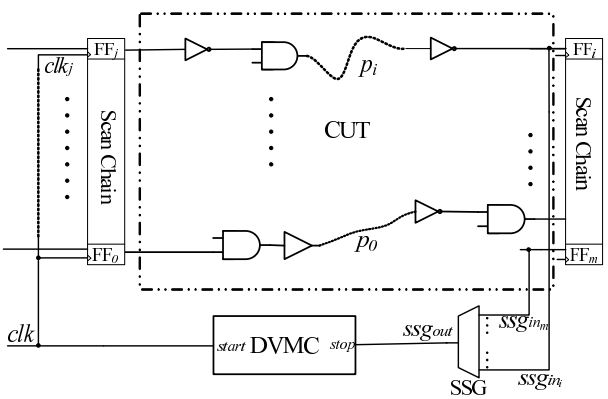


Fig. 1 Architecture of on-chip path delay measurement system.

Before the measurement, a test pattern (which sensitizes p_i) is assigned to the primary inputs and FFs of CUT. SSG is controlled to send the transition propagating to the output of p_i, to ssg_{out}. Then, we start the measurement by launching a positive transition to *start*. The transition propagates to the *start* of DVMC after the rising edge of *clk*, and DVMC starts the measurement. At the moment the clock rising edge reaches FF_j, a transition is launched to p_i. The transition reaches *stop* of DVMC through p_i, ssg_{in_i} and ssg_{out}. Then, DVMC stops the measurement. The measured delay time of p' contains the delay of redundant lines ssg_{in_i} and ssg_{out}. By using the criterion of [8], we can detect small-delay defects occurring on p_i and the segments of clock trees. This method has a good measurement resolution enough to detect defects even if the path delay is short [13]. However, there is an important point for small-delay fault test. When the intended transition reaches the *stop* of DVMC through SSG, DVMC stops the measurement. If the transition on the off-input of a PUM affects measured delay time, an incorrect measurement result will be recorded. The incorrect measurement result leads to false error indications or test escapes.

3. Constraint of Single-Path Sensitization

In this section, we introduce the necessity of single-path sensitization in on-chip delay measurement using an example. We give some simulation results to prove that the robust test is not appropriate for on-chip delay measurement. We also analyze the reasons for low small-delay fault coverage.

3.1 Necessity of Single-Path Sensitization

Unlike the traditional delay fault test, for the on-chip delay measurement method we must consider the single-path sensitization condition. As already known, in traditional delay fault test, we have to test sensitizing PUM robustly and not non-robustly to detect delay faults on PUM regardless of the delay time to off-inputs. The same is true for the delay measurement method. However, this is not sufficient, which is explained below using the example of Fig. 2 and the simulation data in Figs. 3 and 4.

In Fig. 2, PUM is the path p_1-p_2-G-p_4, denoted with a bold line. Assume that all sub-paths p_1, p_2 and p_4 are sensitized, and the values of on-input I_{ON} and the off-input I_{OFF} of the AND gate G are T1. From Table 1, PUM is robustly tested, but is not single-path sensitized. Here, we

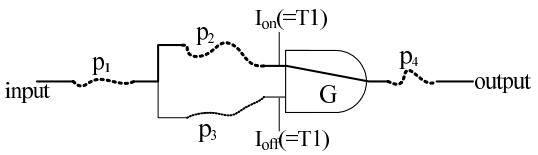


Fig. 2 Example of PUM not satisfying single-path sensitization condition.

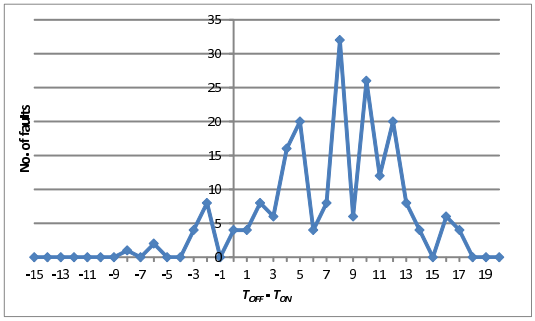


Fig. 3 Relation of number of faults and $T_{OFF}-T_{ON}$ for s5378.

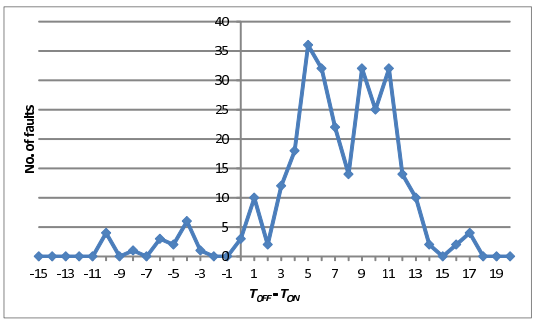


Fig. 4 Relation of number of faults and $T_{OFF}-T_{ON}$ for s9234.

set $T_{ON} = t_1+t_2+t_{ON}+t_4$ is the delay time of the PUM, and $T_{OFF} = t_1+t_3+t_{OFF}+t_4$ is the delay time of the path p_1-p_3-G-p_4, where t_i is the delay times of p_i and, t_{ON} and t_{OFF} are the gate delay of G from I_{ON} and I_{OFF}. By measuring the delay time from the input to the output, we obtain the following time:

$$T = max(T_{ON}, T_{OFF}).$$

If the expected delay time of t_3 is longer than t_2, we need to set the expected path delay to $T_{OFF}+3\sigma$. In other words, we should compare T with $T_{OFF}+3\sigma$. We cannot detect a resistive fault on p_2 with statistical analysis of T, unless the PUM has a sufficiently delay fault such that $T_{ON} \geq T_{OFF}+3\sigma$. It is likely to bring fault escapes in manufacturing testing, as a PUM typically has plenty of off-inputs. Else if the expected delay time of t_2 is longer than t_3, we need to set the expected path delay to $T_{ON}+3\sigma$. We can detect small-delay defects on the PUM, unless the off-input has a delay fault that causes $T_{OFF} \geq T_{ON}+3\sigma$. It is noted that in both of the two cases, we cannot detect small-delay defects on the PUM when $T_{OFF} > T_{ON}$. Even there has just one off-input of the PUM makes t_3 is longer than t_2.

To prove that the robust test is not appropriate for on-chip delay measurement, we show the relationship of T_{OFF} and T_{ON} using some experimental data. We use the test set (robust but not single-path sensitization) of s5378 and s9234; Figures 3 and 4 show results of s5378 and s9234, respectively. In these figures the x axis shows the delay time difference of T_{OFF} and T_{ON} using number of inverters, and the y axis shows the numbers of faults. The results show that in most cases (over than 90%) $T_{OFF} > T_{ON}$. It causes incorrect measurement result, and defects can be masked. These results demonstrate that, in the actual testing, it is difficult to ensure that the transition of on-input earlier than the transitions of all off-inputs. In other words, it is difficult to detect small-delay defects on the PUM under robust but not single-path sensitization. To guarantee that faults on the PUM are detected, we must make the measured time include t_2 regardless of t_3. We can satisfy this by using the single-path sensitization condition, setting S1 to I_{OFF}.

Table 2 Small-delay fault coverage of ISCAS89 benchmark of LOS test.

Circuit	F_{all}	F_{det}	Cov
s5378	8880	3886	43.76%
s9234	16442	7718	46.94%
s13207	23910	12297	51.43%
s35932	60634	36732	60.58%
s38584	68972	21340	30.94%

3.2 Small-Delay Fault Coverage

Because of the constraint of single-path sensitization, small-delay fault coverage of the test method using on-chip delay measurement is very low. Table 2 shows the small-delay fault coverage of ISCAS89 benchmark circuits of LOS test by using on-chip delay measurement. For example fault coverage of s38584 is less than 31%. From this, a method for improving small-delay fault coverage using on-chip delay measurement is strongly required.

4. Techniques for Improving Fault Coverage

This section introduces some techniques for improving fault coverage. These techniques are available for improving small-delay fault coverage of on-chip delay measurement. For higher fault coverage with an acceptable area overhead, we propose a procedure for using these techniques at the same time. The proposed method is a class of DFTs (designs for test), which facilitate testing. Although DFT increase area resulting in increase in the probability of defects, detecting small-delay faults with the DFT accomplishes shipment of dependable chips, which are free from manufacturing defects bringing about early-die.

4.1 Segmented Scan

Segmented scan is one of the techniques for improving the fault coverage [26]. Consider the part of a sequential circuit shown in Fig. 5, assume that the FFs are connected in the order of their numerical indices. Assume that a slow-to-fall small-delay fault occurs on line a. Under the single-path sensitization condition, the path (which includes the line a) cannot be sensitized since the initialization condition $FF_1 = FF_2 = 1$ implies $FF_3 = 1$ by 1 bit shift during the launch cycle. Thus, the off-input of the OR gate (line c) is set to controlling value, and the fault is blocked from being propagated to FF_4 during the capture cycle. Consider again, assume that the scan chain is partitioned into two segments

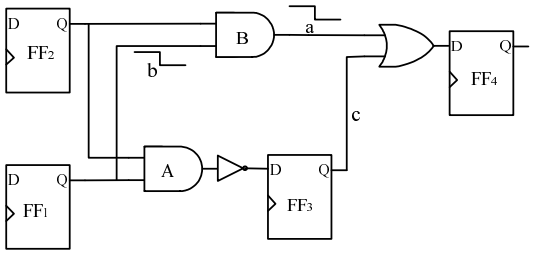


Fig. 5 Example circuit.

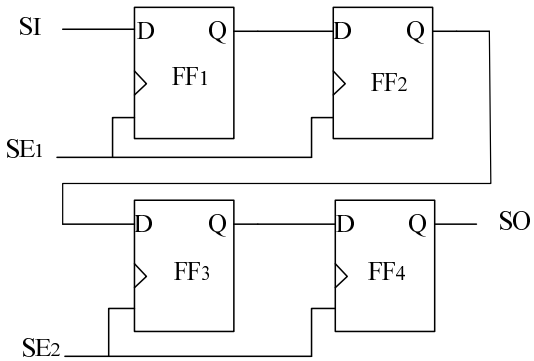
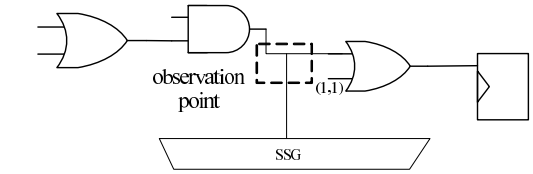


Fig. 6 Segmented scan.

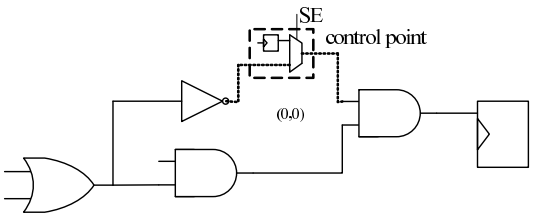
and each segment is connected to independent scan enable signals SE_1 and SE_2 as shown in Fig. 6. The initialization pattern (FF_1, FF_2, FF_3, FF_4) = (1, 1, 0, X) (X = do not care) is scanned in with both the scan enables SE_1 and SE_2 set 1. In the next cycle, we set SE_1 = 1, SE_2 = 0, then FF_3 contains the value of "0" instead of set to "1". Therefore, the fault effect is propagated to FF_4 and single-path sensitization is achieved. The example demonstrates that the path cannot be sensitized under the single-path sensitization condition. However, it can be achieved by using two scan segments. Thus, segmented scan technique can be utilized to improve small-delay fault coverage of on-chip delay measurement method.

4.2 Test Point Insertion

Test point insertion (TPI) for improving fault coverage is popular on design. There are two types of TPI methods, namely observation point insertion and control point insertion [27]. As shown in Fig. 7, observation point insertion involves making a node observable by making it a primary output or connecting it to the SSG. Control point insertion involves a selector with an activation signal where the activation signal is driven by a dedicated FF and can be set to a non-controlling value during the scan operation. Control point is enabled during the test operation and disabled during normal operation. As shown in Fig. 7 (b), when the SE signal is "1", the node is driven by the dedicated FF and set to a non-controlling value. The FFs are inserted into scan chain; the non-controlling values are provided by scan operation. The dedicated FF is driven by the system clock signal just like other scan FFs. In this section, we investigate a strategy for maximizing the fault coverage from a small number of observation points and control points, respectively.



(a) observation point



(b) control point

Fig. 7 Example of the test point insertion.

(1) Observation Point Insertion

Consider the sequential circuit shown in Fig. 5, assume that a slow-to-fall small-delay fault occurs on line *b*. Under the single-path sensitization condition, the path (which includes this fault) cannot be sensitized since the initialization condition FF_1 = FF_2 = 1 implies FF_3 = 1 by 1 bit shift during the launch cycle. Thus, the off-input of the OR gate (line *c*) is set to controlling value, and the fault is blocked from being propagated to FF_4 during the capture cycle. Here we insert an observation point after gate B (on line *a*) to observe the fault. We just need to ensure the values of FF_1 and FF_2 are (1, 1) and (0, 1) in the initialization pattern and launch pattern, respectively. By inserting the observation point, the transition on line *b* can reach *stop* of the DVMC, the sub-path (which includes the fault) is sensitized under the single-path sensitization condition. Thus, by comparing the measured delay time with the expected delay time, the transition fault on line *b* can be detected with criteria based on statistical analysis.

We place observation points in order to detect faults that are not detected by LOS test under the single-path sensitization condition. For keeping the discussion general, we assume a set of all faults in one circuit denoted F, and the set of faults, which are detected by LOS under the single-path sensitization condition, denoted F_{SINGLE}. We denote the set of target faults as U = F−F_{SINGLE}.

We determine the placement of observation points for U as follows. For illustration, we show an example of part of a CUT in Fig. 8. Assume that there are a fault on line G_1 and a fault on line G_2. Target to the fault G_1, because the off-input is controlling value, the fault is blocked at Gate 4 and Gate 5. Thus, the path including the fault G_1 cannot

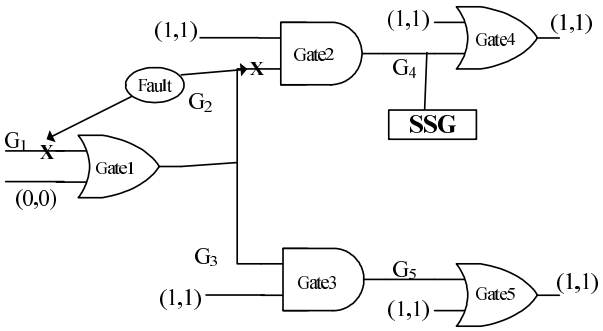


Fig. 8 Example of observation point insertion.

be sensitized under the single-path sensitization condition. The fault effects to lines G_2, G_3, G_4 and G_5. An observation point on any one of these lines allows the fault to be detected under the single-path sensitization condition. However, observation points on some of these lines may allow other faults to be detected. For example, the fault on line G_2 can be detected by inserting an observation point on line G_4. Thus, we should find the line which allows the most faults to be detected.

If we insert a test point on a line l that lies on a critical path, then inserting a test point on l may degrade performance. In order to prevent any possible performance degradation due to this, we do not insert any test point into signal lines on a critical path. Before the test point insertion procedure, we identify all signal lines that lie on a critical path. We delete these signal lines from the potential test point set.

We select a minimal subset of observation points applied for U by using a greedy covering procedure. The procedure for observation point insertion is given next as Procedure 1. To achieve higher fault coverage with an acceptable hardware overhead, the number of inserted observation point (N_O) is decided by results of hundreds of pre-simulation tests.

Procedure 1: Observation Point Insertion

1. Let U be the set of target faults. Let OB be an empty set, it is the set of lines used to insert observation points.
2. For every line g_i, let OBS(g_i) be an empty fault set. Find the set of faults OBS(g_i) such that can be detected with an observation point inserted on the line g_i.
3. Select a line g_j such that OBS(g_j) has the largest number of faults $tf_j \in U$.
4. Add the line g_j to OB. Remove faults $tf_j \in$ OBS(g_j) from U.
5. If U = $\emptyset$, or the number of observation points = pre-set value N_O, stop; else go to Step 3.

(2) Control Point Insertion

Consider the circuit shown in Fig. 5, assume that a slow-to-fall small-delay fault occurs on line a. Under the single-path sensitization condition, the path (which includes this fault) cannot be sensitized since the initialization condition $FF_1 = FF_2 = 1$ implies $FF_3 = 1$ by 1 bit shift during the launch cycle. Thus, the off-input of the OR gate (line c) is set to controlling value, and the fault is blocked from being propagated to FF_4 during the capture cycle. Here we insert a control point after gate A (on line c). Under the single-path sensitization condition, to detect the slow-to-fall delay fault of line a, we just need to set the value of the off-input of OR gate (line c) as non-controlling value (S0) in both the initialization pattern and launch pattern.

We place control points in order to detect faults that are not detected by LOS test under the single-path sensitization condition. We select a minimal subset of control points by using a greedy covering procedure. As the same with observation point insertion, to achieve higher fault coverage with minimal control point, we should consider the number of faults that can be detected by one control point insertion. In other words, control point which detects the largest number of faults will be first inserted. In order to prevent any possible performance degradation, we do not insert any test point into signal lines that are on a critical path. The procedure for control point insertion is given next as Procedure 2. To achieve higher fault coverage with an acceptable hardware overhead, the number of inserted observation point (N_C) is decided by results of hundreds of pre-simulation tests.

Procedure 2: Control Point Insertion:

1. Let U be the set of target faults. Let CO be an empty set, it is the set of lines used to insert control points.
2. For every line g_i, let COS(g_i) be an empty fault set. Find the set of faults COS(g_i) such that can be detected with a control point inserted on the line g_i.
3. Select a line g_j such that COS(g_j) has the largest number of faults $tf_j \in U$.
4. Add the line g_j to CO. Remove faults $tf_j \in$ COS(g_j) from U.
5. If U = $\emptyset$, or the number of control points = pre-set value N_C, stop; else go to Step 3.

(3) Procedure for Segmented Scan and Test Point Insertion

Based on Procedure 1 and Procedure 2, test points are inserted according to the number of faults that can be detected. In other words, test point which detects the largest number of faults will be first inserted. Thus, after a number of test points inserted, the effect for the coverage improvement will be not very notable. Hence, by using only one of the introduced techniques, we may not be able to get an ideal fault coverage under the single-path sensitization condition. For higher fault coverage, we use the segmented scan and test point insertion at the same time. The procedure for using all the introduced techniques is given next as Procedure 3. The values of N_S, N_C and N_O are decided by results of hundreds of pre-simulation tests. We can use these values to get a higher coverage with an acceptable hardware overhead.

Procedure 3: Segmented scan and test point insertion

1. Let L_0 be the length of CUT's scan chain, U be the set

of undetectable faults under the single-path sensitization condition, and N_S be the number of scan segments. We divide the scan chain to N_S segments, the length L of these segments is calculated by $L = L_0/N_S$. Each segment is controlled by a corresponding scan enable signal. After this step, let $U = U - F_S$, where F_S is the set of new detectable faults.

2. Let N_C be the number of control point will be inserted. Insert these control points using procedure 2.
3. Let N_O be the number of observation points will be inserted. Insert these observation points using procedure 1.

This procedure inserts control points before observation point insertions. This is because that the effect of the control point is better than the observation point in term of area overhead; the results of the pre-simulation tests which confirm this fact will be shown in the next section (Fig. 12).

5. Evaluation

In this section, we study the effects of these introduced techniques on the set of faults undetectable by LOS test under the single-path sensitization condition. The corresponding hardware overhead also will be evaluated. In this evaluation, we use ISCAS89 benchmark circuits. The results of cell area are obtained by synthesis with Synopsys design compiler using Rohm 180 μm process [28]. We also reported the maximal core utilization after layout with Synopsys IC Compiler. The maximal core utilization means the maximal value of core utilization that the IC Compiler can make layout (placement and routing) without errors. The chip area depends on both the maximal core utilization and cell area. The smaller value of maximal core utilization means larger chip area resulting from more complex routing. Figures 9~11 show the relation between maximal core utilization (in s5378 and s9234) and the numbers of scan segments, inserted observation points and inserted control points, respectively. These figures are explained later. The test patterns are generated with in-house ATPG based on what is used in [14]. First, we evaluate the effect of segmented scan. Next, we evaluate the effect of test point insertion (observation point insertion and control point insertion, respectively). We also compared the effects of control point insertion and observation point insertion for the pre-simulation tests to explain why the proposed procedure inserts control points before observation point insertions (as noted in the last paragraph of the previous section). For higher fault coverage, we evaluate the effect of segmented scan and test point insertion. Tables 3~6 show the evaluation results. In these Tables, the column *Circuit* shows the circuit name. The columns *[14]* and *Proposed* show the evaluation results of the method of [14] and the methods using introduced fault coverage improving techniques. The column N_T gives the number of the test pattern pairs (As only one path is selected to be measured for each test pattern pair, N_T also gives the number of measurements). Columns $C_0(\%)$ and $C_1(\%)$ report the fault coverage of [14] and methods using introduced fault coverage improving techniques, respectively. Columns $S_0(mm^2)$ and $S_1(mm^2)$ report the cell area of [14] and methods using introduced fault coverage improving techniques, respectively. The column C_{uti} reports the maximal core utilization after layout. Column N_S, N_C and N_O show the numbers of scan segments, inserted control points and inserted observation points, respectively. The column V shows the test data volume in 10^5 bit. The column T shows the test application time in 10^5 clocks. The column $C_{IMP}(\%)$ reports the effect of fault coverage improvement by using fault coverage improving techniques. The column $AO(\%)$ reports the area overhead, which is calculated by $AO = (S_1 - S_0)/S_0 \times 100(\%)$.

In addition, to achieve a still higher fault coverage with the same overall hardware overhead, we implemented

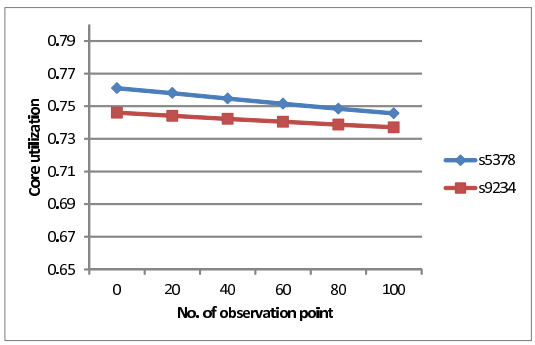


Fig. 10 Maximal core utilization vs. inserted observation points.

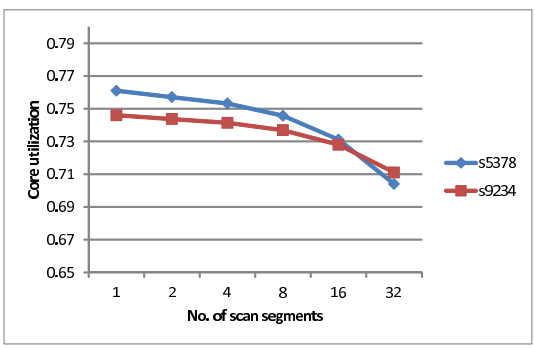


Fig. 9 Maximal core utilization vs. scan segments.

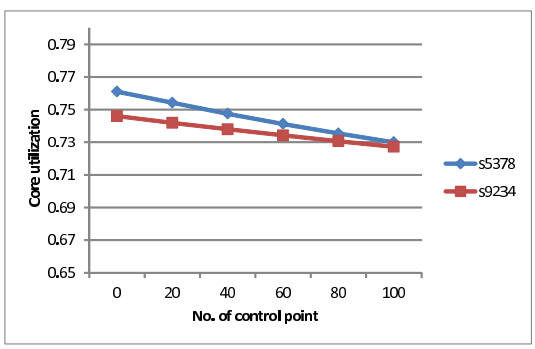


Fig. 11 Maximal core utilization vs. inserted control points.

Table 3 Effect of segmented scan.

Circuit	[14]				*Proposed*						
	N_T	C_0(%)	S_0(mm^2)	C_{uti}	N_S	N_T	C_1(%)	S_1(mm^2)	C_{uti}	C_{IMP}(%)	AO(%)
s5378	160	43.76	0.118	0.76	8	345	66.07	0.118	0.75	22.31	0.56
s5378	160	43.76	0.118	0.76	16	638	77.64	0.119	0.73	33.88	1.11
s9234	199	46.94	0.191	0.75	8	576	64.86	0.191	0.74	17.92	0.35
s9234	199	46.94	0.191	0.75	16	726	68.35	0.192	0.73	21.41	0.68
s13207	330	51.43	0.357	0.73	8	507	66.14	0.358	0.73	14.71	0.18
s13207	330	51.43	0.357	0.73	32	1048	72.45	0.360	0.71	21.02	0.73
s38584	3778	60.58	0.889	0.74	8	5734	73.25	0.890	0.73	12.67	0.07
s38584	3778	60.58	0.889	0.74	64	7208	79.48	0.895	0.72	18.90	0.58
s35932	3213	30.94	0.963	0.75	8	6425	58.49	0.963	0.75	27.55	0.07
s35932	3213	30.94	0.963	0.75	64	6498	59.13	0.968	0.74	28.19	0.54

Table 4 Effect of observation point insertion.

Circuit	[14]				*Proposed*						
	N_T	C_0(%)	S_0(mm^2)	C_{uti}	N_O	N_T	C_1(%)	S_1(mm^2)	C_{uti}	C_{IMP}(%)	AO(%)
s5378	160	43.76	0.118	0.76	100	315	57.11	0.131	0.75	13.35	11.33
s5378	160	43.76	0.118	0.76	200	336	62.43	0.144	0.73	18.67	22.65
s9234	199	46.94	0.191	0.75	100	538	64.24	0.204	0.74	17.30	6.99
s9234	199	46.94	0.191	0.75	200	722	69.00	0.217	0.73	22.06	13.98
s13207	330	51.43	0.357	0.73	100	476	56.26	0.370	0.73	4.83	3.73
s13207	330	51.43	0.357	0.73	300	502	61.34	0.397	0.72	9.91	11.20
s38584	3778	60.58	0.889	0.74	400	4213	66.39	0.943	0.73	5.81	6.00
s35932	3213	30.94	0.963	0.75	400	5947	51.74	1.016	0.75	20.80	5.54

Table 5 Effect of control point insertion.

Circuit	[14]				*Proposed*						
	N_T	C_0(%)	S_0(mm^2)	C_{uti}	N_C	N_T	C_1(%)	S_1(mm^2)	C_{uti}	C_{IMP}(%)	AO(%)
s5378	160	43.76	0.118	0.76	40	622	58.90	0.125	0.75	15.14	6.34
s5378	160	43.76	0.118	0.76	100	717	72.61	0.136	0.73	28.85	15.86
s9234	199	46.94	0.191	0.75	60	527	57.04	0.202	0.73	10.10	5.87
s9234	199	46.94	0.191	0.75	100	564	62.09	0.209	0.73	15.15	9.78
s13207	330	51.43	0.357	0.73	60	1944	77.76	0.368	0.73	26.33	3.14
s38584	3778	60.58	0.889	0.74	100	4568	68.79	0.897	0.73	8.21	0.84
s35932	3213	30.94	0.963	0.75	40	10414	69.23	0.981	0.75	38.29	1.94

Table 6 Effect of segmented scan and test point insertion.

Circuit	[14]						*Proposed*								
	N_T	C_0	S_0	C_{uti}	V	T	(N_S/N_C/N_O)	N_T	C_1	S_1	C_{uti}	V	T	C_{IMP}	AO(%)
s5378	160	43.76	0.118	0.76	0.69	0.31	32/40/20	856	91.50	0.130	0.69	3.80	2.50	47.74	10.81
s9234	199	46.94	0.191	0.75	0.98	0.49	64/70/40	1162	84.56	0.214	0.67	5.93	3.11	37.62	12.35
s13207	330	51.43	0.357	0.73	4.62	2.26	64/100/50	1956	87.66	0.388	0.69	27.78	13.33	36.23	8.54
s38584	3778	60.58	0.889	0.74	110.62	55.44	128/200/100	11425	87.60	0.950	0.70	337.04	165.23	27.02	6.85
s35932	3213	30.94	0.963	0.75	113.29	56.02	128/200/100	10684	72.77	1.024	0.71	379.07	186.02	41.83	6.33

the proposed procedure several times with the same overall hardware overhead (by changing the area ratio of control point insertion and observation point insertion in the same overall hardware overhead). Figures 13 and 14 show the results of s5378 and s9234.

5.1 Effect of Segmented Scan

We improved the fault coverage 12.67~22.31% by only using 8 scan segments. However, with the increasing of scan segment's numbers, the effect for the coverage improvement is not very notable for some circuits. For example, as shown in Table 3 in the circuit s35932, we used 8 scan segments, the coverage improvement was 27.55%. However, when we used 64 scan segments, the fault coverage is just improved 0.64% compared to 8 scan segments. For higher fault coverage, we must consider to use other techniques. Figure 9 shows that the maximal core utilization became smaller with increasing the number of scan segments. It means that increasing the number of scan segments increases the number of redundant lines and makes routing more difficult. As a result, it causes the larger total area.

5.2 Effect of Test Point Insertion

We improved the fault coverage with 4.83~22.06% by inserting 100~400 observation points. The fault coverage can be improved with 8.21~38.29% by inserting 40~100 con-

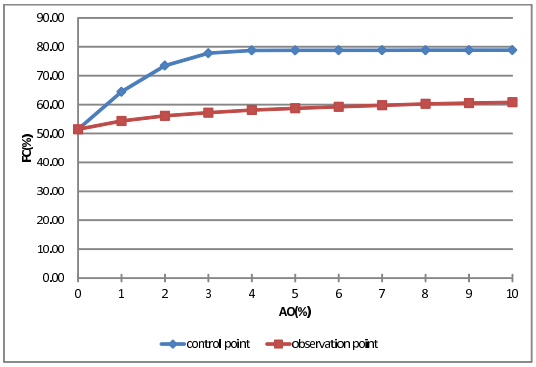


Fig. 12 Comparison of the effects of control point insertion and observation point insertion.

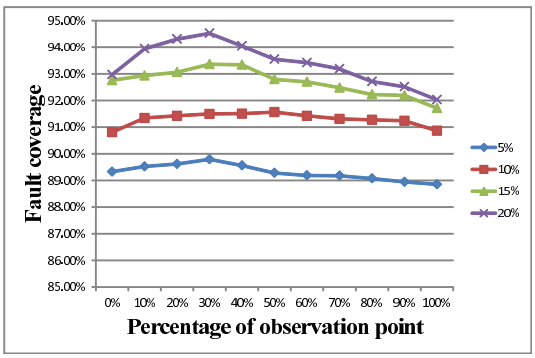


Fig. 13 Result of s5378 using 32 scan.

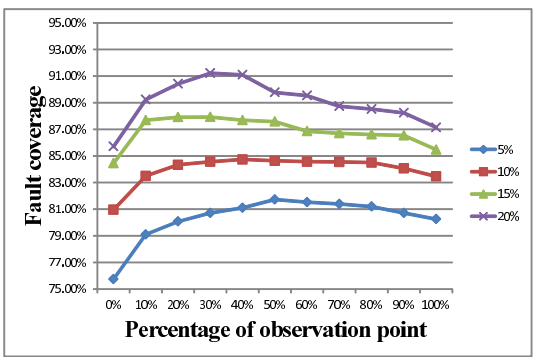


Fig. 14 Result of s9234 using 64 scan.

trol points. However, after a number of test points inserted, the effect for the coverage improvement became not very notable. For the aim of higher fault coverage, we need to insert more test points or use other techniques. From the results in Figs. 10 and 11, the maximal core utilization became smaller with increasing the number of inserted test points. It means that increasing the number of inserted test points leads to more difficult routing, and it causes the total area becomes larger. As shown in Table 4, the decrease of maximal core utilization is 0.03 (from 0.76 to 0.73) when we insert 200 observation points in s5378. However, the decrease of maximal core utilization is only 0.01 (from 0.74 to 0.73) even we insert 400 observation points in s34584. We can get similar results in control point insertion in Table 5. We found that the decrease is smaller in large circuits when we insert the same (or more in large circuit) number of test points. This means that the increases of the redundant lines and difficulty in routing caused by our DFT design are small in large circuits. In other words, the effect on layout of our proposed method is less serious in large circuits.

To decide whether observation point insertion follows control point insertion, we also compared the effects of control point insertion and observation point insertion with the same hardware overhead. Figure 12 presents the results of s13207. In Fig. 12, the y axis and the x axis show the fault coverage(%) and the hardware overhead(%). Here, we set the hardware overhead of test point insertion from 1% to 10%. From the experiment result, we found that the effect of the control point insertion is better than the observation point. For example, the fault coverage is improved more than 36% by using control point insertion with only 3% hardware overhead, while the improvement of the fault coverage is less than 6% by using observation point insertion with the same hardware overhead. Therefore, control points are inserted before observation points in Procedure 3.

5.3 Effect of Segmented Scan and Test Point Insertion

To achieve higher fault coverage with an acceptable hardware overhead, we use segmented scan, observation point insertion and control point insertion at the same time by using procedure 3. The evaluation results are shown in Table 6. As shown in the results, we got an acceptable fault coverage by using these techniques at the same time. Fault coverage can be improved 27.02~47.74% with 6.33~12.35% of hardware overhead.

5.4 Effective Fault Coverage by the Same Overall Hardware Overhead

To achieve a still higher fault coverage with the same overall hardware overhead, we implemented the proposed procedure several times with the same overall hardware overhead (by changing the area ratio of control point insertion and observation point insertion in the same overall hardware overhead). Figures 13 and 14 show the results of s5378 and s9234.

The y axis in Figs. 13 and 14 shows the fault coverage, and the x axis shows the area ratio of the overhead for the observation point insertion to the whole overhead. For example, in the case of the overall hardware overhead is 10% and the area ratio for the observation point insertion is 40%, we insert control points and observation points with area ratio of 6:4. Here, we set the overall hardware overhead as 5%, 10%, 15% and 20%. Figure 13 shows the result of s5378 using 32 scan segments, and Fig. 14 shows the result of s9234 using 64 scan segments.

From the experiment result, we can achieve a still higher fault coverage with the same hardware overhead. For example, for s5378 using 32 scan segments, we can achieve

the most effective fault coverage of 91.57% when the observation point insertion occupies 50% of the overall hardware overhead (when the overall hardware overhead is 10%). For s9234 using 64 scan segments, we can achieve the most effective fault coverage of 91.22% when the observation point insertion occupies 30% of the overall hardware overhead (when the overall hardware overhead is 20%). From the experiment result of Figs. 13 and 14, to achieve a still higher fault coverage with the same hardware overhead, we should set the area of observation point insertion occupies 30∼50% of the overall hardware overhead.

6. Conclusion

Our pre-simulation results show that single-path sensitization is required when using on-chip delay measurement method. This constraint makes the fault coverage very low. To improve small-delay fault coverage under the single-path sensitization condition, this paper introduced techniques using segmented scan and test point insertion. For higher fault coverage, we propose a procedure for using these techniques at the same time. As the evaluation results, the proposed procedure improved the fault coverage 27.02∼47.74% with 6.33∼12.35% of hardware overhead. To achieve a still higher fault coverage with the same hardware overhead, we should set the area of observation point to occupy 30∼50% of the overall hardware overhead.

In this paper, we focus only on the LOS test. However, the proposed procedure can be extended to improve fault coverage for other test designs. As our future work, for higher fault coverage and smaller hardware overhead, we will apply the proposed procedure on LOC test and LOS+LOC. Our future work also includes test data and test application time reduction.

Acknowledgments

This work was supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc., Cadence Design Systems, Inc. and Mentor Graphics, Inc.

References

[1] A. Krstic and K.T. Cheng, Delay fault testing for VLSI circuits, Kluwer Academic Publishers, 1998.
[2] A.D. Singh and G. Xu, “Output hazard-free transition tests for silicon calibrated scan based delay testing,” Proc. IEEE VLSI Test Symp., pp.349–357, 2006.
[3] S.K. Sunter, “BIST vs. ATE: Need a different vehicle?,” Proc. IEEE Int’l Test Conf., p.1148, 1998.
[4] X. Qian and A.D. Singh, “Distinguishing resistive small delay defects from random parameter variations,” Proc. IEEE Asian Test Symp., pp.325–330, 2010.
[5] Semiconductor Research Corporation, Research Challenges in Test and Testability, 2006.
[6] A.D. Singh, “Scan based testing of dual/multi core processors for small delay defects,” Proc. IEEE Int’l Test Conf., pp.1–8, 2008.
[7] M. Tehranipoor and N. Ahmed, Nanometer Technology Designs: High-Quality Delay Tests, Springer Publishing Company, 2007.
[8] K. Noguchi, K. Nose, T. Ono, and M. Mizuno, “A small-delay defect detection technique for dependable LSIs,” Proc. IEEE Symp. VLSI Circuits, pp.64–65, 2008.
[9] R. Datta, A. Sebastine, A. Raghunathan, and J.A. Abraham, “On-chip delay measurement for silicon debug,” Proc. ACM Great Lakes Symp. VLSI, pp.145–148, 2004.
[10] H. Yotsuyanagi, H. Makimoto, and M. Hashizume, “A boundary scan circuit with time-to-digital converter for delay testing,” Proc. IEEE Asian Test Symp., pp.539–544, 2011.
[11] M.C. Tsai, C.H. Cheng, and C.M. Yang, “An all-digital high-precision built-in delay time measurement circuit,” Proc. IEEE VLSI Test Symp., pp.249–254, 2008.
[12] S. Pei, H. Li, and X. Li, “A low overhead on-chip path delay measurement circuit,” Proc. IEEE Asian Test Symp., pp.145–150, 2009.
[13] T. Tanabe, K. Katoh, K. Namba, and H. Ito, “A delay measurement for VLSI circuit by subtraction,” IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J93-D, no.4, pp.460–468, April 2010.
[14] K. Katoh, K. Namba, and H. Ito, “A low area on-chip delay measurement system using embedded delay measurement circuit,” Proc. IEEE Asian Test Symp., pp.343–348, 2010.
[15] R. Datta, A. Sebastine, and J.A. Abraham, “Delay fault testing and silicon debug using scan chains,” Proc. IEEE European Test Symp., pp.46–51, 2004.
[16] R. Datta, G. Carpenter, K. Nowka, and J.A. Abraham, “A scheme for on-chip timing characterization,” Proc. IEEE VLSI Test Symp., pp.24–29, 2006.
[17] K. Namba and H. Ito, “Test sets for robust path delay fault testing on two-rail logic circuits,” IEEE Trans. Comput., vol.60, no.10, pp.1459–1470, Oct. 2011.
[18] N. Ahmed, M. Tehranipoor, and V. Jayaram, “Timing-based delay test for screening small delay defects,” Proc. IEEE/ACM Design Automation Conf., pp.320–325, 2006.
[19] X. Fan, Y. Hu, and L.T. Wang, “An on-chip test clock control scheme for multi-clock at-speed testing,” Proc. IEEE Asian Test Symp., pp.341–348, 2007.
[20] M. Collins and B.M. Al-Hashimi, “On-chip time measurement architecture with femtosecond timing resolution,” Proc. IEEE European Test Symp., pp.103–110, 2006.
[21] X. Wang, M. Tehranipoor, and R. Datta, “Path-RO: A novel on-chip critical path delay measurement under process variations,” Proc. IEEE/ACM Int’l Conf. Computer-Aided Design., pp.640–646, 2008.
[22] A. Jain, A. Veggetti, D. Crippa, and P. Rolandi, “An on-chip flip-flop characterization circuit,” Proc. Int’l Conf. Integr. Circuit and Syst. Design, pp.41–50, 2010.
[23] M. Collins, B.M. Al-Hashimi, and N. Ross, “A programmable time measurement architecture for embedded memory characterization,” Proc. IEEE European Test Symp., pp.128–133, 2005.
[24] S. Ghosh, S. Bhunia, A. Raychowdhury, and K. Roy, “A novel delay fault testing methodology using low-overhead built-in delay sensor,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.25, no.12, pp.2934–2943, Dec. 2006.
[25] A. Raychowdhury, S. Ghosh, and K. Roy, “A novel on-chip delay measurement hardware for efficient speed-binning,” Proc. IEEE Int’l On-Line Testing Symp., pp.287–292, 2005.
[26] Z. Zhang, S.M. Reddy, I.P. Pomeranz, J. Rajski, and B.M. Al-Hashimi, “Enhancing delay fault coverage through low power segmented scan,” Proc. IEEE European Test Symp., pp.21–28, 2006.
[27] J.S. Yang, B. Nadeau-Dostie, and N.A. Touba, “Test point insertion using functional flip-flops to drive control points,” Proc. IEEE Int’l Test Conf., pp.1–10, 2009.
[28] H. Onodera, A. Hirata, T. Kitamura, and K. Tamaru, “P2Lib: Process portable library and its generation system,” IPSJ Journal, vol.40, no.4, pp.1660–1669, 1999.

Wenpo Zhang received B.E. degree from University of Electronic Science and Technology of China in 2004, and the M.E. degree from Chiba University in 2012. Currently, he is working toward the Ph.D. degree at Chiba University. His research interests include very large scale integration (VLSI) testing and design for testability. He is a student member of the IEEE.

Kazuteru Namba received B.E., M.E. and Ph.D. from Tokyo Institute of Technology in 1997, 1999 and 2002, respectively. He joined Chiba University in 2002. He is currently an Assistant Professor of Graduate School of Advanced Integration Science, Chiba University. His current research interests include dependable computing. He is a member of the IEEE and the IPSJ.

Hideo Ito was born in Chiba, Japan, on June 1, 1946. He received the B.E. degree from Chiba University in 1969 and the D.E. degree from Tokyo Institute of Technology in 1984. He joined Nippon Electric Co. Ltd. in 1969 and Kisarazu Technical College in 1971. He had been a member of Chiba University from 1973 until March 2012. He is currently a Professor Emeritus of Graduate School of Advanced Integration Science. He had been a member of the IEEE and the IPSJ.