HAL
open science

# Validation of a Mixed-Signal Board ATPG Method: The TCB case study

Valérie-Anne Nicolas, Bertrand Gilles, Laurent Tchamnda Nana

## To cite this version:

Valérie-Anne Nicolas, Bertrand Gilles, Laurent Tchamnda Nana. Validation of a Mixed-Signal Board ATPG Method: The TCB case study. 2008. hal-00607290

**HAL Id: hal-00607290**

**https://hal.univ-brest.fr/hal-00607290v1**

Submitted on 8 Jul 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Validation of a Mixed-Signal Board ATPG Method : The TCB case study
# Technical report - LISYC

*Valérie-Anne Nicolas  Bertrand Gilles  Laurent Nana*
*Université Europénne de Bretagne*
*Université de Brest*
*Laboratoire d'Informatique des SYstèmes Complexes (LISYC - EA3883)*
*Dpt Informatique - 20 avenue Le Gorgeu - CS 93837 - 29238 Brest Cedex 3 - France*
*{vnicolas|gilles|nana}@univ-brest.fr*

## Abstract

*We present the validation protocol of a mixed-signal board ATPG method [6, 7]. First results confirm the method fitting well with maintenance test, board modeling stage adequacy and test data generation reliability. The essential need for user-defined dedicated test strategies is highlighted in order to ensure meaningful test process and full black-box test.*

## 1 Introduction

During maintenance stage, board test is still in need of semi-automatic, robust and complete tools. This is quite different from the design or production stages for which a large panel of methods and tools exists [3]. This state of things is mainly due to the maintenance context itself, for at least two reasons. First of all, the late location of maintenance stage in product life-cycle. In many cases, this implies reduced knowledge about the board for maintenance people: no designer direct knowledge, partial documentation, confidentiality restrictions. Moreover, in the case of long life systems, even if some test information is available, it may be unusable on aged components whose caracteristics have evolved. The presence of mixed-signal components reinforce the overall complexity of this kind of test. Secondly, decreasing cost of electronic components induce a consumerist inclination: why test, diagnose and repair when replacing is cheaper and quicker? Although replacing faulty boards or entire systems is sensible and interesting for large distribution products, it is less suited for large long-life systems.

Nowadays, due to the wide variety of board maintenance situations encountered, test, diagnosis and repair at maintenance stage are often realized in an empirical way. Clearly, dedicated methods and specialized tools are needed to guide or automate at least a part of the work involved in the maintenance stage.

Our work is in keeping with this maintenance context and focuses more particularly on mixed-signal board test. We have proposed a method to help maintenance testing, providing a functional ATPG for mixed-signal boards [6, 7].

The aim of this paper is to sketch a validation protocol for this method and to discuss first results concerning board modeling choices suitability, signal representation pertinency, test strategies quality.

The second section briefly recalls our method and ATPG. The third section introduces the validation protocol. The application of the validation protocol on an example is detailed in the next section. Trends for future work conclude the paper.

## 2 The mixed-signal board ATPG method

As mentioned in the previous section, maintenance test lacks of specific methods and tools. We have proposed a method dedicated to maintenance test of mixed-signal boards [6, 7]. Its aim is to check a board behavior and to help determine faulty components in case of defective functionality. This method provides some guidelines for modeling and testing mixed-signal boards. It has been implemented in a semi-automatic prototype tool called "Copernicia". This tool also includes an automatic test pattern generator (ATPG). The non-automatic part of the method is the modeling of the board and its components (a library is provided by Copernicia with the most common components). The test of the board and its components is fully automated using some standard test strategies and test models for the different components. One may add some more specific or detailled test strategies from its own. Whereas the definition and use of such strategies are not automated (they have to

be carried on by the test engineer), the test data generation process remains automatic.

Our method is intrinsically based on functional modeling and testing. Since they only make use of the external behavior of the components, functional-based models may address a wide spectrum of situations concerning board maintenance test: they may be adapted to the amount of information available (component specification levels), to the nature of the components (digital, mixed-signal, or analog) and to the goal of the test (go-nogo, fine-grain diagnosis oriented testing). Functional testing of component is not used during design or production stages because test software development is costly. It is mainly achieved at the system level in order to test the interactions between components and to check if the global system meets its specification requirements. Thus, there is no predefined functional tests available at the board level.

The method uses an only generic uniform formalism for modeling and testing: finite state machines (FSM). FSM has been chosen because it allows intuitive modeling of component behaviors (it is especially true for digital components), it is well suited to express test strategies and well known to test practitioners [5, 9]. ATPG is implemented using constraint logic programming (CLP [13]) with the Eclipse tool [4].

One may refer to [6] to have a look at the way a simple case study is modeled in Copernicia and [7, 8] to have an idea of the testing process.

## 3 The validation protocol

We have proposed a method to help board test in maintenance. Now, we have to determine how this method fits maintenance testing requirements, process and objectives. For this purpose, we outline a protocol to validate the method, helped by a simulation tool.

On the one hand, in applying our method, we model a given board and then generate some test data for it. On the other hand, using a simulation tool, we model the board and next, simulate its behavior on the obtained test data. This process has to be iterated on a selected set of representative examples.

The main objectives are:

- evaluation of the adequacy of our modeling,
- verification that outputs obtained by simulation are consistent with the outputs predicted by the method (reliability of test data generation),
- evaluation of our test package (global test method, test levels, test models, test strategies, standard or user-defined facilities...) with respect to maintenance test requirements, process and objectives.

Several tools are available for modeling, simulating and analyzing dynamic systems. Among them, some are well adapted for systems such as boards, including mixed signal ones, and provide graphical editors which can be used to build complex models by interconnecting blocks which are either predefined basic functions (signal generators, filters...) or user defined functions. Without being exhaustive, we can highlight well known commercial tools such as Simulink which is part of the MATLAB Toolset [10], SystemBuild which is part of the MATRIXx Toolset [2], Labview toolkits for simulation [1] and a public domain tool such as Scicos which is part of the Scilab Toolset [12]. Scicos functionalities are similar to Simulink's. Actually, the Scilab Toolset developed by INRIA in France is known as the public domain version of the MATLAB Toolset. Most of the functionalities offered by Simulink are also available in SystemBuild.

In order to validate our test data generation approach, we have chosen Simulink for the modeling and simulation of boards. This choice is made because we already use it for teaching and already have some experience of it.

The detailed validation protocol is:

- defining a set of representative examples of boards to be tested ;
- chosing adapted test strategies for each board ;
- applying the method on each board: i.e. board modeling, test strategies definition, test data (and predicted results) generation ;
- board modeling with Simulink ;
- then for each test data: simulating board behavior with the test data as input, and next verifying the consistency of the output obtained with respect to the output predicted by the method and the test objective.

We present in the next section the experiment on the Test Case Board (TCB) introduced in the Section 4 of [6].

## 4 Protocol application on the TCB

The modeling of the Test Case Board (TCB) with our method is studied in the Section 4 of [6] and recalled in Figure 1. First of all, we present the TCB modeling using Simulink. Next, we expose the chosen test strategies and the test data generated by our ATPG algorithms. Then, we present the simulation with these test data and the results we obtained. A discussion on this experiment concludes the section.

### 4.1 TCB Simulink modeling

The hierachical Simulink modeling of the board is shown in Figures 2 and 3. Figure 2 shows the top level of the modeling. The central rectangle represents the board. The board

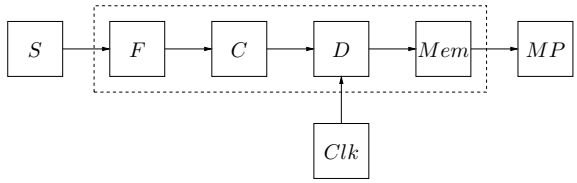


**Figure 1. TCB Copernicia modeling**

primary input (*PI*) is connected to an analog sine wave generator. The board primary output (*PO*) is connected to the *po* block that models the measurement point for the values of the data written into the memory. These values are stored into the MATLAB workspace with their time stamp. Scopes display signals during the simulation. In particular, scopes connected respectively to the filter output, the sampler output, and the comparator output make it possible to observe internal signals of the board. The *threshold* box sets the threshold's value of the comparator.

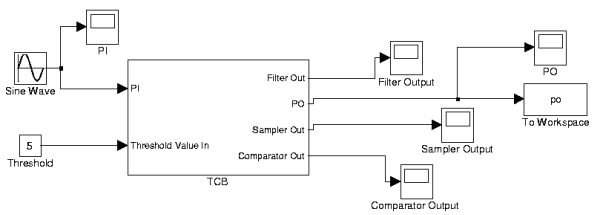


**Figure 2. TCB Simulink modeling: Top level**

Figure 3 shows the second level of modeling. At this level, the modeling of the TCB as a set of interconnected block diagrams looks like the board level modeling we proposed in [6] which is shown in Figure 1 (the TCB is delimited by the dashed rectangle). Each block at the board level matches with a Simulink diagram block:

- the filter (F) is modeled by the built-in transfer function block that implements the transfer function of the analog high-pass filter expressed by

$$H(s) = \frac{s}{s + 1000\,.\,2\pi}$$

  where 1000 (Hz) is the value of the cutoff frequency of the filter,
- the comparator (C) is modeled by a customized block using the S-Functions API [11],
- the digital controller (D) is modeled by the built-in zero-order hold block with a sampling rate of 500 Hz,
- the memory (Mem) is modeled by the general expression block expressed by $u(1)$, meaning that the block output is the same as its input (identity function).

The threshold of the the Simulink comparator is an input of the block whereas the threshold of the block C is a parameter of the block description [6]. In order to make it possible to refine the Simulink model of the comparator, we have added an input for setting the threshold's value rather than using a hard-wired value in the S-function code. So, we do not need to recompile the S-function code of the block each time the value of the threshold changes.

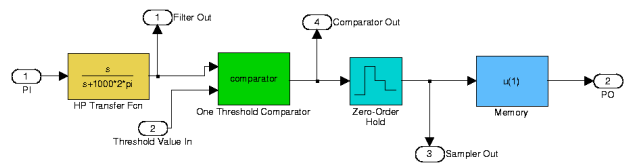


**Figure 3. TCB Simulink modeling: Level 2**

Comparing this Simulink modeling with the one obtained by our method, we note that both approaches rely on two hierarchical levels of specification. For Simulink modeling, first level is used to model the board inputs and outputs. Second level is the specification of the components of the board and of their in-between links.

In our method, first level defines the board inputs and outputs, but also the way components are linked. Second level is dedicated to the specification of component behavior.

Both approaches are thus quite similar. The only real difference is the internal representation of components: Simulink's ordinary differential equations (with MATLAB ode45 solver) vs our method FSM.

## 4.2 Testing process

The black-box board test data set generated with our method is $TDS = \{TD_1, TD_2, TD_3, TD_4, TD_5\}$, made of:

$TDS_{filter} = \{TD_1, TD_2\}$ using $S_{filter1}$ and $S_{filter2}$ test strategies which are recalled below,

$TDS_{comparator} = \{TD_3, TD_4\}$ and

$TDS_{digital} = \{TD_5\}$ using a test strategy $S_{digital1}$ that assumes synchronization between analog input sine signal and digital part.

This test data set follows directly from the application of test models and strategies described in Section 4.4 of [6]. This test data set is the same as the one presented in Section 4.5 of [6] without the two first test data, which had unfortunately been added.

A test data (TD) is made of an input couple and an output singleton. The input couple has the form $(S, Clk)$ and the output singleton has the form $(MP)$ where $S$ is the analog source, $Clk$ the clock signal and $MP$ the digital measurement point (memory state).

For memory, here are the formal test data:

$$TD_1 = \quad (In = (sig(sine, V_{IN}, 10\,f_c, 0), top(z)), \\ Out = ([v, z]))$$

with $z = T_e$ and $V_{IN} - \delta_1 \geq S$ (S$_{filter1}$ test strategy) where $\delta_1$ is a filter tolerance characteristic and S is the threshold of the comparator. The value $v$ is the value of the signal $sig(rw, \Delta T_1, T_0, t_0)$ at time $z$.

$$TD_2 = \quad (In = (sig(sine, V_{IN}, f_c, 0), top(z)), \\ Out = ([v, z]))$$

with $z = T_e$ and $\frac{\sqrt{2}}{2}\,V_{IN} - \delta_3 \geq S$ (S$_{filter2}$ test strategy). The value $v$ is the value of the signal $sig(rw, \Delta T_1, T_0, t_0)$ at time $z$.

$$TD_3 = \quad (In = (sig(sine, S - \delta, F_{IN}, \phi_{IN}), top(z)), \\ Out = ([0, z]))$$

with $z = T_e$. The tolerance $\delta$ is the tolerance used in the test model of the comparator.

$$TD_4 = \quad (In = (sig(sine, S + \delta, F_{IN}, \phi_{IN}), top(z)), \\ Out = ([v, z]))$$

with $z = T_e$. The value $v$ is the value of the signal $sig(rw, \Delta T_1, T_0, t_0)$ at time $z$.

$$TD_5 = \quad (In = (sig(sine, V_{IN}, \tfrac{1}{T_e}, \phi_{IN}), top(z)), \\ Out = ([1, z]))$$

with $z = k\,T_e$ and $T_e = t_0 + \frac{\Delta T_1}{2}$ ($S_{digital1}$ test strategy).

Our ATPG algorithms solve and propagate all the induced sets of constraints with the following parameters:

- the comparator threshold is set to 5 V,
- the filter cutoff frequency is set to 1000 Hz,
- the sampling period $T_e$ is set to 0.002 s,

and compute the effective test data:

$$TD_1 = \quad (In = (sig(sine, 5.15, 10000, 0), top(0.002)), \\ Out = ([v, 0.002]))$$

The value $v = 0$ is the value at time $0.002$ of a rectangular wave signal which frequency is the same as the one of the input sine signal.

$$TD_2 = \quad (In = (sig(sine, 7.21, 1000, 0), top(0.002)), \\ Out = ([v, 0.002]))$$

The value $v = 0$ is the value at time $0.002$ of a rectangular wave signal which frequency is the same as the one of the input sine signal.

$$TD_3 = \quad (In = (sig(sine, 5.12, 2672.87, 0), top(0.002)), \\ Out = ([0, 0.002]))$$

$$TD_4 = \quad (In = (sig(sine, 5.52, 2807.23, 0), top(0.002)), \\ Out = ([v, 0.002]))$$

The value $v = 0$ is the value at time $0.002$ of the signal $sig(rw, 0.001, 0, 0)$.

$$TD_5 = \quad (In = (sig(sine, 11.18, 500, 5.82), top(0.002)), \\ Out = ([1, 0.002])) \tag{1}$$

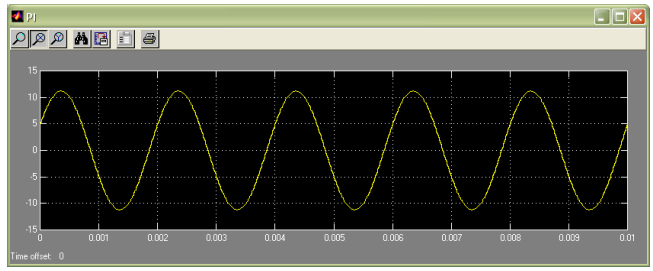


**Figure 4. Primary input test signal**

Let us focus on test data $TD_5$ generated for the digital part using the test strategy $S_{digital1}$. This test data gives the input test stimulus (shown in figure 4) for which the time stamped value of the memory ($Out$) is equal to 1 when time $z$ is a multiple of the sampling period $T_e$ (Clk period). Thus, this test data checks the analog-digital synchronization. Indeed, with test data $TD_5$, the primary output $PO$ is always equal to 1. This allows the detection of stuck-at-zero faults, but not stuck-at-one faults.

Let $S_{digital2}$ be a new test strategy ensuring that the value on the primary output $PO$ is alternately equal to 0 or 1. The new computed value of $TD_5$ using the test strategy $S_{digital2}$ is:

$$TD'_5 = \quad (In = (sig(sine, 20.61, 250, 2.9), top(0.002)), \\ Out = ([v, 0.002]))$$

With the test data $TD'_5$, the $S_{digital2}$ test strategy thus detects both stuck-at-zero faults and stuck-at-one faults. The $S_{digital2}$ test strategy is therefore better than $S_{digital1}$.

Note that without any extra test strategy, the value of $TD_5$ would be:

$$TD''_5 = \quad (In = (sig(DC, 3.0, _, _), top(0)), \\ Out = ([0, 0]))$$

where $sig(DC, 3.0, _, _)$ represents a DC signal of magnitude 3 V.

With test data $TD''_5$, the primary output $PO$ is always equal to 0 : the analog-digital synchronization is not ensured and stuck-at-zero faults are not detected.

These comments about $TD_5$ clearly emphasize the importance of the choice of the test strategy in order to have the most meaningful results.

Now, let us look at $TDS_{filter}$. It relies on $S_{filter1}$ and $S_{filter2}$ test strategies. Both test strategies ensure that the signal at the output of the filter is higher than the threshold's value of the comparator, to guarantee that the value observed on the primary output $PO$ is alternately equal to 0 or 1. Without using these strategies, the primary output $PO$ may always be equal to 0, and the distinction between weakened or not weakened signal becomes impossible.

If a measurement point at the output of the filter is available, no extra test strategy is needed to test the filter (we perform a kind of unitary test of the filter from the primary inputs of the board). If no intermediate measurement point is available, using $S_{filter1}$ and $S_{filter2}$ test strategies is necessary to test the filter (we perform a black-box test of the filter from primary inputs and outputs of the board).

We showed the advantage of test strategies to produce meaningful results. On the filter test case, we point another role of the test strategies which is the forward propagation of results. Anyway, they are all context dependent strategies.

One may note that no particular test strategy is needed for the comparator. The reason is that the data propagation from the comparator output to primary output $PO$ is straightforward (assuming the memory component is correct).

Finally, using standard component test models and test strategies $S_{filter1}$, $S_{filter2}$ and $S_{digital2}$, our method generate the test data set $TDS' = \{TD_1, TD_2, TD_3, TD_4, TD'_5\}$.

The test data generation stopping point is reached because $TDS'$ (like $TDS$) covers all the test models and associated test strategies of the board's components (this is implemented by full FSM transition coverage). We thus consider that this test data set is sufficient and adequate to test the board in a black-box way. This matches with simulation results. Some of them are presented in the next section.

### 4.3 Simulation

We have simulated the Simulink model with the different test data presented in the previous section, using the MATLAB ode45 (Dormand-Prince) solver, starting from time 0 to stop time 0.01s. The results we obtained agree with the test data predicted outputs.

In this section, for the sake of conciseness, we focus on the simulation with test data $TD_5$ generated for the digital part using the test strategy $S_{digital1}$. This test data checks the analog-digital synchronization.

Figure 5 shows the filter output signal. This signal is weakened and has shifted because the frequency of the input test stimulus (500 Hz) is lower than the cutoff frequency of the filter (1000 Hz). The magnitude of the signal is slightly higher than the threshold's value of the comparator (5 V) for times multiple of the sampling rate.

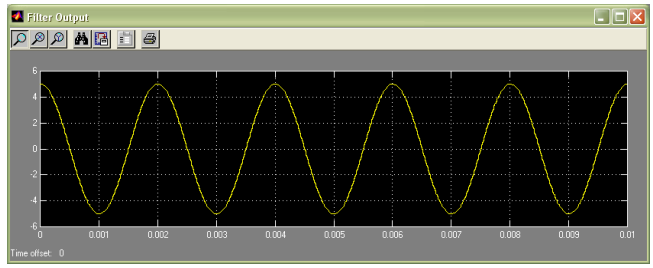


**Figure 5. Filter ouput signal**

Figure 6 shows the digital comparator output signal. Because the magnitude of the input signal of the comparator is slightly higher than its threshold's value for times multiple of the sampling rate, the output signal is a rectangular wave whose magnitude is 1. The intervals for which the logical value is 1 are centered on the corresponding periods.

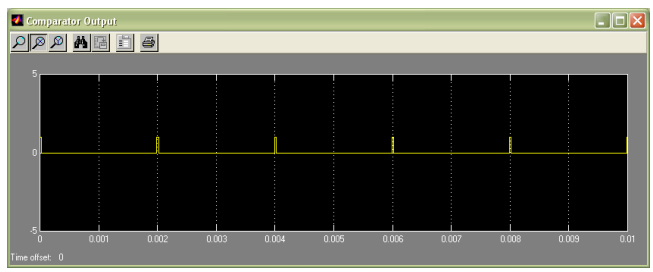


**Figure 6. Comparator ouput signal**

Figure 7 shows the successive time stamped values written into the memory exported to the MATLAB workspace thanks to the po box (see Figure 2). This result matches the computed primary output in expression (1).

The results we obtained in doing all the simulations agree with the test data and therefore confirm the adequacy and reliabilty of test data (inputs and outputs) generation.

### 4.4 Discussion

The TCB validation experiment stated in 4.2 and 4.3 has confirmed the signal representation choice (sine, rectangular wave and DC signals are common basic signals), the adequacy and reliability of the functional models for basic components, the test models and standard test strategies pertinence in terms of data and expected results for unitary tests, the accuracy of test data generation algorithms.

It has also pointed out the limits of the test models in case of embedded components (black-box testing) and the necessity for extra test strategies definition to ensure propagation of meaningful results to the outputs. The accuracy

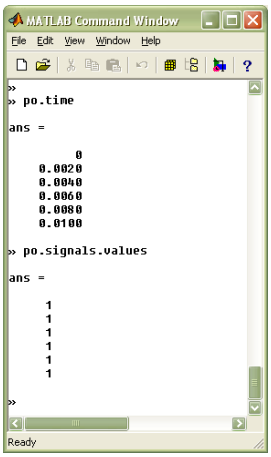


**Figure 7. Primary output signal**

of black-box testing is therefore highly related to the pertinency of the applied test strategies.

In our test approach, the overall black-box testing process is the much complete one. The other test facilites we provide may be viewed as restricted applications of this main testing process. For instance, a component unitary test is easier than the overall board test because its environment is restricted to some input/output points, with no other component interaction.

More generally, this particular experiment has shown the interest of a uniform approach to test mixed-signal boards and the generality of our method caracterized by its ability to mix and deal with different test strategies.

## 5 Conclusion and future work

We have presented a protocol to validate a mixed-signal board ATPG method and its application on a board case example. Results obtained have been discussed in Section 4.4.

The first results confirm that the method fits well with maintenance test, board modeling stage adequacy and test data generation reliability. The usefulness of proposed automatic standard test facilities is real but appears restricted to some specific tasks of maintenance test (go-nogo test, unitary test). The essential need for user-defined dedicated test strategies is highlighted in order to ensure meaningful test process and full black-box test.

This paper has presented a validation of the method supported only by the example of the TCB. Obviously, a meaningful validation relies on a selected set of several representative examples. Experiments with others board examples are ongoing, and preliminary results are consistent with those stated here.

## References

[1] http://www.ni.com/labview.

[2] http://www.ni.com/matrix.

[3] M. L. Bushnell and V. D. Agrawal. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Springer, 2000.

[4] A. M. Cheadle, W. Harvey, A. Sadler, J. Schimpf, K. Shen, and M. Wallace. Eclipse: An Introduction. Technical Report IC-Parc-03-1, IC-Parc, Imperial College London, 2003.

[5] K.-T. Cheng and A. Krishnakumar. Automatic Functional Test Generation Using the Extended Finite State Machine Model. In *30th ACM/IEEE Design Automation Conference*, pages 86–91, 1993.

[6] B. Gilles, L. t. Nana, and V.-a. Nicolas. Modeling and Generation of Test Patterns for Mixed-Signal Boards: Dealing with Basic Signals. In *Proceedings of the 6th IEEE International Board Test Workshop, BTW07*, Sept. 2007.

[7] B. Gilles, V.-a. Nicolas, and L. Lemarchand. Timed functional modeling for mixed-signal boards in maintenance testing. In *Proceedings of the 7th IEEE Latin-American Test Workshop, LATW 2006*, Mar. 2006.

[8] B. Gilles, V.-A. Nicolas, L. Lemarchand, and L. Marcé. Timed functional modeling for mixed-signal boards in maintenance testing: A case study. Technical report, LISYC, http://www.ea3883.univ-brest.fr/ea3883.htm, Feb. 2006.

[9] D. Lee and M.Yannakakis. Principles and Methods of Testing Finite State Machines–A Survey. In *IEEE Transactions on Computers*, volume 84, August 1996.

[10] The MathWorks, Inc. *Using Simulink - Model-Based and System-Based Design*, 2003.

[11] The MathWorks, Inc. *Writing S-Functions - Model-Based and System-Based Design*, 2003.

[12] R. Nikoukhah and S. Steer. SCICOS A Dynamic System Builder and Simulator. Technical Report 0207, INRIA, 1997.

[13] C. Paoli, M.-L. Nivet, and J.-F. Santucci. Use of Constraint Solving in Order to Generate Test Vectors for Behavioral Validation. In *IEEE International High-Level Design Validation and Test Workshop, HLDVT'00*, 2000.