

PAPER

Distribution-Controlled X-Identification for Effective Reduction of Launch-Induced IR-Drop in At-Speed Scan Testing*

Kohei MIYASE^{†a)}, Kenji NODA^{††}, Members, Hideaki ITO^{††}, Nonmember, Kazumi HATAYAMA^{††}, Takashi AIKYO^{††}, Yuta YAMATO[†], Members, Hiroshi FURUKAWA[†], Nonmember, Xiaoqing WEN[†], and Seiji KAJIHARA[†], Members

SUMMARY Test data modification based on test relaxation and X-filling is the preferred approach for reducing excessive IR-drop in at-speed scan testing to avoid test-induced yield loss. However, none of the existing test relaxation methods can control the distribution of identified don't care bits (X-bits), thus adversely affecting the effectiveness of IR-drop reduction. In this paper, we propose a novel test relaxation method, called Distribution-Controlled X-Identification (DC-XID), which controls the distribution of X-bits identified in a set of fully-specified test vectors for the purpose of effectively reducing IR-drop. Experiments on large industrial circuits demonstrate the effectiveness and practicality of the proposed method in reducing IR-drop, without lowering fault coverage, increasing test data volume and circuit size.

key words: ATPG, X-bit, X-identification, X-filling

1. Introduction

As the frequency of VLSI circuits is approaching the GHz domain, delay testing has become mandatory. At-speed testing, which captures a test response at the rated clock speed, is essential in order to guarantee a clock speed given by its specifications [1], [2]. Ideally, circuits-under-test (CUT) should be tested under the same environmental conditions as the functional operation. However, power consumption caused by high switching activity during testing causes several problems for at-speed testing.

For at-speed testing, the scan test scheme is usually utilized [1]. *Shift mode* and *capture mode* of at-speed scan testing vary in terms of the power related problems. Regardless of whether at-speed scan testing or slow-speed scan testing, shift mode operation requires a significant number of clock cycles in order to load test stimuli into the flip-flops being scanned. Power consumption due to switching activity in shift mode may cause heat, circuit damage, and the loading of incorrect test stimuli. Many methods for reducing shift power consumption have been proposed over the last decade [3], [4].

Figure 1 shows a clock diagram of the *Launch-Off-Capture (LOC)* scheme, which is widely used in practice. When the scan enable signal *SE* is 0, two capture pulses

are issued, the first one being the *launch pulse* and the second one being the *capture pulse*. The test cycle, which is the time difference between the launch and capture pulses, should be equal to the rated clock cycle. If excessive *IR-drop* caused by high switching activity occurs at the launch pulse, gate delay increases in the circuit during the test cycle. As a result, incorrect test responses may be loaded by the capture pulse.

Generally, IR-drop caused by high switching activity in testing is much higher than functional operation. This is because functional constraints are often ignored in test vector generation, and test clocking is not often used in functional operations [5], [6]. Therefore, the IR-drop induced by testing, which is *test-induced IR-drop*, may break the functional IR-drop margin causing a functionally normal circuit to fail, and resulting in *test-induced malfunction* [2], [7], [8].

An effective approach for avoiding test-induced malfunction is to reduce the switching activity induced by the launch pulse. Several methods for reducing *launch-induced switching activity* have been proposed. One-hot and multi-capture clocking schemes can reduce the number of clock domains that capture simultaneously [1]. DFT methods such as partial capture [4] can be used to allow only part of a circuit to capture. Switching activity can be directly reduced by generating proper logic values in ATPG [9], [10], assigning logic values to don't care bits (X-bits) by in-ATPG or post-ATPG X-filling [11]–[16]. Generally, the test generation approach is preferable, and *post-ATPG X-filling* in particular is now widely used in practice [12]–[14].

Typical post-ATPG X-filling usually consists of two parts: test relaxation and X-filling. Test relaxation, which is also called X-identification [17]–[19], obtains test cubes

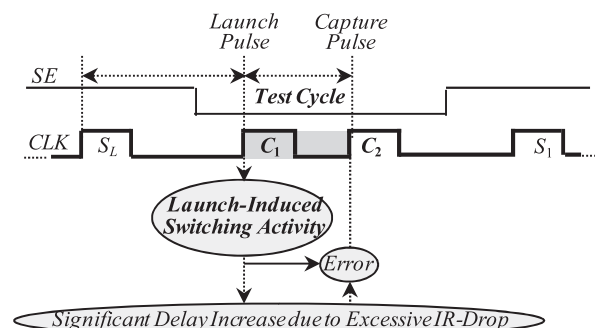


Fig. 1 Launch-off-capture (LOC) scheme.

Manuscript received July 2, 2010.

Manuscript revised November 30, 2010.

[†]The authors are with Kyushu Institute of Technology, Iizuka-shi, 820–8502 Japan.

^{††}The authors are with STARC, Yokohama-shi, 222–0033 Japan.

*A previous version of this paper has been published at International Conference on Computer Aided Design 2008.

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

DOI: 10.1587/transinf.E94.D.1216

including X -bits from a fully-specified test set. It has been found that even for a highly compacted test set, over 50% of X -bits can be identified without changing the fault coverage of the original test set [17], [18]. X -filling fills the identified X -bits with appropriate logic values in order to reduce launch-induced switching activity [12], [13], [15], [16], [26]. Since the post-ATPG X -filling only manipulates test data, it has such unique advantages as (1) no impact on test data volume, (2) no impact on circuit size, and (3) no impact on circuit timing. Therefore, post-ATPG X -filling can be easily implemented into any ATPG flow in order to efficiently reduce launch-induced switching activity.

Effectiveness of post-ATPG X -filling depends on the characteristics of the X -bits in the test cubes. It is obvious that a high percentage of X -bits is likely to lead to greater effect in reducing launch-induced switching activity. Consequently, a low percentage of the X -bits often cause high launch-induced switching activity and/or high launch-induced IR-drop. However, the fact that the distribution of X -bits also has a significant impact on the reduction effect is relatively unknown. The existing X -identification methods [17]–[19] do not take into consideration the distribution of X -bits. Therefore, few X -bits are capable of being identified in a test vector with high launch-induced switching activity. Obviously, ignoring the distribution of X -bits in X -identification will lead to the ineffective reduction of launch-induced switching activity.

X -identification [13] sorts test vectors according to their WSA (Weighted Switching Activity), and then identifies the larger number of X -bits in test vectors causing high WSA. However, it is difficult to accurately estimate the degree of IR-drop with WSA. Cutting-edge EDA tools still need a significant amount of computation time to accurately estimate IR-drop. Therefore, X -identification [13] may not identify larger number of X -bits in test vectors causing high IR-drop

In this paper, we propose a novel test relaxation method, called *Distribution-Controlled X-Identification (DC-XID)*, which controls the distribution of X -bits identified from a set of fully-specified test vectors for the purpose of effectively reducing IR-drop. We first propose a method to control X -bit distribution by controlling fault counts that each test vector targets in X -Identification procedure. Next, we control X -bit distribution for IR-drop reduction. In this work, we suppose a situation where IR-drop estimation tools are not available due to the significant amount of computation time required and/or results are unreliable due to inaccuracy. Hence the proposed method controls the distribution of X -bits so that the percentage of X -bits identified is uniform for each test vector. Furthermore since the proposed method avoids identifying few X -bits from some test vectors, it can average the IR-drop reduction effect for each test vector. By identifying relatively larger percentage of X -bits for each test vector, feasibility of many applications to reduce IR-drop can be increased.

Although another proposed method computes power approximately [20], [21], the method still requires relatively

large amount of computation time. If we can take a great deal of time to estimate accurate IR-drop, or if any other accurate IR-drop estimation technique is available, the proposed method can control the distribution of X -bits based on the estimation to maximize the effectiveness of IR-drop reduction.

After performing distribution-controlled X -identification, we obtain test cubes including X -bits. Note that test cubes include X -bits while test vectors don't. The proposed strategy is also suitable for compression environments using a broadcaster because the basic techniques for post-ATPG X -filling are basically identical for the environments with or without compression.

We apply *Justification-Probability-Based X-filling* [16] which is a sophisticated technique in terms of reduction ratio of launch-induced switching activity and computation time to the test cubes. Experimental results show that the proposed X -identification can control the distribution of X -bits. The results also show that distribution actually helps to reduce the maximum IR-drop as well as the maximum launch-induced switching activity.

This paper is organized as follows: Sect. 2 discusses the background, Sect. 3 presents the results of preliminary experiments, Sect. 4 elaborates upon the distribution-controlled X -identification, and Sect. 5 shows experimental results. Section 6 concludes the paper and outlines future work.

2. Background

2.1 IR-Drop in At-Speed Scan Testing

Generally, IR-drop occurs in functional operation as well as during testing. Its direct effect is increased gate delay [2], [22], [23], which may cause timing violations. IR-drop in testing is usually higher than IR-drop in functional operation, due to (1) high switching activity of test operations such as scan shift, (2) test vectors ignoring functional constraints, and (3) test clocking which is not allowed in functional operation.

Particularly, in at-speed scan testing, test-induced IR-drop has significant impact. It has been reported that switching activity due to the launch pulse for a test vector is usually higher (33% higher as reported in [2]) than a functional vector. This causes excessive IR-drop, which in turn increases gate delay in the test cycle and allows timing violations to occur at the capture pulse. As a result, incorrect values may be loaded by the capture pulse, causing a test-induced malfunction. Since the test cycle is very short, the risk of test-induced malfunction is high in at-speed scan testing, especially for test vectors causing high IR-drop. Therefore, it is important to reduce the maximum IR-drop at the launch pulse in order to avoid test-induced malfunction.

2.2 Post-ATPG X -Filling

Post-ATPG X -filling consists of two parts: test relaxation

(*X*-identification) and *X*-filling. Here we briefly summarize each part before describing a proposed method.

X-Identification is to identify *X*-bits in a set of fully-specified test vectors without lowering fault coverage. The major advantage of obtaining test cubes by *X*-identification, instead of simply keeping unspecified bits in ATPG, is that a compacted initial test set can be obtained through dynamic compaction and random-fill; otherwise, the test vector count may increased even double [12]. Even in a compacted initial test set, *X*-identification is capable of finding over 50% of the *X*-bits [17],[18]. Figure 2 and Table 1 show examples of *X*-bits. Table 1 (a) presents a test set generated for all transition delay faults when the circuit is full-scan designed in Fig. 2. Table 1 (b) presents a set of test cubes including *X*-bits. Although the test cubes include *X*-bits, the all transition delay faults for the circuit are still detected. Several *X*-identification methods have been proposed to find *X*-bits [17]–[19].

X-filling is conducted to assign an appropriate logic value to each *X*-bit in order to achieve a certain purpose, such as test data reduction [18], test power reduction [11]–[16], etc. Methods utilizing *X*-filling have been proposed to reduce test-induced IR-drop [12]–[15]. There are two major approaches. One of them is called *Justification-Based X-Filling* [14], which achieves high effectiveness but may suffer in terms of long computation time. The other is called *Probability-Based X-Filling (Preferred Fill)* [12], which achieves high scalability but its effectiveness depends on probability calculation. Recently a method called *Justification-Probability-Based Fill* [16] has been proposed. Since the method [16] balances effectiveness and scalability, it can be applied to large industrial circuits.

2.3 Contributions of This Work

Existing IR-drop reduction with post-ATPG *X*-filling may suffer in terms of the negative distributions to various purposes, which is the weakness of the post-ATPG *X*-filling. *X*-identification [17],[18] does not take the distribution of *X*-bits into account. Even though the method [17],[18] can change the distribution by sorting the original test vectors proposed in [13], the reduction results may not be sufficient,

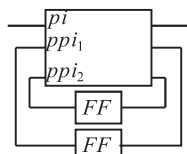


Fig. 2 Example circuit.

Table 1 (a) Given test set (b) test cube with *X*-bits.

	pi	ppi_1	ppi_2		pi	ppi_1	ppi_2
t_1	1	1	0	t_1'	1	1	X
t_2	1	0	1	t_2'	1	0	1
t_3	0	1	0	t_3'	0	1	0
t_4	0	1	1	t_4'	X	1	1

as explained later in Sect. 3.2. This paper attempts to overcome the weaknesses of existing IR-drop reduction methods by proposing *Distribution-Controlled X-Identification*, which can control the distribution of *X*-bits. In this work, we suppose a situation where IR-drop estimation tools are not available due to significant amount of computation time required and/or results are unreliable due to the inaccuracy of the tools. Hence the proposed method can average the percentage of *X*-bits identified from each test vector. Since no test cubes contain extremely minute numbers of *X*-bits, *X*-filling for reducing IR-drop performs almost equally on each test cube. Furthermore, if we can estimate IR-drop accurately, the proposed *X*-identification can identify a higher percentage of *X*-bits in test vectors causing high IR-drop. This issue will be addressed in our future works.

In the *X*-filling phase, we employ justification-probability-based fill in order to reduce launch-induced switching activity. Contrary to other papers [12],[13],[15],[16], the experimental results of this paper demonstrate IR-drop reduction. The results indicate that the proposed method to reduce launch-induced switching activity can also actually achieve the maximum IR-drop reduction.

3. Preliminary Experimentation

3.1 X-Bit-Distribution in Test Cubes

It is obvious that a test cube containing many *X*-bits is more flexible than one containing few *X*-bits. That is to say, we cannot expect a significant effect when conducting *X*-filling on a test cube containing few *X*-bits. None of the existing *X*-identification methods [17]–[19] controls the percentage of *X*-bits for each test cube. We call a distribution of the percentages in a test cube set as “*distribution of X-bits*” in this paper. Although *Specific Bits X-Identification* [24] can minimally control *X*-bit distribution, its effect is relatively small and it is difficult to average the *X*-bit distribution since the framework is the same as that described in [17].

Figure 3 shows the typical distribution of *X*-bits identified by an existing *X*-identification method [17] for an industrial circuit shown in Sect. 6 (Name: “cir2”, Size: 0.6M gates, Test vectors: 0.6K transition delay test vectors). In the example, few *X*-bits are identified in earlier test vectors

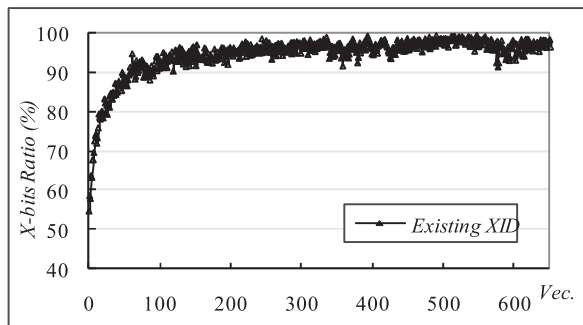


Fig. 3 X-bit-distribution for an industrial circuit “cir2”.

up to the 100th vector, while almost 90% of X -bits are identified in the rest of the vectors. We see that less than 60% of X -bits are identified in the first few vectors.

3.2 X -Bit-Distribution Impact

There is a significant correlation between the X -bit count in a test cube and the effect of X -filling for the test cube. In order to avoid test-induced malfunctions high launch-induced switching activity should be reduced, especially that of vectors causing the greater amount of switching activity. When no accurate IR-drop estimation techniques are available, it is impossible to determine which test vector causes test-induced malfunction. However, when an existing X -identification method identifies X -bits with a distribution like the one shown in Fig. 3, the effect of IR-drop reduction is obviously minimal in earlier vectors, especially when the earlier vectors cause test-induced malfunction. Therefore, test-induced malfunctions are very hard to avoid regardless of which X -filling method.

Figure 4 shows the launch-induced switching activity of the original test vectors and the test vectors after applying the X -filling technique outlined in [16] for the same circuit as in Fig. 3. We see that the more X -bits a test cube has, the more effective X -filling is in launch-induced switching activity reduction for the test cube. It is obvious that high switching activity still occurs for test vectors where few X -bits are identified. Figure 4 also shows the results for when larger numbers of X -bits are identified in test vectors causing high switching activity denoted by “Sorted XID”. Similar to the method described in [13], we sort the initial test vectors in increasing order of switching activities, and then X -identification as outlined in [17] is applied. Although many X -bits identified in the test vectors causing high switching activity, relatively high switching activity still occurs in test vectors where few X -bits are identified.

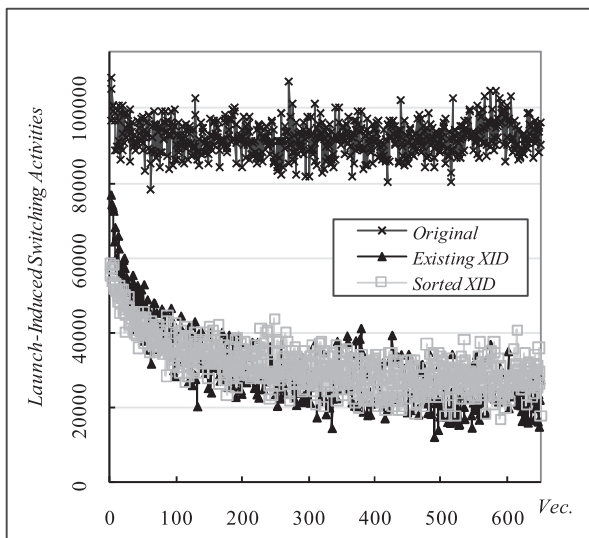


Fig. 4 Launch-induced switching activity.

4. Distribution Controlled X -Identification

4.1 Basic Procedure of X -Identification

Here we introduce the basic procedure of X -identification. Papers of [17],[18],[24] only consider stuck-at fault testing. In this paper, X -identification used is for transition delay fault testing with the LOC (Launch-Off-Capture) clocking scheme. Therefore, we have extended the existing X -identification, although the basic processes are the same. The procedure to identify X -bits consists of the following steps:

(Step-1) Select target faults that each test vector needs to detect so that the final fault coverage is not lowered.

(Step-2) Collect internal values that are required for guaranteeing the detection of the target faults.

(Step-3) Conduct dedicated implication and justification to obtain input values that can detect the target faults.

In Step-1, we select faults that each test vector needs to detect. In order to maintain the fault coverage, each fault has to be detected by at least one vector. When a fault can be detected by more than one vector, we pick one of the vectors to detect the fault. In Step-2, we collect internal values along fault propagation paths to detect the selected faults. When there is more than one fault propagation path, we pick one of them. Such internal values can be easily obtained by fault simulation. In Step-3, the dedicated implication and justification are conducted to justify the collected internal values. Note that unlike ATPG, the dedicated implication and justification do not cause any backtrack since fault detection is guaranteed by the original test vectors. Then we obtain input bits required to detect faults and finally identify the rest of the input bits as X -bits.

Figure 5 shows the concept of extended X -identification for transition delay faults in the LOC at-speed testing scheme. We use the two-timeframe circuit expansion model. In Fig. 5, the black area from the site of transition delay fault F in the before-launch pulse shows the internal logic values to be collected for initialization, and the gray area from the output in the after-launch pulse shows the internal values for fault propagation. From the collected internal values, we then use the dedicated implication and justification techniques to identify input bits in a test vector, whose logic values must be kept in order to detect the transition delay fault. After that, the rest of the input bits can be identified as the X -bits.

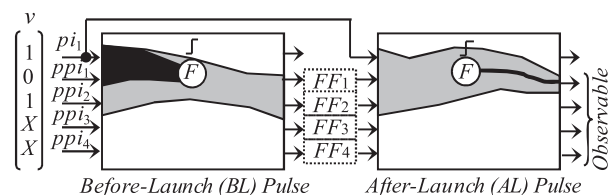


Fig. 5 X -identification for launch-off capture scheme.

Table 2 Detectable faults by each test vector.

	essential	non-essential				
v_1	f_1	f_5	f_6	f_7	f_8	f_{10}
v_2	f_2	f_4	f_5	f_7	f_9	f_{10}
v_3	f_3	f_4	f_5	f_6	f_8	f_9

Table 3 Faults targeted by existing X -identification.

	essential	non-essential				
v_1	f_1	f_5	f_6	f_7	f_8	f_{10}
v_2	f_2	f_4	f_9			
v_3	f_3					

Table 4 Faults targeted by proposed X -identification.

	essential	non-essential	
v_1	f_1	f_6	f_{10}
v_2	f_2	f_4	f_9
v_3	f_3	f_5	f_8

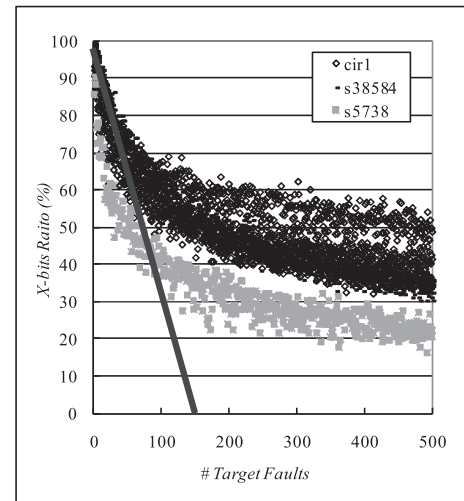
4.2 Controlling Fault Count Targeted

Figure 5 shows an example of how the X -identification targets only one fault per test vector. As shown in the example, some input bits are required to detect the fault. Obviously, more input bits are required when multiple faults are targeted for a single test vector. The number of required bits increases with the number of targeted faults for a single vector, although the correlation is not exactly proportional.

In order to maintain the original fault coverage, a fault must be detected by at least one test vector although it may be detected by multiple test vectors. When such faults are targeted in earlier test vectors, more faults are targeted than in later test vectors. Therefore, fewer X -bits are capable of being identified by the earlier test vectors.

Tables 2 and 3 show examples of this. Table 2 shows detectable faults in a test set composed of three test vectors. The total number of detectable faults is 10. In order to maintain the fault coverage of the given test set, each fault should be detected at least once. We can classify the faults as *Essential Faults* and *Non-essential Faults*. An essential fault is detected by only one test vector but any other vectors cannot detect the fault. In Table 2, f_1 , f_2 , and f_3 are essential faults. Table 3 shows an example of targeted faults. Many faults are targeted by v_1 while only one fault is targeted by v_3 . Since many input bits are required to detect many faults in v_1 , fewer X -bits will be identified. On the other hand, many X -bits will be identified in v_3 since only the bits that detect a single fault are required.

Table 4 shows an example when we identify approximately the same percentage of X -bits in each test vector. In this case, the proposed method averages the number of faults targeted by X -identification as shown in Table 4.

**Fig. 6** Fault count vs % X in a test cubes.

4.3 Fault Count Estimation According to the Desired Amount of X -Bits

The desired percentage of X -bits is determined by users. Once the desired percentage is given, the proposed method assigns the appropriate number of faults to each test cube. The number of faults assigned is different depending on the circuit size, the number of inputs, the complexity of the circuit function, and the number of test vectors. Usually the larger circuits are and the larger number of test vector is, the large number of X -bits is identified. When we determine the number of faults targeted by each test vector, we estimate the number of faults according to the given percentage of X -bits by analyzing the relation between the number of faults detected in a single test cube and the percentage of X -bits.

Figure 6 shows the percentage of X -bits when the number of faults assigned increases for two benchmark circuits (s5378: 3 K gates, 0.2 K FFs, s38584: 10 K gates, 1.4 K FFs) and an industrial circuit (cir1: 50 K gates, 1.1 K FFs). For simple visualization of the graph, we picked only three of test vectors for each circuit, then conduct X -identification as we change the number of faults assigned from 1 to 500. In order to estimate the number of faults for achieving the desired percentage of X -bits, we extract a linear function as shown in Fig. 6.

When the desired percentage of a test vector is high, the expected effectiveness of X -filling is strongly high. Therefore, we fit the function to the data plotted in high percentage of X -bits ratio. Although we are still investigating the optimal relation, the simple estimation technique can actually help to control the X -bits distribution as shown in Sect. 5.

4.4 Considering Accidental Detection

Step-2 (the internal value collection) and Step-3 (the dedicated implication and justification) described in Sect. 4.1 are not conducted on all faults. This is because input bits identi-

Table 5 Fault selection for non-essential faults.

	<i>targeted non-essential</i>		
v_1	f_a	f_b	f_c
v_2	f_d	f_e	f_g
v_3	f_h	f_i	f_j

fied to detect targeted faults accidentally detect other faults.

At first, the proposed distribution-controlled X-identification method conducts Step-2 and Step-3 only on essential faults for each test vector. Then, fault simulation is performed on test cubes detecting at least the essential faults. The fault simulation on these test cubes can drop non-essential faults. Steps-2 and 3 are not conducted on the dropped faults.

In order to control the distribution of X-bits, it is important to maintain the fault count selected in Step-1 (the target fault selection). In this paper, Step-2 and Step-3 are conducted for one-third of non-essential faults targeted by each test vector. Since non-essential faults can be detected by more than one test vector, other undetected faults may be accidentally detected. Although the complete investigation of accidental detections is expensive in terms of computation time, conducting Step-2 and Step-3 for one-third of the faults empirically helps to maintain the fault count selected with Step-1. In case some faults remain to be undetected, Steps-2 and 3 are performed on those undetected faults.

Table 5 shows an example of such fault selection. Faults in Table 5 are non-essential faults undetected by test cubes for detecting essential faults. Faults that each test cube must detect are selected. In Table 5, Steps-2 and 3 are first conducted on the circled faults and in order to obtain test cubes to detect them. Fault simulation on the test cubes can accidentally detect the faults which are squared in Table 5. Next, Steps-2 and 3 are conducted on the undetected fault left, which is f_g . In this way, the number of detections is controlled for each test cube, resulting in the controlled distribution of X-bits.

4.5 Procedure

The procedure of the distribution-controlled X-identification is shown in Fig. 7. Given a test set T and a desired percentage data DP , the distribution-controlled X-identification obtains a test cube set T' . Note that the fault coverage of T' is the same as the one of T .

The 1st part obtains a test cube set T' to detect all essential faults. Essential faults can be obtained by performing fault simulation with double detection mode. For each test vector t_i , we conduct X-identification consisting of Step-2 and Step-3 described in Sect. 4.1 regardless of dp_i (the desired percentage of X-bits for t_i) in DP . This is because each essential fault must be detected by the test vector which can detect the fault in order to maintain the fault coverage of T .

The 2nd part updates T' so that each test cube t'_i detects the specified number of faults based on dp_i . The num-

```

-----
Procedure of DC-XID( $T, DP$ )
  Test set  $T$ ; Desired percentage data  $DP$ ;
{
  /* 1st part */
  fault_simulation( $T$ );
   $EF$  = collect_essential_fault();
  for each test pattern  $t_i$  in  $T$  {
     $f$  = assign_essential_fault( $t_i$ );
     $t'_i$  = identify_X-bits( $t_i, f$ );
  }
  /* 2nd part */
  fault_simulation( $T'$ );
   $UF$  = collect_undetected_fault();
  for each test pattern  $t_i$  in  $T$  {
     $f$  = assign_desired_faults( $t_i, UF, dp_i$ );
     $t'_i$  += identify_X-bits( $t_i, f$ );
  }
  /* 3rd part */
  fault_simulation( $T'$ );
   $UF$  = collect_undetected_fault();
  for each test pattern  $t_i$  in  $T$  {
    if( $dp_i < X$ -ratio( $t_i$ )) {
       $f$  = assign_undetected_fault( $t_i, UF$ );
       $t'_i$  += identify_X-bits( $t_i, f$ );
    }
  }
  /* 4th part */
  fault_simulation( $T'$ );
   $UF$  = collect_undetected_fault();
  for each test pattern  $t_i$  in  $T$  {
     $f$  = assign_undetected_fault( $t_i, UF$ );
     $t'_i$  += identify_X-bits( $t_i, f$ );
  }
}
return  $T'$  composed of  $t'_i$ ;
}
-----

```

Fig. 7 Procedure of distribution-controlled X-identification.

ber of faults is determined with the fault count estimation technique described in Sect. 4.3.

In the 3rd part, undetected faults are assigned to a test cube t'_i which have the higher percentage of X-bits than dp_i . Then, update T' by conducting X-identification.

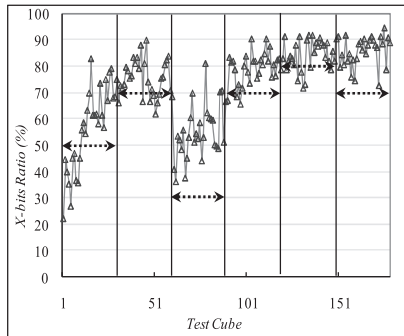
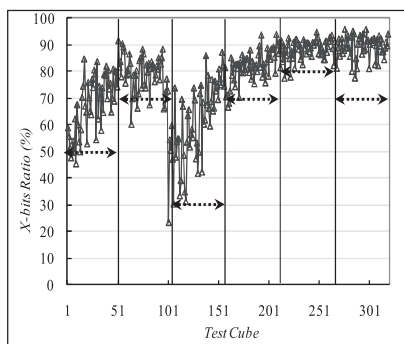
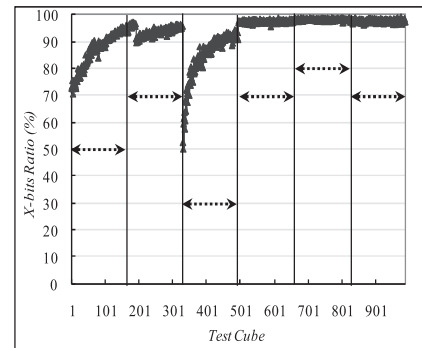
The 4th part updates T' so that all undetected faults are detected regardless of DP . As the results, the larger number of faults may be assigned a test cube than the one obtained with the fault count estimation techniques. However, since the number of undetected faults at the 4th part is few, there is no significant impact on the desired X-bits distribution.

5. Experimental Results

We implemented the proposed DC-XID method using C programming language and applied it to one of ISCAS 89 benchmark circuits and two industrial circuits. In this section we show results of X-bit distribution controlled, switching activity and IR-drop reduced by the propose method.

Table 6 Circuit and test set information.

Circuit	Size (# Gates)	# Scan Chains	# Scan Cells	# Tv	Fault Cov.
s5378	3k	1	179	178	84.8 %
cir1	50k	11	1,077	318	95.3 %
cir2	600k	64	35,851	991	90.0 %

**Fig. 8** Controlled X-bit distribution for s5378.**Fig. 9** Controlled X-bit distribution for cir1.**Fig. 10** Controlled X-bit distribution for cir2.**Table 7** Circuit and test set information.

Circuit	Size (# Gates)	# Scan Chains	# Scan Cells	# Tv	Fault Cov.
cir1	50k	11	1,077	318	90.0 %
cir2	600k	64	35,851	652	79.9 %

in Fig. 8, although the proposed method could obtain X -bits for each test cube closely to the required percentage, the less X -bits were found in some test vectors. The reason of uncontrollability of the X -bit distribution of the circuit is that the circuit size, the number of inputs, and the number of test vector is small.

Figure 9 shows results for the circuit “cir1”. For this circuit, the proposed method could obtain X -bits according to the desired percentage of X -bits. The percentage of X -bits for some test cubes is less than required one. Figure 10 shows results for the circuit “cir2”. In the results, the percentage of X -bits is much larger than the desired one. In terms of X -filling the larger percentage of X -bits is no problem. But these results indicate that some test cubes may have more X -bits. Depending on the circuit size the controllability of X -bit distribution is different.

5.1 Results of Distribution-Controlled X -Identification

We divided test vectors into 6 groups. We used a hypothetical desired percentage of X -bits. Table 6 shows the basic information of the circuits used in the experiments.

For circuit s5378, we generated transition delay test vectors by using our own ATPG tool based on [25]. For two industrial circuits, the test vectors are generated with “TetraMAX™” from Synopsys. Note that “Fault Cov.” for those industrial circuits actually means test coverage which the ATPG tool reported. For comparison between different sizes of circuits, we used the similar desired percentage of X -bits. Also we used the same linear function obtained from Fig. 6 to estimate the fault count according to the percentage of X -bits.

The desired percentage of X -bits is indicated with dotted bidirectional arrows in the graphs (Figs. 8, 9, and 10). It is important to note that there is no fault coverage loss after identifying X -bits with the DC-XID. In terms of the effects of X -filling, there is no problem when the identified percentage of X -bits is over the required one. For circuit “s5378”

5.2 Uniform X -Bit Distribution

The results in Fig. 8, 9 and 10 showed the proposed method could control X -bit distribution. In terms of reduction for IR-drop, it is preferable that many X -bits should be found in test vectors causing high launch-induced IR-drop. However, it is difficult to specify such test vectors since IR-drop analysis requires incredible computation time. One of solutions is that we identify X -bits uniformly in each test vector. As described in Sect. 1, it is important to reduce the high IR-drop in terms of avoidance of test-induced malfunction. Section 5.2, 5.3 and 5.4 show that uniform X -bit distribution can help to avoid high launch-induced IR-drop. Table 7 shows the basic information of the experiments. The test sets were generated with the ATPG tool “TetraMAX™” from Synopsys. The experiments in this section include the switching activity and IR-drop analyses which are for assessment of the proposed method. Since the industrial circuit “cir2” is much larger than the other circuits used in the experiments, the assessment time for “cir2” is significantly

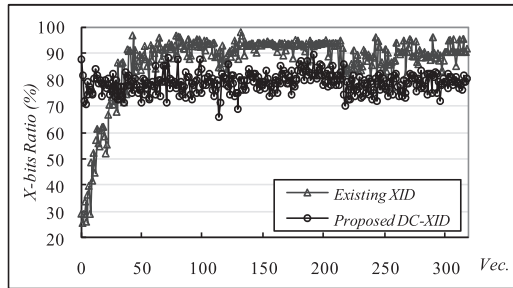


Fig. 11 Uniform X-bit distribution for cir1.

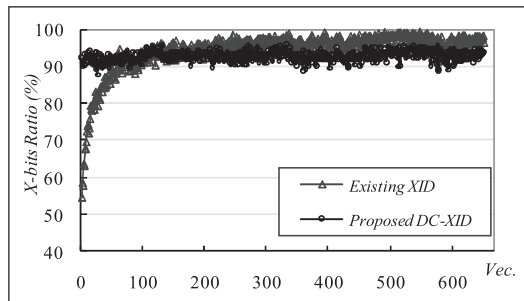


Fig. 12 Uniform X-bit distribution for cir2.

Table 8 Uniform X-bit distribution.

Circuit	Existing XID				Proposed DC-XID			
	X-bits (%)			SD(%)	X-bits (%)			SD(%)
	Ave.	Max.	Min.		Ave.	Max.	Min.	
cir1	86.7	98.1	25.5	12.7	78.7	89.4	66.0	3.7
cir2	94.3	99.3	54.5	5.5	92.8	96.0	87.7	1.3

large. Besides, the percentages of X-bits later than 200th vector show the similar tendency for circuit “cir2” as shown in Fig. 12. Therefore, we used the smaller test set than used in Sect. 5.1. Note that the fault coverage of the test set shown in Table 7 is still practical level and the proposed DC-XID does not decrease the fault coverage. As shown in Sect. 5.3 and Sect. 5.4, the proposed method achieves the high effectiveness even for the smaller test set.

Figure 11 and Fig. 12 show the distribution of X-bits identified by the XID [17] and the proposed DC-XID methods. It is obvious that the DC-XID method is capable of averaging the X-bit distribution. For “cir1”, the minimum percentage of X-bits identified was only 25% for a test vector when the XID method was used. Clearly, it is hard to reduce the launch-induced switching activity and launch-induced IR-drop for the test vector.

We summarized the results in Table 8. Table 8 presents the average, maximum, and minimum percentages of X-bits identified. Although the average percentage of X-bits identified by the DC-XID method is slightly smaller than that of the existing method, the minimum percentage of the DC-XID method is much higher than that of the XID method. The column “SD(%)” shows the standard deviation divided by the number of all inputs (PIs and PPIs). The DC-XID

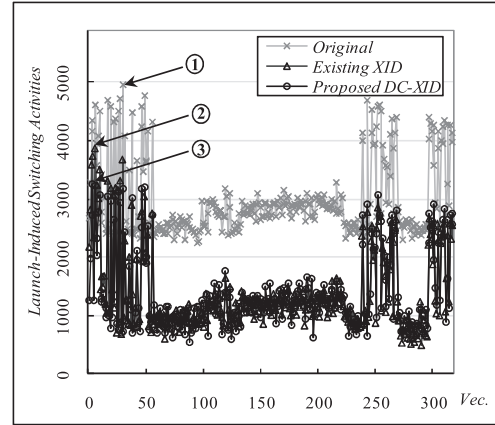


Fig. 13 Launch-induced switching activity for cir1.

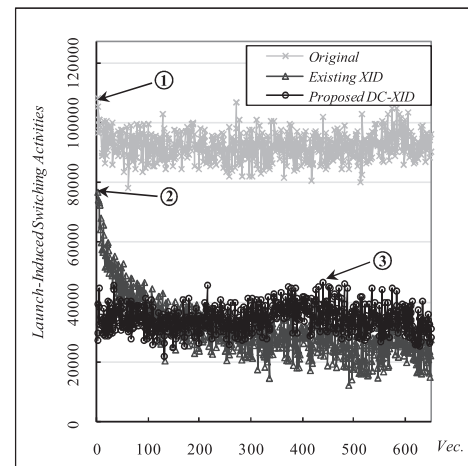


Fig. 14 Launch-induced switching activity for cir2.

method achieved smaller values for both of the two circuits. The results imply that the DC-XID method can indeed uniform the percentage of X-bits.

5.3 Switching Activity Analysis

Figure 13 and Fig. 14 show the switching activity of each test vector for the original test set ①, the one obtained by using the XID method ②, and the one obtained by the DC-XID method ③. The maximum switching activity is indicated by the arrows. Note that we utilized JP-fill [16] as an X-filling technique for test cubes. The maximum switching activity of the XID method ② occurs at test vectors where a smaller number of X-bits are identified, as seen in Fig. 11 and Fig. 12. For circuit “cir2” it is obvious that there are strong correlations between the distribution of X-bits in Fig. 12 and the switching activity in Fig. 14.

5.4 IR-Drop Analysis

We conducted the layout with “SoC Encounter™” from Cadence. We analyzed IR-drop with “RedHawk™” from

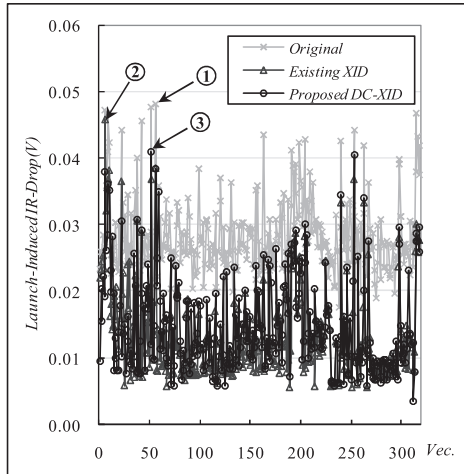
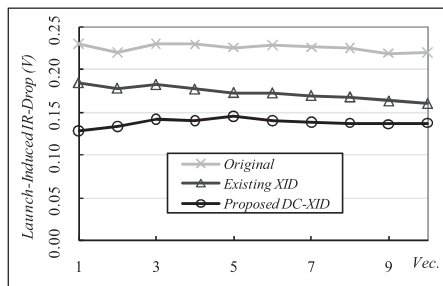
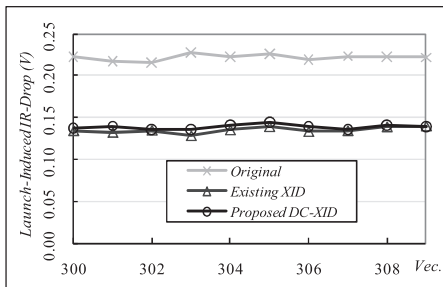


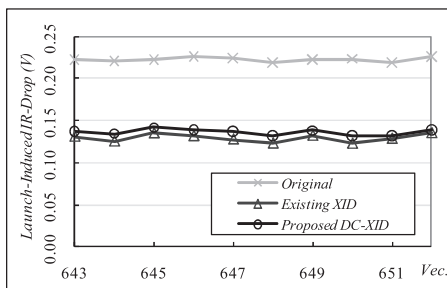
Fig. 15 Launch-induced IR-drop for cir1.



(a)



(b)



(c)

Fig. 16 Launch-induced IR-drop for cir2.

Apache. In Fig. 15, we present the launch-induced IR-drop results for “cir1”. The maximum IR-drops are indicated by arrows. We couldn’t run the tool for every test vector due to the long CPU time for “cir2”. Therefore, we selected several

Table 9 Reduction ratio.

Circuit	Reduction Ratio (%)			
	Max. Switching Activity		Max. IR-Drop	
	XID	DC-XID	XID	DC-XID
cir1	21.9	34.6	5.1	15.2
cir2	29.0	57.1	N/A	N/A

Table 10 CPU time.

Circuit	CPU Time (s)			
	Existing		Proposed	
	XID	JP-Fill	DC-XID	JP-Fill
cir1	25.5	64.91	217.4	129.5
cir2	1993.6	23131.2	13085.7	20579.4

sections of vectors (1–10, 300–309, 643–652). The results of IR-drop in those vector sections are shown in Fig. 16. The results for the first 10 vectors show that the DC-XID reduced more IR-drop than the XID reduced. This is because the XID method identified fewer X-bits from earlier test vectors, as shown in Fig. 12.

The reduction ratio of launch-induced switching activity and launch-induced IR-drop are summarized in Table 9. The left side of Table 9 shows the reduction ratio of maximum launch-induced switching activity. The proposed DC-XID method achieved a roughly 10% improvement for both of the circuits compared to the XID method. Notably, the proposed method achieved a reduction ratio of about 50% for circuit “cir2”. The right side of Table 9 shows the reduction ratio of the maximum launch-induced IR-drop. The proposed DC-XID also could achieve about 10% improvement for “cir1”.

5.5 CPU Time

Table 10 provides information on CPU time. The computer used has a 2.8 GHz CPU and 32 GB memory. The proposed method took a relatively long CPU time. However, shorter CPU times can be achieved by enhancing the program code and aligning the procedure in the justification-probability-based fill (JP-Fill) [16].

6. Conclusions

This paper proposed a novel X-identification method, called *Distribution-Controlled X-Identification*, which is capable of controlling the distribution of identified X-bits. Distribution-controlled X-identification helps to average the number of X-bits identified in each test vector so that test-induced IR-drop can be effectively reduced. Experimental results on industrial circuits demonstrated that the proposed method reduced maximum IR-drop more than the existing XID method. Future works include (1) controlling the distribution of X-bits in order to reduce IR-drop only in specific vectors if power analysis information is available, and (2) applying the proposed method to compression environments.

Acknowledgements

This work was partly supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (B) 22300017.

References

- [1] L.-T. Wang, C.-W. Wu, and X. Wen, eds., *VLSI Test Principles and Architectures: Design for Testability*, Elsevier, San Francisco, 2006.
- [2] J. Saxena, K. Butler, V.B. Jayaram, and S. Kundu, "A case study of IR-drop in structured at-speed testing," *Proc. Int'l Test Conf.*, pp.1098–1104, 2003.
- [3] R. Sankaralingam and N.A. Touba, "Inserting test points to control peak power during scan testing," *Proc. Intl. Symp. DFT*, pp.138–146, 2002.
- [4] S. Wang and W. Wei, "A technique to reduce peak current and average power dissipation in scan designs by limited capture," *Proc. ASP-DAC*, pp.810–816, 2007.
- [5] P. Girard, "Survey of low-power testing of VLSI circuits," *IEEE Des. Test Comput.*, vol.19, no.3, pp.82–92, May/June 2002.
- [6] N. Nicolici and B. Al-Hashimi, *Power-Constrained Testing of VLSI Circuits*, Kluwer Academic Publishers, 2003.
- [7] A.K. Kokrady and C.P. Ravikumar, "Static verification of test vectors for IR-drop failure," *Proc. ICCAD*, pp.760–764, 2003.
- [8] N. Ahmed, M. Tehranipoor, and V. Jayaram, "A novel framework for faster-than-at-speed delay test considering IR-drop effects," *Proc. Int'l Conf. Computer-Aided Design*, pp.439–444, 2005.
- [9] F. Corno, P. Prinetto, M. Redaudo, and M. Reorda, "A test pattern generation methodology for low power consumption," *Proc. VLSI Test Symp.*, pp.35–40, 1998.
- [10] X. Wen, S. Kajihara, K. Miyase, T. Suzuki, K.K. Saluja, L.-T. Wang, K.S. Abdel-Hafez, and K. Kinoshita, "A new ATPG method for efficient capture power reduction during scan testing," *Proc. VLSI Test Symp.*, pp.58–63, 2006.
- [11] W. Li, S.M. Reddy, and I. Pomeranz, "On reducing peak current and power during test," *Proc. ISVLSI*, pp.156–161, 2005.
- [12] S. Remersaro, X. Lin, Z. Zhang, S.M. Reddy, I. Pomeranz, and J. Rajski, "Preferred fill: A scalable method to reduce capture power for scan based designs," *Proc. Int'l Test Conf.*, Paper 32.2, 2006.
- [13] S. Remersaro, X. Lin, S.M. Reddy, I. Pomeranz, and J. Rajski, "Low shift and capture power scan tests," *Proc. VLSI Design 2007*, pp.793–798, 2007.
- [14] K.M. Butler, J. Saxena, T. Fryars, G. Hetherington, A. Jain, and J. Levis, "Minimizing power consumption in scan testing: Pattern generation and DFT techniques," *Proc. Int'l Test Conf.*, pp.355–364, 2004.
- [15] X. Wen, Y. Yamashita, S. Morishima, S. Kajihara, L.-T. Wang, K.K. Saluja, and K. Kinoshita, "Low-capture-power test generation for scan-based at-speed testing," *Proc. Int'l Test Conf.*, Paper 39.2, 2005.
- [16] X. Wen, K. Miyase, S. Kajihara, T. Suzuki, Y. Yamato, P. Girard, Y. Ohsumi, and L.T. Wang, "A novel scheme to reduce power supply noise for high-quality at-speed scan testing," *Proc. Int'l Test Conf.*, Paper 25.1, 2007.
- [17] K. Miyase and S. Kajihara, "XID: Don't care identification of test patterns for combinational circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol.23, no.2, pp.321–326, Feb. 2004.
- [18] A.H. El-Maleh and K. Al-Utaibi, "An efficient test relaxation technique for synchronous sequential circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol.23, no.6, pp.933–940, June 2004.
- [19] R. Sankaralingam and N.A. Touba, "Controlling peak power during scan testing," *Proc. VLSI Test Symp.*, pp.153–159, 2002.
- [20] V.R. Devanathan, C.P. Ravikumar, and V. Kamakoti, "Glitch-aware pattern generation and optimization framework for power-safe scan test," *Proc. VLSI Test Symp.*, pp.167–172, 2007.
- [21] N. Ahmed and M. Tehranipoor, "Transition delay fault test pattern generation considering supply voltage noise in a SOC design," *Proc. DAC*, pp.533–538, 2007.
- [22] A.K. Kokrady and C.P. Ravikumar, "Static verification of test vectors for IR-drop failure," *Proc. ICCAD*, pp.760–764, 2003.
- [23] J. Wang, D. Walker, A. Majhi, B. Kruseman, and S. Eichenberger, "Power supply noise in delay testing," *Proc. ITC*, Paper 17.3, 2006.
- [24] K. Miyase, S. Kajihara, I. Pomeranz, and S.M. Reddy, "Don't care identification on specific bits of test patterns," *Proc. Int'l Conf. Computer Design*, pp.194–199, 2002.
- [25] S. Kajihara, I. Pomeranz, K. Kinoshita, and S.M. Reddy, "Cost-effective generation of minimal test sets for stuck-at faults in combinational logic circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol.14, no.12, pp.1496–1504, Dec. 1995.
- [26] X. Wen, H. Yamashita, S. Kajihara, L.-T. Wang, K. Saluja, and K. Kinoshita, "On low-capture-power test generation for scan testing," *Proc. VLSI Test Symp.*, pp.265–270, 2005.



and fault diagnosis. He is a member of the IEEE.

Kohei Miyase received Ph.D. degrees in Computer Science and Systems Engineering from Kyushu Institute of Technology, Japan 2005. From 2005 to 2007, he was a researcher in Innovation Plaza Fukuoka, Japan Science and technology Agency, Japan. In 2007, he joined the Department of Computer Science and Electronics of Kyushu Institute of Technology, Japan, where he is an Assistant Professor currently. His research interests include test compression, design for testability, low power test,



Kenji Noda received the B.E. from Kyoto University, Japan, in 1995, and the M.E. from Nara Institute of Science and Technology, Japan, in 1997. He joined NEC Corporation in 1997. From 2006, he is temporarily transferred to Semiconductor Technology Academic Research Center. His research interests include low power test and design for testability.



Hideaki Ito received his B.S. degrees in applied physics from Osaka City University, Osaka, Japan, in 1989. He has been with Layout & Test Design Group in OKI SEMICONDUCTOR CO., LTD. Since 1989.



Kazumi Hatayama received his B.S., M.S. and Ph.D. degrees in applied mathematics and physics from Kyoto University, Kyoto, Japan, in 1976, 1978 and 1982, respectively. He is now a Team Leader in Test & Diagnosis Group in Semiconductor Technology Academic Research Center. He is Asia Pacific Regional Chair of Test Technology Technical Council, IEEE Computer Society, IP Track Co-Chair of VLSI Test Symposium and a Program Committee member of International Test Conference and other conferences.

He is a senior member of IEEE and a member of IPSJ, ORSJ and REAJ. His research interests include DFT, BIST, ATPG, fault diagnosis and fault tolerant computing.



Takashi Aikyo received the B.E. degree in electronics from Shibaura Institute of Technology, in 1978, and M.E. degree in electronic and communications engineering from Waseda University, in 1980. He joined Fujitsu Limited in 1980, and he was working on development of computer-aided design tools in LSI testing area. He is currently a senior manager at the Semiconductor Technology Academic Research Center, and he is also a Ph.D. student at the Graduate School of Science and Engineering, Ehime University.

He is a chair of the Dependable Computing Technical Group in IEICE. His research interests include design for testability, and fault diagnosis. He is a member of IEEE Computer society.



Yuta Yamato received his B.E., M.E., and Ph.D. degrees in Computer Science and Systems Engineering from Kyushu Institute of Technology, Japan in 2005, 2007 and 2010 respectively. In 2010, he joined the Fukuoka Industry, Science and Technology Foundation, Japan, where he is currently a researcher. His research interests include low power test, and fault diagnosis. He is a member of the IEEE.



Hiroshi Furukawa received the B.E. from Kumamoto University, Japan, in 1992. He joined NEC Micro Systems (Currently, Renesas Micro Systems Co., Ltd.) in 1992. He is studying towards his Ph.D. degree in the Creation Informatics Program at Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, Japan. His research interests include VLSI test.



Xiaoqing Wen received the B.E. degree from Tsinghua University, Beijing, China, in 1986, the M.E. degree from Hiroshima University, Hiroshima, Japan, in 1990, and the Ph.D. degree from Osaka University, Osaka, Japan, in 1993. He was a Lecturer at Akita University, Akita, Japan, from 1993 to 1997 and a Visiting Researcher at University of Wisconsin, Madison, USA, from October 1995 to March 1996. He joined SynTest Technologies, Inc., Sunnyvale, USA, in 1998, and served as its CTO until

2003. In 2004, he joined the Kyushu Institute of Technology, Iizuka, Japan, where he is currently a full Professor. His research interests include VLSI test, diagnosis, and testable design. He currently holds 15 U.S. Patents, 2 Japan Patents, and 22 pending U.S. and Japan Patents in VLSI testing. He co-authored and co-edited two books on general VLSI testing and power-aware VLSI testing. He received the 2008 IEICE-ISS Best Paper Award. He is a senior member of the IEEE and a member of the REAJ.



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 a Professor currently. His research interest includes 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. Dr. Kajihara is a member of the IEEE and the IPSJ. He serves on the editorial board of the Journal of Electronic Testing: Theory and Applications.