STT MRAM-Based PUFs

Elena Ioana Vatajelu, Giorgio Di Natale, Marco Indaco, Paolo Prinetto

Exploring the Impact of Functional Test Programs Re-Used for Power-Aware Testing

A. Touati, A. Bosio, L. Dilillo, P. Girard, A. Virazel
LIRMM-UM2/CNRS
France
<lastname>.lirmm.fr

P. Bernardi, M.Sonza Reorda
Politecnico di Torino
Italy
<lastname>.polito.it

***Abstract*— High power consumption during at-speed delay fault testing may lead to yield loss and premature aging. On the other hand, reducing too much test power might lead to test escape and reliability problems. Thus, to avoid these issues, test power has to map the power consumed during functional mode. Existing works target the generation of functional test programs able to maximize the power consumption in functional mode of microprocessor cores. The obtained power consumption will be used as threshold to tune the power consumed during testing. This paper investigates the impact of re-using such functional test programs for testing purposes. We propose to apply them by exploiting existing DfT architecture to maximize the delay fault coverage. Then, we combine them with the classical at-speed LOC and LOS delay fault testing schemes to further increase the fault coverage. Results show that it is possible to achieve a global test solution able to maximize the delay fault coverage while respecting the functional power budget.**

Keywords—Power Aware Test; Functional and Structural test; microprocessor test; ATPG.

I. INTRODUCTION

Nowadays, electronic products present various issues that become more important with the CMOS technology scaling and the requisite request of both high operation speed and high frequency [1]. Testing for performance, required to catch timing or delay faults, is therefore mandatory and often implemented through at-speed structural scan testing for digital circuits. Considering at-speed scan testing, two different schemes are used in practice: Launch-off-Shift (LOS) and Launch-off-Capture (LOC). They consist 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/unload cycles) [2].

At-speed structural scan testing may lead to excessive power consumption that can either damage the Circuit Under Test (CUT) or lead to yield loss [3]. Reducing the power during testing is a well-known technique but reducing too much the power consumption may lead to test escape phenomena. Therefore, to cope with the above issues, we have to tune the test power depending on the functional power of the device itself [3]. Stated thus, the knowledge of the actual functional power is mandatory. In this work we target the test of microprocessor cores. In this field, several papers target the functional test of microprocessors [4]. Among all of them, we focused on the power-aware test of microprocessor cores. In [5], authors propose a functional test programs generator for microprocessor cores. The generator aims at maximizing the power consumption of the target microprocessor, and hence the generated programs are good candidates to accurately estimate the functional power limits (i.e., to avoid both over- and under-test).

Since generated functional test programs maximize the power consumption, they are definitively characterized by a high switching activity [5]. In other words, they could be also good candidates for delay fault testing. In this work we investigate the impact of re-using available functional test programs for exploring a global test solution that maximizes the delay fault coverage while satisfying the power consumption constraints.

The use of functional stimuli for manufacturing testing has already been investigated in the literature. Authors in [6] propose a fast fault grading approach to order a pool of functional tests depending on their fault coverage. This approach avoids the use of gate-level fault simulation. In [7], authors present a Design-for-Testability (DfT) architecture applied at RT-level to increase the fault coverage of functional tests. Basically, they add test-points to improve the testability of the device during the application of functional test. In [8], authors present a methodology able to concatenate a set of functional tests to maximize the fault coverage while minimizing the test length. The latter is also reduced thanks to a static compaction algorithm. Finally, in [9], authors present an analysis of functional test programs from the verification and test point of view.

In this paper, we started from the functional test programs generated in [5]. These programs maximize the power consumption of the targeted microprocessor core in functional mode. Then, we investigated the impact of re-using such functional test programs for delay fault testing. In particular we show how these functional tests can be applied to improve the transition delay fault coverage. Conversely to [7], we propose to re-use the DfT circuitry already present in the circuit under test, thus avoiding any further modifications of the device. Basically, we intend to map functional test programs into a structural scan testing scheme (i.e., LOC or LOS). The result will be a test scheme applied to the circuit through existing scan chains.

The remaining of the paper is organized as follows. In section II, we introduce the background of this work. Section III presents the methodology describing how to apply functional test programs by re-using existing DfT architecture.

Section IV analyzes the collected results. Finally, section V concludes the paper.

II. BACKGROUND

This section briefly describes the main concepts of the methodology proposed in the paper. As stated in the introduction, we consider a pool of functional test programs automatically generated as described in [5]. Those programs were generated targeting the *Intel MC8051* non-pipelined *CISC* processor, with 8-bit ALU and 8-bit registers, synthesized with a 65nm industrial technology. All the details about the microprocessor architecture are available in [5]. Table I gives the main characteristics of the synthetized circuit (i.e., number of Gates, Flip-Flops, Scan chains, Primary inputs, Primary outputs and Transition faults).

TABLE I. MC8051 GATE-LEVEL CHARACTERISTICS

#Gates	#FFs	#ScanChains	#PIs	#POs	#TFs
10343	578	1	65	94	31492

A total number of 117 functional test programs have been generated. Fig. 1 depicts the generated programs on the X-axis and the corresponding peak power (in µW) on the Y-axis. By construction, the peak power increases at each new test program, leading to a very effective functional test programs set for maximizing the power consumption of the microprocessor.

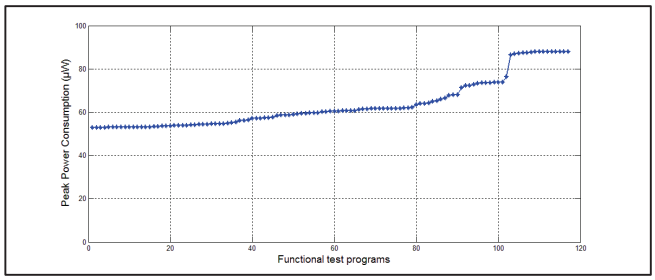

Fig. 1. Functional test programs peak-power consumption

In this paper we performed a comprehensive analysis of such programs. First of all, we considered the achieved transition fault coverage. Basically, we fault simulated each functional test program to obtain the transition fault coverage. The fault simulations were carried out exploiting the commercial tool TetraMax™ [11].

Fig. 2 reports the achieved transition fault coverage. Note that the functional test programs, reported on X-axis, are arranged in ascending order of their peak-power (same as reported in Fig. 1).

The transition fault coverage varies from about 19% up to 24%. We further investigated on the transition fault coverage since our goal is to re-use the functional test programs for at-speed delay testing.

The next evaluation was about the percentage of transition faults that are excited but not observed. For the transition fault model, this percentage varies from 60% to 63.5%, as depicted in Fig. 3. It means that a huge number of transition faults are indeed potentially detectable if we could increase their observability. For example, the first functional test program detects 22% of transition faults while the 61% of the undetected transition faults are not observed.

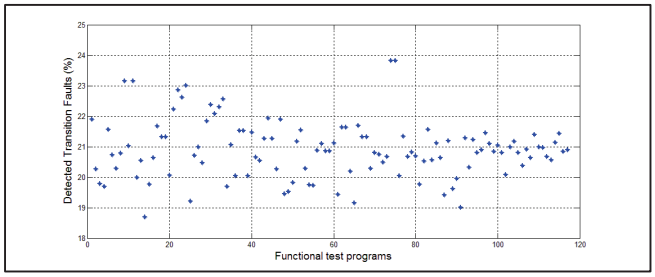

Fig. 2. Transition Fault Coverage (%)

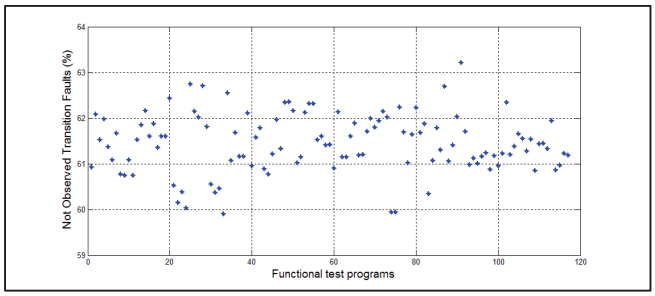

Fig. 3. Not-Observed Transition Faults (%)

In this paper we investigate how to improve the observability to increase the transition fault coverage. Next section will describe the proposed methodology explaining how we can apply the functional test programs to achieve this goal.

III. PROPOSED METHODOLOGY

In order to improve the observability, a simple solution is to add test-points in the circuit. This type of solution has been proposed in [7]. In this paper, instead of modifying the architecture of the microprocessor we intend to re-use the existing DfT circuitry. The main goal is to avoid any extra area overhead. Actually, the microprocessor has been synthetized by inserting one scan chain (see Table 1) that is used for the structural test application. Thus, we would like to exploit any existing scan chain to apply our functional test programs. To do that, we have first to understand how a functional test program is executed in functional mode and then how it can be executed in test mode (i.e., using the scan chain).

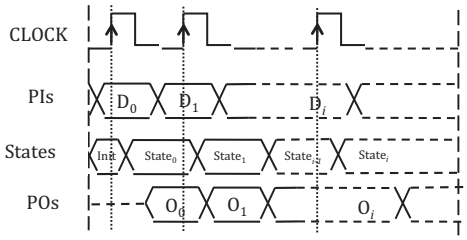

Fig. 4. Functional test program waveforms

In Fig. 4, we report the waveforms corresponding to functional test program execution. We traced the activity of the CLOCK, Primary Inputs (PIs), States and Primary Outputs (POs). The state is stored in the flip-flops of the circuit.

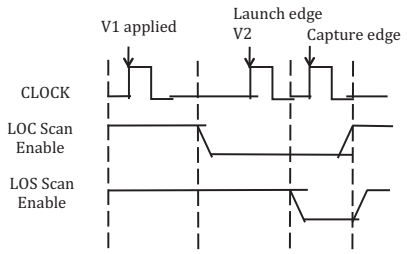

Fig. 5. LOC and LOS test schemes waveforms

Now that the application of the functional test programs is clear, we have to understand how to apply them in test mode. Thus, we have to map them to a structural test scheme and first of all we have to determine the best structural test scheme to use. As well known, two test schemes are used for structural transition fault testing during at-speed scan testing: Launch-off-Shift (LOS) and Launch-off-Capture (LOC) [2]. The typical waveforms of the clock and Scan Enable (SE) signals for LOS and LOC testing schemes are given in Fig. 5. Both schemes use a two-vector test <V1, V2> to detect the targeted transition delay fault. In both schemes, the test vector V1 is shifted into the scan chain(s) at slow speed, while the launch-to-capture cycle is applied at rated speed. In LOS, the SE signal remains at '1' and test vector V2 is obtained by one bit shifting of vector V1. Transitions are launched and propagated in the Circuit Under Test (CUT). Right before the capture cycle, the SE signal is switched from 1 to 0, and then the response to V2 is captured in the scan flip-flops. In LOC, after test vector V1 has been shifted into the scan chain(s), the SE signal has a large time window to be switched from '1' to '0'. This time, the vector V2 is obtained by the functional response of the CUT to vector V1. Transitions are launched and propagated in the CUT, and the response to V2 is captured in scan flip-flops during the capture cycle [2].

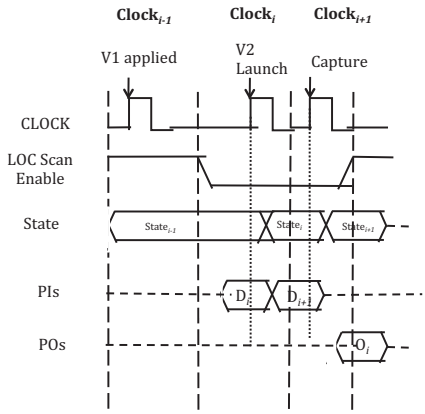

Fig. 6. Mapping functional test program to LOC test scheme

In this work, we consider the LOC as the test scheme to be used for mapping the functional test programs. The main reason is because vector V2 is the functional response to the vector V1. In this sense, the test scheme is more similar to a functional stimuli application.

The problem now is mapping the functional test program into a LOC test scheme. From Fig. 4, it is easy to understand that once we have the knowledge of PIs, FFs and POs values for each clock cycle, we can map them to a LOC test scheme.

Fig. 6 depicts the mapping of a functional test program into a LOC test scheme. Let us consider three generic clock cycles taken from the functional test program waveforms:

- **Clock**$_{i-1}$: it is mapped to the V1 application. Actually we shifted into the scan chain the logic values corresponding to the State$_{i-1}$;
- **Clock**$_i$: it is mapped to the V2 Launch cycle. Actually, before applying the i^{th} clock cycle, the SE signal switches from '1' to '0', thus the circuit is now in functional mode. We have to apply the expected primary inputs (i.e., D$_i$) to obtain V2 as the functional response of the circuit to V1. V2 corresponds to the State$_i$;
- **Clock**$_{i+1}$: it is mapped to the Capture cycle. The circuit is still in functional mode and the response to V2 is captured in the scan flip-flops.

From the above scheme, we can formally define that the functional test program is mapped into a sequence of test patterns. Each test pattern is composed of a couple of test vectors <V1, V2> where:

- V1 = {State$_{i-1}$, PI$_i$};
- V2 = {State$_i$, PI$_{i+1}$}.

The output values are State$_{i+1}$, that are shifted out through the scan chain, and the primary outputs corresponding to the values obtained after the $(i+1)^{th}$ clock cycle (i.e., O$_i$ in Fig. 6). The above scheme is repeated for each clock cycle from 1 to N-1, where N is the last clock cycle of the applied functional test program.

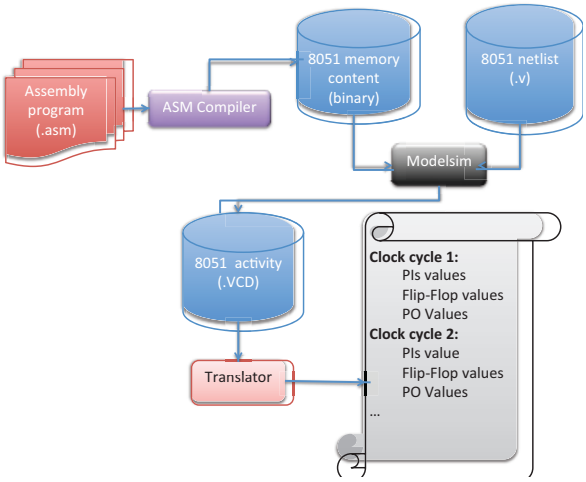

Fig. 7. Functional Programs to Test Patterns

Fig. 7 reports the flow implemented to translate a functional test programs into a LOC-like test scheme. Each functional test program (i.e., the assembly program in Fig. 7) is compiled in order to obtain the memory content to be loaded into the MC8051. The MC8051 with the test program loaded in memory is simulated using ModelSim™ [10] and the switching information related to the MC8051 primary inputs, primary outputs and flip-flops is stored in a VCD file. Thus, in the resulting VCD file we have saved the waveforms depicted in Fig. 4 when the applied functional test program is executed. The last step of the flow actually extracts from the VCD file the information to obtain V1 and V2 for each clock cycle. The latter are stored into a STIL file that will be used during the test application.

IV. Experimental Results

This section presents the results obtained by applying the proposed methodology to the set of functional test programs showing the benefits of mapping them into a LOC-like test scheme. Note that we exploited TetraMax™ [11] to fault simulate the functional test programs. The latters are applied in two different ways. The first one is referred to as "**Sequential**". It means that the programs are executed in the functional mode, so that we applied the sequential transition fault simulation. The second one is referred to as "**Func2LOC**". It means that we applied the methodology described in the previous section to map the functional programs into a LOC-like test scheme. Therefore, the fault simulation procedure is the same than the one used to simulate a LOC test set.

Table II reports the results obtained after the analysis of the functional test programs. We report the transition fault coverage, the length and the peak power. As described above, the transition fault coverage has been estimated in two ways, corresponding to the sub columns Sequential and Func2LOC. Note that once again the length is expressed in terms of patterns (couples of vectors <V1, V2>). As shown in Section III, we associated each couple of functional clock cycles to a LOC pattern. Finally, we give the max and min of each characteristic: TF, Length and Peak Power. The first two rows of the table give the max and min transition fault coverage considering the Func2LOC way and their corresponding Peak Power and Length.

TABLE II. Functional Test Programs Characteristics

	TF%		Length	Peak Power [µW]
	Sequential	Func2LOC		
Max(TF)	25.89	**41.48**	1,166	61.8
Min(TF)	19.84	**33.29**	1,100	61.8
Max(L)	23.7	39.72	**1,188**	57.4
Min(L)	21.44	39.94	**1,100**	87.8
Max(PP)	22.00	39.49	1,144	**88.8**
Min(PP)	23.34	36.31	1,188	**52.8**

The third and fourth rows give the max and min test length and their corresponding TF and Peak Power. Finally and similarly, the last two rows give the max and min peak power and their corresponding TF and Length.

A first comment is that the test length does not vary so much among the functional test programs. The variation is about 8.8% of patterns (from 1,100 up to 1,188). Conversely, the peak power varies of about 41% that is actually expected since these programs have been generated for this reason (see Section II).

The second and most important observation is related to the improvement of the fault coverage. In fact, by applying the proposed methodology (i.e., Func2LOC) we can increase in the average by 15% the transition fault coverage when the test programs are executed in the classical sequential way. It is worth mentioning that we are not so far from the LOC fault coverage (~57%) and LOC test length (1,300 patterns), while we guarantee the functional power budgets.

V. Conclusion and Further Works

In this paper, we presented a comprehensive analysis of the effects of re-using existing functional test programs for transition delay test in the context of power aware microprocessor testing. During this study, we proposed to map the functional programs into a LOC-like test scheme. This mapping allows applying functional programs exploiting the existing scan chain. This increases the transition fault coverage without adding any extra test-point or dedicated DfT architecture. Future works will focus on the analysis of different sets of functional programs targeting different microprocessor cores.

References

[1] ITRS 2013, http://www.itrs.net/home.html.

[2] 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.

[3] M. Valka, A. Bosio, L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel, E. Sanchez, M. Sonza Reorda, "A Functional Power Evaluation Flow for Defining Test Power Limits During At-Speed Delay Testing, " *IEEE European Test Symposium*, pp. 153-158, 2011.

[4] M. Psarakis, G. Gizopoulos, E. Sanchez and M. Sonza Reorda, "Microprocessor Software-Based Self-Testing," *Design & Test of Computers, IEEE* , vol.27, no.3, pp.4-19, 2010

[5] P. Bernardi, M. De Carvalho, E. Sanchez, M. Sonza Reorda, A. Bosio, L. Dilillo, M. Valka, P. Girard, "Fast Power Evaluation for Effective Generation of Test Programs Maximizing Peak Power Consumption," *Journal of Low Power Electronics*, vol. 9, no. 2, pp. 253-263, August 2013.

[6] Fang Hongxia, K. Chakrabarty, A. Jas, S. Patil, C. Tirumurti, "RT-Level Deviation-Based Grading of Functional Test Sequences," *IEEE VLSI Test Symposium*, pp.264-269, 2009.

[7] A. Sanyal, K. Chakrabarty, Mahmut Yilmaz, H. Fujiwara, "RT-level design-for-testability and expansion of functional test sequences for enhanced defect coverage," *IEEE International Test Conference,* pp.1-10, 2010.

[8] I. Pomeranz, "Concatenation of Functional Test Subsequences for Improved Fault Coverage and Reduced Test Length," *IEEE Transactions on Computers*, vol.61, no. 6, pp.899-904, June 2012.

[9] A. Touati, A. Bosio, L. Dilillo, P. Girard, A. Virazel, A. Todri-Sanial, P. Bernardi, "A Comprehensive Evaluation of Functional Programs for Power-Aware Test," *IEEE North Atlantic Test Workshop*, pp. 69-72, 2014

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

[11] Synopsys Inc., TetraMAX®, User Guide 2013.