Effective Launch-to-Capture Power Reduction for LOS Scheme with Adjacent-Probability-Based *X*-Filling

K. Miyase [1], Y. Uchinodan [1], K. Enokimoto [1], Y. Yamato [2], X. Wen [1], S. Kajihara [1]
F. Wu [3], L. Dilillo [3], A. Bosio [3], P. Girard [3], A. Virazel[3]

[1] Kyushu Institute of Technology, Iizuka, Japan
[2] Nara Institute of Science and Technology, Japan
[3] LIRMM, Montpellier, France

Abstract–It has become necessary to reduce power during LSI testing. Particularly, during at-speed testing, excessive power consumed during the ***Launch-To-Capture*** (***LTC***) cycle causes serious issues that may lead to the overkill of defect-free logic ICs. Many successful test generation approaches to reduce IR-drop and/or power supply noise during LTC for the launch-off capture (LOC) scheme have previously been proposed, and several of *X*-filling techniques have proven especially effective. With *X*-filling in the launch-off shift (LOS) scheme, however, adjacent-fill (which was originally proposed for shift-in power reduction) is used frequently. In this work, we propose a novel *X*-filling technique for the LOS scheme, called ***Adjacent-Probability-based X-Filling*** (***AP-fill***), which can reduce more LTC power than adjacent-fill. We incorporate AP-fill into a post-ATPG test modification flow consisting of test relaxation and *X*-filling in order to avoid the fault coverage loss and the test vector count inflation. Experimental results for larger ITC'99 circuits show that the proposed AP-fill technique can achieve a higher power reduction ratio than 0-fill, 1-fill, and adjacent-fill.

Keywords *– test generation, test power, at-speed scan testing, launch-off shift, power supply noise.*

I. INTRODUCTION

Since the fabrication of LSIs, at-speed scan testing has become necessary in order to test timing-related defects. There are two major clocking schemes for at-speed scan testing, which are the launch-off capture (LOC) scheme and the launch-off shift (LOS) scheme. Figure 1 shows the waveforms of Clock and Scan Enable (SE) signals for LOC and LOS. As shown in Figure 1, there are two pulses for launch and capture after last shift operation. Since LOS must disable SE between launch and capture pulses, it is not as easy to implement as LOC. However, LOS can achieve higher fault coverage and a smaller test vector count. Therefore, LOS has advantages in terms of the test quality and test cost [1].

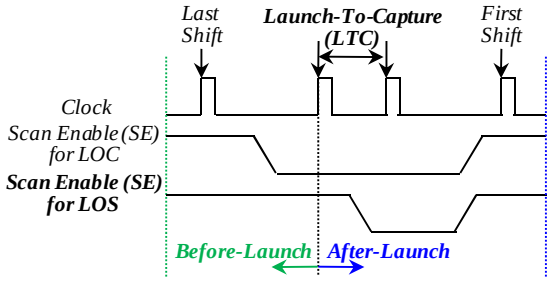

Figure 1. Waveforms for LOC and LOS.

Regardless of at-speed testing schemes, the ***Launch-To-Capture*** (***LTC***) power during at-speed testing, which is represented by the switching activity (transitions) caused by the launch pulse, may cause serious problems. Excessive IR-drop due to many transitions causes excessive delay increase, resulting in possible timing failures. Even if no problems arise in functional mode, timing failures occur in at-speed testing. Therefore, excessive LTC power may finally results in overkill of defect-free LSIs [2,3].

With regards to LTC power reduction, methods based on DFT such as partial capture [2], low LTC power ATPG [4], and *X*-filling [5-10] have been proposed. DFT methods such as partial capture [2] can be used to allow only part of a circuit to capture. Transitions can be reduced by directly generating proper logic values in ATPG [4], assigning logic values to *don't-care* bits (*X*-bits) by in-ATPG or post-ATPG *X*-filling [5-8].

There are three major *X*-filling techniques for LTC power reduction in LOC. One of them is justification-based *X*-Filling (J-Fill) [5], which achieves significant LTC power reduction but may result in long computation time. Another is probability-based *X*-Filling (Preferred-Fill) [6,7], which achieves high scalability but whose effectiveness depends on a probability calculation. The last technique is justification-probability-based *X*-filling (JP-Fill) [8], which balances effectiveness and scalability.

Adjacent-fill [9,10] is often used to reduce LTC power in LOS even though it was originally proposed for shift-in power reduction. It assigns the same logic value to *X*-bits as the logic value on the adjacent flip-flop in the scan path. Since there are a number of scan shift cycles in scan shift mode, adjacent-fill dramatically reduces shift-in power. However, while adjacent-fill can directly reduce LTC power, it does not consider internal gates at all, meaning that it does not always make the optimum decision for logic value assignments to reduce LTC power.

In this paper, we propose a novel technique, ***Adjacent-Probability-based X-filling*** (***AP-fill***), to reduce LTC power in the LOS scheme. AP-fill considers the transitions of internal gates in a circuit by using probability calculations. It is obvious that searching for the optimum assignment for *X*-bits is too expensive; therefore, AP-fill first examines the impact derived by 0 to 1 and 1 to 0 transitions at each flip-flop by using probability calculations as preprocessing. And then, appropriate logic values are assigned to *X*-bits by referring to the results of the calculations.

In our experiments, we incorporate the proposed AP-fill into a post-ATPG test modification flow [5,8,11]. The post-ATPG test flow consists of test relaxation and *X*-filling. Test relaxation [12,13] identifies *X*-bits in a given test

vectors without increasing test vector count or degrading fault coverage, even after *X*-filling. Therefore, this flow is a practical solution to test generation problems. We extend the existing test relaxation to the one in the LOS scheme, and then we compare the results of AP-fill with those for previous *X*-filling techniques. Experimental results show that AP-fill achieves a higher reduction ratio of LTC power for LOS than previous methods.

The rest of this paper is organized as follows; the background of each proposed technique is presented in Section II; test relaxation for the LOS scheme is described in Section III; adjacent-probability-based *X*-filling (AP-fill) is described in Section IV; experimental results for larger ITC'99 benchmark circuits are shown in Section V; and our conclusions are summarized in Section VI.

II. Background

A. Circuit Modeling for X-Filling

A circuit modeling for a target test scheme is generally used for *X*-filling as well as test generation. Figure 2 shows the circuit modeling for stuck-at fault test generation for a full-scan sequential circuit. Since all flip-flops are designed as a scan structure, each flip-flop is controllable and observable. Thus, a full-scan sequential circuit can be treated as a combinational circuit in test generation and *X*-filling as shown in Figure 2.

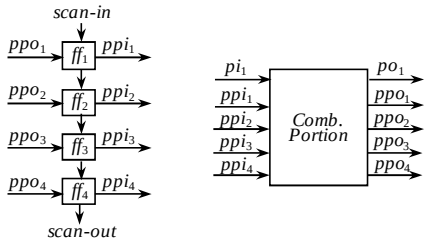

Figure 2. Circuit Modeling for Stuck-at Fault Test.

Two-pattern testing (such as transition delay testing and path delay testing requires initialization vectors and launch vectors. In the LOC scheme, a launch vector is obtained from the output response of a circuit before the launch pulse. Figure 3 shows a simplified circuit modeling for the LOC scheme based on the circuit shown in Figure 2. The combinational portion is duplicated before and after launch pulse. Each pseudo primary output ($b\text{-}ppo_k$) of the before-launch circuit and the corresponding pseudo primary input ($a\text{-}ppi_k$) of the after-launch circuit are connected. Note that the primary input *pi* and primary output *po* have been omitted for the simplicity of the figure.

On the other hand, a launch vector for the LOS scheme is obtained through the shift operation along the scan path. After a launch pulse, the logic value on a *ppi* shifts to the next *ppi*. A circuit modeling for LOS is shown in Figure 4. In this figure, $b\text{-}ppi_k$ is connected to $a\text{-}ppi_{k+1}$ (except $b\text{-}ppi_4$ and $a\text{-}ppi_1$). Note that *pi* and *po* are omitted for simplicity of the figure. With the circuit modeling, we can easily apply *X*-filling to the LOC and LOS schemes.

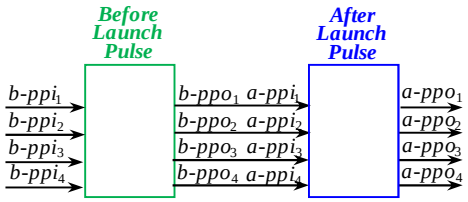

Figure 3. Circuit Modeling for LOC.

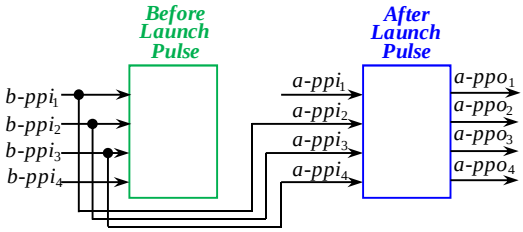

Figure 4. Circuit Modeling for LOS.

B. X-Filling for LOS

Recently, many *X*-filling techniques have been proposed for the LOC scheme [5-8]. Justification-Based *X*-Filling (J-Fill) [5], Preferred-Fill [6,7], and Justification-Probability-Based *X*-Filling (JP-Fill) [8] equalize the logic values on flip-flops before and after launch pulse. In Figure 3, logic values on $b\text{-}ppi_k$ and $b\text{-}ppo_k$ are equalized in order to reduce transitions at as many flip-flops as possible. However, since the circuit modeling for LOS differs from the one for LOC, the major *X*-filling techniques for LOC [5-8] cannot be applied to the LOS scheme.

In the LOS scheme, a transition at a flip-flop before and after launch pulse is caused by the shift operation. In Figure 5, we extract the pseudo primary inputs before launch pulse (*b-ppi*) and after launch pulse (*a-ppi*) from the circuit modeling in Figure 4. The left side of Figure 5 shows that the transition occurs between $b\text{-}ppi_3$ and $a\text{-}ppi_3$. Since logic value on $a\text{-}ppi_3$ is shifted from $b\text{-}ppi_2$, the transition between $b\text{-}ppi_2$ and $b\text{-}ppi_3$ is the cause by the transition at ff_3 (as shown on the right side of Figure 5).

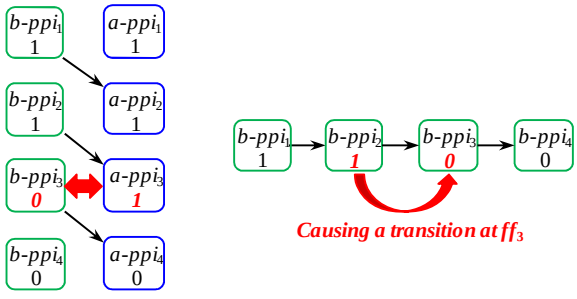

Figure 5. Transition at Flip-Flops for LOS.

Since power reduction during the shift cycle is strongly related to reduction of LTC power for LOS, adjacent-fill [9,10] is often used for LTC power reduction. Originally proposed for shift-in power reduction, adjacent-fill assigns the same logic value to *X*-bits as the logic value for the

adjacent flip-flop on the scan path. For example, assume that there is a test cube, *c*, where <*b-ppi*$_1$, *b-ppi*$_2$, *b-ppi*$_3$, *b-ppi*$_4$> = <0, *X*, *X*, 0> (as shown in Figure 6). Adjacent-fill assigns logic value 0 to all *X*s.

However, if <*b-ppi*$_1$, *b-ppi*$_2$, *b-ppi*$_3$, *b-ppi*$_4$> = <1, *X*, *X*, 0>, conventional adjacent-fill cannot determine the optimum assignment in terms of LTC power reduction for the LOS scheme. When it assigns 0 to *X*s, the transition occurs at flip-flop *ff*$_2$. Similarly, when it assigns 1 to *X*s, the transition occurs at flip-flop *ff*$_4$. Conventional adjacent-fill cannot measure which transition (at *ff*$_2$ or *ff*$_4$) has a greater impact on LTC power reduction. The proposed method determines the logic value that achieves a significant reduction in LTC power through probability calculation.

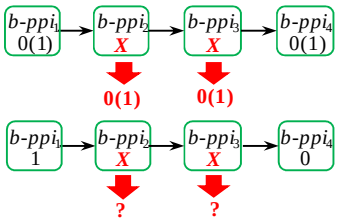

Figure 6. Example of Adjacent-Fill.

C. Post-ATPG Test Modification and Its Benefits

Every *X*-filling technique requires test cubes. There are two major methods of obtaining test cubes. One of them leaves *X*-bits after performing test generation without random-fill [8,9,10]. As a result, the test vector count for this method is higher than that of test generation with random-fill. The other method, called test relaxation, finds *X*-bits in test vectors even if the test vector set is compacted by test compaction techniques [12, 13].

Figure 7 shows general flows for obtaining test cubes and *X*-filling. We call the flow consisting of ATPG, test relaxation, and *X*-filling, as a "post-ATPG test modification". Since test relaxation does not degrade fault coverage or increase the test vector count, the post-ATPG test modification flow produces modified test vectors that can achieve a small test vector count as well as a desired property, such as low power test vectors [5,8,11]. Sections III and IV present test relaxation customized for LOS and the proposed method of *X*-filling, respectively.

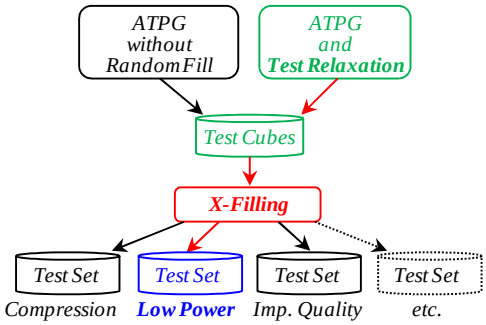

Figure 7. Post-ATPG Test Modification Flow.

III. Test Relaxation For LOS

A. Test Relaxation for Stuck-at Fault Test

Test relaxation identifies *X*-bits in given test vectors without degrading fault coverage of the original test vectors. Previous methods [12, 13] efficiently identify *X*-bits with fault simulation, dedicated implication, and dedicated justification. In this section, we introduce test relaxation for stuck-at fault test vectors [12], and then we extend it for application to LOS.

The method proposed in [12] first assigns every detectable fault to one of the test vectors detecting the fault. Each detectable fault must be detected by at least one test vector in order to maintain the fault coverage of the given test vectors. This method specifies a test vector that detects each fault with fault simulation. Next, the method collects internal logic values to activate and propagate each fault with fault simulation, and then performs the dedicated implication and justification to obtain the necessary input values for the fault detection.

Figure 8 illustrates the input values required for the activation and propagation of fault *F*. Note that *pi* and *po* are omitted for simplicity. When a test vector detects fault *F*, the internal logic values for fault activation and propagation are obtained with fault simulation. Then, the internal values are justified with the dedicated implication and justification. Figure 8 shows that the logic values on *ppi*$_1$, *ppi*$_2$, and *ppi*$_3$ are necessary for the activation and propagation of the fault. Finally, the unnecessary logic value on *ppi*$_4$ is identified as an *X*-bit.

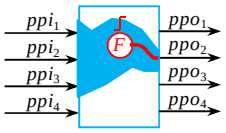

Figure 8. Test Relaxation for Stuck-at Fault Test.

B. Extension to Test Relaxation for LOS

Test relaxation for stuck-at fault testing can easily be extended to LOS test scheme using the circuit modeling presented in Figure 4. As a condition of LOS fault activation, an initializing value must be set at the fault site before launch pulse. As a condition of LOS fault propagation, the faulty value is propagated to at least one pseudo primary output after launch pulse. Figure 9 shows that the pseudo primary inputs for the initialization (*b-ppi*$_1$) and the fault propagation (*a-ppi*$_1$, *a-ppi*$_2$, and *a-ppi*$_3$). As a result, logic values on *b-ppi*$_3$ and *b-ppi*$_4$ are identified as *X*-bits, since the logic values on *a-ppi*$_2$ and *a-ppi*$_3$ are shifted from *b-ppi*$_1$ and *b-ppi*$_2$.

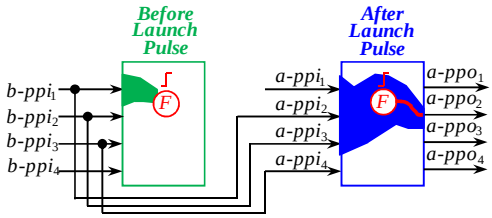

Figure 9. Test Relaxation for LOS.

The basic procedure of test relaxation for LOS is summarized below:

Basic Procedure for Test Relaxation on LOS

Step-1 (*Circuit Expansion for LOS*)
Expand the circuit according to the LOS circuit modeling. The combinational portion of the circuit is duplicated, and each *b-ppi* is connected to the corresponding *a-ppi* in the scan path.

Step-2 (*Target Fault Assignment*)
Conduct fault dropping fault simulation and assign faults to each test vector so that each detectable fault is assigned to at least one test vector.

Step-3 (*Collect Input Values to Activate Faults*)
Collect internal logic values in the before-launch circuit to activate the target faults assigned to each test vector. Then, perform the dedicated implication and justification to justify the internal logic values collected.

Step-4 (*Collect Input Values to Propagate Faults*)
Collect internal logic values in the after-launch circuit to propagate the target faults assigned to each test vector. Then, perform the dedicated implication and justification to justify the internal logic values collected.

Step-5 (*Identify X-Bits*)
For each test vector after Step-4, identify the unnecessary input values to detect the target faults as *X*-bits.

IV. Adjacent-Probability-Based *X*-Filling

This section introduces a novel *X*-filling technique, called ***Adjacent-Probability-Based X-Filling* (*AP-fill*)** for reducing LTC power in LOS. By referring to the transition probability of internal gates in a circuit, AP-fill is capable of assigning more appropriate logic values than adjacent-fill. AP-fill calculates the internal gate transition probability several times as preprocessing, and then assigns appropriate logic values to *X*-bits according to that probability.

A. Transition Probability Calculation for a Test Cube

In order to search for appropriate assignments for *X*-bits in LTC power reduction, we use the transition probability of internal gates in a circuit with test cubes [14]. This transition probability calculation is summarized in the rest of this sub-section.

First, probabilities $prob^0$ and $prob^1$ are assigned to each pseudo primary input in the transition probability calculation. Note that $prob^0$ ($prob^1$) is the probability that a signal line takes 0 (1). The assignment of a probability to each bit (i.e. pseudo primary input) in a test cube is shown in Table I. We assign 0.50 to $prob^0$ and $prob^1$ of *X*s.

TABLE I. Assignment Of The Probability For PPI

PPI logic value	$prob^0$	$prob^1$
0	1.00	0
1	0	1.00
X	0.50	0.50

Next, $prob^0$ and $prob^1$ of each internal gate in the circuit are calculated. This is similar to a well-known controllability calculation in ATPG [15]. Figure 10 shows an example on an AND gate. Based on this calculation, $prob^0$ and $prob^1$ for each signal line in before-launch circuit and after-launch circuit are obtained.

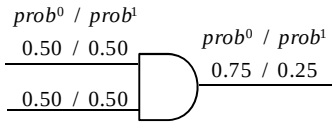

Figure 10. Probability Calculation for AND Gate.

Using probabilities $prob^0$ and $prob^1$ for each signal line, the ***Probabilistic Weighted Transition* (*PWT*)** [14] for a test cube is calculated as follows:

$$PWT(c) = \sum_{i=1}^{i=n} (w_i * p_i)$$

where n is the total number of gates, w_i is the number of fan-out branches for gate i. Transition probability p_i for gate i is expressed as follows:

$$p_i = (\ b\text{-}prob^0(i) * a\text{-}prob^1(i)\ ) + (\ b\text{-}prob^1(i) * a\text{-}prob^0(i)\ )$$

The proposed method uses PWT to find appropriate logic value assignments for test cubes. In the next segment, we present how to use PWT in AP-fill.

B. Transition Impact Calculation

In order to examine the impact that transitions at each flip-flop have on LTC power reduction, AP-fill first examines the transition probability when a 0 to 1 (rising) transition or a 1 to 0 (falling) transition occurs at each flip-flop. In a test cube where all values are *X*s, 1 and 0 are assigned to $b\text{-}ppi_{k-1}$ and $b\text{-}ppi_k$, respectively. The ***Rising Transition Impact* (RTI_k)** for flip-flop ff_k is obtained using PWT for the test cube, as shown in Figure 11. Note that the ***Falling Transition Impact* (FTI_k)** is obtained in a similar manner. The RTI and FTI of each flip-flop are calculated as preprocessing of AP-fill and then referred to in the logic value assignment to *X*-bits.

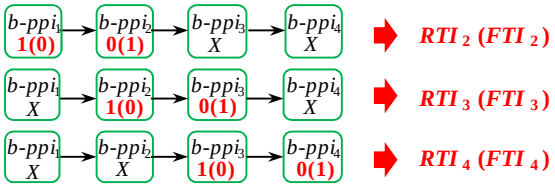

Figure 11. Rising Transition Impact and Falling Transition Impact.

The impact of the 0 to 1 transition (i.e. rising transition impact) for flip-flop ff_j, RTI_j, is expressed by a ***PWT*** for a test cube. This is expressed:

$$RTI_j = PWT(c_{RTI_j})$$

where test cube c_{RTIj} has logic values on *b-ppi*$_1$ through *b-ppi*$_m$, as expressed below.

$$c_{RTIj} = \{\ b\text{-}ppi_1, b\text{-}ppi_2, \ldots, b\text{-}ppi_m\ \}$$

Here, *m* is the number of pseudo primary inputs (flip-flops). Test cube c_{RTIj} consists of the following logic values:

$$b\text{-}ppi_k = \begin{cases} 0 & \text{if } (k == j) \\ 1 & \text{if } (k == j\text{-}1) \\ x & \text{otherwise} \end{cases}$$

Using the RTI and FTI, we can ascertain the transition impact of each flip-flop. These RTIs and FTIs are used in the logic value assignment described in the next sub-section.

C. Logic Value Assignments with Transition Impact

After calculating RTI and FTI, the values obtained for each flip-flop are stored. When assigning logic values to *X*-bits, AP-fill refers to these RTIs and FTIs. When the preceding logic value of a single or sequence of *X*-bits is 0(1) and the next logic value of the *X*-bits is 0(1), AP-fill assigns 0(1) to those *X*-bits, similar to adjacent-fill. On the other hand, when the preceding logic value of the *X*-bits is 0(1) and the next logic value of the *X*-bits is 1(0), AP-fill utilizes RTIs and FTIs to assign appropriate logic values.

The larger the RTI$_j$, the more LTC power is consumed by *ff*$_j$. Therefore, when the logic value on *b-ppi*$_i$ is 1(0), the values on *b-ppi*$_{i+1}$ through *b-ppi*$_{k-1}$ (i<k-1) are *X*s, and the value on *b-ppi*$_k$ is 0(1), AP-fill compares RTI$_{i+1}$ and RTI$_k$. AP-fill then determines the logic values to be assigned to *X*-bits in order to suppress a high impact transition. In the example shown in Figure 12, RTI$_2$ and RTI$_4$ are compared, and then 0 or 1 is assigned depending on their values. Note that RTI and FTI are calculated before the logic value assignment. AP-fill simply refers to the RTI and FTI values in the logic value assignment phase.

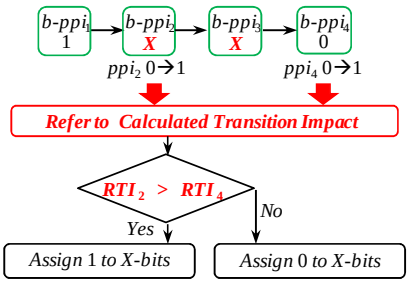

Figure 12. Logic Value Assignment of AP-Fill.

D. Algorithm of Adjacent-Probability Fill

Figure 13 shows the algorithm for Adjacent-Probability Fill (AP-fill). The input is a test cube set, *C*, and the output is a test vector set, *V*, which achieves low LTC power for the LOS scheme. AP-fill first calculates RTI and FTI of each flip-flop, then assigns logic values to *X*-bits based on the RTIs and FTIs (as described in Figure 13).

The time complexity of calculating RTI and FTI for all flip-flops is $O(m*n)$, where *m* is the number of gates, and *n* is the number of flip-flops. It should be noted again that the RTI and FTI for each flip-flop are calculated only once.

The time complexity of logic value assignment for all test cubes is $O(n*c)$, where *n* is the number of flip-flops and *c* is the number of test cubes. The complexity of logic value assignment is the same in both adjacent-fill and AP-fill.

```
Adjacent-Probability Fill (C)
    Input: Test cube set C;
    Output: Test vector set V;
{
  /* Transition Impact Calculation */
  for each b-ppi_j (j=1 to N_ppi){
     RTI_j and FTI_j calculation;
  }

  /* X-Filling according to RTI and FTI */
  for each test cube c_i in C (i=1 to N_tc){
     for each bit b-ppi_j in c_i (j=1 to N_ppi){
        if ( b-ppi_j == 0(1) && b-ppi_k == 0(1) &&
             b-ppi_{j+1, ... ,k-1} == x ){
           assign 0 (1) to each x;
        }
        if ( b-ppi_j == 1 && b-ppi_k == 0 &&
             b-ppi_{j+1, ... , k-1} == x ){
           if ( RTI_{j+1} > RTI_k ){
              assign 0 to each x;
           }
           else {
              assign 1 to each x;
           }
        }
        if ( b-ppi_j == 0 && b-ppi_k == 1 &&
             b-ppi_{j+1, ... , k-1} == x ){
           if ( FTI_{j+1} > FTI_k ){
              assign 1 to each x;
           }
           else {
              assign 0 to each x;
           }
        }
     }
  }
  return V;
}
```

Figure 13. Algorithm of AP-Fill.

V. Experimental Results

The proposed post-ATPG test modification flow incorporating AP-fill was implemented with C programming language and applied to six large ITC'99 benchmark circuits. Experiments were conducted on a workstation (Intel Xeon® 3.33GHz with 64GB main memory). Table II shows the circuit statistics. Note that the initial test vectors for transition delay fault were generated with TetraMAX™.

The average *X*-bit ratio for a test cube is shown in the last column of Table II. More than 80% of *X*-bits in the lager circuits such as b17, b18, and b19 were identified, and even in the smaller circuits such as b20, b21 and b22, over 60% of *X*-bits were identified. We used the test cube obtained to compare the proposed AP-fill with previous *X*-filling techniques.

TABLE II. CIRCUIT STATISTIC AND PERCENTAGE OF *X*-BITS

Circuit	*# Gates*	*# FFs*	*# Tv*	*Fault Cov.*	*X-bits*
b21	14,672	490	670	94.1%	62.7%
b20	16,304	490	860	92.0%	62.8%
b22	22,634	735	788	91.1%	65.7%
b17	40,693	1,415	1,034	93.0%	81.8%
b18	91,508	3,320	1,415	92.8%	86.5%
b19	182,357	6,642	2,172	90.0%	91.0%

Table III presents the average weighted switching activity (WSA) [6] reduction ratio of the various *X*-filling techniques compared to conventional transition delay test generation with random-fill. We applied 0-fill, 1-fill, adjacent-fill and the proposed AP-fill to the test cube set obtained by test relaxation. Although peak WSA should be reduced in order to avoid timing failure during test mode, the average WSA reduction ratio can be used to evaluate the effectiveness of *X*-filling techniques. Our experiments indicated that AP-fill achieved a WSA reduction ratio of almost 70%, while adjacent-fill achieved an average of 52%.

TABLE III. AVERAGE WSA REDUCTION RATIO

Circuit	*0-Fill(%)*	*1-Fill(%)*	*Adjacent -Fill(%)*	*AP-Fill(%)*
b21	41.2	26.5	37.5	60.2
b20	39.0	27.8	36.4	58.8
b22	43.6	30.0	39.8	61.3
b17	63.1	41.9	58.9	73.8
b18	72.0	51.8	66.7	78.6
b19	78.5	58.1	72.9	83.4
Ave.	56.2	39.3	52.1	69.3

Peak WSA reduction is dependent on a given test cube [7]. In the experiments, we used the same test cube for each *X*-filling technique. Therefore, we can fairly evaluate the peak WSA reduction capability of each *X*-filling technique. AP-fill achieved the highest reduction ratio, as demonstrated in Table IV.

TABLE IV. PEAK WSA REDUCTION RATIO

Circuit	*0-Fill(%)*	*1-Fill(%)*	*Adjacent -Fill(%)*	*AP-Fill(%)*
b21	16.7	12.7	15.5	22.2
b20	11.9	6.7	9.1	19.3
b22	17.5	13.6	16.1	21.4
b17	25.5	18.8	25.2	26.9
b18	20.0	20.9	19.5	23.5
b19	22.8	21.6	22.0	26.8
Ave.	19.1	15.7	17.9	23.3

Table V shows the CPU time for test relaxation and each *X*-filling technique. The CPU time for AP-fill includes the calculation for RTI and FTI of each flip-flop. Since RTI and FTI calculations for a flip-flop are totally independent of other flip-flops, these calculations can be processed in parallel, although we did not implement such calculation in this experiment.

Discussion

LOS achieved higher fault coverage and a smaller test vector count than LOC. Furthermore, while it is preferable to reduce shift power with DFT-wise techniques [8,11,14], both shift-in power and LTC power can be reduced by using AP-filling. Therefore, we propose that using AP-fill in the LOS scheme is one successful power-aware solution.

TABLE V. CPU TIME

Circuit	*XID(s)*	*0-Fill(s)*	*1-Fill(s)*	*Adjacent -Fill(s)*	*AP-Fill(s)*
b21	16.7	0.0	0.0	0.0	17.2
b20	23.0	0.0	0.0	0.0	17.0
b22	32.3	0.0	0.0	0.0	38.8
b17	108.9	0.1	0.1	0.1	205.9
b18	317.8	0.2	0.2	0.2	1113.6
b19	827.7	0.6	0.5	0.6	3272.4

VI. CONCLUSIONS

In this work, we proposed a novel *X*-filling technique for the LOS scheme, called ***Adjacent-Probability-based X-Filling*** (***AP-fill***), which utilizes probability calculation in adjacent-fill. Experimental results for larger ITC'99 circuits showed that the proposed AP-fill technique could achieve a higher reduction ratio than 0-fill, 1-fill, and adjacent-fill. It can be concluded that AP-fill could be one of the strongest *X*-filling techniques for LOS. Therefore, AP-fill can be modified and extended for several specific situations and helps in solving power related problems for LOS. Future work includes extending the proposed method to compression environments and large industrial circuits.

ACKNOWLEDGMENT

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

REFERENCES

[1] K. Agarwal, et al., "Power Analysis and Reduction Techniques for Transition Fault Testing," *Proc. Asian Test Symposium*, pp.403-408, 2008

[2] P. Girard, et al., *Power-Aware Testing and Test Strategies for Low Power Devices*, Springer, 2009.

[3] C. P. Ravikumar, et al., "Test Strategies for Low-Power Devices," *J. of Low Power Electronics*, Vol. 4, No.2, pp. 127-138, 2008.

[4] V. R. Devanathan, et al., "A Stochastic Pattern Generation and Optimization Framework for Variation-Tolerant, Power-Safe Scan Test," *Proc. Intl. Test Conf.*, Paper 13.1, 2007.

[5] X. Wen, et al., "On Low-Capture-Power Test Generation for Scan Testing," *Proc. VLSI Test Symp.*, pp. 265-270, 2005.

[6] 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.

[7] S. Remersaro, et al., "Low Shift and Capture Power Scan Tests," *Proc. VLSI Design*, pp. 793-798, 2007.

[8] X. Wen, et al., "A Novel Scheme to Reduce Power Supply Noise for High-Quality At-Speed Scan Testing," *Proc. Int'l Test Conf.*, Paper 25.1, 2007.

[9] K. M. Butler, et al., "Minimizing Power Consumption in Scan Testing: Pattern Generation and DFT Techniques," *Proc. Int'l Test Conf.*, pp. 355-364, 2004.

[10] N. Badereddine et al, "Power-Aware Scan Testing for Peak Power Reduction", *Proc. IFIP VLSI-SOC Conference*, pp. 441-446, 2005.

[11] K. Miyase, et al., "A Novel Post-ATPG IR-Drop Reduction Scheme for At-Speed Scan Testing in Broadcast-Scan-Based Test Compression Environment," *Proc. Int'l Conf. on CAD*, pp. 97-104, 2009.

[12] K. Miyase, et al., "XID: Don't Care Identification of Test Patterns for Combinational Circuits," *IEEE Trans. on CAD*, 23-2, pp. 321-326, 2004.

[13] A. H. El-Maleh, et al., "An Efficient Test Relaxation Technique for Synchronous Sequential Circuits," *IEEE Trans. on CAD*, Vol. 23, No. 6, pp. 933-940, 2004.

[14] X. Wen, et al., "A Highly-Guided X-Filling Method for Effective Low-Capture-Power Scan Test Generation," *Proc. Int'l Conf. on Computer Design*, pp. 251-258, 2006.

[15] L.-T. Wang, et al., *VLSI Test Principles and Architectures: Design for Testability*, San Francisco: Elsevier, 2006.