Modeling and Generation of Test Patterns for Mixed-Signal Boards: Dealing With Basic Signals

Bertrand Gilles, Laurent Tchamnda Nana, Valérie-Anne Nicolas

Modeling and Generation of Test Patterns for Mixed-Signal Boards: Dealing With Basic Signals

Bertrand Gilles Laurent Nana Valérie-Anne Nicolas
EA3883 - LISyC - Département Informatique - Université de Bretagne Occidentale
20 avenue Le Gorgeu - CS 93837 - 29238 Brest Cedex 3 France
{gilles|nana|vnicolas}@univ-brest.fr

Abstract

In the context of maintenance testing and diagnosis of faulty boards, a functional FSM (Finite State Machine)-based model for mixed-signal board has been introduced. The board is broken down into interconnected functional blocks. Each block has an associated functional model which describes its behavior and a test model which specifies how the block can be efficiently tested. A test model for a block is created by merging its functional model and a chosen test strategy. The board checking consists in testing each block individually using its test model and functional models of other blocks. Test patterns for a component are generated by covering the transitions of its test model and propagation to primary inputs/outputs through functional models of other blocks. The board test data set is built by using the test patterns for all the blocks of the board. The current improvement of the method deals with basic signals.

Keywords: *maintenance testing, modeling, mixed-signal boards, basic signals.*

1. Introduction

Numerous test methods and techniques have been developed for circuit test [1, 2] associated to the different stages of product life-cycle, mainly at design and production levels. Surprisingly, not much interest has been thrown into testing during the maintenance stage. However, maintenance testing has its own specificity. Thus, our work is related to maintenance testing and focus more particularly on mixed-signal boards.

The maintenance stage is one of the steps constituting the life-cycle of a board. This stage is specific and complex because of its location after the development/production cycle and because it spawns over long periods of time. First, since boards are embedded into systems in the field, and because of the level of confidentiality often required (military, commercial aspects), the knowledge about the board is reduced for maintenance people (no designer direct knowledge and partial documentation). Second, maintenance stage targets long-life boards.

In-circuit testing and boundary scan [3, 4] enable structural tests at component level and interconnection tests. These techniques are not sufficient in the maintenance stage where functional testing is needed in order to check the board behavior during long time, and to determine and replace faulty components in case of defective functionality (diagnosis). This is especially true in the military field where safety and longevity for boards are highly required. Maintenance stage differentiates from design or production stages where functional test at board level is not used because test software development is costly. Actually, functional tests are mainly achieved at system level in order to check if the whole system meets the requirements.

At present time, maintenance test engineers have to face various situations. Different testing situations exist, ranging from testing of well known boards (original data, documentation and diagrams) to that of unknown boards (no information). To our knowledge, there is no general tool for handling such a wide panel of situations (see section 2). Practitioners need large background and experience since they have sometimes to work empirically, processing large cases by hand with high development costs. Clearly, dedicated tools are needed to guide or automate at least a part of the work involved in the maintenance stage. The work presented in this paper goes in that direction. Our goal is to provide a help to board maintenance testing and diagnosis.

We propose a method supported by a semi-automatic tool allowing the functional specification of the board, the definition of test strategies and the automatic test data set generation. Because automation relies on formalization, a formalism has to be chosen to match practitioners background in order to be really useful. 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 testing. 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). Talking with our industrial partner, we chose the FSM formalism which is well known to test engineers.

We first present some related work, then we introduce the FSM-based functional model for mixed-signal boards and the associated ATPG (Automatic Test Pattern Generation). Thereafter, we present our approach for basic signals through a simple case study as well as the results obtained. The implementation prototype is described next. A discussion on future works ends the paper.

2. Related work

We describe in this section some work about functional test generation. In [5], a functional ATPG based on first order sensitivity calculations has been presented. The first order sensitivity represents the relation between the elements of the circuit and output parameters (performance). Component deviations are deduced by measuring various output parameters, and through sensitivity analysis and tolerance computation. This method considers only analog circuits. An automatic test vector generation approach for functional testing of mixed-signal circuits has been presented in [6]. Analog blocks of the circuit are tested using first order calculations mentioned above. ATPG for digital blocks is achieved using a technique based on binary decisions diagrams and boolean difference. In [7], the authors have developed a functional mixed-signal ATPG algorithm that uses signal flow graphs (SFG) and reverse simulation. SFG is used to represent graphically the equation of a circuit. These two different methods assume that the structure of the circuit is well known. As explained in the previous section, maintenance test engineers may have a few, or even no information about the circuit. Even if assuming that redrawing the diagram of the circuit is feasible, this implies a high cost activity.

Some functional ATPG algorithms based on a high level description language such as VHDL have been developed. In [8], an extended finite state machine (EFSM) is extracted from a behavioral description. Functional tests are generated by traversing all the transitions of the extracted EFSM. This ensures that every statement in the behavioral description is exercised at least once. In [9], a control flow graph (CFG) is built from the VHDL description. Significant control flow paths are selected. Constraints are associated to these paths and test data are generated using constraint logic programming (CLP). These methods target only digital circuits. They do not address the problem of mixed-signal circuits testing.

After this overview of related work, the next section deals with our approach proposed for the modeling and the ATPG of mixed-signal boards.

3. FSM-based board modeling and ATPG

In [10], we proposed a board modeling for maintenance ATPG. It relies on FSM-based functional models for the components of mixed-signal boards. Two hierarchical levels of modeling are considered: Board level and block level. Both are used for our ATPG. The modeling levels are first presented, then the ATPG is described.

3.1. Board level modeling

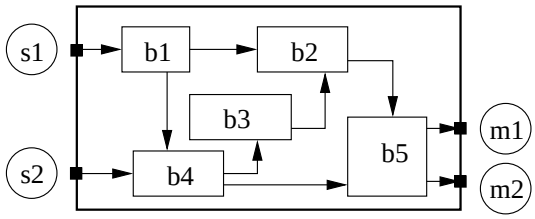

Figure 1. A board is an assembly of blocks

The board is first modeled at the *board level*, as a set of interconnected functional blocks, as depicted in figure 1. In addition to the building blocks of the board, some external blocks are needed to model connections between the board primary inputs/outputs (PI/PO) and an automatic test equipment (ATE): External sources which supply input signals (blocks $S1$ and $S2$ in figure 1) and output measurement points (blocks $m1$ and $m2$ in figure 1).

Blocks are analog, digital or mixed-signal, and may have several inputs and outputs. Oriented links denote data exchanges between components.

3.2. Block level modeling

Each block has an associated *functional model* which describes its behavior and a *test model* which specifies how the block can be efficiently tested. We have chosen communicating finite state machines (CFSM) [11] as the modeling formalism for the blocks. Two CFSM interact when one CFSM produces an output that is placed in the input queue of the other. CFSM are often used for modeling systems involving communicating processes [11, 12]. Each block is modeled by one or more CFSM, depending on the complexity of the functionality of the block. Blocks interactions are embedded into blocks descriptions.

We first present CFSM, then we present the two different models. The use of the functional and test models is described in the model-based ATPG section.

Communicating FSM The blocks of the board are represented by a set of communicating FSM. Since communications are involved, CFSM transitions are decorated with labels of the form $S \rightarrow G[A]$ where S is an optional synchronization condition between CFSM, G an optional boolean guard and A an optional action. The associated semantics is: "when the synchronization condition is verified, the boolean guard is then evaluated. If the guard is true, the transition is crossed and the action is done".

The synchronization condition S may be an expression such as $A\,?\,(d_1, d_2, \ldots)$ which means a blocking receiving of d_i data list from CFSM A.

The communication between two CFSM is realized with a queue. Sendings are allowed in actions since they are not blocking. Expression $cst\ :\ A\,!\,(d_1, d_2, \ldots)$ means a non blocking sending of d_i data list towards CFSM A. cst is an optional constraint which applies on d_i data list.

A d_i data may represent a scalar value (a DC signal for example) or a more complex signal.

The functional model The functional model of each component is made of one or more CFSM. The model may be specified with an appropriate graphical user interface or instantiated from a parameterized functions library. The former is used for digital components and the latter is mainly used for common analog or mixed-signal blocks. Test vector lists are also usable. Sometimes, these are the simplest way for specifying blackbox-like blocks functionalities. All of this is achieved using our prototype tool described in section 5.

These specification techniques may be mixed, according to the nature of the board components, and to the kind and form of available descriptions for the different blocks. Functional specifications may also be adjusted to fit the desired testing grain.

The test model To generate appropriate test vectors for a given component, test strategies are applied to the functional model [13]. This is realized mainly by extending the functional model at input/output points (I/O) with new CFSM implementing the test strategy. The test model for a component results from this merging, as depicted in figure 2.

3.3. Model-based ATPG

With our method (see figure 2), testing the board consists in testing each block individually using its associated

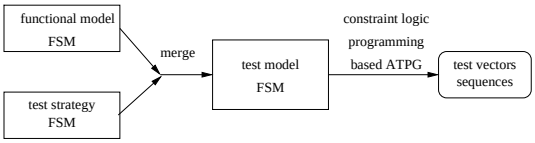

Figure 2. The test pattern generation process

test model. Test patterns for a block (BTP) are then generated by carrying out the transition coverage (a commonly used fault model [14]) of its test model. We chose transition coverage rather than path coverage in order to avoid the path explosion problem and because it provides a reasonable fault coverage [11]. Nevertheless, since a path represents a behavior of the board, it may be of interest to test a given path. We have thus included a facility for generating test patterns that targets a specific path in the board. Transitions identify a set of constraints which are associated to the behavior of a block. These constraints are translated and solved using constraint logic programming (CLP), leading to the test patterns generation.

Since the block under test is often embedded within the board, without any test access mechanism (e.g. block b3 of picture 1), the functional models of adjacent blocks are used for forward propagation to PO and backward propagation to PI. During these propagations, BTP have to satisfy the constraints associated to the adjacent functional models in order to compute the final test patterns. Finally, a board test data set is the union of test patterns for all the blocks of the board.

CLP is very relevant for test data generation [15, 16] since test data are represented in a symbolic way, using ranges of values. These ranges of values deal efficiently with analog and digital data representations in a uniform way. Ranges of vectors are computed for reaching the test requirements. Actual values are defined at the end, making it possible to take into account some ATE specificities.
In our previous work, we dealt only with DC signals. However, most of the time, basic signals like sine or rectangular signals have to be used for board testing. Taking such signals into account requires to model them and to extend our functional and test models, and this leads to more complex models. This is illustrated on the case study presented in the next section where an analog DC input signal is not adapted for the testing.

4. Case study

The modeling technique, extended for dealing with basic signals has been applied to a simple case study. We first give a functional description of the *Test Case Board* (TCB),

then we present its modeling at board and block levels. The test strategies applied are then described and finally the expected board test data are given.

4.1. Board Description

The TCB is a mixed-signal board that has one analog channel. The main function of the board is to check in a cyclic way the instantaneous voltage of the input signal by comparing it to a threshold. The result of the comparison is a time stamped logical value written into a RAM memory. An analog high-pass filter prevents from inputing a DC signal.

4.2. Board level modeling

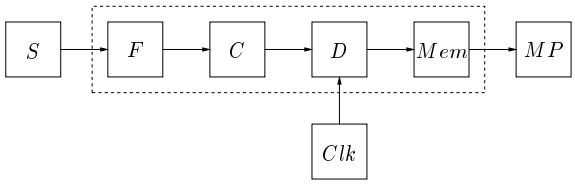

Figure 3. The TCB level modeling

Figure 3 shows the TCB level modeling (see section 3.1). The board is delimited by the dashed rectangle and is made of one analog block F, one mixed-signal block C, one digital block D and one digital block Mem. F is a first order high-pass analog filter, C is a comparator, D is the controller which checks cyclicly the comparator output and Mem is a memory. Block S represents the analog source and block MP represents a digital measurement point. Block Clk represents a clock signal.

4.3. Block level modeling of the TCB

As mentioned in section 3.2, each block's functionality is modeled by a CFSM. By convention, CFSM name is the same as block name. Before presenting the functional models of the different blocks of the TCB, we first present the representation of the signals involved in the board modeling. The modeling of the board inputs/outputs is then presented. Thereafter we present the modeling of the analog part (filter) and the mixed-signal part (comparator), and finally we tackle the modeling of the digital part (controller and memory).

Signals and representation We consider two types of signals involved in the board modeling: the analog sine signal and the digital rectangular wave signal.

$s(t-t_0)$ is T_0-periodic delayed sine signal analytically defined as:

$$s(t-t_0) = A\sin(2\pi F_0(t-t_0)) \quad (1)$$
$$= A\sin(2\pi F_0 t - \phi_0) \quad (2)$$

with $\phi_0 = 2\pi F_0 t_0$, where F_0 is the signal frequency and t_0 the delay time.

$rw(t-t_0)$ is a T-periodic delayed rectangular wave analytically defined on the interval $[t_0, t_0+T[$ as:

$$rw(t-t_0) = \begin{cases} 1 \ if \ t \in [t_0, t_0+\Delta T_1] \\ 0 \ if \ t \in]t_0+\Delta T_1, t_0+\Delta T_0+\Delta T_1[\end{cases} \quad (3)$$

where t_0 is the delay time, with the obvious relation:

$$T = \Delta T_0 + \Delta T_1 \quad (4)$$

This signal is shown figure 4.

Signals are represented by a *structure* called *sig* which is an aggregate with a number of fields called its *arguments*. The number of arguments (called the arity) depends on the type of the signal to be represented. Each argument is a parameter of the signal. By convention, the first argument is named *type* and corresponds to the type of signal.

The structure (5) represents a generic sine signal where the type field is always set to the value *sine*. The other fields correspond to the characteristics of the signal (magnitude, frequency and phase). Hence, signal (2) is represented by an instance of this structure as shown in (6).

$$sig(sine, ampl, frq, phi) \quad (5)$$

$$sig(sine, A, F_0, \phi_0) \quad (6)$$

Similarly, structure (7) represents a generic rectangular wave signal where the type field is always set to the value *rw*. Signal (3) is represented by an instance of this structure, as shown in (8).

$$sig(rw, dt1, prd, dly) \quad (7)$$

$$sig(rw, \Delta T_1, T, t_0) \quad (8)$$

The dot operator is used to access the fields of the structure.

Board inputs/outputs In order to generate test data and program, we also need to model the data sources and measurement tools.

Figure 5 presents the functional model of the analog S source. CFSM S sends the x data (which characterises the signal) to CFSM F.

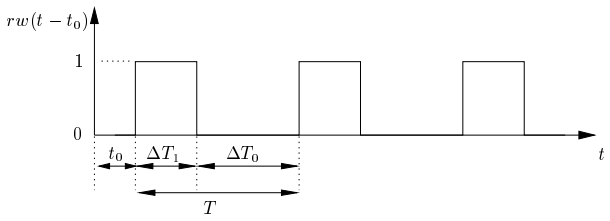

Figure 4. Rectangular wave signal

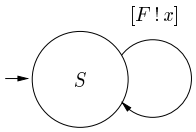

Figure 5. The functional model of the source S

Figure 6 shows the functional model of the clock signal. CFSM Clk sends periodically (with period T_e) a time stamped top event to CFSM D. This behavior is achieved by using the y reference clock and a constraint where $\%$ represents the modulo operator.

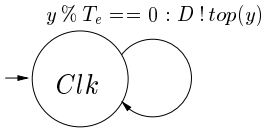

Figure 6. The functional model of the clock signal

Concerning outputs modeling, figure 7 shows the functional model of the measurement point MP. CFSM MP just waits for the time stamped data coming from CFSM Mem (x represents the data and z represents the dating).

Analog part The analog part of the board is modeled by the F CFSM depicted in figure 8. This CFSM describes the behavior of the first order high-pass filter when the input stimulus is an analog sine signal. Thus, CFSM F waits for the x data coming from CFSM S and sends a sine signal to CFSM C with a set of constraints $\Gamma = \{\gamma_1, \gamma_2, \gamma_3\}$ that express the ouput signal characteristics (attenuation, frequency, and phase shift). Elements of set Γ are given by:

$$\begin{cases} \gamma_1 = (V == \frac{x.ampl}{\sqrt{1+\frac{f_c^2}{F_0^2}}}) \\ \gamma_2 = (F_0 == x.frq) \\ \gamma_3 = (\phi_0 == x.phi + \arctan\left(\frac{f_c}{F_0}\right)) \end{cases} \quad (9)$$

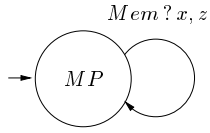

Figure 7. The functional model of the measurement point MP

where $==$ is the equality constraint and f_c is the cutoff frequency.

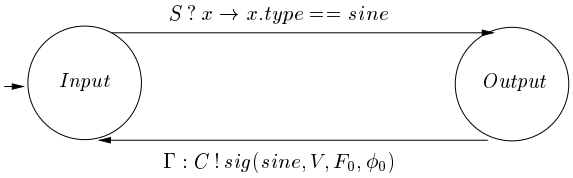

Figure 8. The functional model of the analog first order high-pass filter F

Mixed-signal part The mixed-signal part of the board is modeled by the C CFSM.

Figure 9 presents the functional model of the C comparator. This CFSM C describes the behavior of the comparator when the input stimulus is an analog sine signal. Thus, CFSM C waits for the x data coming from CFSM F. Two different behaviors may occur when the received data x is a sine signal : C sends the digital constant value 0 to CFSM D if the amplitude of x is less than the threshold S. Otherwise, a rectangular wave signal is sent to CFSM D, with the set of constraints Γ defined as:

$$\begin{cases} \gamma_1 = (T_0 == \frac{1}{x.frq}) \\ \gamma_2 = (\Delta T_1 == \frac{1}{2\,x.frq} - \frac{1}{\pi\,x.frq}\arcsin\left(\frac{S}{x.ampl}\right)) \\ \gamma_3 = (t_0 == \frac{1}{2\pi\,x.frq}\arcsin\left(\frac{S}{x.ampl}\right) + \frac{x.phi}{2\pi\,x.frq}) \end{cases} \quad (10)$$

Digital part The digital part of the board is modeled by two communicating FSM: One modeling the digital controller of the board and one for the memory. We explain first the modeling of the controller and next the modeling of the memory.

Figure 10 shows the functional model of the D controller. When a time stamped top event is received from CFSM Clk, output data sent by the comparator C is read. Output data and time stamp are then sent to CFSM Mem and D waits for the next top. As we may see, CFSM D reads in a cyclic way the output of the comparator.

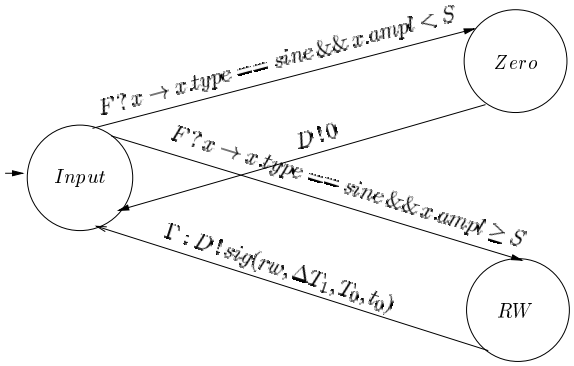

Figure 9. The functional model of the comparator C

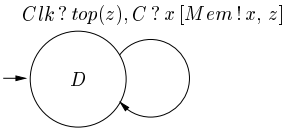

Figure 10. The functional model of the controller D

The functional model of the memory Mem is depicted in figure 11. CFSM Mem waits for a time stamped data from CFSM D and sends the received data (the time stamped output value of the C comparator) to CFSM MP.

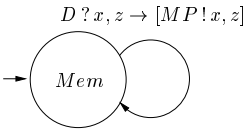

Figure 11. The functional model of the memory Mem

After this presentation of the functional model of the board, the next section deals with test strategies and test models.

4.4. Test strategies and test models

As previously mentioned, a component test model is obtained by merging the block functional model and its possible associated test strategy.

For digital blocks such as the controller D and the memory Mem, no specific test strategy is often needed and the functional model is the default test model. Indeed, the transition covering of the functional model CFSM is often sufficient for testing the component. When the digital default test model is not sufficient, we apply test strategies. An example of such test strategy applied to the measurement point is shown in figure 12. This test strategy assumes the synchronization between the analog input sine signal and the digital part.

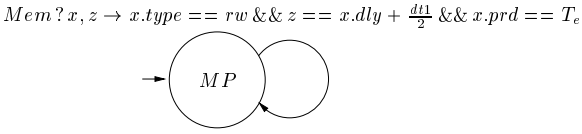

Figure 12. The test model of the measurement point MP (functional model with an added testing strategy)

Unlike digital blocks, analog and mixed-signal blocks often need a test strategy. In the case of the comparator and the filter, the test models result from a restriction of the functional model parameters (choice of two values in the set of possible magnitude (respectively frequency) values).

The test model of the comparator is depicted in figure 13 where δ is a tolerance. The set of constraints Γ is the same as (10). Figure 14 shows the test model proposed for the

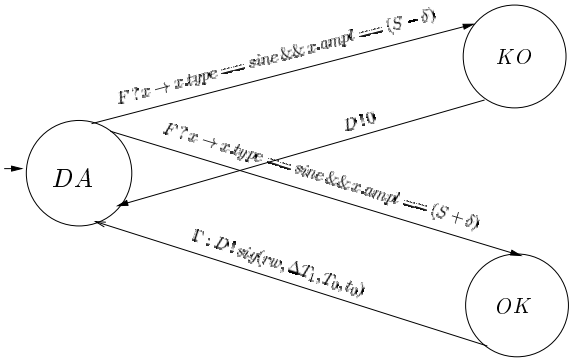

Figure 13. The test model of the comparator C

filter. This test model define two test data (one in the bandwith and one at the cutoff frequency). The associated set of constraints Γ_1 and Γ_2 are respectively defined as:

$$\begin{cases} \gamma_1 = (F_0 == x.frq) \\ \gamma_2 = (V \leq x.max + \delta_1) \\ \gamma_3 = (V \geq x.max - \delta_1) \\ \gamma_4 = (\phi_0 \leq 0 + \delta_2) \\ \gamma_5 = (\phi_0 \geq 0 - \delta_2) \end{cases} \quad (11)$$

and

$$\begin{cases} \gamma_1 = (F_0 == x.frq) \\ \gamma_2 = (V \leq x.max\frac{\sqrt{2}}{2} + \delta_3) \\ \gamma_3 = (V \geq x.max\frac{\sqrt{2}}{2} - \delta_3) \\ \gamma_4 = (\phi_0 \leq \frac{\pi}{4} + \delta_4) \\ \gamma_5 = (\phi_0 \geq \frac{\pi}{4} - \delta_4) \end{cases} \quad (12)$$

where δ_1, δ_2, δ_3 and δ_4 define tolerance boxes.

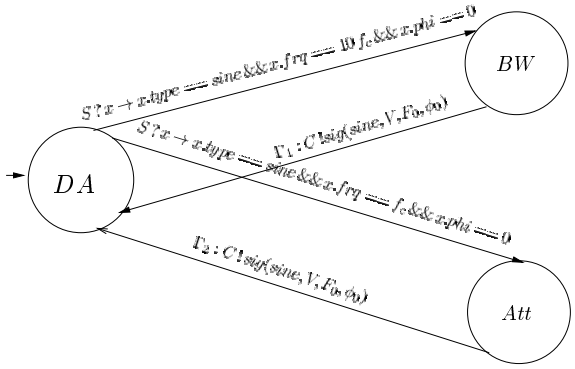

Figure 14. The test model of the filter F

4.5. Board Testing

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).

We present next the test data set obtained respectively for the filter, the comparator and the digital part.

The test data set of the filter is: $TDS_{filter} = \{TD_1, TD_2, TD_3, TD_4\}$ with:

$$TD_1 = \begin{array}{ll} (In = & (sig(sine, V_{IN}, 10\,f_c, 0), top(z)), \\ Out = & ([0, z])) \end{array} \quad (13)$$

with $V_{IN} + \delta_1 < S$ and $z = T_e$, where δ_1 is the tolerance in (11) and S is the threshold of the comparator.

$$TD_2 = \begin{array}{ll} (In = & (sig(sine, V_{IN}, f_c, 0), top(z)), \\ Out = & ([0, z])) \end{array} \quad (14)$$

with $\frac{\sqrt{2}}{2} V_{IN} + \delta_3 < S$ and $z = T_e$, where δ_3 is the tolerance in (12)

$$TD_3 = \begin{array}{ll} (In = & (sig(sine, V_{IN}, 10\,f_c, 0), top(z)), \\ Out = & ([v, z])) \end{array} \quad (15)$$

with $V_{IN} - \delta_1 \geq S$ and $z = T_e$. The value v is the value of the signal $sig(rw, \Delta T_1, T_0, t_0)$ at time z. This signal is computed from the input signal $sig(sine, V_{IN}, 10\,f_c, 0)$ by solving the set of constraints (11) and (10).

$$TD_4 = \begin{array}{ll} (In = & (sig(sine, V_{IN}, f_c, 0), top(z)), \\ Out = & ([v, z])) \end{array} \quad (16)$$

with $\frac{\sqrt{2}}{2} V_{IN} - \delta_3 \geq S$ and $z = T_e$. The value v is the value of the signal $sig(rw, \Delta T_1, T_0, t_0)$ at time z. This signal is computed from the input signal $sig(sine, V_{IN}, f_c, 0)$ by solving the set of constraints (12) and (10).

The test data set of the comparator is: $TDS_{comp} = \{TD_5, TD_6\}$ with:

$$TD_5 = \begin{array}{ll} (In = & (sig(sine, S - \delta, F_{IN}, \phi_{IN}), top(z)), \\ Out = & ([0, z])) \end{array} \quad (17)$$

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

$$TD_6 = \begin{array}{ll} (In = & (sig(sine, S + \delta, F_{IN}, \phi_{IN}), top(z)), \\ Out = & ([v, z])) \end{array} \quad (18)$$

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. This signal is computed from the input signal $sig(sine, S + \delta, F_{IN}, \phi_{IN})$ by solving the set of constraints (10) and (9).

The test data set of the digital part obtained, using the test strategy, is: $TDS_{digital} = \{TD_7\}$ with:

$$TD_7 = \begin{array}{ll} (In = & (sig(sine, V_{IN}, \frac{1}{T_e}, \phi_{IN}), top(z)), \\ Out = & ([1, z])) \end{array} \quad (19)$$

with $z = k\,T_e$. The input signal $sig(sine, V_{IN}, \frac{1}{T_e}, \phi_{IN})$ is computed by solving the set of constraints (10) and (9) with the additional constraint $T_e = t_0 + \frac{\Delta T_1}{2}$.

5. Prototype

We have partially implemented the FSM-based board modeling and the model-based ATPG in a prototype tool named *Copernicia*. This prototype provides a GUI allowing high level description of a mixed-signal board. In addition, the GUI includes some facilities for the choice of the test strategy, for the description of the board-ATE connection and for the description of the data flow. The GUI part of the prototype is written in C++ with the ILOG Views graphic library [17] and the ATPG part is implemented using CLP with the solver ECL^iPS^e [18]. The prototype, which is

still under development, has already been used in simple industrial case studies involving mixed-signal boards [13, 19]. We currently focus on test data generation algorithms.

6. Conclusion and future work

We have presented a method for testing mixed-signal boards in a maintenance context. This method has been improved using basic signals. We are currently focusing on test data generation algorithms. We are also prospecting for improved test strategies. Another objective is to extend the models to take into account more complex boards. Further work is required on industrial cases to validate the approach or exhibit its limits.

Acknowledgements

We wish to thank Bruno Castel and Michel Le Goff from ISIS-MPP compagny for their contribution to this project.

References

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

[2] F. Azaïs, S. Bernard, Y. Bertrand, M.-L. Flottes, P. Girard, C. Landrault, L. Latorre, S. Pravossoudovitch, M. Renovell, and B. Rouzeyre. *Test de circuits et de systèmes intégrés*. Hermes Science, 2004.

[3] *IEEE STANDARD 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture*. IEEE Standards Board, New York, NY 10017, 1990.

[4] *IEEE STANDARD 1149.4-1999, IEEE Standard for a Mixed Signal Test Bus*. IEEE Standards Board, New York, NY 10017, 1999.

[5] N.B. Hamida and B. Kaminska. Analog Circuit Testing Based on Sensitivity Computation and New Circuit Modeling. In *International Test Conf.*, pages 652–661, October 1993.

[6] 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.

[7] R. Ramadoss and M. Bushnell. Test Generation for Mixed-Signal Devices using Signal Flow Graphs. In *9th Internation Conf. on VLSI Design*, pages 242–248, January 1996.

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

[9] 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.

[10] B. Gilles, V.-A. Nicolas, L. Lemarchand, L. Marcé, and B. Castel. Towards a New Modelling of Mixed-Signal Boards For Maintenance Testing. In *11th IEEE International Mixed-Signals Testing Workshop, IMSTW'05*, pages 90–97, June 2005.

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

[12] D. Brand and P. Zafiropulo. On Communicating Finite-State Machines. *Journal of Association for Computing Machinery*, 30(2):323–342, April 1983.

[13] V.-A. Nicolas, B. Gilles, L. Lemarchand, L. Marcé, and B. Castel. A Maintenance-Oriented Board Testing Approach. In *3rd IEEE International East-West Design and Test Workshop, EWDTW'05*, pages 143–147, September 2005.

[14] I.G. Harris. Fault Models and Test generation for Hardware-Software Covalidation. *IEEE Design & Test of Computers*, pages 40–47, July-August 2003.

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

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

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

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

[19] B. Gilles, V.-A. Nicolas, L. Lemarchand, and L. Marcé. Timed Functional Modeling for Mixed-Signal Boards in Maintenance Testing. In *7th IEEE Latin-American Test Workshop, LATW 2006*, March 2006.