Towards a New Modelling of Mixed-Signal Boards For Maintenance Testing

Bertrand Gilles, Valérie-Anne Nicolas, Laurent Lemarchand, Lionel Marcé, Bruno Castel

To cite this version:

Bertrand Gilles, Valérie-Anne Nicolas, Laurent Lemarchand, Lionel Marcé, Bruno Castel. Towards a New Modelling of Mixed-Signal Boards For Maintenance Testing. 11th IEEE International Mixed-Signal Testing Workshop, IMSTW'05, Jun 2005, Cannes, France. p.90, 2005. <hal-00607349>

HAL Id: hal-00607349

http://hal.univ-brest.fr/hal-00607349

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.

Towards A New Modelling of Mixed Signal Boards For Maintenance Testing

Bertrand Gilles[1] Valérie-Anne Nicolas[1] Laurent Lemarchand[1]
Lionel Marcé[1] Bruno Castel[2]

[1]EA3883 - Laboratoire d'Informatique des SYstèmes Complexes (LISYC)
Département Informatique - Université de Bretagne Occidentale
20 avenue Le Gorgeu - BP 809 - 29285 Brest Cédex, France
{vnicolas|gilles|lemarch|marce}@univ-brest.fr

[2] ISIS-MPP
Rue Pierre Rivalon - Technôpole Brest Iroise - 29200 Brest France
bruno.castel@isismpp.fr

Abstract

We are introducing a new method for the test of mixed signal boards. The emphasis is on the functional modelling and test vectors generation. A prototype implementation of the method has been used and validated in an industrial case study.

Keywords

Modelling, functional testing, mixed signal boards, test vectors generation.

1 Introduction

The test is an activity which takes place all along the life cycle of a component under different forms. Testability analysis during the design stage, then, during other stages (development, production, maintenance), conception and generation of test data, application of test data and finally analysis of results obtained [1]. The development of communication and multimedia technologies, the constant use of on-board technologies (transports, mobile telephony, ...) require more and more efficient and safe mixed signal components and therefore are in need of more adapted, reliable and safe tests. In addition, the increasing integration of components (submicronic technologies) and ever tighter design/production deadlines represent many constraints for the testing process. Majhi and Agrawal [2] have given a survey of trends and challenges in mixed signal testing. In this context, it is crucial to master more efficiently the stage relating to testing and maintenance of mixed signal boards (including the diagnosis of errors and their correction). Our task is related to this aspect and concerns more particularly the maintenance testing of mixed signal boards with a view of diagnosing. In practise, a faulty component is returned with a description of the behaviour which has revealed the fault (functional description).

We present a new testing method for mixed signal boards. The board is considered as a set of blocks which are described with functional and test models. These models use Finite State Machines (FSM). Test vectors generation is realised solving constraints expressed by FSM.

Ayari *et al.* [3] have presented an automatic test vector generation approach for functional testing of mixed signal circuits. Ramadoss and Bushnell [4] have proposed a reverse simulation

approach to analogue and mixed signal test generation that parallels the digital test generation process. The main difference between these works and our approach resides in the underlying fault model: physical faults vs behavioural faults.

The paper is organised as follows: in section 2 we present our board modelling. In section 3 we explain the test vectors generation. A first implementation prototype used for a case study and results are presented in section 4. A conclusion is given in section 5.

2 Board modelling

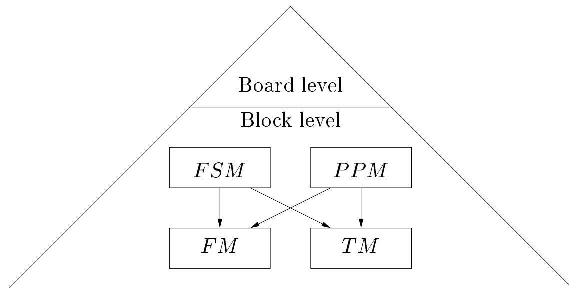

Figure 1: The mixed signal board modelling.

The mixed signal boards that we wish to test are involving analogue components in the board's interfaces only. The digital components exclusively form the control part of the board. Although analogue and digital components coexist, this type of board is not what we call an hybrid system because there is not mixed, underlying discrete-continuous dynamic associated to it. Indeed, the system, developed as such, follows a continuous dynamic in its interfaces and a discrete dynamic in its control part without any interferences whatsoever between these two dynamics. It did not seem necessary to use hybrid formalisms (such as hybrid automata [5] or the MLD logic [6] specific to hybrid systems) in order to create a model of this type of board, although these formalisms offer a uniform format. Indeed, hybrid formalism emphasise the modelling of the interlacing of discrete and continuous dynamics. This is of no interest in our particular case.

In order to generate the test patterns for mixed signal boards, our modelling uses two hierarchical levels of description depicted in figure 1. The first level, called the *board level*, breaks down the board into functional blocks. At the second level, the *block level*, two kinds of functional specifications are used: Finite State Machine (FSM) for digital blocks and predefined parametrised functions (PPM) for analogue and mixed signal blocks. A *functional model* (FM) and a *test model* (TM) are derived from these specifications. These two internal models make use of FSM. The board level and the block level are described next. At the block level, both functional specification and internal models are detailed.

2.1 The board level

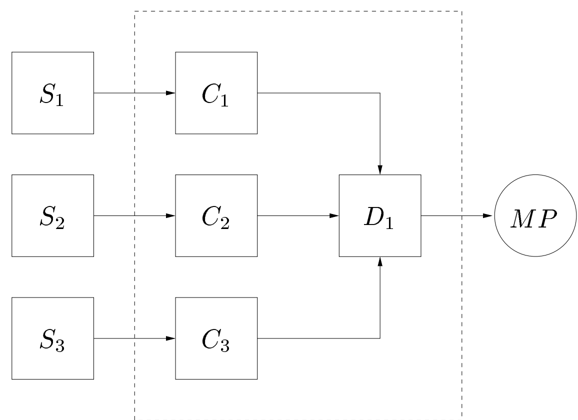

Figure 2: An example of board level specification.

At this level, the board is broken down into a set of interconnected functional blocks (figure 2). They can be analogue, digital or mixed-signal. Oriented links denote data exchanges between blocks. Signals exchanged on a link are characterised by:

- amplitude
- form (DC, sine, square)
- frequency (is equal to zero if the form is DC)
- type (analogue, digital)

A block can have several inputs and outputs. Additional blocks (not part of the board) are used

to model the external sources which supply input signals. Output measurement points are also modelled by external blocks.

Figure 2 shows an example of board level modelling. The board delimited by the dashed rectangle is made of three mixed signal blocks $C_1 \cdots C_3$ and a digital block D_1. C_i are single level comparators and D_1 is a controller which checks cyclicly the comparators outputs. Blocks $S_1 \cdots S_3$ represent analogue sources and block MP an output measurement point. This board is used as example thereafter.

In the next section, we describe the specification of individual blocks of a board. Blocks interactions are embedded in block descriptions.

2.2 The block level

At this level, two kinds of functional specifications are used depending on the nature of the block: predefined parametrised functions for analogue and mixed signal blocks and FSM for digital blocks. This approach works well with full documentated boards. A functional model and a test model are derived from the specification and form the internal representation of the block.

Unfortunately, complete board specifications are not always available. This black-box problem arises particularly in maintenance testing, since the documentation of the faulty board is most often missing. In this case, the internal test model has to be provided. This test model is also used as the functional model of the black-box block.

For each block type, functional specifications and internal functional models are described next. Internal test models for each block type are discussed last.

Since we make intensive use of communicating FSM, we explain this aspect first.

2.2.1 Communicating FSM

The block set of the board corresponds to a set of communicating FSM. Since communications are involved, FSM transitions are decorated with labels of the following form:

$$S \rightarrow G[A]$$

where S is a synchronisation condition between two FSM, G a boolean guard and A an optional action. The associated semantics is: "when the synchronisation condition is verified, the boolean guard is then evaluated. If the guard is true, the transition is being crossed and the action is done if any".

The synchronisation condition S can be one of the following expressions:

$$F\,!\,(d_1, d_2, \ldots) \quad (1)$$

$$F\,?\,(d_1, d_2, \ldots) \quad (2)$$

(1) means a not blocking sending of d_i data list towards FSM F.
(2) means a blocking receiving of d_i data list from F. The communication between two FSM is realised with a queue.

2.2.2 Analogue or mixed signal blocks

Functional specifications When the block type is analogue or mixed signal, we select an available parametrised function in a component library. This function describes the block specifications. The library contains the functions for the most common analogue or mixed signal components like voltage sources, analogue first order filters, comparators, DAC, ADC, ... Parameters are associated with functions. For example, the source function (SF) and the comparator function (CF) are chosen for S_i sources and C_i comparators of the board depicted in figure 2.

SF parameters are:

- the signal form (DC, sine, square, ...)
- the signal level
- the signal frequency

and CF parameters are:

- the number of thresholds
- the threshold values
- the tolerance used for the measurement

Internal functional model The internal functional model is derived automatically from the library function. It is expressed with a FSM instanciated from a pattern according to function parameters values. We have currently written patterns for a few common components. As an example, we describe now the functional models for the S_i sources and the C_i comparators. By convention, FSM name is the same as block name.

Figure 3 shows the internal functional model for a source S_i.

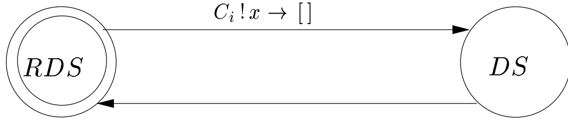

Figure 3: The functional model for a source S_i.

FSM S_i has two states (RDS,DS) and two transitions. RDS and DS stands respectively for ready to send and data sent. S_i goes from the initial state RDS to the DS state, sending the x data (which characterises the signal) to the FSM Ci. Then, S_i goes back to the initial state immediately with a ϵ transition. Thus, S_i provides data to C_i.

Figure 4 shows the internal functional model for a comparator C_i.

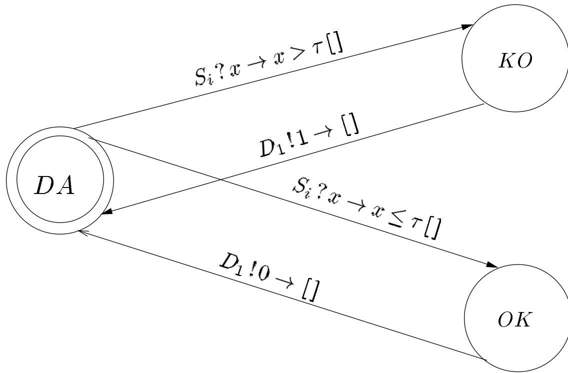

Figure 4: The internal functional model for a comparator C_i.

FSM C_i has three states and four transitions. In the initial state DA, which stands for data acquisition, FSM waits for the x data sent by the FSM S_i. Two different behaviours can occur when the x data is received:

- C_i goes to state OK if x [1] is less than or equal to threshold τ and goes back to the initial state, sending the digital value 0 to the FSM D_1. This case corresponds to allowed values for x.
- C_i goes to state KO if x is greater than threshold τ and goes back to the initial state, sending the digital value 1 to D_1. This case corresponds to forbidden values for x.

2.2.3 Digital blocks

Functional specifications When the block type is digital, the functional block specification is directly written using an FSM. As an example, we describe now the digital part of the board presented in figure 2. Figure 5 shows the functional specification for block D_1, which has three inputs and one output. FSM D_1 has three CC_i states, $1 \leq i \leq 3$, with CC_1 the initial state. CC_i stands for check comparator C_i output. State CC_i has two outgoing transitions towards state CC_{i+1}, $1 \leq i \leq 2$ and state CC_2 has two outgoing transitions towards initial state CC_1. The transitions between state CC_i and state CC_{i+1}, $1 \leq i \leq 2$, and between state CC_2 and CC_1 are the following:

- $C_i?x \rightarrow x == 1[KO_i]$: D_1 goes from state CC_i to state CC_{i+1} when the digital data x has been received and x is equal to one. This means that the input data of C_i has exceeded the threshold τ. The action $[KO_i]$ indicates the faulty behaviour of C_i.
- $C_i?x \rightarrow x == 0[OK_i]$: D_1 goes from state CC_i to state CC_{i+1} when the digital data x has been received and x is equal to zero. This means that the input data of C_i has not exceeded the threshold τ. The action $[OK_i]$ indicates the correct behaviour of C_i.

Thus, D_1 checks cyclicly data sent by the C_i.

[1] In order to present clearer figures in the paper, we now reduce signal characteristics to its amplitude (x). In our method, all signal characteristics are taken into account. This generalisation presents no difficulty.

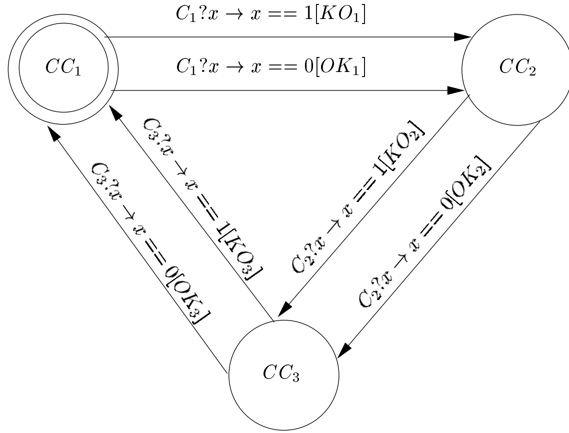

Figure 5: The functional specification and model for block D_1

Internal functional model For a digital block, the internal functional model is the same as its functional description.

2.2.4 Black-box block

When the block specification is unknown, it is not possible to describe it using FSM or parametrised functions. However, by learning on other assumed good boards, a testing engineer can build a sufficient test data set allowing to determine if the block is good or faulty. Each test data describes for the block the output vector according to the input vector. This set of test data makes up the internal functional model for the block which reduced to the test model. Test model is explained next.

2.2.5 Internal test model

Analogue and mixed signal blocks For analogue or mixed signal block, the internal test model is derived from the parametrised function selected in the component library. The internal test model is a FSM using the function parameters. This model integrates the testing engineer skills. For instance, a good single level comparator testing requires two input test data:

- one input test data slightly greater than its threshold value (TD_1)
- one input test data slightly less than its threshold value (TD_2)

Input test data are computed with the threshold and the tolerance for measurement which are parameters of the CF function. Let τ be the threshold and ϵ the tolerance measurement expressed in threshold percent, then input test data are computed as follows:

$$TD_1 = \tau + \delta$$
$$TD_2 = \tau - \delta$$
$$where\ \delta = \tau\epsilon$$

Figure 6 shows the internal test model for C_i comparators.

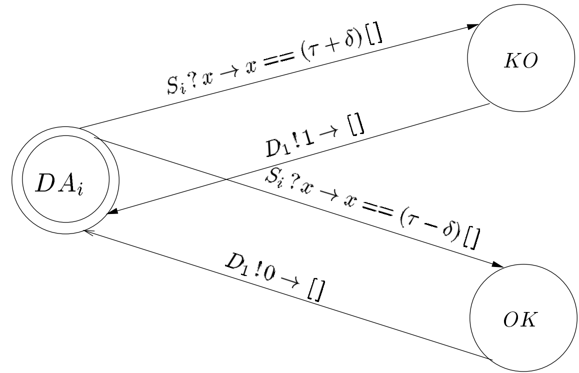

Figure 6: The internal test model for a comparator C_i

The test model thus describes the set of input and output data which are mandatory to decide if comparator C_i is good or faulty. The associated semantics is the transitions coverage of this FSM. In others words, test vectors applied to C_i and associated responses are generated from its test model.

In our example, there is no internal test model for sources since S_i are external to the board and do not need to be tested.

Digital blocks For digital blocks, the internal test model is the same as the internal functional model. Indeed, a digital block can be fully tested using the FSM describing its functional specification. The associated semantics is the transitions coverage of the FSM.

3 Board testing

3.1 Main ideas

Since the board is broken down into a set of blocks, board testing is then realised by testing each block. Nevertheless, that is not unit testing of the blocks because one cannot access directly input/output of blocks. Actually, each block is tested from PIs and POs, using upstream and downstream blocks. It is thus tested in its execution context.

Let DT_{ij} be a test data for a block B_i defined as

$$DT_{ij} = (V_{ij}, R_{ij}) \qquad (3)$$

where V_{ij} is a test vector applied to primary inputs (PIs) and R_{ij} is the associated response at primary outputs (POs).

Let TDS_i be the block B_i test data set defined as the DT_{ij} set :

$$TDS_i = \{DT_{i1}, DT_{i2}, \cdots\} \qquad (4)$$

The board test data set TDS is defined as:

$$TDS = \bigcup_{optimised} TDS_i \qquad (5)$$

The board test data set is the optimised union of all blocks test data sets. Thus, we have to find all TDS_i for board testing. Bock testing is explained next.

3.2 Block testing

Block testing is based on the test model of the block and its associated semantics. Thus, the coverage of the transitions of the FSM determines a path set in the FSM. An inputs/outputs block constraints set is associated to each path. Solving these constraints leads to find ranges of values for block B_i inputs (BV_i) and block B_i outputs (BR_i).

The next step is to propagate upstream BV_i using functional models of upstream blocks. The propagation continues until PIs are reached. In the same manner, BR_i is propagated downstream using functional models of downstream blocks. The propagation continues until POs are reached. This leads to obtain V_i and R_i defined in (3).

As first example, we explain how to obtain the test data set for the comparator C_1 (figure 7). The test model of C_1 has one input and one output. Hatched rounded box represents the internal test model of C_1. Rounded boxes represent the internal functional models of the others blocks. Solid arrows show communications between FSM. Dash arrows show the propagations. Tiny black filled circles and tiny black filled rectangle represent respectively PIs and POs.

Assuming τ is equal to 10 and ϵ is equal to 10%, we find:

$$BV_{11} = 9 \quad ; \quad BR_{11} = 0$$
$$BV_{12} = 11 \quad ; \quad BR_{12} = 1$$

and thus

$$V_{11} = (9, X, X) \quad ; \quad R_{11} = OK_1$$
$$V_{12} = (11, X, X) \quad ; \quad R_{12} = KO_1$$

where X is an unspecified value. The block test data set is:

$$TDS_1 = \{DT_{11}, DT_{12}\}$$

with $DT_{11} = (V_{11}, R_{11})$ and $DT_{12} = (V_{12}, R_{12})$.

That is

$$TDS_1 = \{((9, X, X), OK_1), ((11, X, X), KO_1)\}$$

We give as second example optimised results obtained for the digital block D_1 (figure 8). We find:

$$BV_{21} = (0, 0, 0) \quad ; \quad BR_{21} = OK_1\ OK_2\ OK_3$$
$$BV_{22} = (1, 1, 1) \quad ; \quad BR_{22} = KO_1\ KO_2\ KO_3$$

and thus

$$V_{21} = (9, 18, 27) \quad ; \quad R_{21} = OK_1\ OK_2\ OK_3$$
$$V_{22} = (11, 22, 33) \quad ; \quad R_{22} = KO_1\ KO_2\ KO_3$$

assuming C_i thresholds are respectively equal to 10, 20 and 30 and C_i tolerance is equal to 10 %.

The block test data set is:

$$TDS_2 = \{DT_{21}, DT_{22}\}$$

with $DT_{21} = (V_{21}, R_{21})$ and $DT_{22} = (V_{22}, R_{22})$.

That is

$$\begin{aligned} TDS_2 &= \{((9,18,27), OK_1\ OK_2\ OK_3) \\ &\quad ,((11,22,33), KO_1\ KO_2\ KO_3)\} \end{aligned}$$

In case of black-box blocks, the internal test model is only available and is also used as internal functional model. Consequently, it may occur that data test propagation fails when a black-box block is not the current block to be tested. As a matter of fact, only a restricted part of the behaviour of the block is represented in its test model. This is an important limitation for the black-box modelling.

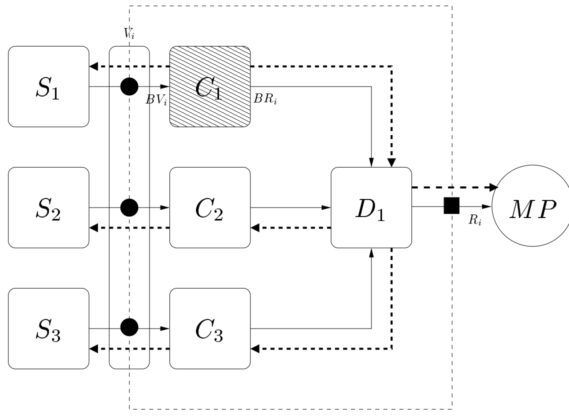

Figure 7: Test data generation for block C_1

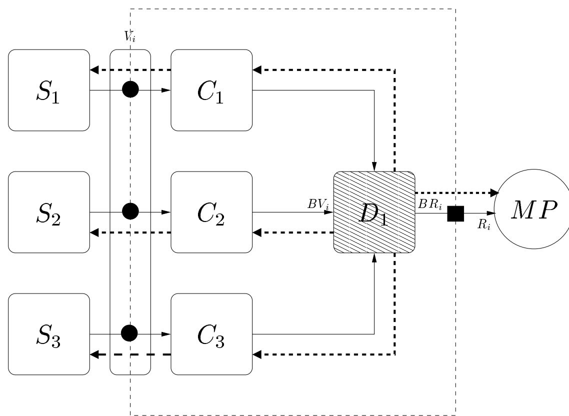

Figure 8: Test data generation for block $D1$

4 Prototype implementation

The method presented in sections 2 and 3 provides a homogeneous framework for the functional modelling and the functional testing of mixed signal boards. From an application point of view, this is translated by an open and homogeneous platform. A prototype of this platform (depicted in Figure 9) is in the process of being realised.

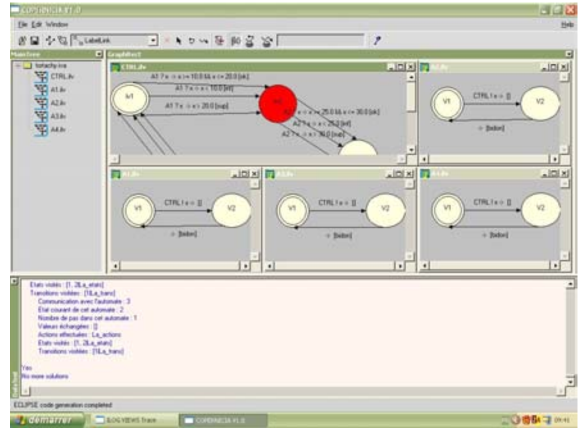

Figure 9: The testing platform prototype

Currently, it allows to model a board as described in section 2. It also allows to conceive and generate test vectors as explained in section 3. The prototype is written mainly in C++.

The part concerning the implementation of test vector generation and propagation is realised in a logical formalism and by using constraint programming, with the ECLiPSe solver [7]. The constraint programming is very relevant for test data generation [8] [9] since it works initially on range of test data values. Test data generated are instanciated the less needed. The automatic test equipment (ATE) may then instantiate particular values taking into account specific hardware requirements.

The part concerning the graphical user interface is realised with the ILOG Views C++ library [10].

The prototype was used for an industrial case study: the Tachy board which is a mixed signal board. This board has fourteen analogue channels receiving signals coming from tachymetric generators. The global function of the board is

to check in a cyclic manner the analogue channels by comparing them to thresholds (tension) and to register in RAM memory the number of each faulty channel. A fault is represented by the fact that the thresholds have been exceeded.

The Tachy board is modelled like the board quoted as example in sections 2.1 and 2.2, but using fourteen sources and fourteen comparators. We have modelled this board and automatically generated its test vectors as explained in sections 2 and 3.

This project benefits from a cooperation with the Brest Subsidiary ISIS-MPP, specialised in test and tools for board testing.

5 Conclusion and future work

We have presented a method for mixed signal boards testing. Some aspects seem to have been accepted such as the homogeneous modelling formalism, the board modelling methodology and the test vectors generation approach. The method and prototype have been validated on an industrial case study. Nevertheless, other examples are needed to investigate more precisely limits and advantages of the method.

We are looking further to generalise our approach to other types of mixed signal boards. This work should lead to the development of automatic (or semi-automatic) tools, managing the testing and maintenance stages and adaptable to different and specific types of boards. This kind of tool would respond to a real need in the industrial community.

Acknowledgements

We wish to thank Michel Le Goff from ISIS-MPP for his availability and his willingness to clarify matters on the industrial practices involving mixed signal tests.

References

[1] Spyros Xanthakis, Michel Maurice, Antonio De Amescua, Olivier Houri, and Luc Griffet. *Test & Contrôle des Logiciels : Méthodes, Techniques & Outils*. EC2, 1994.

[2] A. K. Majhi and V. D. Agrawal. Mixed-Signal Test. In *Proc. 11th International Conf. VLSI Design*, pages 285–288, 1998.

[3] B. Ayari, N. BenHamida, and B. Kaminska. Automatic Test Vector Generation for Mixed-Signal Circuits. In *Proc. The European Design and Test Conference*, pages 458–463, 1995.

[4] R. Ramadoss and M. L. Bushnell. Test Generation for Mixed-Signal Devices Using Signal Flow Graphs. In *Proc. 9th International Conference on VLSI Design: VLSI in Mobile Communication.*, pages 242–247, January 1996.

[5] Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems. *Hybrid Systems I, Lecture Notes in Computer Science*, 736:209–229, 1993. Springer-Verlag.

[6] Alberto Bemporad and Manfred Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.

[7] Andrew M. Cheadle, Warwick Harvey, Andrew J. Sadler, Joachim Schimpf, Kish Shen, and Mark G. Wallace. Eclipse: An introduction. Technical Report IC-Parc-03-1, IC-Parc, Imperial College London, 2003.

[8] Richard A. Demillo and A. Jefferson Offutt. Constraint-Based Automatic Test Data Generation. *IEEE Transactions on Software Engineering*, 17(9):900–910, September 1991.

[9] Bogdan Korel. Automated Software Test Data Generation. *IEEE Transactions on Software Engineering*, 16(8):870–879, August 1990.

[10] ILOG. *ILOG Views Fundation 5.0 User's Manual*, 2002.