At-speed scan test with low switching activity

Conference Paper · May 2010

DOI: 10.1109/VTS.2010.5469580 · Source: IEEE Xplore

4 authors, including:

Janusz Rajski
Mentor Graphics
309 PUBLICATIONS **8,413** CITATIONS

SEE PROFILE

Sudhakar Reddy
University of Iowa
722 PUBLICATIONS **15,902** CITATIONS

SEE PROFILE

At-Speed Scan Test with Low Switching Activity

Elham K. Moghaddam
Department of ECE, University of Iowa, Iowa City, IA 52242
ekhayatm@engineering.uiowa.edu

Janusz Rajski
Mentor Graphics Corporation
Wilsonville, OR 97070. Phone
janusz_rajski@mentor.com

Sudhakar M. Reddy
Department of ECE, University of Iowa, Iowa City, IA 52242
reddy@engineeing.uiowa.edu

Mark Kassab
Mentor Graphics Corporation
Wilsonville, OR 97070
mark_kassab@mentor.com

Abstract

This paper presents a novel method to generate test vectors that mimic functional operation from switching activity point of view. The method uses states obtained by applying a number of functional clock cycles starting from the scan-in state of a test vector to fill the unspecified scan cell values in test cubes. Experimental results presented for industrial circuits demonstrate the effectiveness of the proposed method.

KEYWORDS: At-speed test, scan design, IR-drop, Switching activity, Low power tests.

1. Introduction

As VLSI design sizes and their operating frequencies continue to increase, at-speed test, which captures a test response at the rated clock speed, has become more important in order to maintain sufficient level of outgoing quality [2]. Using functional tests as at-speed tests is expensive due to the cost of generating the tests and the cost of high speed functional testers. Therefore scan-based at-speed tests are the most widely adopted tests for digital integrated circuits. While scan-based tests are less costly to implement and result in higher coverage metrics they may violate various functional constraints on the sequential circuits being tested. This may cause the circuit under test (CUT) to operate in non-functional states during test.

Given a circuit with k scan cells, there are 2^k states some of which are not functional states or legal states. For example in a BCD counter with 4 scan cells, only ten (0-9) of sixteen states are legal states. We use the words legal states, reachable states, and functional states interchangeably to mean the states of the sequential circuit under test that can be reached during normal/functional operation of the circuit.

During deterministic test generation, of scan based tests, the generated test cubes typically have only a small fraction of bits specified. Randomly filling unspecified bits with 0s and 1s may cause test vectors to contain non-functional states. Non-functional characteristics of test stimuli and test responses in scan testing cause higher switching activity at circuit nodes during test. Higher switching activity causes higher peak supply currents as well as higher power dissipation. Excessive peak supply currents may cause higher IR drops, which tend to increase signal propagation delays of affected gates. Increased gate delays during test may cause good chips to fail at-speed tests causing yield loss [2]. Therefore, it is important to reduce IR-drop during at-speed scan testing, in order to avoid IR-drop-induced yield loss.

In this work we propose an effective method to generate test patterns such that the switching activity during capture cycles of the tests is low. The proposed method uses background states obtained by applying a number of functional clock cycles starting with scan-in states to fill unspecified values in test cubes to keep the switching activity of test patterns low.

The rest of the paper is organized as follows. In Section 2, we review previous related works. In Section 3 we describe the proposed method. In Section 4 we give experimental results for industrial circuits and in Section 5 we summarize the contributions in this work.

2. Earlier Works

Several methods have been proposed to reduce overtesting and IR-drop during at-speed scan test.

Scan chain segmentation with gated clocking is proposed in [3, 4] to reduce capture power. The proposed methods divide the scan chains into several segments, enabling only one scan chain segment at a time to capture the test responses. This reduces the number of gates affected by the state change of scan cells, thus reducing the switching activity in the

978-1-4244-6648-1/10/$26.00 ©2010 IEEE

circuit. Although these methods reduce peak and average capture power during test, they increase test pattern counts, in addition to consuming additional chip area and design effort to enable independently clocking scan segments.

The test generation methods proposed in [5, 6] restrict the scanned in states to the set of reachable states to insure that the (CUT) operates only in the functional mode during capture cycles. Thus, such tests not only avoid abnormal switching activity during capture cycles but also avoid detection of certain faults that do not affect normal functional operation. Identifying reachable states has high complexity for large designs. This is addressed in [5, 6] by simulation based procedures that enumerate reachable states only as necessary for detecting targeted faults.

Pseudo-functional test generation procedures [7] determine a subset of illegal states and generate tests that avoid using any of the illegal states as scanned in states. Pseudo-functional tests do not guarantee avoiding non-functional operation. Finding a sufficiently large set of illegal states in large designs and specially in designs with multiple clock domains may be compute intensive. Functional and pseudo functional tests may also reduce achievable fault coverage since they avoid detection of faults that require non-functional scan-in states.

In order to detect faults not detectable by functional tests, partially-functional tests were introduced in [8]. Partially-functional tests use scan-in states that are at a minimal Hamming distance from a reachable state. By keeping the Hamming distance of scan-in states from reachable states to a small value the switching activity caused by the tests is kept close to or the same as that caused by reachable states.

The method to fill unspecified entries in test cubes (X-filling) has the advantage that it can be used as a post-processing step in any ATPG flow and requires no change to the ATPG procedures [9-12].

In [9,10], the authors propose a low-capture-power (LCP) X-filling method for assigning 0s and 1s to the X-bits in a test cube so that the number of transitions at the outputs of scan flip-flops in capture mode, for the resulting fully-specified test vector, is reduced. This approach is effective, however, it increases pattern counts significantly and also it is less scalable due to long run times.

The preferred fill (PF) technique proposed in [10], attempts to reduce the Hamming distance between the initialized and captured patterns by using a procedure based on signal probabilities. This method also increases pattern counts significantly while requiring shorter run times compared to LCP fill.

In order not to increase test pattern counts, a given set of specified test patterns are relaxed to obtain tests with unspecified values and then the unspecified values are filled using preferred fill [11] or LCP [12] values. One can also use the method proposed in this work to fill the unspecified values in the relaxed tests.

Another method that is widely used especially in test data compression procedures to reduce the switching activity is zero-fill. Zero-fill increases 0′s in a test vector, which helps reduce switching activity especially during scan shift, but tends to increase pattern counts significantly.

In this work, we investigate a new procedure to fill unspecified values in test cubes to cause low peak switching activity during test capture cycles. Since we don't identify reachable states to generate tests we cannot insure that the switching activity by the proposed tests is the same or determine how close it is to the functional switching activity. Our observations regarding the switching activity caused by the proposed tests is based on the fact that it is close to the minimum switching activity measured when the circuit under test is operated in functional mode for several clock cycles starting from the scan-in states of the tests in which the don't care bits are randomly filled. We discuss this in detail in the next section. Clocking the circuit in functional mode was proposed in [1] to address over testing. As shown in [6], this does not guarantee that the circuit will enter a reachable state starting from arbitrary scan-in state.

3. A New Method to Fill Unspecified Values in Test Cubes

In this section, we first give the results of an experiment we performed using launch off capture (LOC) tests for transition delay faults (TDFs). The experiments reveal a key observation we used to develop the proposed test generation method. In this work we target generation of LOC tests for TDFs.

3.1. Switching activity caused by LOC tests

In this work we use Weighted Switching Activity (WSA) defined below as a measure of power and supply current demand. The weighted switching activity (WSA) of a gate is the number of state changes at the gate multiplied by (1+ gate fanout). The WSA of the entire circuit is obtained by summing the WSA of all the gates whose state switched; $WSA = \sum_{\forall gate} (1 + gatefanout)$.

We computed WSA caused by different LOC tests for TDFs as follows. Recall that LOC tests for TDFs scan in a state, with the scan enable line at 1 (active) to set up the initialization vector of a two pattern test followed by two capture cycles during which the scan enable line is 0 (inactive). The two capture cycles are called launch and capture cycles.

We simulated LOC tests by applying the launch and capture cycles followed by several additional (capture) cycles with scan enable at 0 and determined WSA caused by each capture cycle. The timing diagram for this simulation is shown in Figure 1. In Figure 2 we show the WSA of up to a total of twenty capture cycles for 32 LOC tests obtained by random fill of the unspecified values in test cubes. The primary inputs were held at the values used in the test during the first two capture cycles and then were randomly set for the next 18 capture cycles. In Figure 2 we give WSA as a percentage of the maximum possible value of WSA which occurs if all the gates in the circuit under test (CUT) switch state.

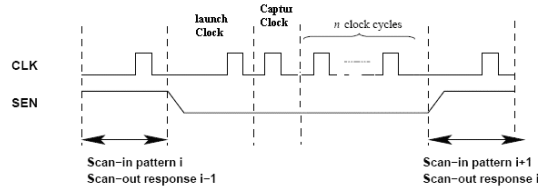

Figure 1: The timing diagram for generating functional background in LOC test

From Figure 2 we can make the following observations;

(i) The WSA reaches a steady state value which is typically much lower than the WSA during the first two capture cycles,

(ii) The steady state WSA value reached depends on the scanned in state, and

(iii) Even when the WSA caused by the first (launch) capture cycle is low the WSA of the second capture cycle could be much higher.

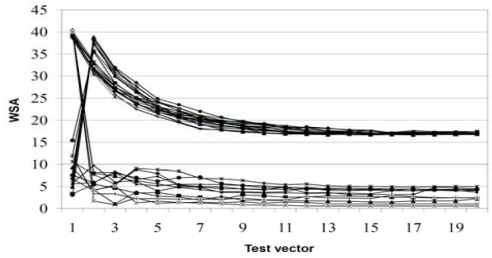

Figure 2: Applying 20 capture cycles to LOC test vectors generated by random fill

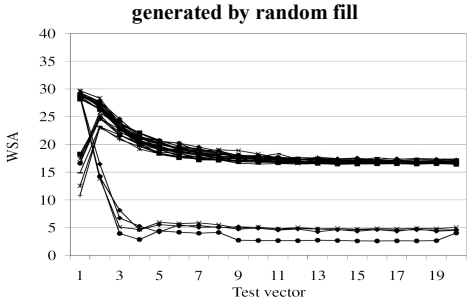

Figure 3: Applying 20 capture cycles to LOC test vectors generated using zero fill

We observed similar behavior for LOC tests obtained using preferred fill [10] and also using zero fill. This is illustrated in Figures 3 and 4. Additionally if the primary input values are held constant at their values during the second capture cycle the WSA during the last 18 cycles behaved similar to what is shown in Figures 2, 3 and 4.

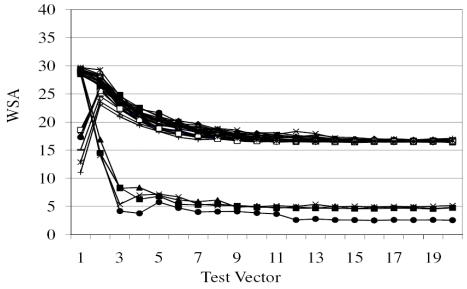

Figure 4: Applying 20 capture cycles to LOC test vectors generated using Preferred fill

3.2. The Proposed Test Generation Method

The proposed method to generate tests with low WSA during capture cycles is based on the observations made in the last section.

The observations made above suggest that if one were to fill the unspecified values in test cubes by the values in the scan cells found by simulating the test cube for some number of cycles with random values for primary inputs and scan enable at 0, after random filling the unspecified values in test cubes, the WSA of the resulting tests can be expected to be low and close to the steady state value determined during simulation. The reason for this expectation is that the number of specified values in test cubes is small and hence the state obtained after filling the unspecified values with the same as those of the state, say S, after simulation, differs from S only in some positions where the test cube had specified values.

We illustrate the proposed test generation using an example. Consider a test cube with T_1= (0, X, 1, X, X, 0, X). After filling the Xs randomly with zeros and ones and simulating the resulting fully specified test for several extra cycles assume that the state obtained is T_2 = (0, 1, 0, 0, 1, 1, 0). We fill the Xs in T1 by the corresponding values in T_2 to obtain the test T = (0, 1, 1, 0, 1, 0, 0). Note that the test T obtained differs from the state after simulation of T_1 in only 2 positions even though three cells contained specified values in T_1.

After extensive experiments using many tests we chose to use five cycles of simulating test cubes as described above and used the states of the scan cells obtained to fill the unspecified values in the test cubes of the LOC tests. Table 1 shows reduction in peak WSA during the first capture cycle for tests in

two industrial circuits, C2020 and C2225 when the unspecified values in the test cubes are filled from the state obtained after different cycles of simulation. (The profiles of the industrial circuits we used in our experiments are given in Table 2 in the next section.) It can be noticed that after five cycles we obtain most of the reduction in WSA.

Table 1: Reduction in peak WSA after different numbers of cycles of simulation

	% Peak WSA Reduction				
Circuit	#cycle=1	#cycle=3	#cycle=5	#cycle=10	#cycle=20
C2020	25.18	30.05	31.28	32.00	32.45
C2225	66.84	70.95	71.8	73.52	74.58

It is important to note that the proposed method to fill unspecified values does not require changes to the ATPG procedures. It only changes the way the unspecified values in test cubes are filled which is done after the test cubes are generated.

The pseudo-code for the test generation procedure using the method to fill unspecified values as described above is shown in Figure 5.

While *(not all faulst targeted)*
Select a fault and generate a test cube using dynamic compaction
Fill unspecified values randomly
Apply 5-10 clock cycles to obtain a pattern that mimic WSA of functional patterns
Use the obtained pattern to fill the unspecified values of the test cube.
Fault simulate the test and drop the detected faults
End while

Figure 5: The proposed low-power test generation procedure

As noted above, since only a small percentage of the scan cells have specified values in a test cube, when the unspecified values are filled using the state obtained after simulating as proposed, the state part of the test obtained will be close to the state obtained after simulation. Thus we can expect that the WSA of the capture cycles for the tests obtained as proposed here will be close to the steady state value illustrated in Figure 1. In order to verify whether this conclusion is valid we computed the WSA for the two capture cycles for all the LOC tests of an industrial circuit C2020 and the results are given in Figures 6 and 7. In Figures 6 and 7 we report the percentage increase in WSA of the two test capture cycles compared to the WSA caused by the background state obtained after five cycles of simulation. In Figures 6 and 7 X-axis gives the test number. It can be seen that the percentage increase in the WSA for the capture cycles of tests is less than 30% for all the tests and for majority of the tests less than 10%. The higher increase is only for the first few tests since in these tests typically higher percentage of test cube inputs are specified.

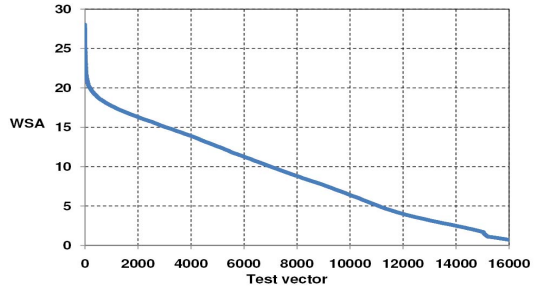

Figure 6: % difference in WSA in the first capture cycle of the proposed tests relative to the WSA when the test cubes were simulated for five cycles after random fill.

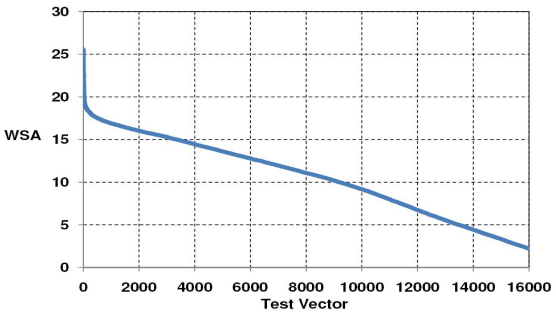

Figure 7: % difference in WSA in the first capture cycle of the proposed tests relative to the WSA when the test cubes were simulated for 5 cycles after random fill.

In the proposed test generation procedure the unspecified values in test cubes were filled to match contents of state elements after simulating randomly filled test cube for several cycles. Instead, one can fill the unspecified values to match any known functionally reachable state. One can also choose a functionally reachable state which minimally differs in the positions where the test cube has specified values. If this is done we will obtain what have been called partially-functional tests [8]. This approach will require computing a sufficient number of reachable states.

Our primary goal in this work is to reduce capture power in order to avoid IR-drop-induced yield loss. Experimental results in Section 4 show that we can achieve our goal with this method. But higher than normal average switching activity caused by scan shifts also could cause excessive heat due to increased power dissipation. We can reduce switching activity during both scan shift and during capture cycles by combining preferred fill method with ACF. Instead of filling all unspecified scan cells after generating test cubes randomly, one can fill a percentage of the unspecified scan cells with preferred values and the remaining ones with random values. Then apply 5-10 clock cycles to obtain the state used to fill the unspecified values in the test

cube. We used 50% fill with preferred and 50% with random values.

4. Experimental Results

The proposed method was implemented using a state of the art commercial ATPG for LOC tests. As discussed earlier we used five cycles of simulation of test cubes to obtain the background state used to fill the unspecified values in test cubes. 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 all unspecified values in the test cube generated for the primary target fault to target detection of additional faults values. The unspecified values remaining after dynamic compaction were filled using the proposed method or preferred fill or zero fill methods, resulting in three test sets for the three methods of fill. All test sets were generated to achieve the same maximum coverage of TDFs. On average the test generation times using the proposed method were approximately 15% higher than that for random fill.

In Table 2, 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 and the numbers of scan cells in the designs, respectively.

Table 2: Characteristics of industrial circuits

Circuits	# Gates (Milions)	# Flts (Milions)	# Scan Cells
C260	0.26	0.39	14k
C305	0.30	0.5	21k
C419	0.41	0.9	19k
C496	0.49	1.3	45k
C844	0.8	1.2	79k
C1498	1.4	3.6	45k
C2020	2.0	3.3	130k
C2225	2.2	3.3	140k
C2511	2.5	4	160k

We also computed the Bridge Coverage Estimate (BCE) of the three test sets. The Bridge Coverage Estimate (BCE) metric of a test set was proposed in [13] and is often used to estimate the bridging fault coverage of test sets. The BCE of a test set is determined from the information on the number of times each stuck-at fault is detected by the test set, and is calculated as follows: $BCE = \sum_{i=1}^{N} (1-2^i)\frac{f_i}{F}$, where $|F|$ is the total number of stuck-at faults, f_i is the number of stuck-at faults detected i times, and N is the maximum number of times any fault is detected.

In Table 3 we show percentage reduction in peak WSA, relative to random fill, caused in the first and the second capture cycles using the proposed test generation method, we call, ACF method, preferred fill method, zero fill method and ACF/PF is the combined ACF and PF method that fills 50% of unspecified scan cells with PF as described earlier.

From Table 3 we note that if ACF is used, peak WSA is reduced, on average, by 49% in the first capture cycle and by 28% in the second capture cycle. These reductions are higher than for the preferred fill and zero fill methods. Peak WSA reduction of ACF/PF is very similar to ACF Method.

Table 3: % Reduction in Peak WSA during first and second capture cycles

	% Reduction in Peak WSA							
	1st Capture				2nd Capture			
Circuits	ACF	PF	0 Fill	ACF/ PF	ACF	PF	0 Fill	ACF/ PF
C260	50	49	47	53	7	0	-4	9
C305	50	48	43	47	7	15	11	8
C419	42	32	49	47	19	23	25	26
C496	34	20	10	32	21	21	17	19
C844	73	73	73	73	8	15	13	12
C1498	31	37	44	40	21	30	35	23
C2020	31	32	33	32	32	31	33	28
C2225	74	61	70	69	74	61	75	73
C2511	52	43	4	53	64	39	5	60
Ave.	49	45	41	49	28	26	23	29

Table 4: % Reduction in State Element Transitions

	% Reduction in Peak SET							
	1st Capture				2nd Capture			
Circuits	ACF	PF	0 Fill	ACF/ PF	ACF	PF	0 Fill	ACF/ PF
C260	88	86	86	91	86	38	23	55
C305	61	64	60	58	65	17	22	15
C419	57	40	69	61	57	30	12	33
C496	61	22	15	55	67	48	15	51
C844	95	92	91	95	95	56	75	68
C1498	55	56	59	57	69	36	41	30
C2020	60	5	57	59	33	54	53	64
C2225	81	75	89	85	93	61	61	87
C2511	85	67	3	86	67	89	69	92
Ave.	71	62	59	72	70	48	41	55

In Table 4 we show the percentage reduction in peak SET (state element transition), which is the maximum number of scan cells whose states change during the capture cycles of the tests. It can be seen that ACF and ACF/PF reduce peak SET more than preferred fill and zero fill. Another point to be noted from Tables 3 and 4 is that peak WSA and SET for circuit C2511 using zero fill is not much reduced compared to the conventional tests that use random

fill of unspecified values. Consequently we conclude that zero fill can reduce overtesting in some circuits but not in all circuits.

Next we compare the average WSA reduction during scan shift. As shown in Table 5, ACF only reduces the average WSA during shift by 38% and is not as effective as preferred fill and zero fill. However it can be seen that ACF/PF achieves better reduction in average WSA during shift.

Table 5: % Reduction in Average WSA during shift cycles

	Average shift power Reduction			
Circuits	**ACF**	**PF**	**0 Fill**	**ACF/PF**
C260	**7.6**	**91.7**	**92.8**	**63.8**
C305	**20.2**	**19.6**	**87.7**	**34.9**
C419	**69.2**	**88.7**	**92.9**	**83.5**
C496	**13.0**	**89.9**	**93.7**	**39.2**
C844	**39.8**	**99.1**	**99.3**	**63.7**
C1498	**37.0**	**87.1**	**87.8**	**69.6**
C2020	**56.3**	**84.9**	**95.9**	**70.7**
C2225	**39.4**	**90.9**	**96.7**	**81.7**
C2511	**63.5**	**63.7**	**92.7**	**65.9**
Average	**38.5**	**79.5**	**93.3**	**63.7**

In Table 6 we give the percentage increase in pattern counts and percentage decrease in BCE relative to the case when random fill is used. From Table 6 one can notice that using ACF method the test pattern counts increase, on average, by 14% instead of 37% when preferred fill is used. Comparing ACF and preferred fill in terms of BCE, it can be seen that ACF always achieved better test quality. Consequently, compared with preferred fill, ACF retains test quality while simultaneously moderating the increase in test pattern counts. ACF/PF method gives similar results.

Table 6: Comparing pattern counts and BCE

	# Pattern Increase %				BCE Reduction %			
Circuits	**ACF**	**PF**	**0 Fill**	**ACF/ PF**	**ACF**	**PF**	**0 Fill**	**ACF/ PF**
C260	**7**	**10**	**9**	**6**	**0.6**	**1.0**	**0.9**	**0.1**
C305	**23**	**42**	**41**	**24**	**0.2**	**1.4**	**1.2**	**0.3**
C419	**21**	**49**	**38**	**23**	**0.1**	**2.1**	**3.5**	**1.3**
C496	**5**	**10**	**5**	**6**	**0.2**	**0.8**	**0.7**	**0.3**
C844	**16**	**88**	**76**	**38**	**1.9**	**16.8**	**17.1**	**7.3**
C1498	**13**	**47**	**51**	**26**	**0.5**	**2.2**	**2.9**	**1.3**
C2020	**31**	**50**	**55**	**32**	**3.9**	**6.2**	**6.0**	**4.4**
C2225	**4**	**15**	**15**	**1**	**1.5**	**4.6**	**5.5**	**2.7**
C2511	**7**	**23**	**15**	**14**	**1.5**	**5.5**	**3.0**	**5.7**
Ave.	**14**	**37**	**34**	**19**	**1.1**	**4.5**	**4.5**	**2.6**

5. Conclusions

Power-aware at-speed tests have become essential for ensuring the quality of submicron designs. In this paper we present a new technique called ACF for generating test vectors for at-speed delay tests that reduce switching activity during test. ACF uses background states obtained by applying a number of clock cycles to test vectors to fill unspecified values in test cubes.

The time to obtain background states depends on the number of additional clock cycles simulated. Experimental results show that after few clock cycles the WSA of background states become stable and applying more clock cycles has no significant effect on reducing WSA.

ACF is shown to achieve substantial reductions in peak WSA of capture cycles of LOC tests for industrial circuits without losing test coverage and only minimally increasing test pattern counts. Experimental result shows that BCE of tests obtained using ACF method is very similar to the BCE of conventional tests.

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. M. Rosinger, et.al., "Scan Architecture with Mutually Exclusive Scan Segment Activation for Shift and Capture Power Reduction", *IEEE TCAD*, July 2004, pp. 1142-1153.

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

[5] 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.

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

[7] Z. Zhang, et al, "On Generating Pseudo-Functional Delay Fault Tests for Scan Design", *Proc. DFTS 2005*, pp. 398-405.

[8] I.Pomeranze and S. M. Reddy, "Definition and Generation of Partially-functional Broadside Tests", IET Computer and Digit. Jan. 2009, pp.1-13.

[9] X. Wen, et al, "Low-capture-power test generation for scan-based at-speed testing," *Proc. ITC 2005*, pp. 1019-1028.

[10] S. Remersaro, et al, "Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs," in Proc. of Intl. Test Conf., Oct. 2006, pp. 1–10.

[11] S. Remersaro, et al, "Low Shift and Capture Power Scan Tests," Proc. VLSI Design Conf., Jan. 2007, pp. 793–798.

[12] X. Wen, et al, "A New ATPG Method for Efficient Capture Power Reduction During Scan Test," Proc. VLSI Test Symp., pp.58-63, 2006.

[13] B. Benware, C. Schuermyer, N. Tamarapalli, Kun-Han Tsai, S. Ranganthan, R. Madge, J. Rajski and P. Krishnamurthy, "Impact of multiple-detect test patterns on product quality", Proc. ITC 2003, pp. 1031- 1040.