Effective IR-Drop Reduction in At-Speed Scan Testing Using Distribution-Controlling X-Identification

Kohei Miyase[1], Kenji Noda[2] Hideaki Ito[2], Kazumi Hatayama[2], Takashi Aikyo[2],
Yuta Yamato[1], Hiroshi Furukawa[1], Xiaoqing Wen[1], Seiji Kajihara[1]

1: Kyushu Institute of Technology
Iizuka, Japan

2: STARC
Yokohama, Japan

ABSTRACT – **Test data modification based on test relaxation and X-filling is the preferable 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-Controlling 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. Experimental results on large industrial circuits demonstrate the effectiveness and practicality of the proposed method in reducing IR-drop, without any impact on fault coverage, test data volume, or test circuit size.**

1. INTRODUCTION

As circuit speed goes toward the GHz domain, delay testing has become mandatory for current VLSI circuits. Particularly 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 scan flip-flops. 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].

Fig. 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.

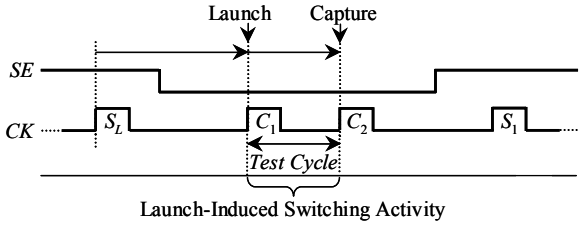

Fig. 1: Launch-off-Capture (LOC) Scheme.

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 often not allowed in functional operations [5, 6]. Therefore, 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 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 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]. Since the post-ATPG *X*-filling only manipulates test data, it has such unique advantages as (1) no impact on test data

978-1-4244-2820-5/08/$25.00 ©2008 IEEE

volume, (2) no impact on test 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 well known that a high percentage of *X*-bits tend 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 not only high WSA but also high IR-drop

In this paper, we propose a novel test relaxation method, called ***Distribution-Controlling 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. 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 nearly equivalent percentages of *X*-bits are identified in each test vector. Further more 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-controlling *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 almost identical even in the environment. We are currently working on it, and will report the results in the immediate future.

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: Section 2 discusses the background, Section 3 presents the results of preliminary experiments, Section 4 elaborates upon the distribution-controlling *X*-identification, Sections 5 describes *X*-filling with the proposed method, and Section 6 shows experimental results. Section 7 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]. Fig. 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].

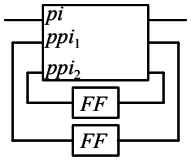

Fig.2: Example Circuit

Table 1: (a) Given Test Set (b) Test Cube with *X*-Bits

(a)

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

(b)

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

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 decreased scalability. 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, as explained later in Section 3.2. This paper attempts to overcome the weaknesses of existing IR-drop reduction methods by proposing ***Distribution-Controlling 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] takes the distribution of *X*-bits into consideration. 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].

Fig. 3 shows the typical distribution of *X*-bits identified by an existing *X*-identification method [17] for an industrial circuit shown in Section 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 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.

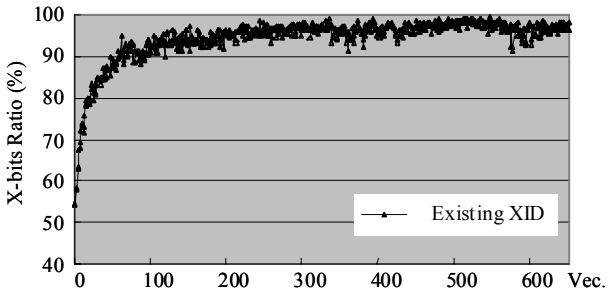

Fig.3: *X*-bit-Distribution for an Industrial Circuit

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.

Fig. 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. Fig. 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.

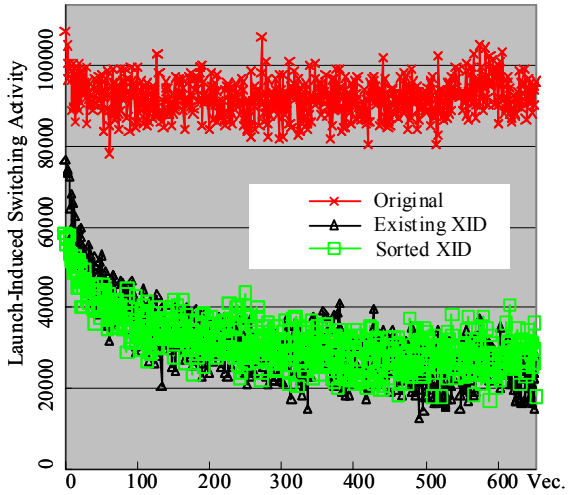

Fig.4: Launch-Induced Switching Activity

4. Distribution Controlling *X*-Identification

4.1 Basic Procedure of *X*-Identification

Here we introduce the basic procedure of *X*-identification. Papers of [13, 15, 22] 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.

Fig. 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 gray area from the transition delay fault *F* in the before-launch pulse shows the internal logic values to be collected, 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.

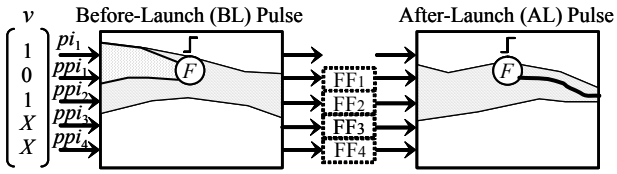

Fig.5: *X*-Identification for Launch-off Capture Scheme

4.2 Controlling Fault Count Targeted

Fig. 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 can 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 vector 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.

In this work, we control the distribution of *X*-bits in order to identify approximately the same percentage of *X*-bits in each test vector. Therefore, the proposed method averages the number of faults targeted by *X*-identification as shown in Table 4.

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_7 f_9
v_3	f_3	f_5 f_8

4.3 Considering Accidental Detection

Step-2 (the internal value collection) and Step-3 (the dedicated implication and justification) described in Section 4.1 are not conducted on all faults. This is because input bits identified to detect targeted faults accidentally detect other faults.

At first, the proposed distribution-controlling *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 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. 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.

Table 5: Fault Selection for Non-Essential Faults

	targeted non-essential		
v_1	(circled) f_a	[squared] f_b	[squared] f_c
v_2	(circled) f_d	[squared] f_e	f_g
v_3	(circled) f_h	[squared] f_i	[squared] f_j

5. IR-drop Reduction with *X*-Filling

In order to effectively reduce IR-drop without power analysis information, we average the percentage of *X*-bits by the proposed distribution controlling *X*-identification method. Since we cannot concentrate the IR-drop reduction in certain test vectors due to lack of power analysis information, we reduce IR-drop as much as possible for each test vector.

In this work, we employ the justification-probability-base fill (JP-fill) method detailed in [16], which is capable of achieving both effectiveness and scalability in a balanced manner. Therefore the method is applicable for large industrial circuits. JP-fill directly reduces launch-induced switching activity.

6. EXPERIMENTAL RESULTS

We implemented the proposed DC-XID method using C programming language and applied it to two industrial circuits. The computer used has a 2.8GHz CPU and 32GB memory. The test set was generated with the ATPG tool "TetraMAX™" from Synopsys. Table 6 detail the information on the circuits and test sets used.

Table 6: Basic Information of Industrial Circuits

Circuit	*Size (# Gates)*	*# Faults*	*# Scan Chains*	*# Scan Cells*	*# Tv*	*Fault Cov.*
*cir*1	50k	92,718	11	1,077	318	90.0 %
*cir*2	600k	1,670,706	64	35,851	652	79.9 %

Table 7 shows results of *X*-identification. In this Table, we compare the results of the existing *X*-identification (*XID*) method [17] to those of the proposed distribution-controlling *X*-identification (*DC-XID*) method. In this experiment, we control the distribution so as to identify an equal percentage of *X*-bits from each test vector. Table 7 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 method achieved smaller values for

both of the two circuits. The results imply that the DC-XID method can indeed average the percentage of *X*-bits. Fig. 6 and Fig. 7 show the distribution of *X*-bits identified by the XID and 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.

Table 7: *X*-Identification

Circuit	*Existing XID*				*Proposed DC-XID*			
	X-bits (%)			*SD(%)*	*X-bits (%)*			*SD(%)*
	Ave.	*Max.*	*Min.*		*Ave.*	*Max.*	*Min.*	
*cir*1	86.7	98.1	25.5	12.7	78.7	89.4	66.0	3.7
*cir*2	94.3	99.3	54.5	5.5	92.8	96.0	87.7	1.3

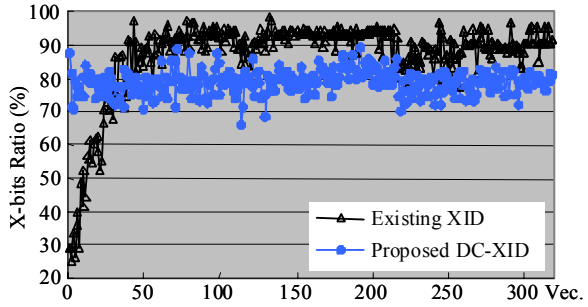

Fig. 6: Distribution of *X*-Bits for "cir1"

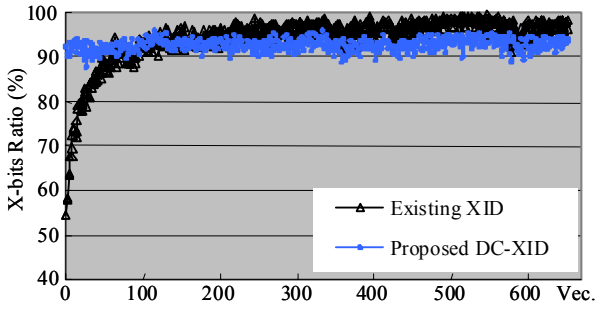

Fig. 7: Distribution of *X*-Bits for "cir2"

The left side of Table 8 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". Fig. 8 and Fig. 9 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. 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. 8 and Fig. 9. For circuit "cir2" it is obvious that there are strong correlations between the distribution of *X*-bits in Fig. 7 and the switching activity in Fig. 9.

The right side of Table 8 shows the reduction ratio of the maximum launch-induced IR-drop. We conducted the layout with "SoC Encounter™" from Cadence. We analyzed IR-drop with "RedHawk™" from Apache. We achieved a roughly 10% improvement in the switching activity and the IR-drop reduction of "cir1". In Fig. 10, we present the launch-induced IR-drop results for "cir1". The maximum IR-drops are indicated by arrows. Note that the results for "cir2" are not shown in Table 8 since we couldn't run the tool for every test vector due to the long CPU time. Therefore, we selected several sections of vectors (1-10, 300-309, 643-652). The results of IR-drop in those vector sections are shown in Fig. 11. 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. 11.

Table 8: Reduction Ratio

Circuit	*Reduction Ratio (%)*			
	Max. Switching Activity		*Max. IR-Drop*	
	XID	*DC-XID*	*XID*	*DC-XID*
*cir*1	21.9	34.6	5.1	15.2
*cir*2	29.0	57.1	N/A	N/A

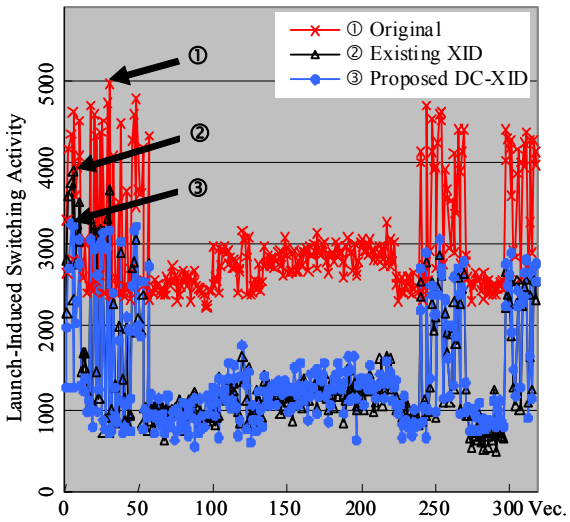

Fig. 8: Launch-Induced Sw. Activity for "cir1"

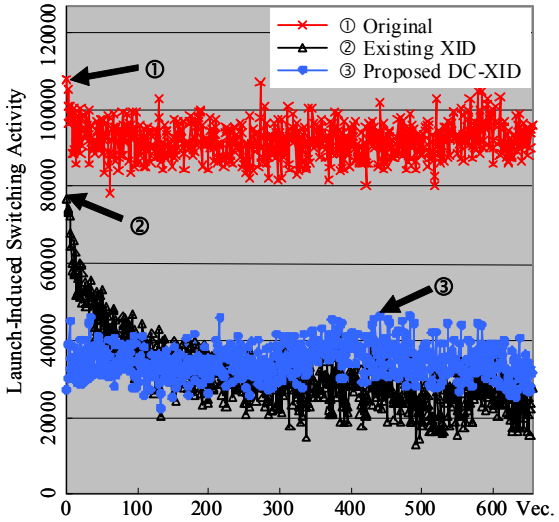

Fig. 9: Launch-Induced Sw. Activity for "cir2"

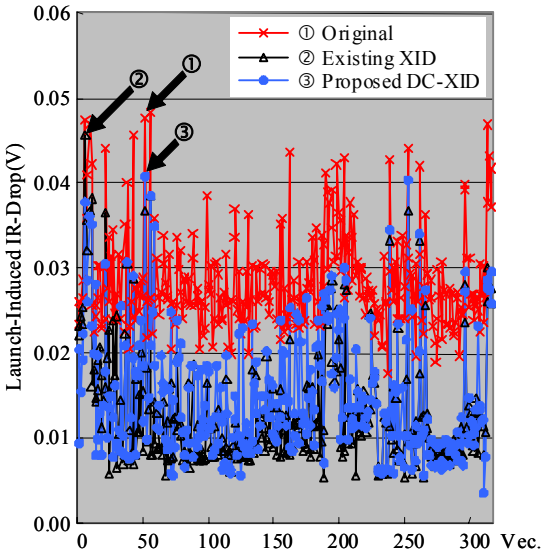

Fig. 10: Launch-Induced IR-Drop for "cir1"

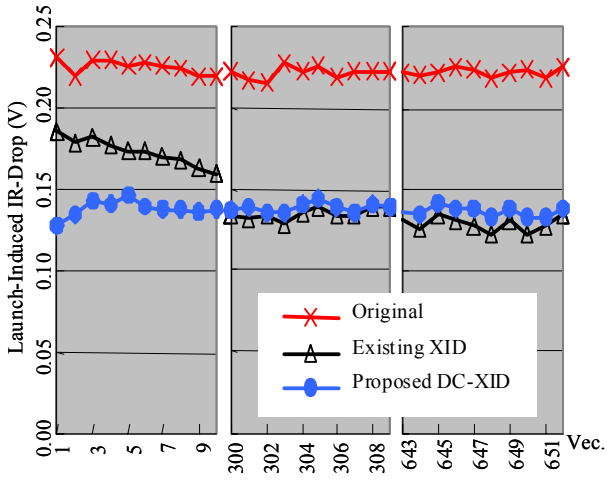

Fig. 11: Launch-Induced IR-Drop for "cir2"

Table 9 provides information on CPU time. 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).

Table 9: CPU Time

Circuit	*CPU Time (s)*			
	Existing		*Proposed*	
	XID	*JP-Fill*	*DC-XID*	*JP-Fill*
*cir*1	25.5	64.91	217.4	129.5
*cir*2	1993.6	23131.2	13085.7	20579.4

7. CONCLUSIONS

This paper proposed a novel *X*-identification method, called ***Distribution-Controlling X-Identification***, which is capable of controlling the distribution of identified *X*-bits. Distribution-controlling *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 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.

REFERENCES

[1] L.-T. Wang, C.-W. Wu, and X. Wen, (Editors), *VLSI Test Principles and Architectures: Design for Testability*, San Francisco: Elsevier, 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. on 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 Design & Test of Computers*, 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. on Computer-Aided Design*, pp. 439-444, 2005.

[9] F. Corno, P. Prinetto, M. Redaudengo, 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, 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, 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. on Computer-Aided Design*, 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. on Computer-Aided Design*, 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, 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. on Computer Design*, pp. 194-199, 2002.