RTL-based Functional Test Generation for High Defects Coverage in Digital SOCs

M. B. Santos, F.M. Gonçalves, I.C. Teixeira and J. P. Teixeira
IST / INESC
R. Alves Redol, 9, 1000-029 Lisboa, Portugal
mabs(fmg)@zorba.inesc.pt, ict(jct)@cezanne.inesc.pt

Abstract

Functional test is long viewed as unfitted for production test. The purpose of this contribution is to propose a RTL-based test generation methodology which can be rewardingly used both for design validation and to enhance the test effectiveness of classic, gate-level test generation. Hence, a RTL-based Defect-Oriented (DO) test generation methodology is proposed, for which a high Defects Coverage (DC) and a relatively short test sequence can be derived, thus allowing low-energy operation in test mode. The test effectiveness, regarding DC, is shown to be weakly dependent on the structural implementation of the behavioral description. The usefulness of the methodology is ascertained using the `VeriDOS` *simulation environment and the* CMUDSP *ITC'99 benchmark circuit.*

1. Introduction

Functionality, performance and quality requirements of electronic products continue to increase. Therefore, electronic design and test *productivity* needs to continuously increase. Here, we are concerned with the productivity of the test process (test planning, preparation and application).

Design *complexity* drives the design process towards higher levels of abstraction, and to HDL (Hardware Description Languages). Design *reuse* is increasing, as it can be seen in the extensive use of IP (Intellectual Property) cores in SOCs (Systems-On-a-Chip) [1]. System designers become system architects, reusing more and more proven components and their test processes. Test *reuse* must follow the same path. Stringent *quality* requirements are driving the need to complement traditional test preparation, based on structural test and the single LSA (Line Stuck-At) fault model, for digital systems, with DBT (Defects-Based Test) [2], or DOT (Defect-Oriented Test) [3].

HDL-based test preparation (test pattern generation (TPG) and fault simulation (FS)), namely at RTL (Register Transfer Level), including structural reconfiguration for DFT (Design For Testability), is thus a requirement [4]. In order to allow comparisons, new public domain, benchmark systems (ITC'99 benchmarks [5]) are being released, as a natural follow-up of the old ISCAS'85 (combinational) and '89 (sequential) benchmarks. However, can a functional test be effective in uncovering physical defects? How is its effectiveness dependent on the synthesized structure? These are important questions, not only for test reuse, but also due to the fact that soft cores can be synthesized by different EDA (Electronic Design Automation) systems, and mapped in different cell libraries and manufacturing technologies.

The purpose of this contribution is to show that a RTL-based, guided functional test can be reused to improve the coverage of physical defects in a production environment. Therefore, a methodology for RTL-based test generation is proposed, leading to RTL fault coverage and high Defects Coverage (DC) [3]. The methodology also provides guidelines for BIST (Built-In Self Test) implementation, leading to low-energy, high DC solutions. The preliminary results are validated using the CMUDSP benchmark [5] and the proprietary Defect-Oriented (DO) `VeriDOS` environment [6], expanded to support TPG.

The paper is organized as follows. In sections 2 and 3, the context of the work is presented, and major RTL fault models revisited. The proposed methodology is presented in section 4. The case study and experimental set-up are described in section 5. Finally, section 6 presents the results and conclusions of the work.

2. Context

Test preparation for low escape rates requires *structural testing*. It is a well known fact that, in general, *functional test* is a poor approach to derive a HQ (high quality) production test. As a consequence, functional test is mainly used for design validation. However, could it be

rewarding to *reuse* functional test to enhance production test quality? This could also be useful for *lifetime testing*, as functional tests are usually easy to apply even through remote activation. Finally, it would also enable at-speed test, which may additionally uncover dynamic faults.

As product quality requirements increase, setting DPM (Defects per million) goals down to 10 DPM, LSA-based test preparation is no longer sufficient [7-9]. Additional detection methods, such as i_{DDX} (power supply) current testing and delay testing have been added to reinforce test effectiveness. We refer as *test effectiveness (TE)* the ability of a given test pattern to uncover physical defects, likely to occur in a target system or component [8]. DBT is now examined as an enabler for HQ test [2].

Design complexity drove the need to enhance RTL-based test preparation. However, if functional test reuse is to be considered, the key issues are:

- Which RTL fault models should be used for HQ test, in terms of test effectiveness?
- How to generate a RTL functional test that can simultaneously support design (and prototype) validation and production / field testing?
- Is the functional test usefulness strongly dependent on the structure synthesized to implement the system's functionality?
- Assuming that some modules (e.g., IP cores in SOCs) are self-testable, how can an effective BIST solution be derived, leading to high DC and low-energy BIST operation (thus restricting the power consumption costs of the self -test procedure)?

These issues also include *design quality*, as, at some extent, testability requirements may require design reconfiguration, or DFT.

3. RTL Fault Models

Several RTL fault models [10-14] (see Table 1) and quality metrics have been proposed in the literature [15-23]. Controllability and observability are key metrics, as well as *branch coverage*, as a measure of the thoroughness by which the functional control and data paths are activated and, thus, considered in the functional test. In some cases, their effectiveness in covering LSA faults on the circuit's structural description has been ascertained. However, this does not guarantee their effectiveness to uncover physical defects. In [24] a functional test, applied as a complement to a LSA test set, has been reported to increase the Defects Coverage of an ALU module embedded in a public domain PIC controller.

In this work, several RTL fault models have been used in the context of the `VeriDOS` environment [6] (see Table 1). Fault models in variables and constants are present in all RTL fault lists and are a natural heritage from the structural level LSA fault model. The exception is the fault model based in a software test tool where constant and variables are replaced [11]. Such fault model leads to a significant increase in the size of the RTL fault list, without ensuring that it significantly contributes to the increase of the Defects Coverage.

RTL Fault Model	**[10]**	**[11]**	**[12]**	**[13]**	**[14]**	**This Work**
LSA type	X		X	X	X	**X**
Logic/ Arithmetic Operations		X	X		X	
Constant/ Variable Switch		X				
Null Statements	X	X			X	**X**
IF/ELSE	X	X	X	X	X	**X**
CASE	X	X	X	X		**X**
FOR	X	X		X		**X**

Table 1 - Considered RTL fault models

Two groups of RTL faults for logical and arithmetic operators can be considered: replacement of operators [11,14] and functional test of building blocks [12]. Replacement of operators can lead to huge fault lists, with no significant coverage gains. Exhaustive functional test of all the building blocks of every operator is an implementation dependent approach, which leads to good LSA fault coverage as showed in [12]; nevertheless, it may not add significantly to Defects Coverage.

The Null statement fault consists in not executing a single statement in the description. This is a useful fault model because fault detection requires three conditions: (1) the statement to be reached, (2) the statement execution reflecting on a value change of some state variable or register, and (3) this change being observable at the outputs. Conditions (1) and (2) are controllability conditions, while (3) is an observability condition. Nevertheless, redundancy occurs when both condition (IF, CASE) faults and Null Statement faults are included in the RTL fault list, as Null Statement faults within conditions are already considered in the condition fault model. Moreover, Null Statement full list is prohibitive; hence, only sampled faults are considered.

The IF/ELSE fault model consists of forcing an IF condition to be stuck-at true or stuck-at false. More sophisticated IF/ELSE fault models [11] also generate the dead-IF and dead-ELSE statements. CASE fault models include the control variable being stuck to each of the possible enumerated values (CASE stuck-at), disabling one value (CASE dead-value), or disabling the all CASE execution. FOR fault models were suggested in [10], by

adding and subtracting one from the extreme values of the cycle.

RTL fault models usefulness is generally limited by the fault list size and by the fault simulator mechanisms available for fault injection. The first limitation is responsible for the absence of variable and constant replacement faults in our RTL fault list. The second is responsible for the Null statement fault sampling in the final fault list.

4. Methodology

As referred, the RTL test pattern generation methodology aims at supporting system designers in deriving (and validating) HQ functional tests which can be reused, through the product development cycle, in design validation, production test and field test.

The difficulty of the TPG process is the detection of *hard faults*, those for which only a very limited subset of the input vector space (if not empty) enables detection. Starting at high level, these hard faults must be associated to parts of the functionality seldom revisited. We call them the "dark corners" of the functionality. Such "dark corners" will ultimately be mapped in some structural part of the chip, difficult to control and/or observe [25]. *The advantage of using RTL information is that these hard corners can easily be identified, using RTL fault models and random vectors.* Hence, RTL faults are being used for three purposes:

- To locate design flaws, namely unused functionality (e.g., empty IF conditions). This is a useful spin-off of the proposed methodology, which may prove valuable after frequent design modifications;
- To identify the "dark corners" of the functionality, which need to be exercized, as early as possible in the test application, in order to reduce test length and its energy consumption;
- To generate a functional test sequence which may lead to high N-detection of RTL faults and to high Defects Coverage.

As a consequence, the proposed methodology uses RTL fault simulation (with `VeriDOS`) to identify design flaws / dark corners and to guide the identification of partially constrained test vectors, referred as *masks*, which may significantly increase the single or multiple (N) detection of RTL faults in these dark corners. Multiple RTL fault detection will increase the probability of detection of defects on the synthesized structure.

Hence, a fast RTL fault simulation using random vectors (typically, 5,000 vectors) is used to locate the dark corners. Condition faults (IF, CASE) have been identified as crucial in dark corners location. Following the recognition of a dark corner, RTL code analysis is performed to determine the input variables that need to be forced, in order to *activate* the hard RTL faults. Masks are refined in an iterative process, until N-*detection* of the hard RTL faults is achieved. N is user's defined. Fault simulation is carried out by using nested sets of random vectors and *customized* mask vectors, i.e., mask vectors for which unconstrained positional bits (X, in the mask vectors) are randomly transformed in 0s or 1s.

For each mask *i*, m_i fixed positional bits in possible input words ($n>>m_i$ wide) are identified. Performing this for all hard RTL faults drives the "illumination" of all dark corners, and the generation of a set of n_{cv} constrained vectors, X_i (i = 1,2,..., n_{cv}), or masks. Typically, n_{cv} <20. Note that N-detection of RTL faults also guarantees complete branch coverage, which is a requisite for a HQ functional test. We refer a test set build with nested sets of random vectors and customized mask vectors able to perform at least N detection of RTL faults as *RTL_N test pattern.*

The proposed methodology is suitable for a BIST implementation, since random test generation needs only to be complemented with few masks, constraining few m_i positional bits; hence, only a weak deterministic self-test generation is required. As relative short test sequences can ensure high TE, low-energy BIST may thus be achieved.

As mentioned, LSA-based test patterns, generated from the synthesized structure, are not sufficient to reach very low Defect Level (DL) values. Therefore, RTL-based test vectors (RTL_N test patterns) are reused at structural level, being applied *after* LSA test vectors to increase DC (and thus decrease DL).

5. Case Study. Experimental Set-up

The case study used as test vehicle is one of ITC'99 benchmark circuits, the CMUDSP [26] (Fig. 1). It is a synthesizable digital signal processor (DSP) core written in Verilog. The CMUDSP is modeled after the Motorola DSP56002 processor. However, it does not include all of the instructions available to the DSP56002 processor, nor does it implement necessary functions such as interrupts.

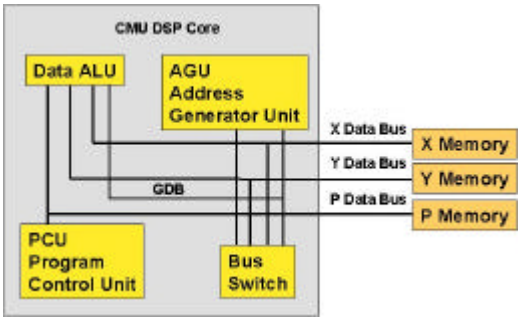

Fig. 1 - CMUDSP ITC'99 benchmark block diagram

The CMUDSP's Harvard architecture consists of three sets of separated data and address busses (X and Y, *data* busses, P, program memory). Address generation is done primarily by two different units. The AGU (Address Generator Unit) generates the addresses for the X and Y data busses. The AGU module is capable of doing complex addressing tasks independently of the ALU to maintain a datum per cycle per bus throughout program execution. The program address bus is generated by the PCU (Program Control Unit) module in most circumstances. Jump instructions in which the target address doesn't fit into the instruction word is a notable exception.

The simulation environment uses a commercial design system and `DOTLab`, a proprietary set of DO tools, including `lobs` (a defect extractor) and `VeriDOS` (Verilog Defect-Oriented Simulator). The DO simulator can perform mixed-level (behavioral/structural) fault simulation, using VHDL or Verilog behavioral descriptions, and Verilog structural descriptions. Gate-level Verilog fault models for BRI (bridging) and LOP (Line Open) defects, both for interconnection and cell faults, are resident in the `VeriDOS` tool, for CMOS physical implementations. `VeriDOS` has been expanded to generate RTL fault lists, and to perform RTL fault simulation. The simulation process uses as kernel the Cadence Verilog™ tool.

6. Results and Conclusions

A first set of experiments was carried out using the AGU module. The RTL fault simulation, performed with RTL fault models (see Table 1), didn't reach 100% fault coverage with a 5,000 random test sequence. A detailed analysis of the simulation results revealed design flaws. In fact, part of the functional description can never be observed (e.g., an empty IF condition). These findings have been reported to the ITC'99 benchmark user's forum. Hence, the experiment highlight a major result: *RTL fault models can rewardingly be used for design verification*. RTL fault simulation using random vectors additionally unveiled 14 (IF, CASE) conditions with low N-activation of RTL faults. *For this type of module, the (IF, CASE) fault model prove to be the most efficient to use for the purpose of identifying the dark corners.* In the case study, 14 *masks* were thus identified. RTL TPG was then carried out using these 14 vectors and random vectors associated with the unconstrained inputs.

At structural level, two topologies for the AGU module were automatically generated using our proprietary IDlib10 CMOS cell library, one synthesized for TIME and another for AREA optimization. For the TIME (AREA) version, a LSA test set of 136 (129) vectors is generated by a commercial EDA system.

The sequence of LSA-based test vectors and RTL-based test vectors was applied, and fault simulation of the AGU module, using `VeriDOS` and realistic open and bridging defects extracted with `lobs`, was performed for the two synthesized structures. Results are shown in Figs. 2 and 3, for the AREA and TIME physical layouts, respectively. Results show that, for AREA, only 43 additional RTL vectors are needed to reach the same DC values than those obtained with additional 14555 random vectors. For TIME (Fig. 3), the same DC is obtained with additional 143 RTL vectors, as compared with the additional 2441 random vectors. For a reasonable number of random vectors, the results also show that higher DC values can be obtained with RTL-based TPG than with random vectors. These results point out that *the proposed RTL guided TPG can enhance the effectiveness of a structural test with few additional vectors.* Moreover, *this conclusion is valid for different synthesized structures.*

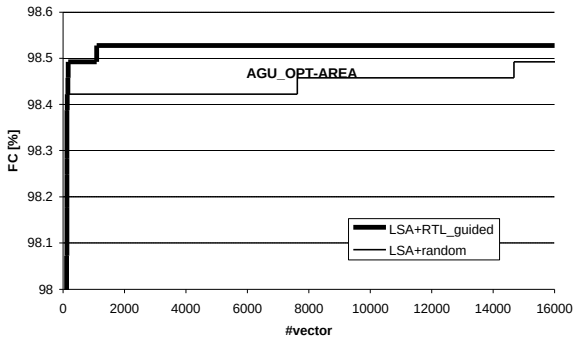

Fig. 2 - Defects Coverage for the AGU, using the AREA structure and the proposed RTL-based functional test.

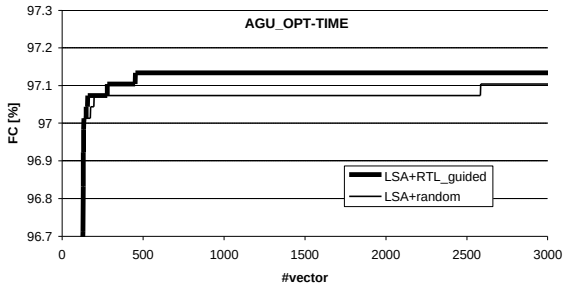

Fig.3 - Defects Coverage for the AGU, using the TIME structure and the proposed RTL-based functional test.

Experiments reveal that in every design there are, definitely, *"dark corners"*, which may either require additional TPG effort, or make random testing (often used in low-cost hardware solutions for BIST) ineffective.

For the PCU module, with 350 inputs and low logic depthness (thus, assumed easy to test), the proposed methodology allowed us to identify 6 constrained input vectors (*masks*), 4 involving m_i=51 deterministic bits, and

2 involving m_i=19 input bits. Hence, with limited additional logic, one can force these lines and allow LFSR-based TPG to carry out the pseudo-random generation. Simulation results are shown in Fig. 4. With 2,000 BIST vectors (instead of 20,000) a DC=94.5% value is reached (instead of DC=91.9% for the full random generation).

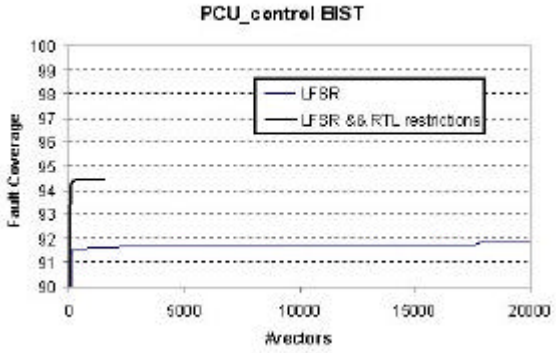

Fig. 4 - Defects coverage for the PCU control module, using the proposed BIST solution.

Additionally, mixed-level fault simulation was performed with `VeriDOS`, using the Verilog structure of the AGU *embedded* in the RTL behavioral description of the complete CMUDSP. Results are shown in Fig. 5. Although lower DC values are (naturally) obtained, still the RTL-guided TPG methodology leads to higher DC values than the ones obtained with random vectors.

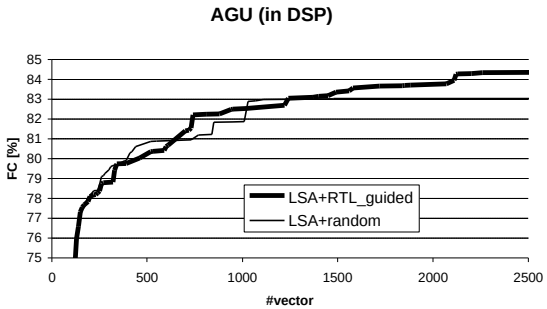

Fig. 5 - Defects coverage for the AGU module, embedded in the CMUDSP benchmark.

Two additional experiments were carried out with the AGU module. First, what are the results if *only* RTL-based tests are reused? Second, how much speed-up (or test length reduction) can be obtained when N-detection of RTL faults are targeted, in functional test generation? Results are shown in Fig. 6, for the AREA structure and for realistic (DC) fault coverage. Here, "RTL-guided" refers to a test sequence build of pairs of (random vector, customized mask vector). Results show that, if no LSA test sets are first applied, worse DC figures and test lengths result, which is to be expected, as no structural information has been used in the TPG process. When LSA test sets are first applied, the use of masks (LSA+RTL-guided curve) leads to gains in DC and test length, as compared with LSA+random vectors. However, test patterns fully defined at RTL level can lead to high DC values and relatively short test lengths, especially if N-detection of RTL faults are targeted in the TPG process. Here N=1 and N=10 have been selected. Using RTL_single (N=1), 157 test vectors are generated, and DC=96.35% is reached, while for RTL_10 (N=10), 1187 vectors are generated, leading to DC=97.9%, already close to the maximum reachable DC value (DC=98.6%) with voltage detection.

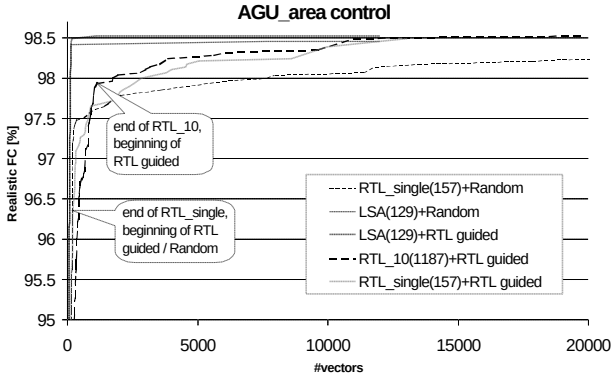

Fig. 6 - Defects Coverage for the AGU (AREA) module, reusing different RTL-based functional tests.

In order to observe the correlation between defects and LSA fault coverage, Fig. 7 displays the LSA fault coverage simulation results for the same AGU physical structure (AREA). Comparing Fig. 6 and 7 the correlation is observable, although 100% LSA fault coverage only leads to DC=98.42%. Similar conclusions regarding the advantage of using RTL_N test patterns can be drawn. However, Fig. 6 shows that maximum DC values are obtained with 12,500 RTL_N+RTL-guided vectors, while Fig. 7 shows that such vectors do not lead to 100 LSA fault coverage. In Fig. 8, DC results similar to the ones depicted in Fig. 6 are shown for the TIME structural synthesis (larger Si area). Although the numerical values are different, conclusions hold. In Fig. 8 the results obtained with a random test are also shown, demonstrating that insufficient (89.4%) DC values are reached.

In conclusion, RTL-based test generation does not substitute structural test generation, for high quality products. However, the proposed RTL-based TPG methodology allows the designer to spot design flaws and/or dark corners of the functionality, derive masks to enhance defect detection and generate, at RTL, test patterns with high DC and low test lengths (results weakly dependent on the circuit's final structure), to be reused either to complement the LSA test, or to drive a low-energy BIST scheme.

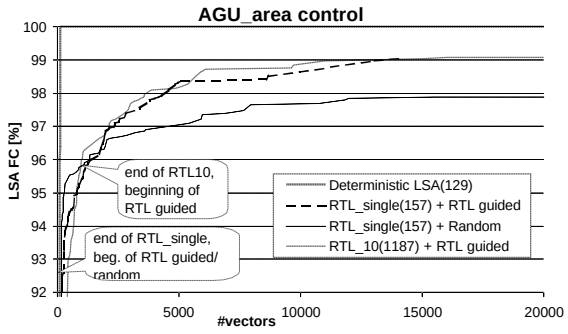

Fig. 7 - LSA Fault Coverage for the AGU (AREA) module.

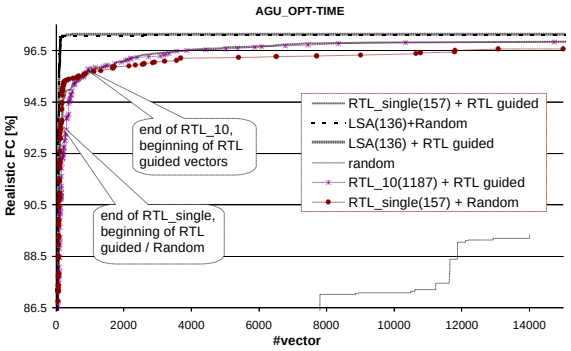

Fig. 8 - Defects Coverage for the AGU (TIME) module, using different RTL-based functional tests.

References

[1] Y. Zorian, E. van Marinissen, S. Dye, "Testing Embedded-Core Based System Chips", Proc. Int. Test Conf. (ITC), pp. 130-143, 1998.

[2] S. Sengupta, S. Kundu, S. Chakravarty, P. Parvathala, R. Galivanche, G. Kosonocky, M. Rodgers, T.M. Mak, "Defect-Based Test: a Key Enabler for Success Migration to Structural Test", Intel Technology Journal, Q1'99, available at http://www.developer.intel.com/ITJ, 1999.

[3] J.J.T. Sousa, F.M. Gonçalves, J.P.Teixeira, C. Marzocca, F. Corsi, T.W. Williams, "Defect Level Evaluation in an IC Design Environment", IEEE Trans. on CAD, vol. 15, nº. 10, pp. 1286-1293, 1996.

[4] Mahesh A. Iyer, moderator and organizer, "High Time for High-Level ATPG", Panel 1 session, Proc. ITC, pp. 1113-1119, 1999.

[5] Scott Davidson, moderator and organizer, "ITC'99 Benchmark Circuits - Preliminary Results", Panel 6 session, Proc. ITC, pp. 1126-1130, 1999.

[6] M. B. Santos, F.M. Gonçalves, I.C. Teixeira and J. P. Teixeira, "Defect-Oriented Verilog Fault Simulation of SoC Macros using a Stratified Fault Sampling Technique", Proc. of the IEEE VLSI Test Symp. (VTS), pp. 326-332, 1999.

[7] J.P. Shen, W. Maly and F.J. Ferguson, "Inductive Fault Analysis of MOS Integrated Circuits", IEEE Design and Test of Computers, n.2, pp. 13 – 26, December 1985.

[8] L.C. Wang, R. Mercer, T.W. Williams, "On the Decline of Testing Efficiency as Fault Coverage Approaches 100%", Proc. IEEE VLSI Test Symp. (VTS)., pp. 74-83, 1995.

[9] P.C. Maxwell, R.C. Aitken, "Test Sets and Reject Rates: All Fault Coverages Are Not Created Equal", IEEE Design & Test of Comp., vol. 10, pp. 42-51, Sept., 1993.

[10] Teresa Riesgo Alcaide, "Modelado de Fallos y Estimación de los Processos de Validación Funcional de Circuitos Digitales Descritos en VHDL Sintetizable", PhD. Thesis, Escuela Téc. Sup. Ing. Industriales, U.P. Madrid, 1996.

[11] Ghassan Al Hayek, Chantal Robach, "From Specification Validation to Hardware Testing: a Unified Method", Proc. Int. Test Conf. (ITC), 1996.

[12] Ronald J. Hayne and Barry W. Johnson, "Behavioral Fault Modeling in a VHDL Synthesis Environment", Proc. Int. Test Conf. (ITC), pp.333-340, 1999.

[13] Fabrizio Ferrandi, Franco Fummi, Donatella Sciuto, "Implicit Test Generation for Behavioral VHDL Models", Proc. Int. Test Conf. (ITC), pp. 587-596, 1998.

[14] P.C.Ward, J.R.Armstrong, "Behavioral Fault Simulation in VHDL", Proc. 27th. ACM/IEEE Design Automation Conf. (DAC), pp.587-593, 1990.

[15] Kurt Thearling and Jacob Abraham, "An Easily Functional Level Testability Measure", IEEE Int. Test Conference (ITC), pp. 381-390, 1989.

[16] Xinli Gu, Krzysztof Kuchcinski, Zebo Peng. "Testability Analysis and Improvement from VHDL Behavioral Specifications", Proc. EuroDAC, pp. 644-649, 1994.

[17] Chung-Hsing and Daniel G. Saab, "A Novel Behavioral Testability Measure" IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, vol. 12, nº. 12, December 1993.

[18] Mahsa Vahid and Alex Orailoglu, "Testability Metrics for Synthesis of Self-Testable Designs and Effective Test Plans", Proc. IEEE VLSI Test Symp. (VTS), pp. 170-175, 1995.

[19] Christos Papachristou, Joan Carletta, "Test Synthesis in the Behavioral Domain". Proc. Int. Test Conference (ITC), pp. 693-702, 1995.

[20] M. H. Gentil, D. Crestani, A. El Rhalibi and C. Durant, "A New Testability Measure: Description and Evaluation", IEEE VLSI Test Symp. (VTS), pp., 421-426, 1994.

[21] V. Chickermane, et al., "Addressing Design for Testability at the Architectural Level". IEEE Transactions on CAD of Int. Circs. and Syst., vol. 13, Nº. 7, 1994.

[22] John R. Wallack and Ramaswami Dandapani, "Coverage Metrics for Functional Tests", Proc. IEEE VLSI Test Symp. (VTS), pp. 176-181, 1994.

[23] Yves Le Traon and Chantal Robach, "From Hardware to Software Testability". Proc Int. Test Conference (ITC), pp. 710-719, 1995.

[24] M. B. Santos and J. P. Teixeira, "Defect-Oriented Mixed-Level Fault Simulation of Digital Systems-on-a- Chip Using HDL", Proc. of the Design Autom. and Test in Europe (DATE) Conf., pp. 549-553, 1999.

[25] J. Dworak, M.R. Grimaila, S. Lee, L.C. Wang, M.R. Mercer, "Modeling the Probability of Defect Excitation for a Commercial IC with Implications for Stuck-at-Fault-based ATPG Strategies", Proc. Int. Test Conf. (ITC), pp. 1031-1037, 1999.

[26] CMUDSP benchmark (I-99-5, ITC 99 5]), http://www.ece.cmu.edu/~lowpower/benchmarks.html