

# Scan Test Planning for Power Reduction

Michael E. Imhof, Christian G. Zoellin,  
Hans-Joachim Wunderlich

Institut fuer Technische Informatik  
Universitaet Stuttgart  
Pfaffenwaldring 47, D-70569 Stuttgart, Germany  
email: {imhof, zoellin, wu}@iti.uni-stuttgart.de

Nicolas Maeding, Jens Leenstra

IBM Deutschland Entwicklung  
Schoenaicherstr. 220, D-71032 Boeblingen, Germany  
email: {nmaeding, leenstra}@de.ibm.com

**Abstract**—Many STUMPS architectures found in current chip designs allow disabling of individual scan chains for debug and diagnosis. In a recent paper it has been shown that this feature can be used for reducing the power consumption during test. Here, we present an efficient algorithm for the automated generation of a test plan that keeps fault coverage as well as test time, while significantly reducing the amount of wasted energy. A fault isolation table, which is usually used for diagnosis and debug, is employed to accurately determine scan chains that can be disabled. The algorithm was successfully applied to large industrial circuits and identifies a very large amount of excess pattern shift activity.

**Categories and Subject Descriptors**—B.8.1 [Hardware]: Performance and Reliability - Reliability, Testing and Fault-Tolerance

**General Terms**—Algorithms, Reliability

**Keywords**—Test planning, power during test

## 1 INTRODUCTION

During test, the switching activity and consequently the average dynamic power consumption of integrated systems is higher by almost a magnitude than power consumption during functional mode [1]. In addition to the increasing static power consumption, this effect has to be taken into account to avoid any impact on both yield and reliability [2], [3]. The commonplace solutions for high volume manufacturing test include reduction of shift speed, circuit partitioning and dedicated cooling equipment. These solutions incur high cost and may also affect the quality of the test. For systems-on-a-chip, test scheduling and test planning strategies were proposed to implement an efficient test of all modules while keeping the maximum power budget [1], [4], [5].

For scan based testing, a plethora of techniques has been proposed to reduce the switching activity during pattern shifting and capture. They include using special types of flip-

flops which suppress output toggling during shift, masking patterns which do not contribute to fault coverage, and gating the shift clock during useless cycles [6], [7], [8]. Scan path segmentation [9] and reduced shift clock frequency also reduce switching without sacrificing fault coverage [10]. Modifications of the scan chains such as reordering or insertion of logic gates have the same goal but may exhibit high design, area and routing overhead [11], [12]. In scan based testing, usually multiple scan chains are employed for implementing a built-in self-test (BIST), embedded deterministic testing or external test. If during certain times only a subset of these scan chains are enabled, significant power savings may be derived [13], a similar strategy has been proposed for testing the Cell Broadband Engine™ (Cell Processor) [14]. Scan chain disabling is independent of the aforementioned power saving techniques and can be combined with them. However, the underlying test planning problem is more complex than SoC test planning, as fault coverage, test time and test power have to be taken into account at the same time. Whereas IP cores can be tested independent of each other, scan chains usually have to be activated in groups for targeting certain faults.

The scheduling technique presented in [14] uses rather coarse information derived at the structural level for generating a suboptimal schedule. In contrast to this, the present paper shows that significantly better solutions can be obtained, if additional information is used that is available but often hidden in the EDA tools for test and diagnosis. The test architecture under consideration is based on STUMPS (Self-Test Using MISR and Parallel Shift Register Sequence Generator, Figure 1) [15] with reseeding [16], [17] as used in the Cell processor. To increase the diagnostic capabilities of such a structure, it has been proposed to allow for masking or disabling certain sets of scan chains [18], [19]. In the Cell processor, disabling information is stored on a seed by seed basis in a register *scanena*.

Diagnosis is usually performed by using a so called fault isolation table or response dictionary [20], [21]. For each fault, we can extract a number of patterns and corresponding flip-flops from the fault isolation table which tells when and where the fault is observable.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2007, June 4-8, 2007, San Diego, California, USA  
Copyright 2007 ACM 978-1-59593-627-1/07/0006...5.00

In the rest of the paper it is shown how this information is used for generating a power optimized test plan. We give a short overview of the basic BIST architecture and the required extension in Section 2. In Section 3, we present a formal definition of the problem, a mapping to the set covering problem and a specific algorithm that can determine an optimized solution in adequate time. This is followed by experimental results, which show that the proposed method is efficient and scalable for the popular benchmark circuits as well as large industrial circuits.

## 2 SELF-TEST WITH PARALLEL SCAN CHAINS

The STUMPS configuration (Figure 1) was proposed in [15] and is the most widespread structure for logic BIST [22], [23] today. Test patterns are generated by a pseudo-random pattern generator (PRPG) and loaded in parallel into multiple scan chains. The PRPG consists of a linear-feedback shift register (LFSR), an XOR network as a phase shifter and a weighting logic to control the distribution of ones and zeros in the test patterns [24]. The circuit responses are evaluated by a multiple-input signature register (MISR), for which a dedicated masking logic allows to mask defective scan chains or those containing unknown states for diagnostic purposes.

Extensions of this self-test structure allow for complete deactivation of single scan chains and the corresponding clock tree [25]. For example, the Cell processor uses multiple test registers (Figure 1), through which automated test equipment has access to the values of the LFSR (*seed*), the weighting (*weight*), the masking (*mask*), the signature (*signature*) and the scan chain activation (*scanena*). A detailed description of the self-test environment of the Cell processor was presented

in [14], and similar features are implemented in other large industrial circuits.

## 3 COMPUTING AN OPTIMIZED TEST PLAN

The goal of determining an optimized test plan is the detection of a given set of faults with minimized power dissipation. For every seed of the LFSR, a set of scan chains is determined, which can be switched off without affecting the fault coverage.

A test block  $b$  is a tuple  $(s, SC_b)$  consisting of seeds for the pattern generator and a set of activated scan chains  $SC_b \subset SC$  called configuration. A seed corresponds to an associated test set of fixed size  $N$ . A fault set  $F_b$  can be determined for every test block  $b$  which contains all faults that can be detected by this block. We are looking for a set  $B$  of blocks, which detects the complete set of faults  $F$  with minimal power dissipation.

A fault  $f$  can be detected by different configurations, and there may be several blocks for one seed. They only differ in their configuration and may cover different fault sets. The block  $(s, SC_b)$  is called minimal regarding a fault  $f$ , if  $f$  would not be detected by the block if any scan chain is removed from  $SC$ .

The cost of a covering is an estimation of its energy dissipation, which is determined by summing up for every seed the number of active chains. A more fine grained estimation may be obtained by computing the weighted switching activity of each block during shift and capture. This extension is straight forward but computationally very expensive. For large circuits with balanced scan chains the additional costs do not provide significant gains of estimation accuracy. Let  $S_B$  be the set of seeds from  $B$ . Then for every seed  $s \in S_B$ ,  $B_s$  is

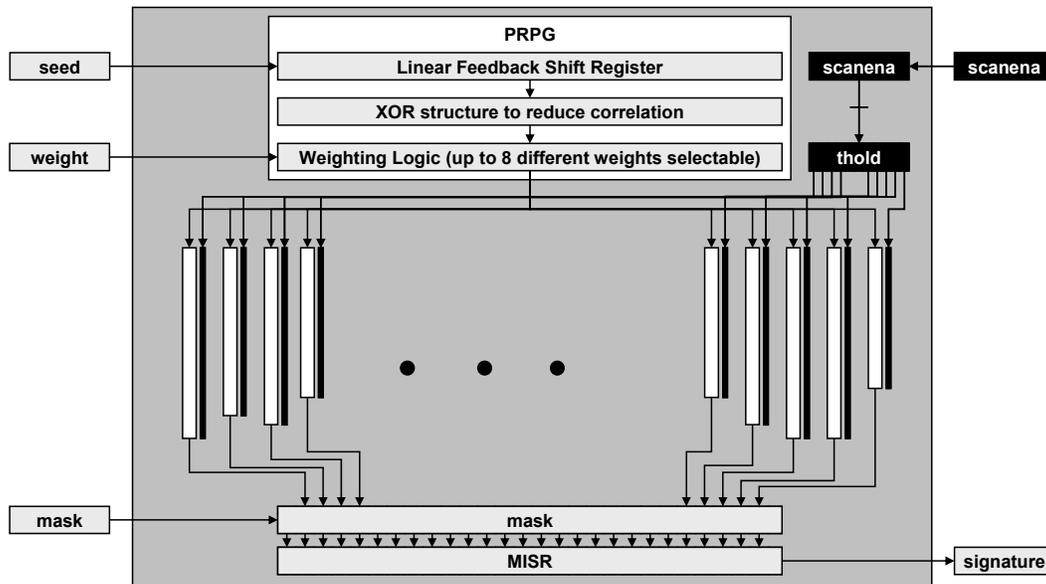


Fig. 1. STUMPS BIST architecture implemented in the Cell processor

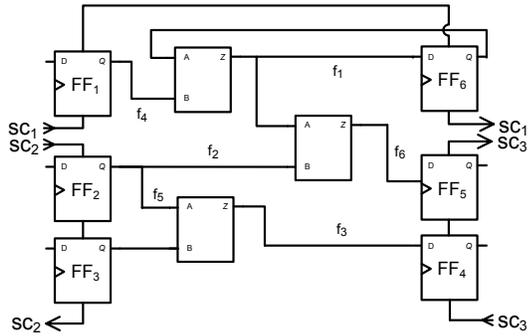


Fig. 2. Example: We assume  $S = \{s_1, s_2, s_3\}$ ,  $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$ .

the associated set of blocks. The cost of a set of blocks can now be estimated as

$$cost(B) = \sum_{s \in S_B} \left| \bigcup_{(s, SC_b) \in B_s} SC_b \right|.$$

If the scan chains are not balanced, weights are associated with each chain. For industrial sized circuits, the determination of a global optimum is not feasible, as solving the set covering problem and the required fault simulations are computationally expensive.

### 3.1 Optimization Algorithm

Every block  $b = (s, SC_b)$  determines a set of faults  $F_b$  detected while executing the block and found in the fault isolation table. If  $F_b$  contains faults that cannot be detected by any other block, then  $b$  is part of the optimal solution and is called essential block. We get  $B_0 := \{b | b \text{ essential}\}$  as an intermediate result and only have to cover the faults in  $F_0 = F \setminus \bigcup_{b \in B_0} F_b$ .

For the remaining faults  $F_0$ , the complexity is further reduced by a „divide and conquer“ approach. For this purpose, the set of faults is partitioned into three classes:

- 1) Hard faults are faults that can only be detected by one seed out of  $S$ . The corresponding seeds are called essential.
- 2) Difficult faults can only be detected by a number of seeds, which is below a user defined constant  $lim$ .
- 3) All other faults are easy to detect.

Those three classes are tackled with different methods and heuristics.

**3.1.1 Hard Faults:** For every hard fault  $f \in F_0$  with corresponding seed  $s_f$ , the function  $c(f, s_f) = \{SC_b | f \text{ is detected by block}(s_f, SC_b)\}$  defines the sets of scan chains and consequently the set of minimal blocks  $B_f = \{(s_f, SC_b) | SC_b \in c(f, s_f)\}$ .

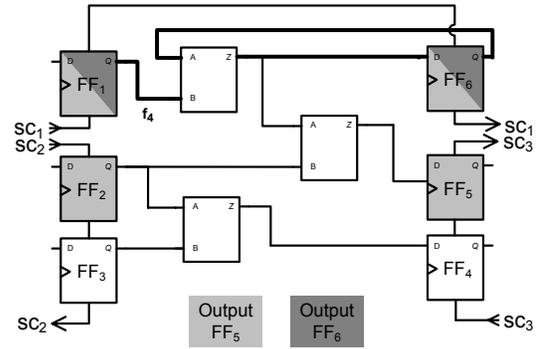


Fig. 3. Example: Let  $f_4$  be a hard fault detected by seed  $s_1$  either in  $FF_5$  or in  $FF_6$ .  $c(f_4, s_1) = \{\{sc_1\}, \{sc_1, sc_2, sc_3\}\}$ .  $B_{f_4} = \{(s_1, \{sc_1\}), (s_1, \{sc_1, sc_2, sc_3\})\}$ . Solution of the covering problem  $B_1 = \{(s_1, \{sc_1\})\}$  has  $cost(B_1 \cup B_0) = 1$ .  $f_1$  is detected in fault simulation of  $B_1 \cup B_0$ , which now detects all faults with  $cost(B_1 \cup B_0) = 1$ .

Since a single output flip-flop is sufficient for detecting a fault, each set in  $c(f, s_f)$  contains just one output scan chain and a minimal number of input chains. For all the patterns generated by a seed  $s_f$ , the output information is found in the fault isolation table. Afterwards, for every flip-flop the flip-flops from the corresponding input cone are collected and the set of required scan chains is determined.

For every block  $b \in \bigcup_{f \text{ hard}} B_f$ , the set  $F_b$  of all the faults, which are detected by  $b$  is determined. Because the number of blocks is small, a branch-and-bound method [26] can be used to find a subset  $B_1 \subset \bigcup_{f \text{ hard}} B_f$ , such that  $\bigcup_{b \in B_1} F_b$  covers all the hard faults from  $F_0$  and  $cost(B_0 \cup B_1)$  is minimal. The remaining set of faults is  $F_1 = F_0 \setminus \bigcup_{b \in B_1} F_b$ . (Example see Figure 3.)

**3.1.2 Difficult Faults:** The next class contains difficult faults that can be detected by a number of seeds, below a fixed limit  $lim$ . To restrict the complexity, only one detecting configuration is considered per seed.

Hence, the function  $c(f, s_f)$  that provides sets of scan chains to be activated for fault  $f$  is calculated independent of the seed  $s_f$ . The used function  $c(f, s_f) = c_{wc}(f)$  delivers a superset of the necessary flip-flops and uses the „Support Region“ [27] of a fault  $f$ . It contains all flip-flops of the output cone of  $f$ , as well as all flip-flops that are part of the input cones of these output flip-flops. This approximation is independent of the generated test set. Again,  $B_f = \{(s_f, c_{wc}(f)) | s_f \text{ seed for } f\}$  is the corresponding set of blocks and a subset  $B_2 \subset \bigcup_{f \in F_{lim}} B_f$  is generated so that  $\bigcup_{b \in B_2} F_b$  covers all difficult faults  $F_{lim}$  from  $F_1$  and  $cost(B_0 \cup B_1 \cup B_2)$  is minimal. The remaining set of faults  $F_2 = F_1 \setminus \bigcup_{b \in B_2} F_b$  only contains easy to detect faults and will be very small or even empty, since they may be already detected in the previous steps. (Example see Figure 4.)

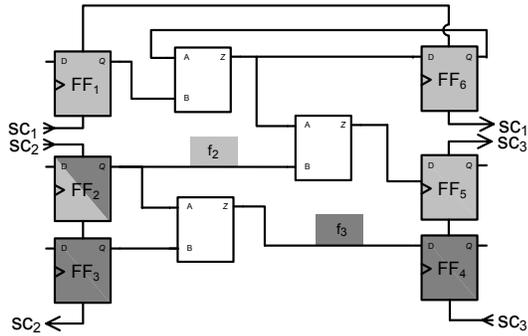


Fig. 4. Example: Let  $lim$  be 2.  $F_{lim} = \{f_2, f_3\}$ .  $S_{f_2} = \{s_2, s_3\}$ .  $S_{f_3} = \{s_1, s_3\}$ .  $c_{wc}(f_2) = \{sc_1, sc_2, sc_3\}$ ,  $B_{f_2} = \{(s_2, \{sc_1, sc_2, sc_3\}), (s_3, \{sc_1, sc_2, sc_3\})\}$ .  $c_{wc}(f_3) = \{sc_2, sc_3\}$ ,  $B_{f_3} = \{(s_1, \{sc_2, sc_3\}), (s_3, \{sc_2, sc_3\})\}$ . Optimal solution  $B_2 = \{(s_3, \{sc_1, sc_2, sc_3\}), (s_3, \{sc_2, sc_3\})\}$  has  $cost(B_2 \cup B_1 \cup B_0) = 4$ .  $f_5, f_6$  are detected in fault simulation of  $B_2 \cup B_1 \cup B_0$ , which now detects all faults with  $cost(B_2 \cup B_1 \cup B_0) = 4$ .

**3.1.3 Other Faults:** To cover all the remaining faults we re-execute step 2 with  $lim = \infty$ , so that blocks for all remaining faults are generated. As many of those faults are detected by a very large set of seeds, the space for the optimization is very large as well. Hence, the optimization with the branch-and-bound method is aborted as soon as the time for finding the next (improved) intermediate solution exceeds a certain time limit. This affects the result only marginally.

## 4 RESULTS

All described steps and methods were implemented in Java as part of an in-house electronic design automation tool and experiments were conducted for a number of benchmark circuits.

### 4.1 Benchmarks and industrial circuits

For evaluation of the presented method, circuit models from the following sources were used:

- International Symposium on Circuits and Systems (ISCAS89)
- International Test Conference (ITC99)
- Industrial circuits from NXP
- Processor core of the Cell processor

The circuits from ISCAS89 (s38417 and s38584) and ITC99 (b17, b18 and b19) are the largest circuits from each collection of benchmark circuits. They do not contain any kind of design for test (DFT) and were extended with the required BIST architecture. The flip-flops of each circuit were arranged into 32 parallel scan chains.

The circuits provided by NXP (p286k, p330k, p388k, p418k and p951k) already contained DFT with parallel scan chains and are several times larger than the circuits from ISCAS89 and ITC99. Moreover, they represent the typical properties of industrial circuits, namely shorter paths and smaller output

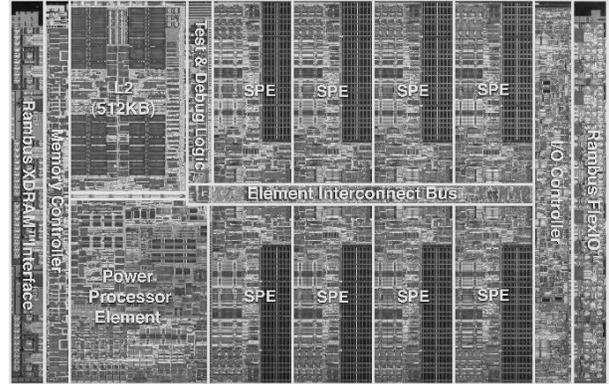


Fig. 5. Die photo of the Cell-Processor

cones as a consequence of the stronger optimization for high clock rates and low area.

As an example for the application of the presented method to a circuit with partial scan, the current implementation of the Cell processor (Figure 5) was used that consists of about 250 million transistors on a  $235\text{ mm}^2$  die. Its top-level self-test architecture consists of 15 self-test domains (so called BIST-satellites), each with its own STUMPS instance [28]. A detailed description of the Cell processor and especially its design for test can be found in [14]. The Synergistic Processing Elements (SPE) was chosen as a representative BIST-satellite, as 70% of the chip area is covered by the 8 identical SPEs.

The characteristics relevant for the test of the SPE are:

- 1.8 million logic gates, 7 million transistors in the logic, 14 million transistors in memory arrays
- 150,000 flip-flops
- 82,500 flip-flops arranged in 32 STUMPS channels
- Memory arrays are not part of the logic BIST and are covered by a separate self-test

### 4.2 Experiments

This section presents results for various combinations of seeds and numbers of patterns per seed. The number of randomly chosen seeds is 200, and per seed a fixed number of 512 or 1024 patterns is generated by the PRPG. Hence the total amount of applied test patterns varies between 102,400 and 204,800. For step 2 of the algorithm, the parameter  $lim$  was set to a value of 3. The first group of columns of Table 1 show the circuit name as well as the number of stuck-at faults and the number of faults detected by the chosen seeds.

In the second column group, reference values for the method described in [14] are given, whereby the cost of the final result ( $cost()$ ) as well as the overall saving ( $\% Red.$ ) is given. The cost function  $cost()$  of a covering is determined by summing up the number of active chains for every seed. The same sets of seeds are used for the method from [14] and for the algorithm in this article.

200,512	# Faults		[14]		Hard Faults		FSIM	Difficult + Hard F.	FSIM	All Faults	
	200,1024	All	Detected	cost()	% Red.	# Faults	cost()	# Faults	cost()	# Faults	cost()
s38417	32320	31144	2723	57.44	555	1171	30330	1760	31023	2081	67.48
		31634	2809	56.10	467	1207	30632	1917	31347	2496	60.99
s38584	38358	36370	1852	71.05	45	208	21458	568	29570	1384	78.36
		36395	1489	76.73	15	94	16502	345	24896	1276	80.05
b17	81330	68598	4097	35.97	3263	2750	68445	3059	68570	3143	50.89
		70256	4379	31.57	2148	2678	70056	3116	70229	3210	49.84
b18	277976	234852	5431	15.14	9754	3957	234464	4348	234780	4418	30.96
		239652	5488	14.25	6080	3915	239321	4343	239601	4406	31.15
b19	560696	468017	5660	11.55	20460	4693	467639	4933	467956	4955	22.57
		479119	5807	9.26	13135	4648	478591	5039	479044	5064	20.87
p286k	648044	605342	10698	2.75	7874	9970	598746	10071	598792	10093	8.25
		609701	10696	2.76	6171	9814	603119	9906	603162	9918	9.84
p330k	547808	488889	9772	23.65	4066	3790	471817	8384	479429	8932	30.21
		491305	9384	26.68	3040	6968	480774	7962	481874	8550	33.19
p388k	856678	836398	6233	37.66	3656	3857	794256	4688	825105	5450	45.49
		838914	5947	40.52	3176	3532	799977	4408	817713	5100	49.00
p418k	688808	632941	9400	26.55	9477	6982	597112	7892	614594	8470	33.82
		638702	9050	29.29	9005	6364	600086	7514	630422	8088	36.81
p951k	1590490	1538987	10924	33.39	9621	8553	1443351	9574	1462481	10335	36.97
		1544946	10599	35.37	8964	8246	1468258	9115	1468258	10104	38.38
SPE	1065190	903645	3388	47.06	2207	1323	897242	1917	900753	2569	59.85
		904317	3180	50.32	2081	1251	898322	1721	901870	2340	63.43

Table 1

200 SEEDS, 512 RESPECTIVELY 1024 PATTERNS

The last column group shows the results achieved with the presented algorithm. According to the presented optimization algorithm, it is split into several intermediate results. For hard faults, it lists their absolute quantity as well as the cost of the generated blocks  $cost(B_1 \cup B_0)$ . The column *FSIM* provides information on the number of faults detected by those blocks. For the difficult faults, the result of the cost function  $cost(B_2 \cup B_1 \cup B_0)$  after optimization is followed by the coverage after fault simulation of the intermediate result. Finally, the cost needed for the final result and the reached saving (% *Red.*) is presented.

The algorithm gives consistently better results than the one introduced in [14] for the complete bandwidth of circuit types. Especially for the circuits b18, b19 and p286k a drastic

improvement is achieved. Differences to the results for the Cell SPE published in [14] are caused by the use of different models and EDA software. For circuits with a large number of hard faults, additional degrees of freedom in the subsequent optimization steps cannot be fully exploited. For example, when considering p951k at  $200 * 512$  patterns, already 83% of the cost is required in order to detect the hard faults (for p286k this is even 99%). It is obvious, that the achieved reduction in power dissipation is dependent on the parameters „number of seeds“ and „patterns per seed“.

Therefore, in Figure 6, the number of seeds is varied for the benchmarks from ISCAS and ITC, while the overall pattern count is kept constant at 102,400. It is shown that through finer granularity, the reduction of the power dissipation can be seriously improved.

In Figure 7, we show the details of a single test plan and plot the number of activated flip-flops per seed. It can be seen that with the method used in [14], no or only few scan chains can be deactivated during the first seeds. In contrast, the test plan computed by the presented algorithm is not affected by such a bias. The test plan can be further improved by reordering seeds (together with their corresponding configurations) in order to adapt the power dissipation to a given envelope. For example, for an envelope that allows for high power dissipation in the beginning of the test, all seeds with high activity would be executed first.

## 5 CONCLUSION

Scan chain disabling structures are found in many circuits to support debug and diagnosis. These structures can be employed for power reduction during test. An efficient test

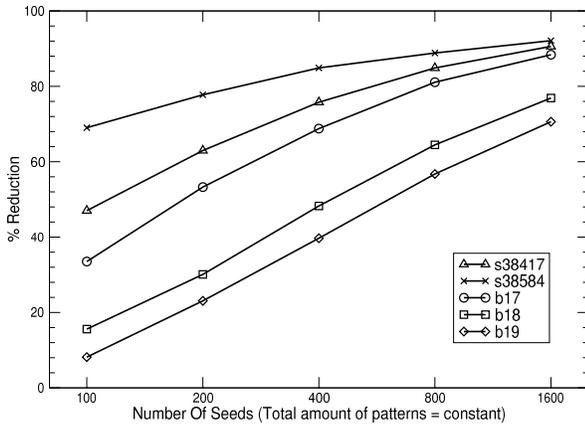


Fig. 6. Dependency on the number of seeds for ISCAS and ITC

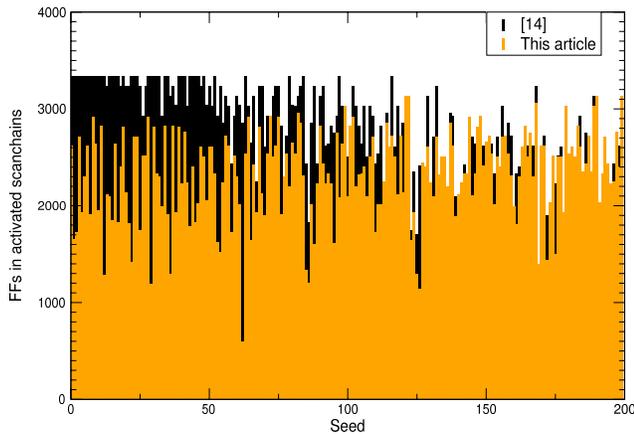


Fig. 7. Solution for b18, 200 seeds, 512 patterns/seed

planning algorithm has been developed which is able to handle large circuits like the Cell processor. The obtained savings of average power consumption during test range from 8% to 80% and exceed the results of known solutions by far.

#### ACKNOWLEDGMENT

This work was supported by the IBM CAS project „Improved Testing of VLSI Chips with Power Constraints“, as well as by the „Deutsche Forschungsgemeinschaft“ within the DFG project „Realtest“ Wu245/5-1. The circuits from NXP were provided in the context of „Realtest“.

Cell Broadband Engine is a trademark of Sony Computer Entertainment Inc.

#### REFERENCES

- [1] Y. Zorian, “A distributed BIST control scheme for complex VLSI devices,” in *Proceedings of the 11th IEEE VLSI Test Symposium (VTS)*, 1993, pp. 4–9.
- [2] C. F. Hawkins and J. Segura, “Test and reliability: Partners in ic manufacturing, part 1,” *IEEE Design & Test of Computers*, vol. 16, no. 3, pp. 64–71, 1999.
- [3] C. F. Hawkins, J. Segura, J. M. Soden, and T. Dellin, “Test and reliability: Partners in ic manufacturing, part 2,” *IEEE Design & Test of Computers*, vol. 16, no. 4, pp. 66–73, 1999.
- [4] Y. Huang, S. M. Reddy, W.-T. Cheng, P. Reuter, N. Mukherjee, C.-C. Tsai, O. Samman, and Y. Zaidan, “Optimal core wrapper width selection and soc test scheduling based on 3-d bin packing algorithm,” in *Proceedings IEEE International Test Conference 2002, Baltimore, MD, USA, October 7-10, 2002*, 2002, pp. 74–82.
- [5] N. Nicolici and B. M. Al-Hashimi, “Power-conscious test synthesis and scheduling,” *IEEE Design & Test of Computers*, vol. 20, no. 4, pp. 48–55, 2003.
- [6] P. Girard, “Survey of low-power testing of VLSI circuits,” *Design & Test of Computers, IEEE*, vol. 19, no. 3, pp. 80–90, 2002.
- [7] S. Gerstendoerfer and H.-J. Wunderlich, “Minimized power consumption for scan-based BIST,” in *Proceedings IEEE International Test Conference 1999, Atlantic City, NJ, USA, 27-30 September 1999*, 1999, pp. 77–84.
- [8] S. Manich, A. Gabarro, M. Lopez, and J. Figueras, “Low power BIST by filtering non-detecting vectors,” *Journal of Electronic Testing Theory and Applications (JETTA)*, vol. 16, no. 3, pp. 193–202, 2000.
- [9] L. Whetsel, “Adapting scan architectures for low power operation,” in *Proceedings IEEE International Test Conference 2000, Atlantic City, NJ, USA, 2000*, pp. 863–872.

- [10] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, and H.-J. Wunderlich, “A modified clock scheme for a low power bist test pattern generator,” in *19th IEEE VLSI Test Symposium (VTS 2001), Test and Diagnosis in a Nanometric World, 29 April - 3 May 2001, Marina Del Rey, CA, USA, 2001*, pp. 306–311.
- [11] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, “Techniques for minimizing power dissipation in scan and combinational circuits during test application,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 12, pp. 1325–1333, 1998.
- [12] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, and A. Virazel, “Design of routing-constrained low power scan chains,” in *2004 Design, Automation and Test in Europe Conference and Exposition (DATE 2004), 16-20 February 2004, Paris, France, 2004*, pp. 62–67.
- [13] R. Sankaralingam, N. A. Touba, and B. Pouya, “Reducing power dissipation during test using scan chain disable,” in *19th IEEE VLSI Test Symposium (VTS 2001), 29 April - 3 May 2001, Marina Del Rey, CA, USA, 2001*, pp. 319–325.
- [14] C. Zoellin, H.-J. Wunderlich, N. Maeding, and J. Leenstra, “Bist power reduction using scan-chain disable in the cell processor,” in *IEEE International Test Conference (ITC 2006), Santa Clara, CA, USA, October 24 - 26, 2006*, 2006.
- [15] P. H. Bardell and W. H. McAnney, “Self-testing of multichip logic modules,” in *Proceedings International Test Conference 1982, Philadelphia, PA, USA, November 1982*, 1982, pp. 200–204.
- [16] B. Koememann, “Lfsr-coded test patterns for scan designs,” in *Proceedings of the European Test Conference, Munich, Germany, 1991*, pp. 237–242.
- [17] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois, “Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers,” *IEEE Trans. on Computers*, vol. 44, no. 2, pp. 223–233, 1995.
- [18] P. Wohl, J. A. Waicukauski, and S. Patel, “Scalable selector architecture for x-tolerant deterministic bist,” in *Proceedings of the 41th Design Automation Conference, DAC 2004, San Diego, CA, USA, June 7-11, 2004*, 2004, pp. 934–939.
- [19] G. Mrugalski, J. Rajski, and J. Tyszer, “Test response compactor with programmable selector,” in *Proceedings of the 43rd Design Automation Conference, DAC 2006, San Francisco, CA, USA, July 24-28, 2006*, 2006, pp. 1089–1094.
- [20] B. Chess and T. Larrabee, “Creating small fault dictionaries,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 18, no. 3, pp. 346–356, 1999.
- [21] C. Liu and K. Chakrabarty, “Design and analysis of compact dictionaries for diagnosis in scan-bist,” *IEEE Trans. VLSI Systems*, vol. 13, no. 8, pp. 979–984, 2005.
- [22] B. L. Keller and T. J. Snethen, “Built-in self-test support in the IBM engineering design system,” *IBM Journal of Research and Development*, vol. 34, no. 2/3, pp. 406–415, 1990.
- [23] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. S. M. Hassan, and J. Rajski, “Logic BIST for large industrial designs: real issues and case studies,” in *Proceedings IEEE International Test Conference 1999, Atlantic City, NJ, USA, 27-30 September 1999*, 1999, pp. 358–367.
- [24] H.-J. Wunderlich, “Multiple distributions for biased random test patterns,” in *Proceedings International Test Conference 1988, Washington, D.C., USA, September 1988*, 1988, pp. 236–244.
- [25] D. Pham, S. Asano, M. Bolliger, M. N. Day, H. P. Hofstee, C. Johns, J. Kahle, A. Kameyama, J. Keaty, Y. Masubuchi, M. Riley, D. Shippy, D. Stasiak, M. Suzuoki, M. Wang, J. Warnock, S. Weitzel, D. Wendel, T. Yamazaki, and K. Yazawa, “The design and implementation of a first-generation Cell processor,” in *International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers, 6-10 Feb. 2005, San Francisco CA, 2005*, pp. 184–185.
- [26] O. Coudert, “On solving covering problems,” in *Proceedings of the 33rd Conference on Design Automation, Las Vegas, Nevada, USA, June 3-7, 1996*, 1996, pp. 197–202.
- [27] I. Hamzaoglu and J. H. Patel, “New techniques for deterministic test pattern generation,” in *16th IEEE VLSI Test Symposium (VTS '98), 28 April - 1 May 1998, Princeton, NJ, USA, 1998*, pp. 446–452.
- [28] M. Riley, L. Bushard, N. Chelstrom, N. Kiryu, and S. Ferguson, “Testability features of the first-generation Cell processor,” in *Proceedings of the IEEE International Test Conference (ITC), 8-10 Nov. 2005, Austin TX, 2005*, p. 6.1.