A Functional Power Evaluation Flow for Defining Test Power Limits during At-Speed Delay Testing

M. Valka A. Bosio L. Dilillo P. Girard
S. Pravossoudovitch A. Virazel
LIRMM – Université de Montpellier II / CNRS
161, rue Ada – 34095 Montpellier Cedex 5, France
Email: <lastname>@lirmm.fr

E. Sanchez M. De Carvalho M. Sonza Reorda
Dipartimento di Automatica e Informatica
Politecnico di Torino, Italy
Email: <lastname>@polito.it

***Abstract*—High power consumption during test may lead to yield loss and premature aging. In particular, excessive peak power during at-speed delay fault testing represents an important issue. In the literature, several techniques have been proposed to reduce peak power consumption during at-speed LOC or LOS delay testing. On the other hand, some experiments have proved that too much test power reduction might lead to test escape and reliability problems. So, in order to avoid any yield loss and test escape due to power issues during test, test power has to map the power consumed during functional mode. In literature, some techniques have been proposed to apply test vectors that mimic functional operation from the switching activity point of view. The process consists of shifting-in a test vector (at low speed) and then applying several successive at-speed clock cycles before capturing the test response.**

In this paper, we propose a novel flow to determine the functional power to be used as test power (upper and lower) limits during at-speed delay testing. This flow is also used for comparison purpose between the above-mentioned test scheme and power consumption during the functional operation mode of a given circuit. The proposed methodology has been validated on an Intel MC8051 microcontroller synthesized in a 65nm industrial technology.

Keywords—At-speed delay fault testing; Power-aware Testing; Functional power

I. INTRODUCTION

Nowadays, electronic products present various issues that become more important with CMOS technology scaling. High operation speed and high frequency are mandatory requests. On the other hand, power consumption is one of the most significant constraints due to large diffusion of portable devices. These needs influence not only the design of devices, but also the choice of appropriate test schemes that have to deal with production yield, test quality and test cost.

Testing for performance, required to catch timing or delay faults, is therefore mandatory, and it is often implemented through at-speed scan testing for logic circuits. At-speed scan testing consists of using a rated (nominal) system clock period between launch and capture for each delay test pattern, while a longer clock period is normally used for scan shifting (load and unload cycles).

In order to test for transition delay faults, two different schemes are used in practice during at-speed scan testing: Launch-off-Shift (LOS) and Launch-off-Capture (LOC) [1].

Although at-speed scan testing is mandatory for high-quality delay fault testing, its applicability is severely challenged by test-induced yield loss, which may occur when a good chip is declared as faulty during at-speed scan testing [2]. Both schemes (LOS and LOC) may suffer from this problem, whose the major cause is Power Supply Noise (PSN), *i.e.*, IR-drop and Ldi/dt events, caused by excessive switching activity (leading to excessive power consumption) during the launch-to-capture cycle [3] of delay testing schemes. In order to deal with this problem, dedicated techniques to reduce the risk of artificial yield loss induced by excessive PSN during at-speed scan testing have been proposed in the literature [1][4]. These techniques are mainly based on test pattern modification or power-aware Design-for-Testability (DfT).

Despite the fact that reduction of test power is mandatory to minimize the risk of yield loss, some experimental results have proved that too much test power reduction might lead to test escape and reliability problems because of the under-stress of the circuit during test [1]. So, in order to avoid any yield loss and test escape due to power issues during test, test power has to map the power consumed during functional mode. To this purpose, the knowledge of functional power for a given CUT is required and may be used as a reference for defining the power consumption (upper and lower) limits during power-aware delay test pattern generation for LOS or LOC.

A solution has been proposed in [8] and [9], where authors introduce a new process to generate test vectors that mimic functional operation from the switching activity point of view. The process consists of shifting-in a test vector (at low speed) and then applying several successive at-speed clock cycles before capturing the test response. These tests are based on a LOC scheme with the insertion of multiple functional cycles between launch and capture operations. The idea behind this process is that after a certain number of clock cycles, the CUT reaches a pseudo-functional state (*i.e.,* a state *similar* to a functional state) which insures that test power will mimic functional power, thus preventing any possible yield loss or test escape due to power issues.

In this paper, our goal is to evaluate the accuracy of the above-mentioned solution presented in [8] and [9]. We propose a methodology in which several power estimation flows are built to compare the power consumption of pseudo-functional patterns (used in [8] and [9]) and LOS patterns (used in a conventional LOS scheme with a single launch-to-capture

1530-1877/11 $26.00 © 2011 IEEE
DOI 10.1109/ETS.2011.21

cycle) with the power consumption of actual functional patterns. For this purpose, we have used the framework presented in [5] that deals with stimuli generation for design validation. In this framework, functional patterns are generated to maximize the switching activity of a given design, which can be used to determine the test power limits during at-speed delay testing. The proposed methodology has been validated on the Intel MC8051 microcontroller synthesized in a 65nm industrial technology. Experimental results show that the peak power consumption of pseudo-functional patterns obtained as in [8] and [9] is about 9 % lower than the maximum functional peak power consumption of the MC8051 microcontroller. The output of this comparative study is a new flow to determine the functional power to be used as test power limits during at-speed scan delay fault testing.

The paper is structured as follows. Section II details each step of the proposed methodology. Section III describes the adopted case study and presents the experimental results. Section IV concludes the paper.

II. THE PROPOSED METHODOLOGY

The methodology proposed in this paper mainly relies on the framework presented in [5]. The goal is still to generate functional stimuli (*i.e.*, test programs) for design validation purpose that maximize power consumption of the design. In the remainder of the paper, we refer to such stimuli as *functional patterns.*

These stimuli are here used for defining test power limits during the application of at-speed delay test patterns. Then, power limits are first compared with the power consumed during the application of the LOS test patterns. We focus on LOS since it is the test scheme providing a higher transition fault coverage but also a higher power consumption compared to LOC [10]. In the remainder of the paper, we refer to such stimuli as *LOS patterns.*

Next, we compare power consumption limits obtained with functional patterns with those obtained from patterns generated as described in [8] and [9]. In the remainder of the paper, we refer to such stimuli as *pseudo-functional patterns*. In order to generate pseudo-functional patterns, we use a LOS-based ATPG program in which many clock cycles are considered between launch and capture.

In the next subsections we detail how the functional stimuli are generated and what are the metrics used to perform the power analysis.

A. Test program generation

In the literature, main research works focus on the generation of methodologies oriented to minimize power consumption of both hardware and software. Conversely, as described in [5] and [6] the authors consider the mirror problem, that is, how to generate test programs oriented to maximize power consumption of the considered processor. In [5] the authors propose a methodology that deals with stimuli generation for design validation. Specifically, the main objective of the test program is not just verifying the circuit functionality, but rather, the robustness of the whole system by applying test programs that maximize power consumption. More intuitively, increasing the power consumption can be seen as a kind of self-burn-in process, where the functionalities of the circuit are tested on extreme power conditions. In fact, exploiting similar approaches, in [6] authors propose a methodology for the automatic generation of stress programs to be used during the reliability characterization process of microprocessors.

In this paper, we exploit the methodology proposed in [6], in order to generate power-aware test programs. The proposed approach exploits the logic simulation of a RTL description of the processor core, integrated in the generation loop. Supported by an evolutionary optimization tool, introduced in [7], syntactically correct assembly programs are generated, forming a population of programs; then, a high-level logic simulator evaluates every test program providing to the evolutionary optimizer a feedback value representing the test program goodness. This value, also known as fitness value, is computed by measuring the switching activity of the RTL signals that correspond to flip-flops in the synthesized version of the processor core. The evolutionary optimizer improves test programs by mimicking the Darwinian concepts of evolution. The higher the fitness value, the better the test program is.

The generation process iterates up to reaching a steady condition, and then, the processor RTL description is substituted with the gate-level one, starting a new generation phase that improves the initial results.

B. Evaluation metrics

Power consumption has been evaluated during the application of functional patterns, LOS patterns and pseudo-functional patterns [8][9].

The power analysis has been done considering two power metrics: cycle average power and peak power consumption. Both cycle average and peak power consumption are measured by means of WSA (Weighted Switching Activity).

In general, cycle average refers to the average of instantaneous power consumed by the device during a specified time window. Peak power refers to the highest power value measured during the same time window. The definition of the time window varies according to the stimuli applied to the device. In the following, we define for each type of applied stimuli the time window and its meaning.

1) Functional patterns

The applied functional patterns correspond to a test program executed by the microprocessor as described in Section II.A. Before the test program execution, the microprocessor has to execute a small piece of Operating System (OS) to ensure the correct initialization of the microprocessor itself.

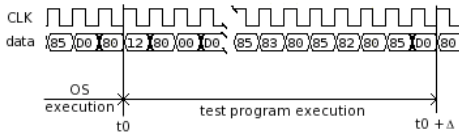

Figure 1 Functional patterns time window

Figure 1 depicts a snapshot of a test program execution waveform. For the sake of readability only the clock signal and the I/O data are reported. After the OS execution, the test program execution starts at time t_0 and ends at time $t_0+\Delta$. For each test program we consider the time window (t_0, $t_0+\Delta$) as the period to estimate the cycle average and the peak power.

2) LOS patterns

Power consumption can be evaluated during the launch and capture cycles of the LOS testing schemes. Launch power is the power consumed during the launch cycle (also called "launch-to-capture" or "test" cycle), which is performed with an at-speed clock cycle. Capture power is the power consumed right after the capture edge, during a time interval also equal to the rated clock period of each experimented circuit. Figure 2 shows the time windows in which these power measures are made.

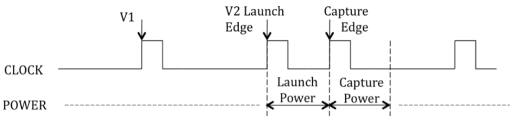

Figure 2 LOS patterns time window

In [10], it has been proved that launch power is greater than capture power. For this reason, in this paper we only consider launch cycle as the time window to estimate the cycle average and the peak power. Both values are evaluated for the whole set of LOS test patterns. Then, the most power-consuming pattern concerning cycle average and peak power is considered.

3) Pseudo-functional patterns

Pseudo-functional patterns are based on a LOS test scheme modified by adding *n* capture cycles. Since in [8][9] the number of capture cycles necessary to reach a pseudo-functional state is not specified, we generate several pseudo-functional patterns having $n = \{5, 10, 50, 100\}$.

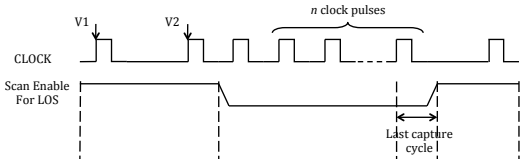

Figure 3 Pseudo-functional patterns time window

Figure 3 shows the *'n capture method'*. This method considers the power consumed in the last capture cycle as a good approximation of the real functional power consumption. Thus, the time window used to measure both the cycle average and the peak power corresponds to the last capture cycle. The calculation of the average and peak power values is repeated for the whole set of test vectors generated by the ATPG tool. Since we are interested only in the critical values of the power, we will take into account only the most power-consuming pseudo-functional patterns, in terms of cycle average and peak power.

III. CASE STUDY

The adopted case study is the Intel microcontroller MCS-51 (MC8051) synthesized under a 65nm industrial technology library. MC8051 represents a classical Harvard architecture non-pipelined CISC architecture, with 8-bit ALU and 8-bit registers. Table I gives the details of the microcontroller synthesis in terms of number of primary inputs/outputs, flip-flops, logical gates and number of transition faults. During the synthesis one scan chain has been added.

TABLE I. MC8051 SYNTHESIS

#Primary Inputs	65
#Primary Outputs	94
#Scan Chains	1
#FFs	578
#Logical Gates	9,451
#Transition Faults	37,752

The MC8051 is embedded into a SoC [11] composed of three cores: (i) the MC8051 microcontroller, (ii) a 64Kx8 bit SRAM memory and (iii) a 16x16 parallel multiplier.

During its mission mode, the microcontroller core reads the program to be executed from an external RAM memory, it communicates with the outside through its parallel port and it drives the multiplier for arithmetic computations. The SoC structure is shown in Figure 4 .

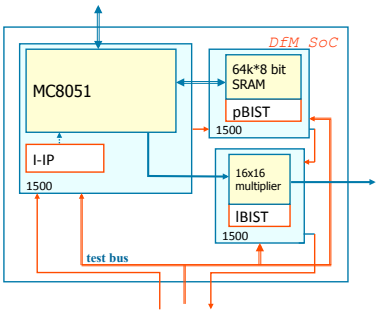

Figure 4 SoC Architecture

In order to estimate power consumption (in terms of both cycle average and peak power) for Functional, LOS and

Pseudo-functional patterns three different flows have been implemented.

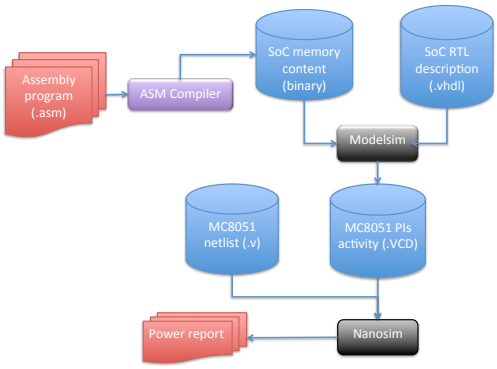

Figure 5 Functional patterns power estimation flow

Figure 5 shows the implemented flow to estimate power consumption for the functional patterns. Each functional pattern corresponds to a generated test program (see Section II.A). The test program is compiled in order to obtain the memory content to be loaded into the SoC. The SoC with the test program loaded in memory is simulated using ModelSim™ [13] and the switching information related to the MC8051 primary inputs is stored in a VCD file. So, in the resulting VCD file we have the stimuli applied to the microcontroller when the target test program is executed. The last step of the flow actually estimates the power consumed during the test program execution. Since our goal is to evaluate the power consumed by the microcontroller, we simulate only the MC8051 microcontroller gate level netlist by applying the stimuli stored in the VCD. The tool used to perform such analysis is NanoSim™ [14]. The result is the power report containing both cycle average and peak power evaluated during the time window (t_0, $t_0+\Delta$) as described in Section II.B.1.

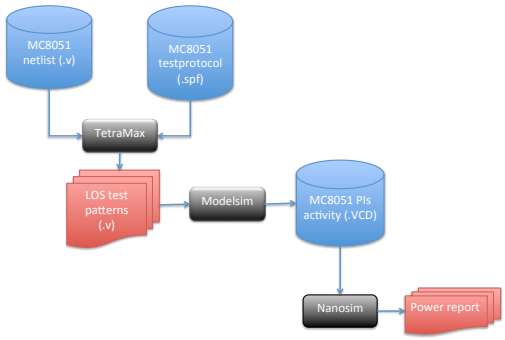

Figure 6 LOS patterns power estimation flow

Figure 6 shows the implemented flow to estimate power consumption for the LOS patterns. The LOS pattern set is generated using TetraMAX™ [12] targeting the transition delay fault model. The required inputs are the MC8051 gate level netlist and the test protocol file (.spf) in which the timing to be applied during the LOS test scheme is described. Note that we used the default setting and random filling options for the ATPG. Static and dynamic compactions were used during test set generation. The number of generated LOS patterns is 1405 and the achieved fault coverage is 79%. The generated LOS test patterns set is then simulated using ModelSim™ [13] and the switching information for the patterns is stored in a VCD file. Finally, the VCD file is used with NanoSim™ [14] to perform the power analyses. The result is the power report containing both cycle average and peak power evaluated during the time window (the launch-to-capture cycle) as described in Section II.B.2.

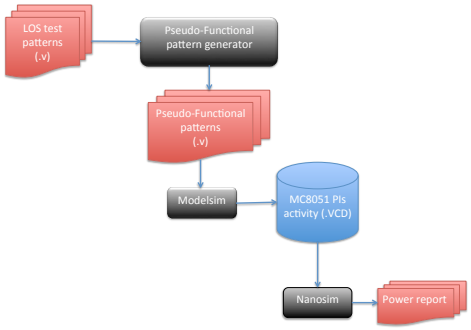

Figure 7 Pseudo-functional patterns power estimation flow

Figure 7 shows the implemented flow to estimate power consumption for the Pseudo-functional patterns. As described in Section II.B.3 Pseudo-functional patterns are based on a LOS test scheme modified by adding *n* captures cycles, where $n = \{5, 10, 50, 100\}$. Each pseudo-functional pattern is then simulated using ModelSim™ [13] and the switching information for the patterns is stored in a VCD file. Finally, the VCD file is used with NanoSim™ [14] to perform the power analyses. The result is the power report containing both cycle average and peak power evaluated during the time window as described in Section II.B.3.

A. Experimental results

This section describes the experimental results. All the experiments were carried our on an Intel Xeon@3.16GHz with 8GB of RAM.

Experiments have been done on a MC8051 description synthesized with an industrial 65nm technology, considering a power supply voltage of 1.2V. Only dynamic switching power due to switching capacitances has been considered (short-circuit power and leakage power consumptions were neglected). This is a valid assumption for the considered technology with the purpose of comparing between Functional, LOS and Pseudo-functional patterns. Similarly, although in deeper technologies the impact of leakage power on the overall power consumption is greater than in the 65nm technology, we may assert that it will not affect the main conclusions of our comparison between Functional, LOS and Pseudo-functional patterns. This forecast is based on the fact that we expect that a higher leakage power will possibly impact the three sets of

patterns in the same manner (as the circuit and numbers of scan FFs are the same, a similar level of leakage currents can be predicted).

Power consumption measured with NanoSim™ during the time window (t_0, $t_0+\Delta$) for the functional patterns is reported in Table II. Results are expressed in milliWatt. In our experiments we generated 30 different functional patterns (i.e., test programs). The second column ("Clock cycles") reports the length of each test program in terms of the number of clock cycles that corresponds to the size of the time window. The value reported for each functional pattern in the "Peak" column represents the highest peak power measured over the considered time window (t_0, $t_0+\Delta$). The value reported for each functional pattern in the "Average" column represents the Average power measured over the considered time window (t_0, $t_0+\Delta$). Note that all measurements represent the overall circuit power consumption, including both logic power (power in the combinational logic) and sequential power (power in scan flip-flops). Clock power (power in the clock tree) is not included.

TABLE II. FUNCTIONAL PATTERNS POWER ESTIMATION

Functional Patterns	Clock cycles	Peak (mW)	Average (mW)
test_program_01	4349	106.25	3.56
test_program_02	4721	106.86	3.88
test_program_03	5697	106.58	4.11
test_program_04	6433	106.24	4.50
test_program_05	6561	106.21	4.58
test_program_06	8613	107.80	5.44
test_program_07	12185	107.51	1.30
test_program_08	14732	107.09	1.34
test_program_09	162567	106.97	7.01
test_program_10	19973	106.03	7.20
test_program_11	20415	108.49	5.14
test_program_12	10851	107.04	4.79
test_program_13	23851	111.29	7.27
test_program_14	27056	107.46	2.23
test_program_15	26980	107.50	2.22
test_program_16	42463	106.24	1.08
test_program_17	42943	107.64	6.53
test_program_18	52113	105.84	8.45
test_program_19	53835	107.35	8.02
test_program_20	71947	109.02	7.69
test_program_21	79892	107.51	2.37
test_program_22	94349	106.11	3.39
test_program_23	277617	110.27	8.34
test_program_24	1048261	106.14	12.61
test_program_25	1050438	119.10	9.41
program_stress_alu	553	107.96	1.30
program_stress_fsm	421	106.73	0.79
program_stress_mem	609	108.48	1.68
program_stress_uP	391	106.79	1.02
program_stress_ram	407	107.78	1.00

Functional patterns considered in Table II have been developed recurring to two different approaches. Both strategies guarantee a comprehensive excitation of the processor core, though they exploit different generation schemes. The first 25 test programs, labeled *test_program_01*, *test_program_02*, etc., were developed by hand, targeting the stuck-at fault coverage of the processor core. These programs achieved about 95% fault coverage on the targeted fault model. Interestingly, the highest peak value was obtained by *test_program_25*, which repeatedly sweeps the addressable memory for exciting the program counter circuitry. On the other hand, patterns labeled *programs_stress_alu*, *fsm*, etc., were developed following the strategy described in section II.A. In this case, functional programs stress mainly the single part of the processor core highlighted in the program name.

TABLE III. LOS AND PSEUDO-FUNCTIONAL PATTERNS POWER ESTIMATION

	Patterns	Peak (mW)	Average (mW)
	LOS	146.93	14.09
Pseudo-functional patterns	*LOS – 5*	108.78	12.15
	LOS - 10	108.89	12.19
	LOS - 50	108.62	11.97
	LOS – 100	108.73	11.63

Table III reports the power consumption for LOS and Pseudo-functional patterns. In the first column ("Patterns") we report LOS patterns and the set of pseudo-functional patterns denoted by the number of capture cycles (5, 10, 50, 100). Results are expressed in milliWatt. The value reported for each functional pattern in the "Peak" column represents the highest peak power measured during the launch-to-capture cycle (see Section II.B.2) for LOS patterns. The peak power of the pseudo-functional patterns is measured during the last capture cycle (see Section II.B.3). The value reported for each pattern in the "Peak" column represents the highest peak power measured over the complete set of patterns. The value reported for each pattern in the "Average" column represents the Average power measured during the launch-to-capture cycle (see Section II.B.2) for LOS patterns. For the pseudo-functional patterns the average is measured during the last capture cycle (see Section II.B.3). As for peak values, the average power is measured over the complete set of patterns. Note that similarly to the experiments carried out on functional patterns, all measurements represent the overall circuit power consumption, including both logic power (power in the combinational logic) and sequential power (power in scan flip-flops). Clock power (power in the clock tree) is not included.

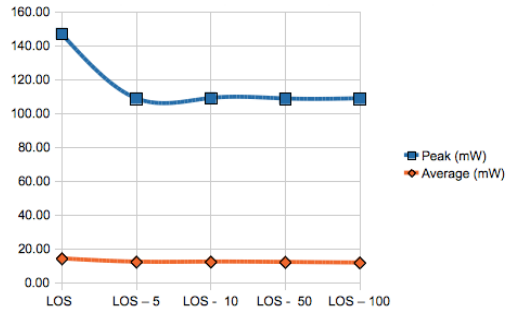

Figure 8 Behavior of LOS and pseudo-functional patterns

The results in Table III are further elaborated in the graph of Figure 8 where both Peak power and cycle Average power are reported.

As predicted in [8] and [9] both peak power and average power reach a steady state value after a certain number of capture cycles. In our experiments, the peak power converges at a stable value after 10 capture cycles, while the average power converges after 5 capture cycles. The peak power convergence value is lower than that during the launch-to-capture cycle for LOS patterns. The reduction of peak power achieved by applying more than 10 capture cycles is about 26% compared to the peak power during the launch-to-capture cycle for LOS patterns. The same conclusions can be drawn for the average power. In this case, the reduction is about 17% compared to the cycle average power during the launch-to-capture cycle for LOS patterns. The trend depicted in Figure 8 confirms the data given in [8] and [9].

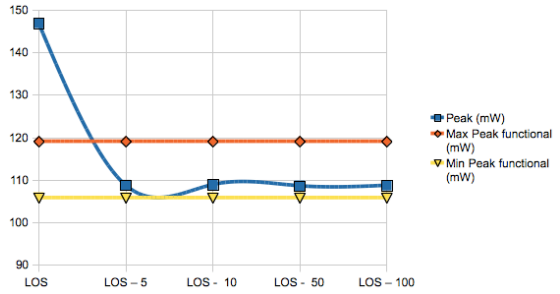

Figure 9 Behavior of Functional and Pseudo-functional patterns

Figure 9 shows a comparison between the functional peak power and the peak power obtained with LOS and Pseudo-functional patterns. In the figure is reported the two bounds (upper and lower) representing respectively the maximum and the minimum functional peak power. From the analysis of data reported in Figure 9 we can make the following observations:

- As reported in the literature [1], the power consumed during test application is higher than the power consumed during functional mode. The gap between LOS peak power (the peak power consumed by LOS patterns during the launch-to-capture cycle) and the maximum functional peak power is about 23.37%. The gap between LOS peak power and the minimum functional peak power is 38.82%.
- The comparison between the functional peak power and the pseudo-functional peak power shows that the latter is not a very accurate prediction of the actual functional peak power (at least in our case study). The pseudo-functional peak power is about 9% lower than the maximum functional peak power. This is also confirmed by the fact that it is really close to the minimum functional peak power (only 2.8% lower). Thus, only considering pseudo-functional peak power for defining test power limits may lead to test escapes due to under stress.

IV. Conclusion and Perspective

In this paper we proposed a novel flow to estimate functional power to be used as power limits during at-speed delay fault test. A first (predictable) conclusion shown by this study is that power consumed during test is higher than power consumed during functional mode. A second (unpredicted) conclusion of this study is that the peak power consumption of pseudo-functional patterns is not so accurate compared to the real functional peak power. Thus, as a final conclusion we have shown that a procedure to generate meaningful functional patterns to estimate the functional power is mandatory to avoid test escape and yield loss.

Our ongoing research activities are devoted to validate the proposed procedure on a larger group of microprocessors (both CISC and RISC) synthesized with several technology libraries.

References

[1] P. Girard, N. Nicolici and X. Wen (editors), Power-Aware Testing and Test Strategies for Low Power Devices, Springer, ISBN 978-1-4419-0927-5, 2009.

[2] J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreeprakash, and M. Hachinger, “A Case Study of IR-Drop in Structured At-Speed Testing”, IEEE Int’l Test Conf., pp. 1098-1104, 2003.

[3] K. Arabi, R. Saleh, and X. Meng, “Power Supply Noise in SoCs: Metrics, Management, and Measurement,” IEEE Design & Test of Computers, vol. 24, no. 3, May-June 2007.

[4] S. Ravi, R. Parekhji, and J. Saxena, “Low Power Test for Nanometer System-on-Chips (SoCs)”, ASP Journal of Low Power Electronics, vol. 4, no. 1, pp. 81-100, April 2008.

[5] A. Calimera, E. Macii, D. Ravotto, E. Sanchez, M. Sonza Reorda, “Generating power-hungry test programs for power-aware validation of pipelined processors,” in Proc. of the 23rd symposium on Integrated circuits and system design, pp 61-66, Septembre 2010.

[6] M. de Carvalho, P. Bernardi, E. Sanchez, M. Sonza Reorda, “An Enhanced Strategy for Functional Stress Pattern Generation for System-on-Chip Reliability Characterization”, [Accepted for publication] IEEE Microprocessor Test and Verification, 2010.

[7] F. Corno, et al. , “Automatic Test Program Generation - a Case Study”, IEEE Design & Test of Computers, 2004, Vol. 21, n. 2, pp. 102-109

[8] liE. K. Moghaddam, J. Rajski, S. M. Reddy, X. Lin, N. Mukherjee, M. Kassab, “Low Capture Power At-Speed Test in EDT Environment,” in Proc. IEEE Int. Test Conf, paper 24-2, November, 2010.

[9] E. K. Moghaddam, J. Rajski, S. M. Reddy, M. Kassab, “At-Speed Scan Test with Low Switching Activity,” in Proc. IEEE VTS Test Symp, pp. 177-182, April, 2010.

[10] F. Wu, L. Dilillo, A. Bosio, P. Girard, S. Pravossoudovitch, A. Virazel, M. Tehranipoor, X. Wen and N. Ahmed, “A comprehesive Analysis of Power Consumption and Transition Fault Coverage for LOS and LOC Testing Schemes”, ASP Journal of Low Power Electronics, Vol.6, N° 2, August 2010.

[11] D. Appello et al. “On the Automation of the Test Flow of Complex SoCs”, IEEE VLSI Test Symposium, pp. 166-171, 2006.

[12] Synopsys Inc., TetraMAX®, User Guide 2010.

[13] MentorGraphics, ModelSim®, http://www.model.com

[14] Synopsys Inc., NanoSim™, User Guide 2009.