SAT-based Capture-Power Reduction for At-Speed Broadcast-Scan-Based Test Compression Architectures

Michael A. Kochte[†,‡], Kohei Miyase[†], Xiaoqing Wen[†], Seiji Kajihara[†],
Yuta Yamato[§], Kazunari Enokimoto[†], Hans-Joachim Wunderlich[‡]
[†] Kyushu Institute of Technology, Iizuka, Japan
[‡] ITI, University of Stuttgart, Stuttgart, Germany
[§] Fukuoka Industry, Science and Technology Foundation, Fukuoka, Japan

***Abstract*—Excessive power dissipation during VLSI testing results in over-testing, yield loss and heat damage of the device. For low power devices with advanced power management features and more stringent power budgets, power-aware testing is even more mandatory. Effective and efficient test set post-processing techniques based on *X*-identification and power-aware *X*-filling have been proposed for external and embedded deterministic test. This work proposes a novel *X*-filling algorithm for combinational and broadcast-scan-based test compression schemes which have great practical significance. The algorithm ensures compressibility of test cubes using a SAT-based check. Compared to methods based on topological justification, the solution space of the compressed test vector is not pruned early during the search. Thus, this method allows much more precise low-power *X*-filling of test vectors. Experiments on benchmark and industrial circuits show the applicability to capture-power reduction during scan testing.**

Keywords-Low capture-power test, X-filling, ATPG

I. INTRODUCTION

For low and ultra-low power devices, functional power can be effectively reduced by power management at different levels in hard- and software [1]. Correspondingly, the power budget of these devices is very stringent. However, structural VLSI test aims to exercise the chip as thoroughly as possible and in shortest-possible time to minimize test costs. The high switching activity all over the chip during the test may easily exceed the tight functional power budget of the device by up to a factor of 5x, especially in the case of low power devices [2],[3], and cause over-testing, circuit-malfunction and eventually yield loss. Furthermore, heat damage and related reliability and lifetime degradation of the device under test may reduce product quality [4]. Special power-aware automatic test pattern generation (ATPG) algorithms are required to counter these problems by keeping the power budget.

In scan-based testing, shift power is dissipated during shift-in and shift-out of test stimuli and responses, while capture power is dissipated during capture of the circuit response. High average shift power dissipation causes heat damage; excessively high peak power both during shifting and capture may result in IR-drop and related timing failures [5],[6]. Scan shift power can be effectively reduced by scan chain segmentation [7], circuit modification [8], or test stimulus manipulation [9]. The method in [10] is able to reduce shift power in test compression environments. However, *capture-power* reduction is still difficult. Fig. 1 shows the waveforms of a launch-off capture (LOC) and launch-off shift (LOS) scheme for at-speed testing. In both schemes, the switching activity in the launch cycle may exceed the functional power budget of the device and result in increased and excessive delay due to large IR-drop. Consequently, a wrong test response may be captured in the capture cycle and delivered to the external test equipment (ATE) [5],[6].

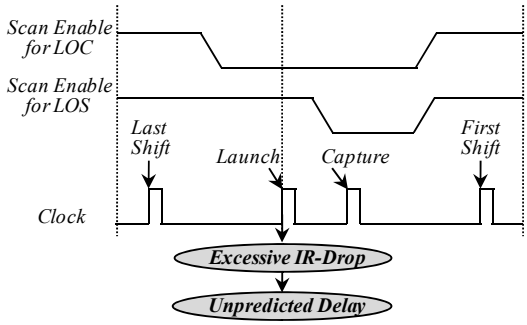

Fig. 1 Waveforms of LOC and LOS Scheme.

Dedicated low capture-power (LCP) aware ATPG causes a significant test pattern count increase with impact on test time and cost [11]. *X*-filling, a test set post-processing method, is a very successful approach to reduce capture-power by assigning appropriate binary values to unspecified signals (*X*-bits) in the test cubes such that switching activity in the launch cycle is reduced [12],[13]. High quality test sets from conventional ATPG with static and dynamic compaction can be processed using *X*-filling without increasing the test time.

LCP *X*-filling can also be applied to test compression environments. Yet, test compression already exploits the *X*-bits present in test cubes to increase the compression ratio as much as possible. Thus, the resulting compressed test vectors contain no or only very few *X*-bits which reduces the effectiveness of *X*-filling. For linear-decompression-based architectures [14], the method in [15] applies *X*-filling on uncompressed test cubes and then solves linear equations to obtain low power compressed test vectors. Although combinational broadcast-scan-based test compression typically achieves lower compression ratios than the sequential linear-decompression-based architectures, it has great practical relevance due to its much simpler design, easy integration into standard ATPG flows and higher ATPG efficiency [16]. The *X*-filling method of [17] achieves a capture-power reduction in Illinois broadcast-scan architectures [18]. The approach requires addi-

tional DFT hardware. Also, Illinois scan may suffer from low fault coverage. The approach of [19] allows power reduction in state-of-the-art combinational broadcast-scan architectures. It exploits clock gating infrastructure in the design to silence multiple scan flip-flops. Since the employed *X*-filling algorithm is based on a simple topological procedure without any backtracking, it searches only a small subspace of all possible *X*-filling solutions and may miss a more optimal one.

In this paper we propose a novel method to post-process test vectors for combinational broadcast-scan-based test compression schemes. The proposed method reduces capture power even if there are no *X*-bits present in the compressed test vectors. The algorithm uses a SAT-solver to check if intermediate *X*-filling solutions are still compressible for the particular decompressor logic of the device under test. Since the decompressor logic is typically small, the problem instances are small as well and this method performs very fast. No hardware modification of the circuit under test is required.

The next section describes the typical ATPG flow for broadcast-scan-based test compression. Section III presents the proposed method for capture-power reduction, followed by experimental results on benchmark and industrial circuits in section IV. Section V concludes this paper.

II. ATPG for Broadcast-Scan-Based Test Compression

This section describes the typical ATPG flows for broadcast-scan-based test compression schemes. Fig. 2 shows a decompressor used in test compression, an external test vector *E* and an internal test vector *I*. The external test vector is delivered by the ATE. The internal test vector is obtained by expanding the external test vector with the decompressor.

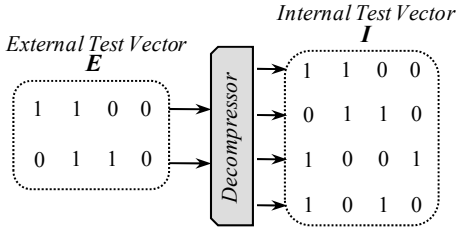

Fig. 2 External & Internal Test Vector of Test Compression.

The broadcast-scan-based technique directly generates external test vectors by adding constraints to ATPG as shown in Fig. 3. The constraints are constructed based on the function of the decompressor logic. Fig. 4 shows a simple MUX-based decompressor used for broadcast-scan-based technique.

Broadcast-scan-based schemes are widely accepted because the decompressor circuit is much simpler than linear decompressors and can be easily integrated into the design. Therefore,

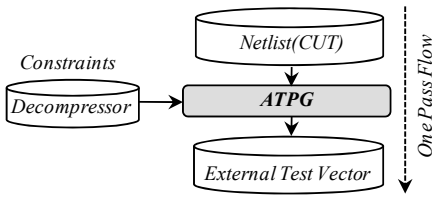

Fig. 3 One Pass ATPG Flow.

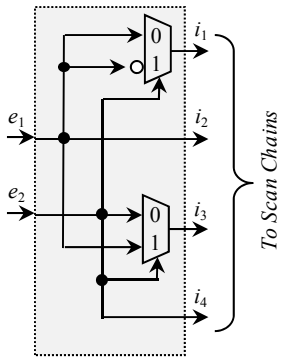

Fig. 4 MUX-Based Decompressor.

effective capture-power reduction techniques are strongly required for this type of test compression as well.

III. Low-Capture-Power Test Post Processing for Test Compression

Many compression techniques exploit *X*-bits in order to achieve a high compression ratio. Therefore, there are no or only a few *X*-bits in the resulting compressed external vectors, and consequently *X*-filling techniques for LCP testing cannot be effectively applied.

Fig. 5 gives an example of three generated external test vectors with only a few *X*-bits and the corresponding internal vectors. Typically, only a subset of the specified bits in the internal vectors is required for fault detection. This subset may be produced with different external vectors. By selecting one particular external vector, the values of the bits not required for fault detection can be influenced. Obviously, the lower the number of specified values required for fault detection, the more possibilities for compression exist and the higher is the control over the remaining values. This freedom can be exploited for power-aware compression.

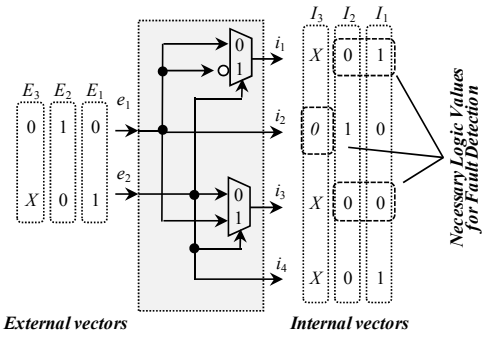

Fig. 5 Example of Power-Aware-Compression.

For the given example, there exist two possibilities to compress the bits required for fault detection in internal vector I_2, either by setting E_2 to (1,0) or to (0,0). The external vector should be selected such that the resulting switching activity does not exceed the functional power budget.

To assess whether a vector causes excessive transitions and thus power dissipation in the launch cycle, we use the weighted switching activity (WSA) of a vector pair v,v^* as metric due to good correlation with power dissipation [4] and

related IR-drop [20]. In the LOC scheme, the second vector v^* is the circuit response to v.

$$wsa(v, v^*) := \sum_{i \in Gates} fanout(i) \cdot (val(i, v) \oplus val(i, v^*)) ,$$

with *fanout(i)* being the number of fanouts at gate i, and *val(i,v)* being the output value of gate i under v. More precise metrics, e.g. simulations at electrical level, are applicable in principle, yet require very high computational resources.

A vector v is called *risky* if *wsa(v,v*)* exceeds a given threshold value pwr_{th}, which can be chosen by the designer. In this paper, we use threshold values of pwr_{th}=15% and 20% of the theoretical maximum WSA of the circuit. To avoid a reduction of test quality and under-testing, it is not desirable to reduce test power too much. The proposed algorithm thus focuses only on test power reduction of risky vectors.

The next section presents an overview of the proposed algorithm, followed by a detailed explanation of the handling of external compressed vectors and subsequent LCP *X*-filling.

A. Overview of the Test Post-Processing Flow

The proposed algorithm for broadcast-based test architectures processes highly compressed test patterns. Using *X*-identification and *X*-filling methods, it generates compressible internal LCP test vectors and finally extracts the external compressed vectors.

Fig. 6 depicts the overall flow. In Phase I, the set of internal vectors I is computed from the external compressed test vectors E by logic simulation of E using the logic model of the decompressor (c.f. section III.B). The subset $I_R \subseteq I$ of risky vectors w.r.t. the power budget pwr_{th} is identified using the WSA of the internal vectors.

Since the initial external vectors typically contain only a few or no *X*-bits, the resulting vectors I_R are highly specified. We apply an extended version of the *X*-identification (XID) algorithm of [21] to uncover as many *X*-bits in the internal vectors as possible without compromising fault coverage. By focusing *X*-identification only on I_R, more *X*-bits can be uncovered, which increases *X*-filling effectiveness. The resulting sparsely specified vectors $I_{R,X}$ are always compressible with the given decompressor and are used as basis for LCP *X*-filling. To ensure that *X*-filling of $I_{R,X}$ generates compressible vectors, the *X*-filling algorithm takes the decompressor logic into account as explained in section III.C.

Depending on the circuit and vectors, it may not be possible to reduce the WSA of each risky vector below pwr_{th}, especially if the number of *X*-bits is low. To further reduce the number of risky vectors at this point, Phase II targets vectors that are still risky after completion of Phase I. These vectors I'_R are duplicated. In the next step the targeted faults of the vectors in I'_R are distributed to the two duplicates. With a lower number of target faults, fewer specified bits are required for fault detection and the number of *X*-bits in $I'_{R,X}$ increases. *X*-filling is performed on the vectors $I'_{R,X}$ to obtain $I'_{R,LCP}$.

Finally, the external compressed vectors E_{LCP} of the fully specified LCP vectors of phase I and II are extracted.

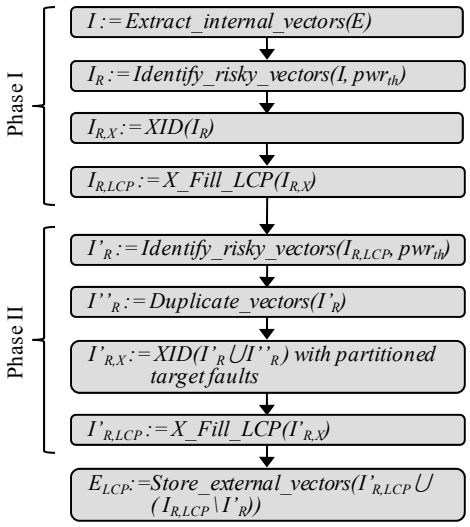

Fig. 6 LCP Test Post-Processing Flow.

B. Expansion of External Test Vectors

To obtain internal vectors from external ones, a logic simulation of the combinational decompressor logic is performed. In this step (c.f. Fig. 7), the *vector slices (VS)* of the external vector, i.e. the test data delivered by the ATE in one scan cycle, is decompressed into the *scan slices (SS)* of the internal vector. The scan slices are then delivered to the scan chains.

For a fast simulation of the whole (full-scan) circuit under test, an entirely combinational model is generated. In this model, a single decompressor is duplicated n times, where n is the length of the longest scan chain in the circuit. Fig. 8 shows the combinational model for the example of Fig. 7. In the model, four decompressor instances are created since the scan chain length is four flip-flops and there are four scan slices. After logic simulation of the external vector slices, the internal signal values I of the scan slices are extracted.

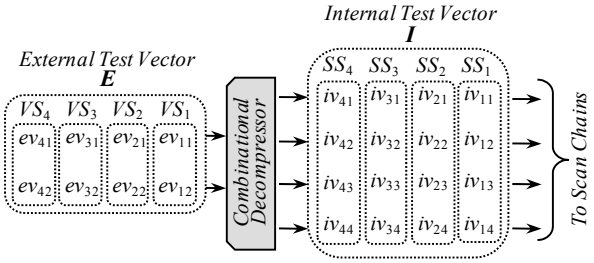

Fig. 7 Test Expansion with Decompressor.

C. X-Filling for Compressible LCP Test Vectors

The expanded and sparsely specified internal vectors $I_{R,X}$ are known to be compressible. However, during conventional *X*-filling flip-flops may be assigned values which render this vector incompressible. Thus, after determining a target value for a flip-flop, a compressibility check is performed based on the decompressor logic and the currently specified values. If the internal vector is still compressible with the new assignment, the *X*-filling algorithm proceeds to the next

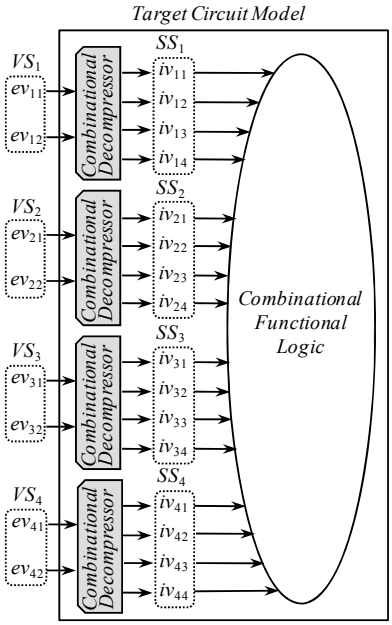

Fig. 8 Decompression Expansion for Scan Slices.

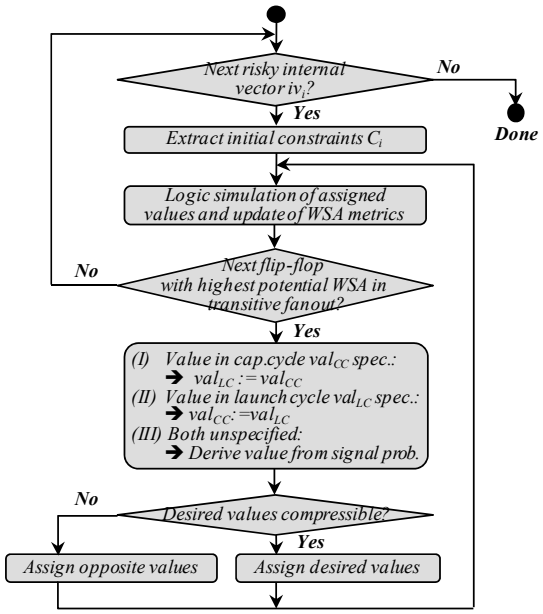

Fig. 9 Overview of the Proposed *X*-Filling Step.

objective signal or flip-flop. If it is not compressible, the opposite value is assigned to the flip-flop, and *X*-filling proceeds to the next objective.

The resulting compressible solution may cause a higher switching activity than a test vector generated without taking compression into account because some signals may have to be assigned unfavorable values due to constraints of the decompressor. This problem could be partially solved by a backtracking-based search for a compressible and optimal solution. The algorithm proposed here trades off optimality of the solution and runtime and does not employ backtracking.

The compression aware *X*-filling algorithm is depicted in Fig. 9. For each risky vector *iv*, the initial constraints C_i, i.e. the specified logic values in the scan slices SS_i, are extracted and stored for each scan cycle. Flip-flops with *X*-values are ordered w.r.t. a WSA metrics that counts the potential activity in the transitive fanout of each flip-flop. Processing then starts with the flip-flop with highest potential WSA resp. highest capture-power reduction impact. To reduce switching activity in the launch cycle, the Hamming distance between the values in the launch and capture cycle is reduced as much as possible. If the corresponding flip-flop is *X* in the launch cycle, but has a defined binary value $val_{CC} \in \{0,1\}$ in the capture cycle, then it should also take this value $val_{LC}=val_{CC}$ in the launch cycle. If the value in the launch cycle is specified, the algorithm tries to find an assignment to flip-flops in the launch cycle which justifies val_{CC}. If a flip-flop is *X* in both cycles, the target value is selected based on the signal probability of the flip-flop in the capture cycle. This way, the number of switching flip-flops and thus, the switching activity in their transitive fanouts are reduced.

To ensure compressibility of the scan slice SS_i with the targeted values, we perform a SAT-based check based on a SAT-model consisting of the set of characteristic equations of the gates of the decompression logic (c.f. Fig. 10). For each target value val_{LC} in the launch cycle, we check if there is solution of the SAT model given the constraints $C_i \cup \{iv_{ij}=val_{LC}\}$. If that is the case, scan slice *i* is still compressible and we set $C_i := C_i \cup \{iv_{ij}=val_{LC}\}$. Otherwise, this additional assignment is not consistent with the current set of constraints and cannot be compressed. Thus, the opposite value is added to the set of constraints $C_i := C_i \cup \{iv_{ij}=\neg val_{LC}\}$. Then, event based logic simulation is performed for the newly assigned value and the WSA metrics is updated. The algorithm proceeds with the next flip-flop to be justified according to the metrics.

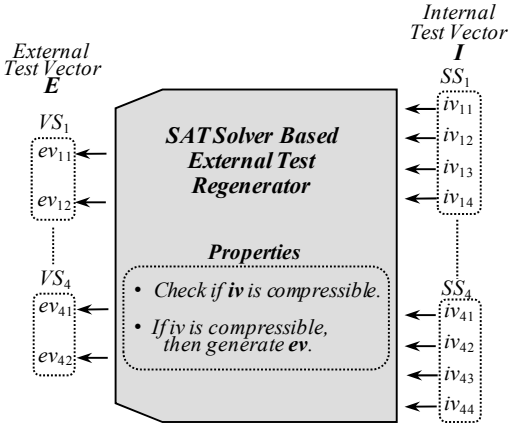

Fig. 10 SAT Solver Based External Test Regenerator.

After performing LCP *X*-filling in phase I and II of the algorithm, the compressed external test vectors are extracted and stored.

IV. Experimental Results

The proposed algorithm has been implemented in our CAD framework. To check for vector compressibility, the MiniSAT solver [22] is used. All experiments were conducted on a 3 GHz Intel Core i7 workstation. The algorithm is evaluated on the larger circuits of the ITC'99 benchmark suite and a

wide variety of industrial circuits (named p*k) with up to 630000 gate primitives, kindly provided by NXP. Table 1 gives the number of gate primitives, the number of flip-flops and the number of transition faults of the considered circuits.

For these circuits, 4x, 8x and 16x broadcast-based full-scan compression DFT hardware was synthesized using a commercial Design-for-Test tool. The number of internal scan chains ranges from 32 to 128 depending on the compression configuration. Test vectors for transition faults were generated using a commercial ATPG. Table 2 lists the transition fault coverage and number of test vectors for the circuits and the different compression ratios in column 3 and 4. The results were obtained without additional test point insertion which could explain low fault coverage for some of the circuits. In general, with increasing compression ratio the number of vectors grows as well.

We investigated the effectiveness of the proposed LCP *X*-filling algorithm for a 15% and 20% power budget threshold. For the two cases the table shows the initial number of risky vectors w.r.t. the power threshold, the number of risky vectors after application of the algorithm, and the per-cent change in vector count due to duplication.

Of the three ITC'99 benchmark circuits, only b17s has risky vectors when considering a power budget of 15%. The algorithm effectively reduces the WSA of these vectors below the threshold, so that finally no more risky vectors exist.

For the larger industrial circuits the number of risky vectors is also reduced. For many circuits and compression configurations, all risky vectors are eliminated with no or only marginal increase of the vector count. However, for some circuits and compression configurations, not all risky vectors could be rendered power-safe. With increasing compression ratio, the freedom to assign targeted values to the scan flip-flops is reduced and thus, the algorithm does not always succeed in limiting the WSA of risky vectors below the power threshold. To achieve power safety for these cases, masking risky test responses as in [23] could be applied.

In a few cases with the targeted tight power budget of 15%, the number of remaining risky paths is high even for a low compression ratio, as for example for p259k. The table cells marked with a dash indicate cases where no reduction in the number of risky vectors could be achieved. In these cases, the targeted power threshold is very close or even below the average WSA of the initial test set. In consequence, *X*-identification is not focused on a small subset of vectors, but has to be applied to almost all vectors, resulting in a lower average number of *X*-bits per vector. With the given compression constraints, this number of *X*-bits does not leave enough freedom to sufficiently reduce WSA below the threshold.

For the larger industrial circuits and the power budget of 20%, all risky vectors could be rendered completely power-safe for about 80% of the compression configurations with no or only a marginal vector increase (max. 1.2%). For circuit p418k, a very high number of risky vectors could be rendered power-safe w.r.t. a 15% power budget at no cost at all.

The runtime of the approach depends on the circuit size, the decompressor, and the number of test vectors to process. For the case of a 20% power threshold, the runtime is given in minutes in column 11. For smaller circuits, the runtime does not exceed a few minutes even if multiple thousands of patterns are processed. For larger circuits, runtimes reach multiple hours and are in the order of ATPG. Depending on the circuit, 30-50% of the runtime is spent on XID.

Table 1 Circuit Information

Design	# Gates	# FFs	# TF (in 1000)
b17s	30540	1415	168
b18	73747	3320	423
b19	145828	6642	840
p77k	84314	3386	394
p81k	95235	3877	492
p89k	104588	4301	493
p100k	117644	5735	532
p259k	321205	18389	1485
p295k	363381	18465	1626
p330k	408450	16775	1863
p418k	543724	28616	2363
p533k	630708	32409	2963

V. CONCLUSIONS

Excessive test power dissipation is a severe issue especially for low power devices with tight power budget, causing yield loss and potentially heat damage. This paper proposed a novel capture-power reduction method for broadcast-scan-based test compression environments. The method is a post-processing technique that targets test power reduction of risky test vectors by use of an *X*-filling algorithm that takes the compression constraints into account. The experimental results for a wide variety of circuits showed that the number of risky vectors can be effectively and efficiently reduced. For a test power budget of 20%, the method eliminates all risky test vectors in about 80% of the circuits and compression configurations at no or only marginal cost in terms of vector count.

ACKNOWLEDGEMENTS

This work was supported in part by JSPS Grant-in-Aid for Scientific Research (B) 22300017. M. Kochte was a Visiting Researcher at the Kyushu Institute of Technology in 2010, supported by the German Academic Exchange Service (DAAD).

REFERENCES

[1] J. Rabaey, "Low Power Design Essentials (Integrated Circuits and Systems)", Springer, 2009.

[2] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," Proc. IEEE VLSI Test Symp., pp. 4-9, 1993.

[3] S. Sde-Paz and E. Salomon, "Frequency and Power Correlation between At-Speed Scan and Functional Tests," Proc. IEEE Intl. Test Conf., Paper 13.3, 2008.

[4] P. Girard, "Survey of Low-Power Testing of VLSI Circuits," IEEE Design & Test of Computers, vol. 19, no. 3, pp. 82-92, 2002.

[5] J. Wang, D.M.H. Walker, A. Majhi, B. Kruseman, G. Gronthoud, L. E. Villagra, P. Wiel, S. Eichenberger, "Power Supply Noise in Delay Testing," *Proc. Int'l Test Conf.*, Paper 17.3, 2006.

[6] S. Ravi, "Power-Aware Test: Challenges and Solutions," *Proc. Int'l Test Conf.*, Lecture 2.2, 2007.

Table 2 Experimental Results

				15% power budget			20% power budget			
Design	Compression ratio	# Vectors	Fault coverage	# Risky vectors	# Risky vectors final	Vector increase %	# Risky vectors	# Risky vectors final	Vector increase %	Runtime (min)
b17s	4x	1836	91.86%	18	0	0	0	0	0	0.4
	8x	2028	91.74%	8	0	0	0	0	0	0.7
	16x	2800	91.40%	2	0	0	0	0	0	1.2
b18	4x	2280	80.04%	0	0	0	0	0	0	1.2
	8x	3038	79.95%	0	0	0	0	0	0	2.2
	16x	4743	79.72%	0	0	0	0	0	0	4.6
b19	4x	3232	80.91%	0	0	0	0	0	0	3.8
	8x	4000	80.88%	0	0	0	0	0	0	6.6
	16x	5953	80.65%	0	0	0	0	0	0	10.9
p77k	4x	3544	90.98%	440	0	0	227	0	0	19.6
	8x	3717	90.89%	375	4	0	164	0	0	20.1
	16x	3685	90.28%	394	129	2.71	212	2	0	38.4
p81k	4x	2637	88.57%	1007	83	6.75	15	0	0	8.9
	8x	4331	88.41%	1492	223	8.4	44	3	0.02	15.1
	16x	6828	87.91%	2695	588	11.28	98	12	0.28	33.5
p89k	4x	6439	80.49%	1161	0	0	21	0	0	14.7
	8x	6879	80.51%	1158	1	0	26	0	0	16.4
	16x	7352	80.46%	1145	17	0.34	18	0	0	21.8
p100k	4x	4138	96.56%	1993	2	0	502	0	0	27.9
	8x	4251	96.51%	1695	17	0.35	487	0	0	32.5
	16x	4390	96.46%	1988	64	1.66	471	0	0	100.8
p259k	4x	4585	75.59%	2956	350	7.48	979	0	0	257.8
	8x	4268	75.64%	2801	1399	24.04	975	3	0	259.2
	16x	4417	75.64%	2963	–	–	937	44	1.22	745.4
p295k	4x	18344	93.99%	0	0	0	0	0	0	35.4
	8x	18636	93.98%	0	0	0	0	0	0	35.9
	16x	18662	94.00%	0	0	0	0	0	0	72.0
p330k	4x	13951	98.18%	10584	–	–	639	0	0	303.9
	8x	14248	98.20%	10756	–	–	651	0	0	321.6
	16x	14603	98.17%	11072	–	–	657	0	0	572.3
p418k	4x	5726	94.58%	3886	0	0	11	0	0	120.4
	8x	5877	94.57%	3856	0	0	5	0	0	121.8
	16x	6070	94.57%	3569	2	0.03	0	0	0	44.9
p533k	4x	9096	74.68%	4851	–	–	476	0	0	464.5
	8x	8952	74.71%	5072	–	–	463	0	0	367.9
	16x	9278	74.71%	5125	–	–	529	0	0	789.3

[7] L. Whetsel, "Adapting Scan Architectures for Low Power Operation," *Proc. Int'l. Test Conf.*, pp., 863-872, 2000.

[8] R. Sankaralingam, R. Oruganti, and N. Touba, "Reducing Power Dissipation during Test Using Scan Chain Disable," *Proc. VLSI Test Symposium*, pp. 319-324, 2001.

[9] R. Sankaralingam and N. A. Touba, "Controlling Peak Power during Scan Testing," *Proc. VLSI Test Symposium*, pp. 153-159, 2002.

[10] D. Czysz, M. Kassab, X. Lin, G. Mrugalski, J. Rajski, J. Tyszer, "Low Power Scan Shift and Capture in the EDT Environment," *Proc. Int'l Test Conf.*, Paper 13.2, 2008.

[11] K. Agarwal, S. Vooka, S. Ravi, R. Parekhji, A. S. Gill, "Power Analysis and Reduction Techniques for Transition Fault Testing," *Proc. Asian Test Conf.*, pp. 403-408, 2008.

[12] X.Wen, H. Yamashita, S. Kajihara, L.-T. Wang, K. Saluja, and K.Kinoshita, "On Low-Capture-Power Test Generation for Scan Testing," *Proc. VLSI Test Symp.*, pp. 265-270, 2005.

[13] S. Remersaro, et al., "Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs," *Proc. Int'l Test Conf.*, Paper 32.2, 2006.

[14] J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, "Embedded Deterministic Test," *IEEE Trans. on CAD*, Vol. 23, No. 5, pp. 776-792, May 2004.

[15] M. –F. Wu, J. –L. Huang, X. Wen, K. Miyase, "Power Supply Noise Reduction for At-speed Scan Testing in Linear-Decompression Environment," *IEEE Trans. on CAD*, Vol. 28, No. 11, pp. 1767-1776, 2009.

[16] N. A. Touba, "Survey of Test Vector Compression Techniques," *IEEE Design & Test of Computers*, vol. 23, pp. 294-303, July-Aug. 2006.

[17] C.-Y. Liang, M.-F. Wu, J.-L. Huang, "Power Supply Noise Reduction in Broadcast-Based Compression Environment for At-Speed Scan Testing," *Proc. IEEE Asian Test Symp.*, pp. 361-366, 2010.

[18] I. Hamzaoglu and J.H. Patel, "Reducing test application time for full-scan embedded cores," *Proc. 29th Int. Symp. On Fault-Tolerant Computing (FTCS-29)*, pp.260-267, 1999.

[19] 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. ICCAD*, pp. 97-104, 2009.

[20] K. Noda, et al., "Power and noise aware test using preliminary estimation," *Proc. VLSI Design, Automation and Test*, pp. 323-326, 2009.

[21] K. Miyase and S. Kajihara, "XID: Don't Care Identification of Test Patterns for Combinational Circuits," *IEEE Trans.CAD*, Vol. 23(2), pp. 321-326, 2004.

[22] N. Een and N. Sörensson, "An Extensible SAT Solver," *SAT*, 2003.

[23] X. Wen, et al., "Power-Aware Test Generation with Guaranteed Launch Safety for At-Speed Scan Testing", *Proc. IEEE VTS*, pp. 166-171, 2011.