Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs

Santiago Remersaro[1], Xijiang Lin[2], Zhuo Zhang[1], Sudhakar M. Reddy[1],
Irith Pomeranz[3] and Janusz Rajski[2]

1. Department of ECE, University of Iowa, Iowa City, IA 52242
2. Mentor Graphics Corporation, Wilsonville, OR 97070
3. School of ECE, Purdue University, West Lafayette, IN 47907

Abstract

When the response to a test vector is captured by state elements in scan based tests, the switching activity of the circuit may be large resulting in abnormal power dissipation and supply current demand. High supply current may cause excessive supply voltage droops leading to larger gate delays which may cause good chips to fail tests. This paper presents a scalable approach called Preferred Fill to reduce average and peak power dissipation during capture cycles of launch off capture delay fault tests. Experimental results presented for benchmark and industrial circuits demonstrate the effectiveness of the proposed method.

1. Introduction

Scan based test has become the standard method for manufacture test. Earlier it has been observed that scan tests may cause switching activity far exceeding the activity during normal operation of the circuit [1]-[3]. Excessive switching activity is caused by scan tests requiring the circuit under test (CUT) to operate outside of the normal functional operation. Excessive switching activity during the application of scan tests are caused both during scan chain shifts to load tests and unload test responses as well as when the scan cell contents are updated using functional clocks in what are referred to as capture cycles. Abnormal switching activity causes abnormal peak as well as average power dissipation and supply currents. Excessive power dissipation may cause hot spots that could damage the CUT. Excessive peak supply currents may cause supply voltage droops resulting in increased gate delays during test. Increased gate delays during test may cause good chips to fail at-speed tests causing yield loss [2].

Several methods have been proposed to reduce switching activity in a CUT during scan based tests. Earlier methods to reduce switching activity during scan shift include adding additional logic [4]-[8], scan chain segmentation [5][9][10], ordering of tests [11] and scan elements [12], and reduced transition tests [13]. Earlier approaches to reduce switching activity during capture cycles include segmented scan with gated clocking [9][10], test generation methods [14]-[18], and methods to fill unspecified values in test cubes [19]-[22].

The test generation method proposed in [15] that restricts the scanned in states to the set of reachable states insures that the CUT operates in the functional mode only during capture cycles. Thus, such tests not only avoid abnormal switching activity during capture cycles but also avoid detection of faults that do not affect normal functional operation. Methods to generate tests with reachable scanned in states for transition delay faults (TDFs), called functional tests, were proposed in [17]. The requirement that scanned in state of a test is a reachable state may be difficult to ascertain in larger designs. For this reason pseudo-functional tests that attempt to operate a CUT close to its normal functional operation were investigated in [16][18]. Pseudo-functional test generation procedures determine a subset of non-reachable states and generate tests that avoid using any of the non-reachable states as scanned in states. Pseudo-functional tests do not guarantee avoiding non-functional operation that may cause excessive switching activity. Additionally finding a sufficiently large set of un-reachable states in large designs and specially those with multiple clock domains may not be practical.

The methods to fill unspecified entries in test cubes to reduce circuit switching activity have the advantage that they require only minimal changes to the existing ATPGs. However, in order to make such methods scalable, the specific procedure used to fill the unspecified entries must achieve substantial reduction

1-4244-0292-1/06/$20.00 © 2006 IEEE

in switching activity while not increasing the run times of ATPGs substantially. In this work we propose a scalable and effective method, called *preferred fill,* to fill unspecified entries in test cubes such that the switching activity during capture cycles of the tests is reduced. The proposed method is applied to generate launch off capture tests for TDFs. However, the method can be applied to generate tests for other fault models such as stuck-at and path delay faults.

The remainder of the paper is organized as follows. In Section 2 we briefly review earlier related works. In Section 3 we describe the proposed procedure and give experimental results on ISCAS-89 benchmark circuits. In Section 4 we describe an adaptation of the proposed method in to a commercial ATPG and give experimental results on industrial circuits. Section 5 concludes the paper.

2. Preliminaries

In this section we first briefly review launch off capture (LOC) also called broadside test method [23] used to detect delay faults in standard scan designs. We discuss the issues related to supply current demands and power dissipation during the application of LOC tests. Next we review earlier works that proposed methods to fill unspecified values in test cubes to reduce supply current and power dissipation during test application. We also discuss the shortcomings of the known methods that are addressed by the method proposed in this work and described in the following sections. Since our focus in this paper is transition delay faults (TDFs), all discussions assume TDFs.

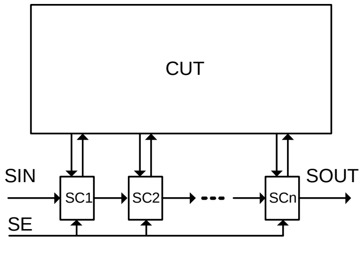

Figure 1: Standard scan

2.1. Launch off Capture Tests

For standard scan designs illustrated in Figure 1 two methods have been used to test TDFs. One is called launch off capture (LOC) [23] and the other is called launch off shift [24]. Both methods use a two-pattern test $<V_1, V_2>$ to detect a targeted TDF. In both methods, the first pattern V_1, called the initialization pattern, is scanned in, with the scan enable (SE) signal asserted. For a slow to rise (slow to fall) TDF, V_1 sets the fault site to 0(1). The second pattern V_2 is generated differently in the two methods. In LOC the second pattern is generated through the combinational logic of the circuit under test (CUT) by applying a clock pulse with SE de-asserted. This is referred to as the launch cycle and is illustrated in Figure 2 for a scan chain of length *n*. Application of V_2 activates the fault by launching a transition at the fault site and also propagates the fault effect to an observed output (primary output or a scan cell). For a slow to rise (slow to fall) TDF V_2 is a test for a stuck-at-0 (stuck-at-1) fault at the fault site. Following the application of V_2, another clock pulse is applied with SE still de-asserted to capture the CUT response to the test. This is also shown in Figure 2 where the corresponding clock pulse is labeled *C* for capture. Often the two clock pulses used to launch a transition and capture test responses are both called capture cycles. Thus a standard LOC test uses two capture cycles.

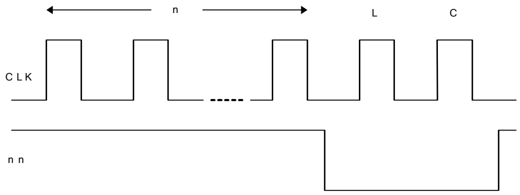

Figure 2: Timing diagram of LOC Tests

During the application of LOC tests, circuit nodes switch states due to scan shifts as well as capture cycles. The switching activity caused by the changes in the circuit inputs (primary inputs as well as the scan cells) initiated by the capture cycle clocks may be considerably higher than during normal circuit operation. High supply current demand may cause supply voltage to droop which tends to increase signal propagation delays of effected gates. Increased delay due to supply voltage droops may lead to capturing faulty responses, especially during the second capture cycle. This causes good chips to fail tests leading to yield loss [2]. Thus to reduce the potential yield loss it is critically important to reduce peak switching activity caused by the first capture cycle changing the state of the scan cells. It is also important to reduce the switching activity caused by the second capture cycle to prevent excessive power dissipation.

In this work we use *Weighted Switching Activity* (WSA) defined next. WSA was also used to represent instantaneous power and current in earlier works [4]. The *weighted switching activity* (WSA) of a node is the number of state changes at the node multiplied by (1+node fan-out). The WSA of the entire circuit is obtained by summing the WSA of all the nodes in the circuit.

2.2. Related Earlier Works

A test pattern for a target fault or a set of target faults typically contains many unspecified entries. The unspecified values can be arbitrarily specified to binary values. This is often referred to as filling of unspecified values or simply fills. Among the works that target reduced WSA some target reducing WSA during scan shifts [19] and others target reducing WSA during capture cycles [20]-[22]. Experimental results in [19] showed that a method called adjacent fill not only reduces WSA during scan shift but also the average WSA (averaged over all tests) caused by capture cycles. However, it was observed that this method may increases the peak or maximum value of WSA [21]. As noted earlier increased peak WSA during capture cycles increases the possibility of failing good chips due to increased supply voltage droops [2]. For this reason recent works proposed methods to reduce peak WSA during capture cycles of LOC tests [20]-[22]. The objective of this work is also to reduce peak WSA and average WSA caused during capture cycles in LOC tests.

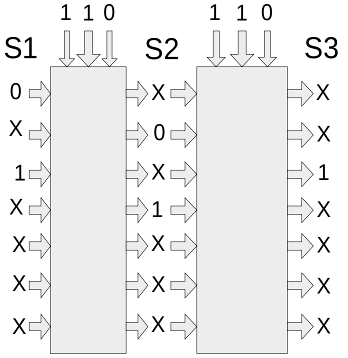

Figure 3: A LOC test

Next we briefly describe the methods used in [21][22] to fill unspecified entries in LOC test cubes. The two methods are essentially the same with some differences in details. We discuss the steps used in [22]. Consider the two-pattern test illustrated in Figure 3. In Figure 3 we show two frames. The vertical inputs at each time frame are primary inputs and the horizontal inputs and outputs are the present and the next states of the CUT, respectively. A vector in a two-pattern test has two components. One component corresponds to the primary inputs and the other component corresponds to the state variables. In Figure 3, S_1 (S_2) is the state part of the initializing (test) vector V_1 (V_2) of the two-pattern test <V_1, V_2>. S_3 is the state captured by the second capture cycle. If the state and the primary inputs of V_1 and the primary input part of V_2 are fully specified then S_1, S_2 and S_3 will all be fully specified. Further more S_2 and S_3 are determined entirely by S_1 and the primary inputs. Thus the unspecified values available to fill to reduce WSA are only in S1 and the primary inputs. For the sake of simplicity of explanation, assume that the primary input values are held constant during the application of the two-pattern test and are also fully specified. Then the switching activity in the circuit is caused by the changes in the state from S_1 to S_2 after the first capture cycle and next due to the state change from S_2 to S_3 after the second capture cycle. The method proposed in [22] fills the unspecified values in S_1 such that S_1 and S_2 as well as S_2 and S_3 differ in as few places as possible. In other words the method tries to fill the unspecified values in S_1 such that the Hamming distances between S_1 and S_2 and S_2 and S_3 are as small as possible. The procedure called LCP-fill from [22] to fill unspecified values in S_1 to reduce WSA caused by capture cycles of LOC tests is briefly reviewed next.

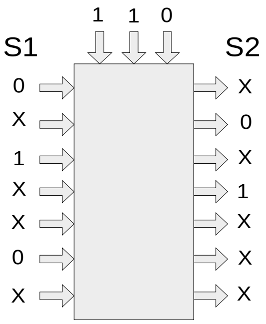

Figure 4: LCP filling

Consider the example illustrated in Figure 4 which shows a test cube with $S_1 = (0, X, 1, X, X, 0, X)$ which results in $S_2 = (X, 0, X, 1, X, X, X)$. Notice that the first, the third and the sixth elements of S_1 are specified and the corresponding elements of S_2 are unspecified. Similarly the second and the fourth components of S_2 are specified while the corresponding elements of S_1 are unspecified. In order to reduce the places in which S_1 and S_2 differ or equivalently to keep the Hamming distance between S_1 and S_2 small, the procedure of [22] iteratively specifies few selected unspecified entries in S_1 at a time. In the example being considered, one can specify the second and the fourth elements of S_1 to 0 and 1, respectively, to match the specified values in the corresponding elements of S_2. One can also attempt to set the first, the third and the sixth elements of S_2 to 0, 1 and 0, respectively, to match the values in the corresponding elements of S_1, through implication and line justification procedures of ATPGs. These steps will specify some of the originally unspecified values in S_1. S_1 is simulated with the newly specified values to obtain S_2 and the procedure is iterated until all components of S_1 are specified. One can also fill the unspecified values in S_1 such that the Hamming distance between S_2 and S_3 is small also. The procedure in [22] attempts to balance between the two objectives of keeping the Hamming distances between S_1 and S_2 and S_2 and S_3 small. The LCP-fill procedure leads to considerable reduction in peak and average WSA of LOC tests compared to random fill of the unspecified values in the test cubes. However the run time of this procedure could be potentially high. The reasons for this are the repeated simulations of incrementally updated test cubes and use of implications and line justification steps as part of the procedure.

In this work we investigate a simple and scalable procedure to fill unspecified values to reduce WSA during capture cycles of LOC tests. The procedure fills all the unspecified values in the initialization vector of a two-pattern test in one step rather than iterative incremental fill and simulation. It also does not use implications and line justification procedures. Experimental results presented for benchmark circuits show that the reductions in the peak and average WSA using the proposed procedure are similar to those obtained using LCP-fill while the run times are substantially smaller.

3. The Proposed Method

In this section the proposed method called *preferred fill* (*PF*) to fill unspecified values in test cubes is described. For the sake of simplicity we first describe the basics of the method in the context of circuits with 100% scan and present experimental results on ISCAS-89 benchmark circuits. In the next section we discuss extensions to the basic method used for industrial designs.

3.1. Preferred Fill

Consider a two-pattern LOC test $<V_1, V_2>$ for TDFs. It can be written as $V_1 = (PI_1, S_1)$ and $V_2 = (PI_2, S_2)$, where PI_1 and S_1 correspond to the PI values and the PPI values in the initialization vector and PI_2 corresponds to the PI values in the test vector and S_2 is the PPO values implied by V_1. Our goal, as in [20]-[22] is to reduce the Hamming distance between V_1 and V_2 by reducing the Hamming distance between PI_1 and PI_2 as well as between S_1 and S_2.

The Hamming distance between PI_1 and PI_2 can be minimized in a straightforward manner as done in [21]. Where ever possible we first fill the unspecified values in PI_1 (PI_2) to match the specified values in PI_2 (PI_1). This step is not needed in case the primary inputs are held constant over the two cycles during test generation. After this step, all the remaining unspecified values in PI_1 and PI_2 will be in the same positions. We randomly fill these values to have the same specified value. To illustrate filling of the values in PIs consider the following. In a two pattern test cube let PI_1 = (1XXX01X) and PI_2 = (01X0XXX). We fill the unspecified values in PI_1 (PI_2) in positions 2 and 4 (5 and 6) with 1 and 0 (0 and 1) to match the specified entries in these positions in PI_2 (PI_1). After this step we get PI_1 = (11X001X) and PI_2 = (01X001X). Next we fill positions 3 and 7 in $PI_1$and PI_2 randomly say by 0 and 1, respectively and get (1100011) and PI_2 = (0100011). The Hamming distance between PI_1 and PI_2 after the proposed fill is 1.

It should be pointed out that instead of randomly filling unspecified values in PI_1 and PI_2 in the second step one can use a better procedure. For example we can determine preferred values as we do for PPIs which is discussed next.

Next we describe the procedure we used to reduce the Hamming distance between S_1 and S_2 of a two-pattern test. Recall that we can arbitrarily fill the unspecified values in S_1 since it is the state that is scanned in. Let $S_1 = (s_{11}, s_{12}, s_{13}, ..., s_{1n})$ and $S_2 = (s_{21}, s_{22}, s_{23}, ..., s_{2n})$. If s_{1j} is unspecified then we should fill it with 1 (0) if the probability of s_{2j} taking the value 1 (0) is higher than it taking the value 0 (1). In other words we should fill s_{1j} with a value that is more likely to be held in the jth scan cell. With this in mind, we define $HP_j(v)$, $v \in \{0, 1\}$, as the probability of the jth flip-flop holding the value v under the assumption that all PIs and PPIs other than the jth PPI are applied random inputs. Note that $HP_j(v)$ is simply the conditional probability that the jth scan cell will hold the value v if it is loaded with v and all other inputs are applied random inputs.

Definition: 0 (1) is the *preferred value* for the jth scan cell (i.e. s_{1j}) if $HP_j(0) > HP_j(1)$ ($HP_j(1) > HP_j(0)$). Note that if $HP_j(0) = HP_j(1)$ then s_{1j} does not have a preferred value.

The procedure we used to fill the unspecified values in S_1 is the following. First, in every position of S_1 in which it has unspecified value and S_2 has a specified value we fill the unspecified value in S_1 by the specified value in S_2. After this step in every position in which S_1 has an unspecified value S_2 will also have unspecified value (S_2 may have additional unspecified values). This method is called *dynamic preferred fill*. We illustrate below this step using an example.

Let S_1 = (1XXX01X) and S_2 = (01X0XXX). We fill the second and the fourth position of S_1 to match the specified values 1 and 0 in S_2 and obtain S_1 = (11X001X) and S_2 = (01X0XXX). Notice that now every place in which S_1 is unspecified S_2 is also unspecified. In these positions, S_1 is filled with the preferred values if they exist. Otherwise, they are filled randomly.

The hold probabilities defined above or equivalently the conditional probabilities as discussed above can be computed in two ways. One is to use symbolic methods to accurately compute probabilities and the other is to estimate the probabilities by simulating a large number of random patterns. The first method is known to be NP-hard and the second method may also require a large computation effort for modern designs.

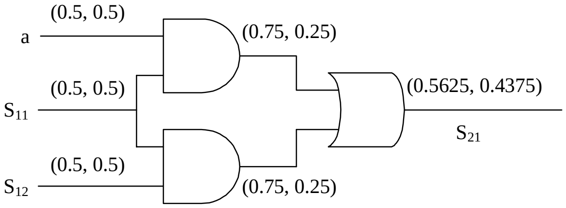

Figure 5: Signal probabilities

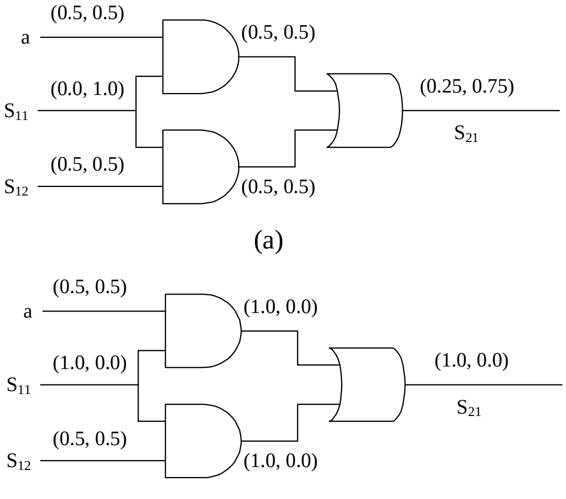

Figure 6: Conditional probabilities

A simpler but less accurate method is to use the standard signal probability calculation procedures ignoring statistical correlation between gate inputs [25]. However even this approach may take large run times if the number of scan cells in a circuit is large as is the case with modern VLSI designs. This is because the conditional probabilities must be computed separately for each PPI. For this reason we used a much simpler procedure. It uses simpler procedures to compute signal probabilities ignoring correlation between gate inputs together with a simplifying assumption. The simplifying assumption we used is that the probability of s_{2j}, the jth component of S_2, assuming the value 1 or 0 is independent of the state of s_{1j}. Under this assumption, we get, $HP_j(1) > HP_j(0)$ ($HP_j(0) > HP_j(1)$) if and only if $P_j(1) > P_j(0)$ ($P_j(1) > P_j(0)$), where $P_j(1)$ ($P_j(0)$) is the probability that s_{2j} takes the value 1 (0). Thus one can determine the preferred values for s_{1j} by simply computing the signal probabilities of s_{2j}. Signal probabilities of all s_{2j}, $1 \le j \le n$, can be simultaneously found in one pass using the standard signal probability calculation procedures [25] ignoring the correlation between gate inputs. In our experiments we used this simplified method of determining preferred values for s_{1j}.

Next we give an example to illustrate how we computed signal probabilities and preferred values for s_{1j}, $1 \le j \le n$. Consider the circuit in Figure 5. On each line we show two numbers. The first number is the probability of the line having the value 0 and the second number is the probability of the line having value 1. The circuit has two PPIs s_{11} and s_{12} and one PI *a*. We assign to these inputs equal probabilities for 0 and 1 as shown. Next we compute the probabilities for the other lines using the standard formulae [25] and ignoring the correlation between gate inputs. For example the two inputs to the OR gate of the circuit are correlated. The output of the circuit is the PPO $s_{21.}$ The probability of the scan cell holding a 1(0) is computed by computing the conditional probability of a corresponding PPO being at 1(0) given that the corresponding PPI is at 0(1). This is illustrated in Figure 6 for the circuit of Figure 5. In Figure 6(a) we set the 1 probability of PPI s_{11} to 1.0 and the 0 probability to 0.0. The probability of s_{21} being 1 under this condition is 0.75 and hence $HP_1(1) = 0.75$. We can compute $HP_1(0) = 1.0$ in a similar manner as illustrated in Figure 6(b). In Figure 6(b) we set the 0 probability of s_{11} to be 1.0 and its 1 probability to be 0.0. Since $HP_1(0) > HP_1(1)$ the preferred value for s_{11} is 0. As this example illustrates computing $HP_j(0)$ and $HP_j(1)$ requires two passes through the circuit for each j, $1 \le j \le n$. Instead, as discussed above, we use the signal probabilities $P_j(1)$ and $P_j(0)$ of the jth PPO being at 1 and 0, respectively, to determine the preferred value for the jth PPI. For the example under consideration, we have from Figure 5 $P_1(0) = 0.5625$and $P_1(1) = 0.4375$. $P_1(0) > P_1(1)$ we conclude that the preferred value for s_{11} is 0. In this example the preferred value found for s_{11} using hold probabilities as well as signal probabilities turns out to be the same. However in general it may be different.

We performed extensive experiments on benchmark circuits using all the methods discussed above to determine preferred values for s_{1j}. All the methods to determine preferred values yielded essentially the same results for the reduction of WSA. For this reason we used the computationally less demanding procedure of computing the signal probabilities for s_{2j}, $1 \le j \le n$, illustrated in Figure 5.

3.2. Experimental Results for Benchmark Circuits

The proposed method was implemented in C language and two experiments were conducted on ISCAS-89 benchmark circuits using Pentium 4 2.8 GHz PC with 1 GB RAM using Linux. For the first experiment we chose an experimental set up for the

Table 1: WSA reduction with preferred fill for ISCAS-89 circuits

Circuit	Average WSA Reduction				Peak WSA Reduction				CPU (Sec.)		# Pat.		
	1st Capture		2nd Capture		1st Capture		2nd Capture		LCP	Pref.	LCP	Pref.	Rand.
	LCP %	Pref. %	LCP %	Pref. %	LCP %	Pref. %	LCP %	Pref. %					
s1423	19.37	37.95	27.75	27.06	18.49	40.78	6.65	16.69	0.1	0	135	112	82
s5378	49.88	45.07	23.35	46.27	36.89	30.08	22.84	26.3	0.7	0	248	206	167
s9234	18.07	34.59	8.56	15.20	12.25	19.13	2.25	-1.07	2.5	0	350	392	328
s13207	23.95	45.77	9.40	29.55	15.42	26.62	4.51	22.81	13.1	0	356	385	377
s15850	45.82	41.82	31.29	31.56	31.68	35.57	6.78	21.9	5.9	0	220	202	183
s35932	60.38	30.78	53.22	28.56	25.05	23.5	19.88	11.83	27.6	0	72	96	40
s38417	20.07	20.75	18.20	15.86	13.02	29.24	10.96	21.88	87.3	0	227	324	222
s38584	43.96	31.56	37.97	18.57	50.17	27.63	51.29	35.68	302.4	0	444	320	292
Average	35.19	36.04	26.22	26.58	25.37	29.07	15.65	19.5	54.95	0	257	255	211

Table 2: WSA reduction by post-processing for ISCAS-89 circuits

Circuit	# ATPG Pat.	# Pat. After Post-Processing	WSA Reduction %				CPU (Sec.)
			1st Capture		2nd Capture		
			Average	Peak	Average	Peak	
s1423	82	82	33.59	28.65	26.60	18.01	8.02
s5378	167	166	45.42	34.81	45.49	23.50	7.50
s9234	328	325	33.19	36.68	16.99	11.47	36.87
s13207	377	367	47.29	28.51	31.36	28.01	63.27
s15850	183	181	43.05	43.13	35.34	3.48	37.43
s35932	40	40	22.12	19.53	18.57	19.84	40.07
s38417	222	222	19.40	25.44	16.36	6.34	106.27
s38584	292	292	33.46	38.65	19.42	33.21	165.47
Average	211	209	34.69	31.92	26.32	17.98	45.58

results on the benchmarks so as to be identical to the one used in [22] with which we compare the obtained results. Test cubes were obtained using an academic ATPG. The test cubes were generated using dynamic compaction procedures to detect all detectable TDFs using LOC tests. The dynamic compaction procedure was allowed to use up to 20% of the unspecified values in the test cube generated for the primary target fault. For each test cube the state captured after the application of the first capture cycle (i.e. S_2) was also provided by the ATPG. The unspecified values were filled using the preferred fill proposed in this paper.

In Table 1 we give the test generation results. After the circuit name we give the reduction percentages in average WSA and in peak WSA in the CUT caused by the first and the second capture cycle using LCP-fill of [22] and using the preferred fill proposed in this paper, respectively. Both reduction percentages are relative to the case when the unspecified values are filled randomly. From the table, it can be seen that both the preferred fill and the LCP-fill achieve similar reduction percentages in peak and in average WSA compared to random fill.

In Table 1, we also show the run times for the LCP-fill and the preferred fill under the column *CPU* in seconds. The run time given for LCP-fill is for only the fill procedure used in [22] and was provided to us by its authors [26]. The run times reported for preferred fill includes computation of the signal probabilities. It can be seen that the run times for preferred fill were unmeasurably small.

Direct comparison of test set sizes will not be meaningful since the test pattern generators used are not the same. However for the sake of completeness we report the test set sizes for LCP-fill from [22] and preferred fill under the column *# Pat*. We also give the pattern count when the unspecified entries are filled randomly and the dynamic compaction procedure is allowed to use all unspecified values to target detection of additional faults. These pattern counts are given in the last column of Table 1. It can be seen that over all the circuits the number of test patterns are similar as can be seen by the average numbers reported in the last row of Table 1. Also when preferred fill is used to fill unspecified entries in test cubes on the average the number of tests increases by approximately 20% to 25% from 211.

In the next experiment we used a post-processing step applied to (completely specified) test vectors generated by an ATPG. For this experiment we used completely specified test vectors generated using dynamic compaction with no limit thus obtaining test sets of minimal size. In the post-processing step we relaxed the given test vectors one at a time after ordering them by the WSA value of the first capture.

Relaxing a fully specified test unspecifies some specified entries in it without decreasing fault coverage. We adapted the test vector relaxation method of [27] for the two pattern TDF tests. The unspecified entries in the relaxed tests were then filled using preferred fill values. Results of this experiment are given in Table 2.

In Table 2 after the circuit name, the number of test patterns in the original test set followed by the number of tests after the post-processing step are given. In the next four columns, the percentage reduction in average and peak WSA in the first and the second capture cycles are given. Run times are given in the last column. Comparing the percentage reduction in WSA in Tables 1 and 2 we conclude that test relaxation followed by preferred fill achieves WSA reduction similar to those achieved by filling test cubes during test generation. The advantage of post-processing approach is that the number of test patterns does not increase. As a matter of fact for some circuits the number of patterns can decrease as shown in the second column of Table 2. However, a disadvantage of post-processing is that it requires additional run times beyond test generation times.

4. Applications to Industrial Designs

In this section, we give a brief sketch of the modifications we made to the basic preferred fill procedure for industrial designs.

4.1. Signal Probability Calculations

An industrial design often contains non-scan cells, RAMs, ROMs, undriven pins, unmodeled cores, buses, bidirectional pins, and combinational loops. Additionally we need to accommodate a larger set of signal values to include 0, 1, U and Z, where U represents unknown value and Z represents high impedance state. Let $P(v)$ be the probability that a signal line takes the value $v \in \{0, 1, U, Z\}$.

Undriven pins and unmodeled core outputs are treated as U and their signal probabilities are set as $P(0) = P(1) = P(Z) = 0$ and $P(U) = 1$. For the PIs which can take Z value, the initial signal probabilities are set as $P(0) = P(1) = P(Z) = 1/3$ and $P(U) = 0$. The signal probabilities at other signal lines are determined in a manner similar to the case when only Boolean gates and signal values 0 and 1 are needed. Now we need to compute probabilities of all four signals.

To determine the signal probabilities of the gates involved in a combinational loop, we need to calculate them iteratively. For example, the signal probabilities at gates G_1 and G_2 in Figure 7 are shown by using the evaluation order G_1 followed by G_2. For each line we show three signal probabilities corresponding to $P(0)$, $P(1)$ and $P(U)$. We illustrate the computations for six iterations. However, the calculation results give optimistic estimates for $P(0)$ and $P(1)$ when sequential behavior of the loop is not allowed during test generation. In our implementation, ATPG does not consider the sequential behavior of the loop. To reduce the optimistic calculation of $P(0)$ and $P(1)$, we apply the procedure given below to estimate the signal probabilities of the gates in a loop.

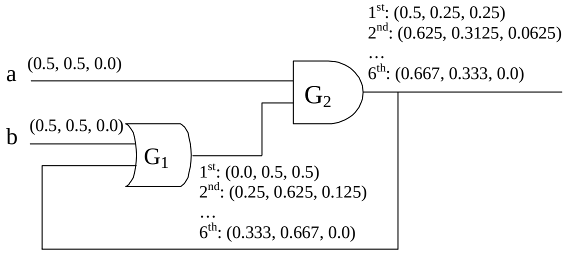

Figure 7: Calculate signal probabilities for a combinational loop

Procedure *estimate_ probabilities_in_loop*()

1. Find minimum cuts to break the loops. A cut is a signal line that breaks the feedback connections among gates.
2. Levelize the gates in the loop by assuming that the cut points are at the first level.
3. Assign probability ($P(0)$, $P(1)$, $P(U)$)=(0.0, 0.0, 1.0) to all cut points.
4. Initialize two change flags cf_0 and cf_1 to be false for every cut point.
5. Calculate the probabilities iteratively:
   a. Evaluate the probabilities of every gate in the loop according to their level order.
   b. For each cut point: If cf_v, where $v \in \{0, 1\}$, is true, do not update the probability at the cut point for value v. Otherwise, if the probability $P(v)$ at the cut point's driving gate is greater than 0.0, set the probability at the cut point to be $P(v)$ and set cf_v to be true.
   c. Stop iteration if there is no change in any cut point.

For example, let us assume that the cut point for the loop shown in Figure 7 is at the connection from G_2 to G_1. After the first iteration, the probabilities at the cut point are (0.5, 0.25, 0.25) and both change flags are set. The second iteration sets the probabilities at G_1 and

G_2 to be (0.25, 0.625, 0.125) and (0.625, 0.3125, 0.0625) and the calculation stops.

4.2. Preferred Fill for Industrial Designs

For the benchmark circuits an unspecified value in the jth PPI s_{1j} was filled with 1 if $P_j(1) > P_j(0)$ and filled with 0 if $P_j(0) > P_j(1)$, where $P_j(0)$ ($P_j(1)$) is the probability that PPO s_{1j} is 1(0). For industrial designs we filled an unspecified value in the jth PPI, s_{1j} with 1 if $P_j(1) > P_j(0) + \varepsilon$ and filled with 0 if $P_j(0) > P_j(1) + \varepsilon$, where ε was set to 0.025 in our experiments.

As noted earlier methods to fill unspecified values in test cubes to reduce WSA typically increases the test pattern counts compared to the case when the unspecified values are randomly filled [20]-[22]. The same was observed when preferred fill is used for industrial designs. In order to reduce the increase in test pattern count using preferred fill we experimented with first filling some percentage of unspecified values randomly followed by preferred fill. We call this technique *limited preferred fill*.

4.3. Use of Signal Probabilities for Test Generation

We also experimented with the use of signal probabilities to guide the ATPG to generate tests that cause lower WSA during capture cycles. We used signal probabilities to guide line justification step of the ATPG. Among the signals which when set to a value justifies a desired line value we pick the signal with the highest probability. The idea behind this heuristic is similar to the idea behind preferred fill. We report the results of using the test generation procedure with this modification.

4.4. Calculation of WSA with Unknown Values

The X-sources in the industrial designs make some gates have unknown values during good machine simulation. The unknown values can potentially create transitions at the internal gates. Ignoring them completely will underestimate the power dissipation in capture cycles. In this work, we computed WSA at a gate *g* using the formula given below.

$$WSA_g = \alpha * (1 + n\_fanout_g)$$

- When *g* has unknown values in both the previous time frame and the current time frame, α is equal to 0.125.
- When the gate *g* has unknown value in the previous time frame and known value in the current time frame, α is equal to 0.0625.
- When the gate *g* has known value in the previous time frame and unknown value in the current time frame, α is equal to 0.0625.
- If the gate has known values in both the time frames that create a transition then α is equal to 1.

4.5. Experimental Results

The proposed techniques were integrated into a commercial ATPG tool and several industrial designs were used to evaluate their effectiveness.

In Table 3, we show the characteristics of the industrial designs used in our experiments. After the circuit names, we show the numbers of gates in millions, the numbers of transition faults in millions, the numbers of scan chains and the numbers of scan cells in the designs under the columns *# Gates*, *# Flts*, *# Scan Chains* and *# Scan Cells*, respectively.

Table 3: Characteristics of Industrial Circuits

Circuit	# Gates (Millions)	# Flts (Millions)	# Scan Chains	# Scan Cells
ckt1	0.229	0.473	16	11900
ckt2	0.313	0.745	20	20200
ckt3	0.496	1.317	32	45000
ckt4	1.096	2.207	4	70300
ckt5	0.913	2.284	16	56400
ckt6	1.498	3.689	8	45000
ckt7	2.01	5.202	32	133500

The results using preferred fill to reduce the average WSA and the peak WSA for the two capture cycles are shown in Table 4 under the columns *1st Capture* and *2nd Capture*. Two sets of data are given in Table 4. The first set of data, given under *Preferred Fill* columns, corresponds to the case when preferred fill alone is used without modifying the test generation procedure. The second set of data, given under *Preferred Fill+ATPG* columns, is for the case when test generation procedure is modified by employing signal probabilities to guide line justification step of the test generation procedure. In Table 4 we give the results for pattern counts and WSA as a percentage increase or decrease relative to their respective values when random fill is used. In column *Pat. Inc. %* we give the percentage increase in pattern count. The percentage decreases in average WSA and peak WSA for each capture cycle are given under the columns *Average Red. %* and *Peak Red. %*, respectively. It can be seen that, on average, when only preferred fill is used, the average WSA and the peak WSA are reduced by 64.29% and 58.43% in the first capture cycle and by 46.90% and 53.55% in the second capture cycle. The test pattern count is increased by 144.83% on average. These results illustrate that random fill helps hold down test pattern counts, but it creates a larger amount of internal gate switching activity leading to much higher power dissipation during test. Preferred fill reduces the switching activity dramatically. However pattern counts are increased substantially.

Table 4: WSA reductions with preferred fill and ATPG

Circuit	Preferred Fill					Preferred Fill + ATPG				
	Pat. Inc. %	1st Capture		2nd Capture		Pat. Inc. %	1st Capture		2nd Capture	
		Average Red. %	Peak Red. %	Average Red. %	Peak Red. %		Average Red. %	Peak Red. %	Average Red. %	Peak Red. %
ckt1	66.61	45.77	36.74	25.81	32.42	80.07	47.87	33.51	27.80	24.99
ckt2	133.13	60.04	62.47	33.80	46.16	29.20	58.31	66.65	38.20	44.07
ckt3	158.07	70.16	56.51	69.10	61.82	81.99	74.00	44.29	69.72	55.47
ckt4	170.83	79.35	66.95	43.32	51.95	182.20	80.61	66.41	44.60	55.13
ckt5	219.12	67.21	66.38	50.69	66.88	303.39	63.32	66.65	44.26	68.67
ckt6	95.57	59.94	51.63	47.56	44.69	181.33	71.62	63.20	52.76	46.88
ckt7	170.51	67.56	68.32	58.04	70.94	25.88	69.54	71.96	58.62	72.95
Average	144.83	64.29	58.43	46.90	53.55	126.29	66.47	58.95	47.99	52.59

Table 5: WSA reductions with limited preferred fill

Circuit	Preferred Fill + 10% Random Fill					Preferred Fill + ATPG + 10% Random Fill				
	Pat. Inc. %	1st Capture		2nd Capture		Pat. Inc. %	1st Capture		2nd Capture	
		Average Red. %	Peak Red. %	Average Red. %	Peak Red. %		Average Red. %	Peak Red. %	Average Red. %	Peak Red. %
ckt1	23.99	36.87	26.49	21.61	25.28	34.18	41.78	26.46	25.86	24.14
ckt2	28.10	38.40	40.10	4.31	22.11	10.11	36.46	43.96	17.36	20.21
ckt3	25.63	58.50	41.42	54.20	48.31	27.15	62.43	38.30	60.36	49.47
ckt4	36.23	59.05	52.27	27.79	43.69	90.80	73.28	52.06	39.37	52.95
ckt5	59.76	40.72	51.90	38.52	54.44	105.85	48.45	54.45	39.32	58.86
ckt6	51.03	51.75	44.63	42.55	35.78	111.25	66.56	61.57	50.39	43.58
ckt7	38.00	49.19	52.63	47.85	65.85	-14.53	57.41	60.88	51.42	67.66
Average	37.53	47.78	44.21	33.83	42.21	52.12	55.20	48.24	40.58	45.27

Table 6: WSA reduction by post-processing

Circuit	# ATPG Pat.	# Pat. After Post-Processing	WSA Reduction %			
			1st Capture		2nd Capture	
			Average	Peak	Average	Peak
ckt1	15092	13491	49.76	31.68	24.81	9.92
ckt2	17135	16381	58.56	66.42	66.73	50.67
ckt3	6377	6357	76.73	50.79	66.89	22.03
ckt4	21761	21268	78.59	52.41	43.34	41.26
ckt5	60160	51013	75.95	14.44	58.11	67.61
ckt6	12521	12520	50.37	35.86	49.53	38.21
ckt7	50639	50406	69.42	60.30	59.42	62.94
Average	26241	24491	65.63	44.56	53.12	41.81

The data in Table 4 for the case using signal probabilities in test generation together with preferred fill given under *Preferred Fill+ATPG* columns shows that both peak and average WSA can be further reduced if signal probabilities are used to guide ATPG. Additionally pattern counts do not increase as much as when only preferred fill is used.

To moderate the pattern count increase when preferred fill is used, we apply the limited preferred fill where we first fill a certain percentage of unspecified values of PPIs randomly and fill the remainder of unspecified values using preferred fill. We report the results of these experiments in Table 5 for the case when up to 10% of the unspecified values are filled using random fill. The data in Table 5 is arranged in a manner identical to that in Table 4. From Table 5 one can notice that using limited preferred fill only the test pattern counts increase, on average, by only 37.53% instead of 144.83% when preferred fill is used without limited random fill, the average percentage reduction in average WSA for the first capture cycle is however reduced to 47.78% from 64.29%. Similar reductions in the percentage reductions of peak as well as the percentage reductions of WSA during second capture cycle can be noted. From the second set of data in Table 5, one can also note that the percentage increase in pattern counts is moderated when limited preferred fill is used with modified ATPG procedure. These results show that a good tradeoff between power dissipation and test pattern count is achieved using limited preferred fill.

Similar to the experiment done in Section 3.2, a post-processing step was applied to the test set generated by the ATPG with random fill. The post-processing step reduced the capture power through test vector relaxation and preferred fill and the results are given in Table 6. The data in Table 6 is arranged in an identical manner to that in Table 2. Comparing the WSA reductions in Table 6 with those under the column *Preferred Fill* in Table 4, it can be seen that the test relaxation followed by preferred fill achieves WSA reductions similar to those achieved by filling test cube during test generation. However, test pattern counts reduce when the post-processing step is used.

5. Conclusions

In this paper we presented a new technique called *preferred fill* to address the problem of larger than normal peak current and power dissipation during the fast capture cycles of broadside delay fault testing. Preferred fill uses circuit signal probabilities to fill unspecified values in test cubes. Since the signal probabilities can be computed once in a preprocessing step preferred fill fills all the unspecified values in a test cube simultaneously. The time to compute signal probabilities and hence preferred fill are negligible and hence the method is scalable to large designs. Since preferred fill is used only to fill unspecified values in test cubes, achievable fault coverage is not effected. Preferred fill is shown to achieve substantial reductions in peak and average power dissipation in benchmark and industrial circuits.

References

[1] J. Rearick, “Too Much Delay Fault Coverage Is a Bad Thing”, *Proc. ITC*, 2001, pp. 624 – 633.

[2] J. Saxena, K. M. Butler, V. B. Jayaram, et.al., “A Case Study of IR-drop in Structured At-Speed Testing”, *Proc ITC 2003*, pp. 1098 – 1104.

[3] P. Girard, “Low Power Testing of VLSI Circuits: Problems and Solutions”, *Proc. ISQED 2000*, pp. 173-179.

[4] S. Gerstendorfer and H. J. Wunderlich, “Minimized Power Consumption for Scan-Based BIST”, *Proc. ITC 1999*, pp. 77 – 84.

[5] L. Whetsel, “Adapting Scan Architectures for Low Power Operation”, *Proc. ITC 2000*, pp. 863-872.

[6] S. Bhunia, H. Mahmoodi, D. Ghosh, S. Mukhopadhyaay, K. Roy, “Low-Power Scan Design Using First-Level Supply Gating”, *IEEE TVLSI*, March 2005, pp. 384-395.

[7] R. Sankaralingam and N. A. Touba, “Inserting Test Points to Control Peak Power During Scan Testing”, *Proc. Intl. Symp. on DFT*, 2002, pp. 138 – 146.

[8] S. Sharifi, J. Jaffari, M. Hosseinabady, A. Afsali-Kusha and Z. Navabi, “Simultaneous Reduction of Dynamic and Static Power in Scan Structures”, *Proc. DATE 2005*, pp. 846-851.

[9] P. M. Rosinger, B. M. Al-Hashimi, and N. Nicolici, “Scan Architecture with Mutually Exclusive Scan Segment Activation for Shift and Capture Power Reduction”, *IEEE TCAD*, July 2004, pp. 1142-1153.

[10] K.-J. Lee, S.-J. Hsu and C.-M. Ho, “Test Power Reduction with Multiple Capture Orders”, *Proc. ATS 2004*, pp. 26 – 31.

[11] V. Dabholkar, S. Chakravarty, I. Pomeranz and S.M. Reddy, “Techniques for Minimizing Power Dissipation in Scan and Combinational Circuits During Test Application,” *IEEE TCAD 1999*, pp. 1325-1333.

[12] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, A. Virazel, “ Design of routing-constrained low power scan chains”, *Proc. DATE 2004*, pp. 62-67.

[13] S. Wang and S. Gupta, “LT-RTPG: A New Test-Per-Scan BIST TPG for Low Heat Dissipation,” *Proc. ITC 1999*, pp. 85-94.

[14] X. Liu and M. S. Hsiao, “Constrained ATPG for Broadside Transition Testing”, *Proc. Intl. Symp. on DFT*, 2003, pp. 175 – 182.

[15] I. Pomeranz, “On the Generation of Scan-based Test Sets with Reachable States for Testing under Functional Operation Conditions”, *Proc. DAC 2004*, pp. 928 – 933.

[16] Y.-C. Lin, F. Lu, K. Yang, and K.-T. Cheng, “Constraint Extraction for Pseudo-Functional Scan-based Delay Testing”, *Proc. ASP-DAC*, Jan. 2005, pp. 166 – 171.

[17] I. Pomeranz and S. M. Reddy, “Generation of Functional Broadside Tests for Transition Faults”, *IEEE TCAD*, 2005.

[18] Z. Zhang, S.M. Reddy and I. Pomeranz, “On Generating Pseudo-Functional Delay Fault Tests for Scan Design”, *Proc. DFTS 2005*, pp. 398-405.

[19] K.M. Butler, J. Saxena, A. Jain, T. Fryars, J. Lewis and G. Hetherington,“Minimizing power consumption in scan testing: pattern generation and DFT techniques”, *Proc. ITC 2004*, pp. 355-364.

[20] X. Wen, Y. Yamashitam, S. Kajihara, L. T. Wang, K. K. Saluja and K. Kinoshita, “On Low-Capture-Power Test Generaion for Scan Testing”, *Proc. VTS 2005*, pp. 265 – 270.

[21] W. Li, S.M. Reddy and I. Pomeranz,”On Reducing peak current and power during test,” *Proc. ISVLSI 2005*, pp. 156-161.

[22] 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. ITC 2005*, pp. 1019-1028.

[23] J. Savir and S. Patil, “Broad-Side Delay Test”, *IEEE TCAD*, 1994, pp. 1057-1064.

[24] J. Savir, “Skew-Load Transition Test: Part I, Calculus”, *Proc. ITC 1992*, pp. 705-713.

[25] M. Abramovici, M.A. Breuer and A.D. Friedman, *Digital Systems Testing and Testable Design*, IEEE Press 1990.

[26] X. Wen, Private Communication, November 2005.

[27] A. El-Maleh and A. Al-Suwaiyan, “An Efficient Test Relaxation Technique for Combinational & Full-Scan Sequential Circuits”, Proceedings of the 20th IEEE VLSI Test Symposium, pp. 53-59.