

PAPER

A GA-Based X-Filling for Reducing Launch Switching Activity toward Specific Objectives in At-Speed Scan Testing*

Yuta YAMATO^{†a)}, Xiaoqing WEN^{††}, Kohei MIYASE^{††}, *Members*, Hiroshi FURUKAWA^{††}, *Nonmember*, and Seiji KAJIHARA^{††}, *Member*

SUMMARY Power-aware X-filling is a preferable approach to avoiding IR-drop-induced yield loss in at-speed scan testing. However, the ability of previous X-filling methods to reduce launch switching activity may be unsatisfactory, due to low effect (insufficient and global-only reduction) and/or low scalability (long CPU time). This paper addresses this reduction quality problem with a novel GA (Genetic Algorithm) based X-filling method, called GA-fill. Its goals are (1) to achieve both effectiveness and scalability in a more balanced manner and (2) to make the reduction effect of launch switching activity more concentrated on critical areas that have higher impact on IR-drop-induced yield loss. Evaluation experiments are being conducted on both benchmark and industrial circuits, and the results have demonstrated the usefulness of GA-fill.

key words: X-filling, genetic algorithm, launch switching activity, IR-drop, at-speed scan testing

1. Introduction

With shrinking feature sizes, growing clock frequency, and decreasing power supply voltage, modern integrated circuits are increasingly suffering from the impact of timing related defects, usually of small delays [1]. As a result, delay testing has become critical for reducing the defect level of deep submicron (DSM) circuits.

Since most integrated circuits are based on scan design, *at-speed scan testing* is widely used for delay testing in practice [2]. Figure 1 shows the timing diagram of at-speed scan testing based on the *launch-on-capture (LOC)* scheme [2]. In *shift* mode ($SE = 1$), a test vector is applied by operating scan flip-flops (FFs) as one or more shift registers with multiple shift clock pulses, with S_L being the last shift clock pulse. In *capture* mode ($SE = 0$), two capture pulses are applied: C_1 (*launch capture*) for launching a transition at the start-point of a path and C_2 (*response capture*) for capturing the circuit response to the launched transition at the end-point of the path. Note that the *test cycle* is equal to the functional clock cycle in at-speed scan testing, which is extremely short for a high-speed design.

Switching activity due to the launch capture C_1 for a

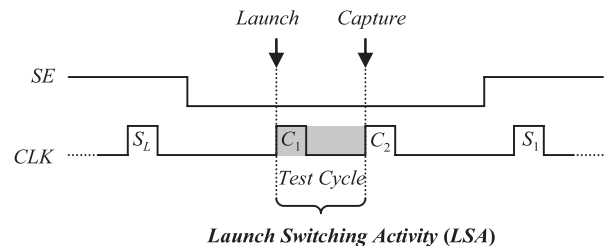


Fig. 1 Launch-on-capture (LOC) scheme for at-speed scan testing.

test vector, referred to as *launch switching activity (LSA)* in this paper, is usually higher (5 times higher as reported in [3]) than that of a functional vector. This can lead to excessive *IR-drop* in the power distribution network, which may increase gate delays along paths in the test cycle to such an extent that timing violations may occur at the response capture C_2 only during at-speed scan testing even for defect-free circuits. As a result, test-induced yield loss may occur, which is costly and unacceptable [4].

The IR-drop-induced yield loss problem is especially severe for high-speed circuits, whose test cycles are short. In addition, low-power circuits are also highly vulnerable to excessive launch switching activity due to narrowed noise margins. Therefore, there is a strong need to reduce launch switching activity during at-speed scan testing.

Previous techniques for reducing launch switching activity can be classified into two general categories: *circuit modification* and *data manipulation*. Circuit-modification-based techniques include *additional circuitry insertion* [5], [6], *scan chain segmentation* [7], *scan re-ordering* [8], and *partial capturing* [9]. On the other hand, data-manipulation-based techniques include *test vector reordering* [10], *test generation* [11], [12], and *X-filling* [13]–[18]. Generally, the data-manipulation approach is preferable, since it does not introduce any circuit overhead or performance degradation, which is inevitable with the circuit-modification approach.

Especially, *X-filling* techniques are now widely used in practice for reducing launch switching activity, since they are compatible with any existing ATPG flow, and have no impact on circuit overhead and circuit timing [19]. X-filling is the process of assigning proper logic values to the X-bits in a partially-specified test cube for reducing launch switching activity in at-speed scan testing. This is realized by minimizing either the Hamming distance between a test vector and its response at flip-flops [13]–[16] or the *weighted*

Manuscript received July 8, 2010.

Manuscript revised November 22, 2010.

[†]The author is with Fukuoka Industry, Science & Technology Foundation, Fukuoka-shi, 810-0001 Japan.

^{††}The authors are with Kyushu Institute of Technology, Iizuka-shi, 820-8502 Japan.

*A previous version of this paper has been published at IEEE 15th Pacific Rim International Symposium on Dependable Computing, Nov. 2009.

a) E-mail: yamato@lab-ist.jp

DOI: 10.1587/transinf.E94.D.833

switching activity (*WSA*) for all nodes in the whole circuit [17], [18].

However, previous *X*-filling methods for reducing launch switching activity in at-speed scan testing suffer from a severe quality problem, in terms of *low effect* (insufficient and global-only reduction) and *low scalability* (long CPU time), as described below:

(1) *Low Effect*

This issue manifests itself in two ways: First, some *X*-filling methods cannot achieve sufficient effect in reducing launch switching activity. This is usually the case when high scalability of *X*-filling is required. Second, most of *X*-filling methods can only reduce launch switching activity globally in a gross manner. That is, the effect of *X*-filling cannot be concentrated on the critical areas that have higher impact on IR-drop-induced yield loss. Obviously, this issue of low effect severely limits the actual usefulness of *X*-filling in practice.

(2) *Low Scalability*

Many *X*-filling methods are computation-intensive, and thus time-consuming. This is usually the case when high effectiveness of *X*-filling is required. Obviously, this issue of low scalability makes it impractical to apply such *X*-filling methods to large industrial circuits.

Therefore, there is a strong need to improve the quality of *X*-filling for reducing launch switching activity. This is important for effectively and efficiently avoiding test-induced yield loss in at-speed scan testing.

This paper addresses this quality issue of *X*-filling with a novel GA (Genetic Algorithm) based *X*-filling method, called *GA-fill*. Its goals are (1) to achieve both effectiveness and scalability in a more balanced manner and (2) to make the reduction effect of launch switching activity more concentrated on critical areas that have higher impact on IR-drop-induced yield loss.

The rest of this paper is organized as follows: Section 2 describes some typical *X*-filling methods to highlight the quality problem in *X*-filling. Section 3 proposes the *GA-fill* as a solution to the problem. Section 4 shows experimental results, and Sect. 5 concludes the paper.

2. Background

X-filling for reducing launch switching activity in at-speed scan testing is conducted by assigning proper logic values to the *X*-bits in a partially-specified test cube, in such a manner that the Hamming distance between a test vector and its response at FFs [13]–[16], or the weighted switching activity for all nodes in the whole circuit [17], [18], are reduced. There are three basic approaches to *X*-filling, namely *justification-based*, *probability-based*, and *hybrid*.

LCP (Low-Capture-Power) X-filling [14] is a typical justification-based technique. PPI-PPO bit-pairs of the form $\langle X, val \rangle$ are processed first (*val* is a logic value), by assigning *val* to the *PPI* (*Pseudo Primary Input*) *X*-bit. After that, PPI-PPO bit-pairs of the form $\langle val, X \rangle$ are processed, one at a time, by justifying *val* to the *PPO* (*Pseudo Primary Out-*

put) *X*-bit. Finally, PPI-PPO bit-pairs of the form $\langle X, X \rangle$ are processed, which may require two passes of justification for each such pair. That is, 1 is first assigned to the PPI *X*-bit and 1 is justified on the PPO *X*-bit. If this fails, 0 is then tried for the PPI and PPO *X*-bits. *LCP X-filling* is highly effective. However, it may take long CPU time, since only one PPI-PPO bit-pair is processed at a time and justification is conducted.

Preferred Fill [15] and *PWT (Probability Weighted Capture Transition) Fill* [17] are typical probability-based techniques. The 0 and 1 probabilities of each node are first calculated by setting 0.50 as the 0 and 1 probability at input *X*-bits and conducting probability propagation. Then, the logic value for a PPI *X*-bit is determined by comparing the 0 and 1 probabilities of its corresponding PPO *X*-bit. In Preferred Fill [12], once 0 and 1 probabilities are calculated, all PPI *X*-bits in a test cube are assigned at one time in order to achieve high scalability. On the other hand, in PWT Fill [14], 0 and 1 probabilities are iteratively calculated after each new *X*-bit assignment so as to achieve high effectiveness. However, the effect of probability-based techniques may be damaged by inaccurate probability results due to approximation in probability calculation. In addition, if the difference between the 0 and 1 probability at a PPO *X*-bit is negligibly small, the effect of logic value determination for its corresponding PPI *X*-bit becomes highly unpredictable.

JP Fill [13] is a typical hybrid technique employing two basic operations, namely *limited justification* and *probability recalculation*. The first operation is conducted for PPI-PPO bit-pairs of the form $\langle \text{logic value}, X \rangle$, but not for any PPI-PPO bit-pair of the form $\langle X, X \rangle$ for which time-consuming, multiple passes of justification are needed. This is to achieve higher scalability. The second operation is conducted for PPI-PPO bit-pairs of the form $\langle X, X \rangle$, but in multiple passes. That is, the logic value for a PPI *X*-bit is determined only if its corresponding PPO *X*-bit has significantly different 0 and 1 probabilities; otherwise, signal probabilities are re-calculated in the next pass. This is to achieve higher effectiveness through improved accuracy in logic value determination.

Recently, *X*-filling methods have been proposed to reduce both shift and capture power [20], [21], and some can reduce both shift-in and shift-out power [20]. It should also be noted that test cubes in *X*-filling can be either obtained by disabling random-fill in ATPG or identified after ATPG from fully-specified test vectors by *test relaxation* or *X-identification (XID)* [22].

However, previous *X*-filling methods for reducing launch switching activity may suffer from a serious issue, i.e. *low quality*. The major reason is that these *X*-filling methods try to directly determine logic values for PPI *X*-bits, which may cause problems in terms of *low effect* and *low scalability*, as described below:

(1) Some *X*-filling methods try to improve the effectiveness by determining logic values for one PPI *X*-bit at a time, while others try to improve the scalability by determining logic values for multiple PPI *X*-bits simultaneously.

This makes it difficult to achieve both effectiveness and scalability in a balanced manner. This problem manifests itself as either insufficient reduction effect or prohibitively long CPU time, leading to the low quality of X -filling.

(2) The direct effect of previous X -filling methods is the reduction of launch switching activity at FFs or in the whole circuit, instead of in any specific area. That is, their reduction effect is global and unfocused. However, it has been shown that high IR-drop around sensitized critical paths is the major reason for test-induced yield loss [23], [24]. This means that the LSA reduction effect achieved by previous X -filling methods may not be what is actually needed from the view-point of avoiding yield loss. Obviously, this problem also leads to the low quality of X -filling.

Note that the low quality problem of previous X -filling methods for reducing launch switching activity is caused by the fact that direct logic value determination is conducted for PPI X -bits according to the relation between logic value or probability of PPI and those of PPO. This motivates us to solve the problem by guiding X -filling in a manner that is not restricted by PPI-PPO bit-pairs.

X -filling can be considered as a kind of combinational optimization problem. The number of solutions for a test cube is 2^n , where n is the number of X -bits in the test cube. It becomes larger and larger in modern circuit designs, which may consist of thousands of FFs. In order to achieve both high effectiveness and good scalability in X -filling, we choose to use the Genetic Algorithm (GA) approach [25], [26], which generally can find optimal or approximate solutions in less time, and propose a novel X -filling method, called **GA-fill**. Its advantage is that a flexible evaluation function, instead of restrictive PPI X -bits, is used to guide X -filling.

Since an evaluation function is free from the restrictions caused by PPI-PPO bit-pairs, this makes it possible to find better solutions that previous X -filling methods can never search and find. In addition, an evaluation function can be made to represent the regional launch switching activity in a critical area that has higher impact on IR-drop-induced yield loss. This makes it possible to achieve better actual effect of launch switching activity reduction through such fine-grained X -filling.

3. GA-Based X -Filling

3.1 Basics

GA is a search technique that is used to find exact or approximate solutions to optimization and search problems. A GA-based search process starts with a set (**population**) of solutions (**individuals**), represented by **chromosomes**. **Fitness** is calculated for each individual in the population. Individuals are taken and used to breed a new population according to their fitness, measured by a certain evaluation function. This process is repeated until a preset termination condition, e.g. the number of total populations or the amount of improvement of the best individual, is satisfied.

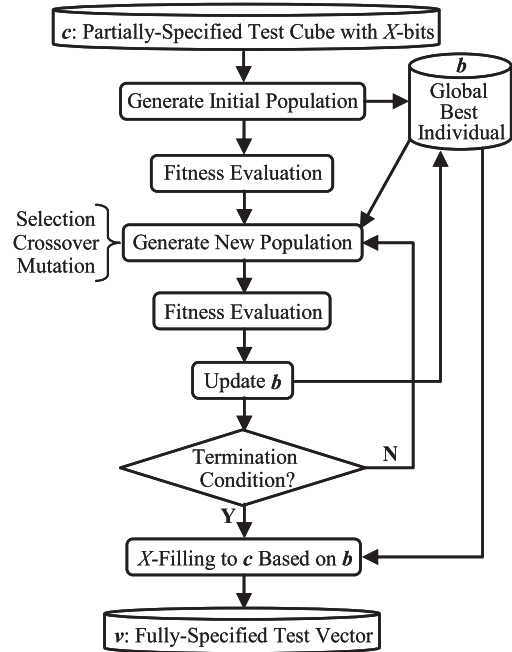


Fig. 2 General flow of GA-fill.

Obviously, the GA approach is ideal to solve the problem of finding optimal logic values assignments in X -filling for reducing launch switching activity in at-speed scan testing. This is because finding a good solution for X -filling needs to search through an extremely large solution space. In addition, this search process is guided by an evaluation function, which is free of restrictions caused by PPI X -bits and can qualitatively express any reduction goal, such as the launch switching activity in a critical area. As a result, the GA approach has the potential of improving the quality of X -filling for reducing launch switching activity in at-speed scan testing.

3.2 General Flow

The general flow of the proposed GA-based X -filling method, called **GA-fill**, is shown in Fig. 2. It consists of 5 basic genetic operations as described in the following.

(1) Initialization

At the beginning of GA-fill, a certain number of random assignments are made for all X -bits in a partially-specified test cube. This results in initial individuals that are used to form the initial population. Figure 3 (a) shows an example, in which the test cube has 9 X -bits, and n initial individuals are created.

(2) Evaluation

The fitness of each individual is evaluated in GA-fill. Here, pattern-parallel 2-valued logic simulation is used to collect information for multiple individuals simultaneously in order to reduce computation time. Then, a proper evaluation function is calculated to quantify the fitness of each individual. Evaluation functions will be discussed in Sect. 3.3.

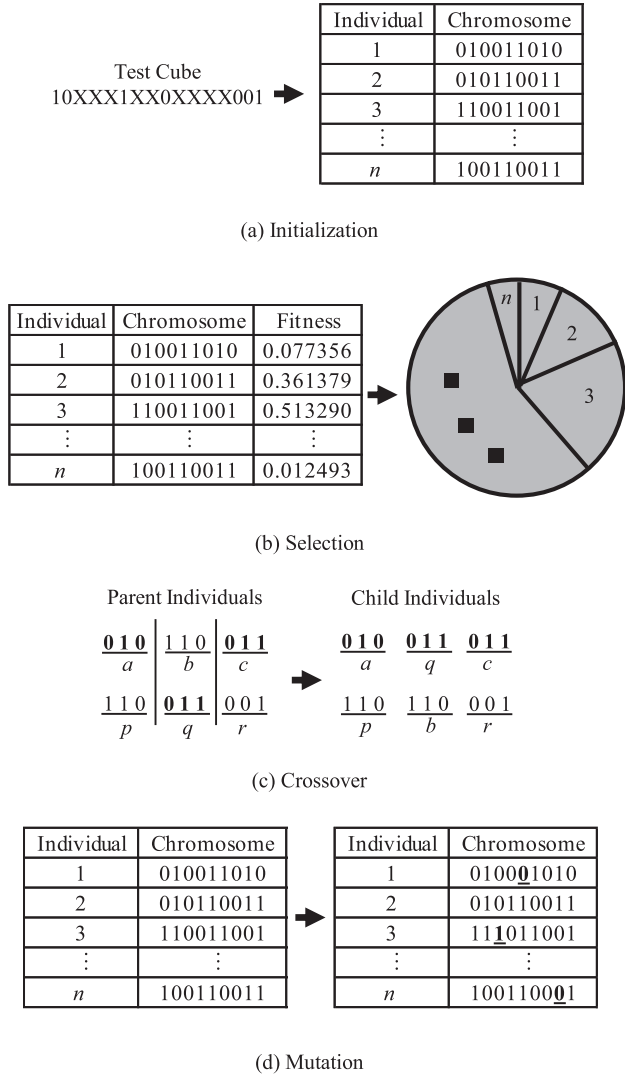


Fig. 3 Examples of basic operations in GA-fill.

(3) **Selection**

A new individual is created from two parent individuals, which are selected based on a certain criterion. Generally, fitter individuals are made more likely to be selected. This can be achieved by using such selection methods as the roulette wheel selection. An example is shown in Fig. 3 (b), where an individual with higher fitness is assigned with a larger area, making it more likely to be selected.

(4) **Crossover**

From two parent individuals, a child individual can be created by crossing-over parts of the chromosomes in the parent individuals. In Two-point crossover, two children can be created by selecting adjacent chromosomes of the parents as shown in Fig. 3 (c). Obviously, a child individual created this way typically shares many of the characteristics of its parent individuals.

(5) **Mutation**

After a child individual is created, some of its chromosomes are flipped with a low probability in order to promote the

diversity of individuals. An example is shown in Fig. 3 (d).

As shown in Fig. 2, GA-fill keeps generating new populations and evaluating new individuals to improve the quality of the global best individual. Note that an individual with the highest fitness throughout the whole population is kept. This process is repeated until the number of generations reaches a preset limit or the fitness of the global best individual becomes greater than a desired value. At this time, all X -bits in the test cube are assigned with the logic values corresponding to the global best individual.

3.3 Fitness Evaluation

3.3.1 Balanced Effectiveness and Scalability

Basically, GA-fill uses the metric similar to the **WSA (Weighted Switching Activity)** metric [8], [15] to estimate the launch switching activity since it has been found to have good correlation with power consumption than un-weighted switching activity while they take almost same computation time.

In order to achieve balanced effectiveness and scalability, the following two evaluation functions are used selectively. Here, the fitness functions of each individual i , $fitness(i)$, is defined as follows:

$$fitness(i) = 1 - \frac{\sum_{k \in FF} (t_k \times c_{wk})}{\sum_{k \in FF} (c_{wk})} \tag{1}$$

if the standard score of the initial test vector is less than a preset threshold; otherwise,

$$fitness(i) = 1 - \frac{\sum_{k \in node} (t_k \times g_{wk})}{\sum_{k \in node} (g_{wk})} \tag{2}$$

where c_{wk} is the number of gates in the fanout cone of FF k and t_k is 1(0) if a transition occurs (does not occur) at FF k in Expression (1), and g_{wk} is the weight of gate k and t_k is 1(0) if a transition occurs (does not occur) at gate k in Expression (2). Here, g_{wk} is set to be the number of its fanouts plus 1. If a gate is fanout-free, its weight is set to 1. Note that the standard scores of each test vector in the initial test set are calculated using Expression (2) before processing GA-fill.

In other words, Expression (1) indicates the ratio of non-switching FFs, while Expression (2) indicates the ratio of non-switching gates. It should be noted that the weight definition is different for each expression. The goal of GA-fill here is not to reduce the hamming distance between PPI and PPO but to reduce the WSA in the whole circuit. Therefore, the number of gates in the fanout cone of a FF is used as the weight, whose definition is different from that of the original WSA, in Expression (1) to contribute in reducing WSA as much as possible. Expression (2) precisely computes the WSA of the circuit nodes with the described weight at the cost of higher runtime compared to Expression (1). The higher precision over Expression (1) increases the effectiveness of the GA-fill algorithm. For larger circuits, the runtime requirements of Expression (2) make its application practically infeasible. To trade-off the precision of

WSA prediction and computation time, both of the expressions are used selectively on a per-pattern basis depending on the standard score threshold value as explained above. The larger the threshold value, the higher the effectiveness and the lower the scalability. However, it is difficult to find an optimal threshold value to achieve both the best quality and scalability without any knowledge or experiences. In this paper, we set 70 as the threshold value in the experiments to carefully reduce the WSA of test vectors whose WSA significantly exceeds the average WSA of a test set. As a result, both high effectiveness and high scalability can be achieved in a balanced manner since the number of test vectors with excessive WSA is small in general.

3.3.2 Critical Area Concentration

By using a proper fitness function, GA-fill can also achieve concentrated reduction effect of launch switching activity in a given critical area. Generally, a *critical area* is a set of nodes (FFs and gates), for which IR-drop due to its launch switching activity has significant impact on timing of the circuit.

Critical area identification [23], [24], [27] can be conducted dynamically for each test vector or statically for the whole circuit, based on gate-level information or even layout information. An example is shown in Fig. 4, in which two activated critical paths go from G_1 to G_{17} and G_4 to G_{19} , respectively. Since launch switching activity near these paths has high impact on timing, it is reasonable to use all nodes on the paths, $G_1, G_6, G_{12}, G_{17}, G_4, G_9, G_{14},$ and G_{19} , as well as nodes near the path, $G_2, G_7, G_{15}, G_5, G_{10},$ and G_{16} , to form a critical area [23].

When the goal is to achieve concentrated reduction effect of launch switching activity in a given critical area (CA), Expression (2) can be modified as follows:

$$fitness(i) = 1 - \frac{\sum_{k \in CA} (t_k \times g_{wk})}{\sum_{k \in CA} (g_{wk})} \quad (3)$$

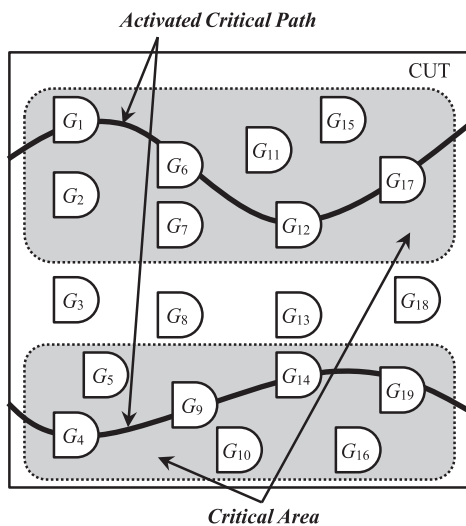


Fig. 4 Example of critical area identification.

Note that the standard score calculation as the preprocessing can also be changed from Expression (2) to Expression (3).

4. Experimental Results

GA-Fill was implemented in C, and experiments were conducted on two ISCAS'89 and six ITC'99 benchmark circuits included in IWLS 2005 benchmarks. The computer used had a 3.0GHz CPU and 24 GB memory. Fully-specified transition delay test vectors were first generated with TetraMaxTM from Synopsys, and partially-specified test cubes were then obtained from them with an internal test relaxation tool [22]. Circuit and test set information is shown in Table 1.

As for the parameters of GA-fill, the population size was set to 64 since the pattern-parallel logic simulation required in fitness evaluation can simulate up to 64 patterns at a time in our 64-bit computing environment. The roulette wheel selection is employed for parent selection to avoid initial convergence. The two-point crossover is employed for crossover to avoid destruction of advantageous characteristics of individuals. These characteristics may result from adjacent bits (chromosomes) that control a larger block of logic. This, for example, is the case for a set of bus lines of a data path. Uniform crossover destroys this relationship of bits. The restoration of such a characteristic by the randomized uniform crossover is very unlikely. From the two children generated during two-point crossover, one child is chosen randomly for the next generation. Mutation probability was set to 0.1%. GA-fill terminated when the number of generations reached 100 or the fitness of the global best individual became greater than the value of 80 of the standard score in the initial test set.

First, we used the fitness function targeted for balanced effectiveness and scalability in GA-fill. The threshold value for choosing an expression was set to 70. For comparison, the justification-based LCP X-filling [14] with high effectiveness, the probability-based preferred fill [15] with high scalability, and the hybrid JP-fill [16] with both high effectiveness and scalability were also implemented. Table 2 shows the reduction ratio of peak WSA for all nodes ("Peak WSA Reduction Ratio (%)") and the CPU time ("CPU (sec.)").

As shown in Table 2, for effectiveness in terms of the

Table 1 Circuit and test set information.

Circuit	#Gates	#FF	#Tv.	Fault Cov. (%)	X(%)
s38417	6724	1564	196	98.3	74.5
s38584	8278	1172	240	82.2	89.9
b17	37117	1317	999	76.7	88.6
b18	92048	3064	2038	69.7	89.1
b19	174157	6130	2763	71.7	91.4
b20	19107	430	1514	94.3	55.4
b21	18718	430	1509	94.9	56.5
b22	28317	645	1913	95	62.3

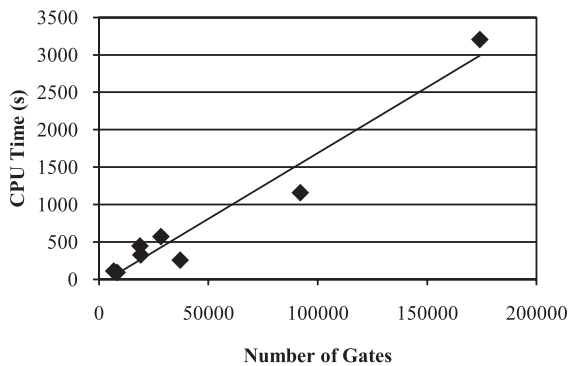
Table 2 Results for balanced effectiveness and scalability.

(a) Effectiveness

Circuit	Peak WSA Reduction Ratio(%)			
	<i>J</i> [14]	<i>P</i> [15]	<i>JP</i> [16]	GA
<i>s38417</i>	9.4	4.9	9.2	10.3
<i>s38584</i>	4.8	4.4	4.8	5.2
<i>b17</i>	16.4	12.2	16.4	22.5
<i>b18</i>	19.5	16.8	19.5	20.0
<i>b19</i>	18.5	16.4	18.5	20.9
<i>b20</i>	3.5	3.2	3.5	5.4
<i>b21</i>	4.8	2.4	4.8	5.2
<i>b22</i>	8.3	7.0	8.7	10.2
<i>Ave.</i>	10.7	8.4	10.7	12.4

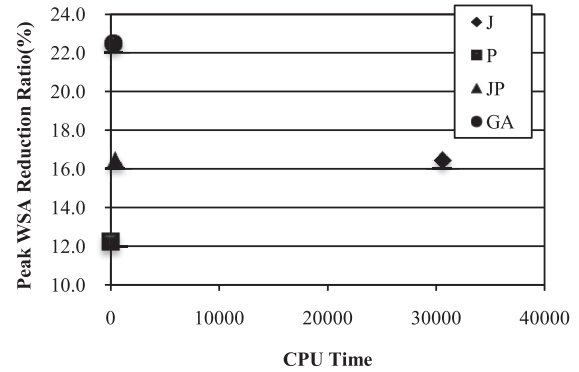
(b) Scalability

Circuit	CPU(sec.)			
	<i>J</i> [14]	<i>P</i> [15]	<i>JP</i> [16]	GA
<i>s38417</i>	3751.4	0.04	256.6	111.6
<i>s38584</i>	3281.6	0.05	110.8	93.9
<i>b17</i>	30593	0.2	418.5	257.7
<i>b18</i>	339179	3.1	5133.7	1159.3
<i>b19</i>	1396032	9.3	22048.5	3207.2
<i>b20</i>	7415.3	0.1	408.8	327.2
<i>b21</i>	7597.5	0.1	395.9	447
<i>b22</i>	23740.9	0.1	895.9	572.5

**Fig. 5** CPU time vs. circuit size for GA-fill.

reduction ratio of peak WSA for all nodes in a circuit, GA-fill outperformed other X-filling methods. In addition, for scalability in terms of CPU time, GA-fill was much closer to that of the best-performing probability-based preferred fill [15] than the justification-based LCP X-filling [14] and hybrid JP-fill [16]. The CPU time required by GA-fill is approximately proportional to the number of gates in a circuit as shown in Fig. 5.

The above results clearly demonstrate that GA-fill can indeed achieve both effectiveness and scalability in a much more balanced manner than previous X-filling methods for reducing launch switching activity in at-speed scan testing. This can also be seen in Fig. 6 from the relation between scalability (measured by CPU time) and effectiveness (measured by reduction ratio of peak WSA for all nodes) for each X-filling method on b17.

**Fig. 6** Effectiveness vs. scalability for b17.**Table 3** Results for critical area concentration.

Circuit	#Gates in Critical Area	Peak WSA Reduction Ratio for Critical Area (%)		CPU(sec.)	
		GA-1	GA-2	GA-1	GA-2
<i>s38417</i>	672	10.6	13.4	111.6	55.1
<i>s38584</i>	828	9.3	10.2	93.9	41.5
<i>b17</i>	3712	25.5	25.1	257.7	133.8
<i>b18</i>	9205	17.5	21.8	1159.3	741.7
<i>b19</i>	17416	17.3	19.6	3207.2	2157.2
<i>b20</i>	1911	3.9	6.7	327.2	167.8
<i>b21</i>	1872	8.2	7.4	447	159.6
<i>b22</i>	2832	6.1	6.1	572.5	352.3
<i>Ave.</i>		12.3	13.8		

Next, we used the fitness function targeted for concentrated reduction effect of launch switching activity in a given critical area. 10% of nodes in each circuit were randomly selected to form a critical area. For comparison, GA-fill using the fitness function targeted for balanced effectiveness and scalability was also executed. The same test sets in Table 2 were used. The experimental results are shown in Table 3.

As shown in Table 3, for most of the circuits, GA-fill using the fitness function targeted for concentrated reduction effect of launch switching activity (GA-2: GA-Concentrated) did outperform GA-fill (GA-1: GA-Balanced) using the fitness function targeted for balanced effectiveness and scalability for all nodes in the circuit. However, for b17 and b21, GA-1 achieved better results than GA-2. Test vectors that have the highest WSA in the critical area after applying GA-2 for b17 and b21 originally had low WSA in the critical area. Therefore, Expression (1), targeting reduction of WSA in the FFs, was selected as the evaluation function for those test vectors in GA-2. As a result, WSA in the critical area was increased instead of the reduction of WSA in the FFs.

In this work, nodes in critical areas were selected randomly, and only gate-level information was used in the experiments. We are planning further evaluation experiments using large industrial circuits and physical layout information for critical area selection [27].

5. Conclusions

This paper proposed a novel *X*-filling method, *GA-fill*, for reducing launch switching activity in at-speed scan testing. *GA-fill* guides *X*-filling with flexible fitness functions, so as to achieve effectiveness and scalability in a more balanced manner, and to make the effect of launch switching activity reduction more concentrated on critical areas that have higher impact on IR-drop-induced yield loss. Experimental results clearly demonstrated the usefulness of *GA-fill*.

There are various other methods in each operation in *GA* such as the tournament selection, uniform crossover, etc. In addition, parameters in each operation such as the threshold value in evaluation, the mutation probability, etc., may affect the result of *GA-fill*. These indicate that *GA-fill* can be improved by choosing proper operation methods and/or parameters.

We are currently conducting additional experiments on large industrial circuits to evaluate whether the result obtained by *GA-fill* is sufficient to avoid test induced yield loss using IR-Drop analysis tool, and the results will be used to fine-tune *GA-fill* by finding a better combination of operation methods and parameters.

Acknowledgements

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

References

- [1] Y. Sato, S. Hamada, T. Maeda, A. Takatori, Y. Nozuyama, and S. Kajihara, "Invisible delay quality - SDQM model lights up what could not be seen," Proc. ITC, Paper 47.1, 2005.
- [2] L.-T. Wang, C.-W. Wu, and X. Wen, ed., VLSI Test Principles and Architectures: Design for Testability, first ed., Morgan Kaufmann, San Francisco, 2006.
- [3] S. Sde-Paz and E. Salomon, "Frequency and power correlation between at-speed scan and functional tests," Proc. ITC, Paper 13.3, 2008.
- [4] J. Saxena, K. Butler, V. Jayaram, and S. Hundu, "A case study of IR-drop in structured at-speed testing," Proc. ITC, pp.1098–1104, 2003.
- [5] 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.
- [6] S. Gerstendoerfer and H. Wunderlich, "Minimized power consumption for scan-based BIST," Proc. ITC, pp.77–84, 1999.
- [7] K. Lee, T. Huang, and J. Chen, "Peak-power reduction for multiple-scan circuits during test application," Proc. ATS, pp.453–458, 2000.
- [8] P. Girard, "Survey of low-power testing of VLSI circuits," IEEE Des. Test Comput., vol.19, no.3, pp.82–92, Feb. 2002.
- [9] 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.
- [10] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, "Techniques for minimizing power dissipation in scan and combinational circuits during test application," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.17, no.12, pp.1325–1333, 1998.
- [11] F. Corno, P. Prinetto, M. Redaengio, and M. Reorda, "A test pattern generation methodology for low power consumption," Proc. VTS, pp.35–40, 1998.
- [12] N. Ahmed, M. Tehranipoor, and V. Jayaram, "Supply voltage noise aware ATPG for transition delay faults," Proc. VTS, pp.179–184, 2007.
- [13] W. Li, S.M. Reddy, and I. Pomeranz, "On reducing peak current and power during test," Proc. ISVLSI, pp.156–161, 2005.
- [14] X. Wen, Y. Yamashita, K. Kajihara, L.-T. Wang, K.K. Saluja, and K. Kinoshita, "Low-capture-power test generation for scan testing," Proc. VTS, pp.265–270, 2005.
- [15] 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. ITC, Paper 32.2, 2006.
- [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. ITC, Paper 25.1, 2007.
- [17] X. Wen, K. Miyase, T. Suzuki, Y. Yamato, S. Kajihara, L.-T. Wang, and K.K. Saluja, "A highly-guided *X*-filling method for effective low-capture-power scan test generation," Proc. ICCD, pp.251–258, 2006.
- [18] J.-L. Yang and Q. Xu, "State-sensitive *X*-filling scheme for scan capture power reduction," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.27, no.7, pp.1338–1343, July 2008.
- [19] P. Girard, X. Wen, and N.A. Touba, Low-Power Testing (Chapter 7) in Advanced SOC Test Architectures - Towards Nanometer Designs, first ed., Morgan Kaufmann, San Francisco, 2007.
- [20] J. Li, Q. Xu, Y. Hu, and X. Li, "iFill: An impact-oriented *X*-filling method for shift- and capture-power reduction in at-speed scan-based testing," Proc. DATE, pp.1184–1189, 2008.
- [21] S. Remersaro, X. Lin, S.M. Reddy, I. Pomeranz, and J. Rajski, "Low shift and capture power scan tests," Proc. VLSI Design, pp.793–798, 2007.
- [22] K. Miyase and S. Kajihara, "XID: Don't care identification of test patterns for combinational circuits," IEEE Trans. Comput.-Aided Des., vol.23, no.2, pp.321–326, Feb. 2004.
- [23] X. Wen, K. Miyase, T. Suzuki, S. Kajihara, Y. Ohsumi, and K.K. Saluja, "Critical-path-aware *X*-filling for effective IR-drop reduction in at-speed scan testing," Proc. DAC, pp.527–532, 2007.
- [24] A.K. Kokrady and C.P. Ravikumar, "Static verification of test vectors for IR drop failure," Proc. ICCAD, pp.760–764, 2003.
- [25] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.
- [26] J. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan, 1975, and MIT Press, 1992.
- [27] N. Ahmed, M. Tehranipoor, and V. Jayaram, "Transition delay fault test pattern generation considering supply voltage noise in a SOC design," Proc. DAC, pp.533–538, 2007.
- [28] M.-F. Wu, J.-L. Huang, X. Wen, and K. Miyase, "Reducing power supply noise in linear-decompressor-based test data compression environment for at-speed scan testing," Proc. ITC, Paper 13.1, 2008.
- [29] P. Girard, N. Nicolici, and X. Wen, ed., Power-Aware Testing and Test Strategies for Low Power Devices, Springer, New York, 2009.



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.



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.



Kohei Miyase received Ph.D. degree 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,

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.



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.