



# Maintenance Testing of Mixed-Signal Boards: the FCB case study

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

## ► To cite this version:

Valérie-Anne Nicolas, Bertrand Gilles, Laurent Lemarchand, Lionel Marcé, Bruno Castel. Maintenance Testing of Mixed-Signal Boards: the FCB case study. 2005. <hal-00607368>

**HAL Id: hal-00607368**

**<https://hal.univ-brest.fr/hal-00607368>**

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.

# Maintenance Testing of Mixed-Signal Boards: the FCB case study

Technical Report LiSys, UBO Brest (France), June 2005

VALERIE-ANNE NICOLAS<sup>1</sup> BERTRAND GILLES<sup>1</sup> LAURENT LEMARCHAND<sup>1</sup>  
LIONEL MARCÉ<sup>1</sup> BRUNO CASTEL<sup>2</sup>

<sup>1</sup>EA3883-LISYC/ Département Informatique, Université de Bretagne Occidentale  
20 avenue Le Gorgeu - CS 93837 - 29238 Brest Cedex3, France  
Tel:+33 298 016 957, Fax: +33 298 018 011  
{vnicolas|gilles|lemarch|marce}@univ-brest.fr

<sup>2</sup>ISIS-MPP  
Rue Pierre Rivoalon - Technopôle Brest Iroise - 29200 Brest, France  
Tel:+33 298 341 600, Fax: +33 298 340 986  
bruno.castel@isismpp.fr

**Keywords:** Maintenance testing, mixed-signal board testing, modelling, automatic test pattern generation, diagnosis.

## Abstract

We present an ongoing work in the domain of mixed-signal board maintenance testing, supported by an industrial case study. We propose a method providing a semi-automation and an help for the board maintenance testing and diagnosis stages. It is validated by the implementation of a prototype tool.

## 1 MOTIVATION

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. The development of communication and multimedia technologies, the constant use of onboard technologies (transports, mobile telephony, ...) require more and more efficient and safe mixed-signal components. There is therefore a need for more adapted, reliable and safe tests. In this context, it is crucial to master more efficiently the testing stages of mixed-signal boards.

A large amount of work has already been done concerning testing during conception phases (design for testability) and production. The main part of these works deals with digital boards, some others with analogue boards, and quite a few with mixed-signal boards [1,2]. As a result, tools exist for digital boards testing, analogue boards testing, or some specific electronic components testing. One may notice that, surprisingly, not much

interest has been thrown into testing during the maintenance stage. Indeed, the maintenance process has its own specificities and testing problematic, different from these of conception or production. Our work is related to this aspect and concerns more particularly the maintenance testing of mixed-signal boards.

First, we explain the specificity of maintenance testing. Then, we sketch the main ideas of our method in this context. Next, some more details on the method and its application via a prototype for mixed-signal boards testing are given. Results on an industrial case study are then presented. A discussion on future work ends the paper.

## 2 MAINTENANCE CONTEXT

The maintenance stage is one of the step constituting the life cycle of a manufactured product, which begins after the development/production cycle, and thus does not belong to it. This cut is not artificial and explains why methods conceived for the development stage are not well suited to maintenance. For example, when a failure occurs in a product in use for several years, it is often long and difficult to repair it. Although it is always possible to throw away a faulty component/board and just replace it by a new good one in the case of cheap large series, there are still some more complex, particular technology, valuable components/boards that really need to be repaired. Actually, in most cases, specifications and test data for the product are missing. It is also rare that the development/production team for this product is still there to help. The one in charge of the diagnosis and repair has then to face a complex

problem, often in an empirical way, with no tool available to guide or automate at least a part of his work.

In maintenance, one may distinguish preventive maintenance from corrective maintenance. Corrective maintenance is, as in the above example, to be able to deal with failures when they occur, sometimes very quickly. Preventive maintenance consists in checking periodically the correct behaviour of components, and may seem more time-consuming than corrective maintenance. However, by anticipating incorrect behaviours, it allows to guarantee a higher level of confidence in the product and thus a lower need for corrective maintenance.

As far as testing is concerned, corrective maintenance may be viewed as a local approach whereas preventive maintenance is more global. In the first case, tester is interested in one behaviour (to reproduce the target failure), and in the second case, he has to check the set of all behaviours of the board. Independently of their nature, these two kinds of maintenance are in great need of generic testing methods and tools.

The method we introduce next was first designed for corrective maintenance needs, and offers a well-adapted precision level according to this aim. In adds, it may also constitute an help for preventive maintenance.

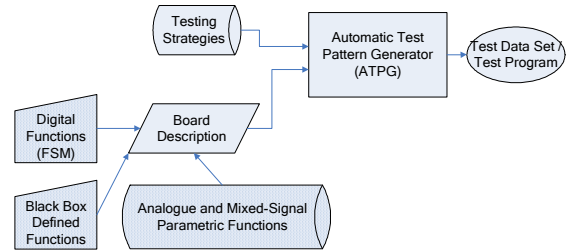
### 3 MAIN IDEAS

Our final goal is to propose a method, supported by a (semi-)automatic tool, providing an help to the board maintenance testing and diagnosis stages. We thus have to formalise the testing and diagnosis data and processes, taking into account the industrial context and practices. Two main aspects are to be considered. First, mixed-signal boards testing practitioners have some know-how related to their experience and skills in the testing process. We think that this knowledge has to be included in the tool. Second, the tool must match the background of testing practitioners in order to be really useful and used. This implies formalisms which are intuitive or well-known to testing engineers. Because of the lack of tools dealing with mixed-signal components, we pay special attention to the modelling and testing of mixed-signal boards.

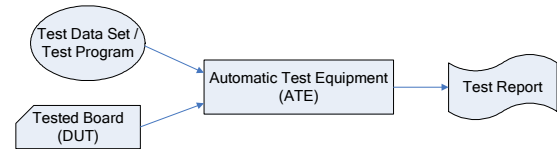
Figure 1 and Figure 2 sketch our complete method process flow.

Figure 1 shows how to generate a test data set for a board given some testing strategies and a description of the (correct) board. This test data set may then be automatically translated into a test program expressed in an Automatic Test Equipment (ATE) language. The test program is

the test data set instrumented with ATE primitives and an oracle. The oracle compares the real outputs of the test with the expected ones, and is thus able to decide and list which tests fail.



**Figure 1: The test data set generation process flow**



**Figure 2: The board testing process flow**

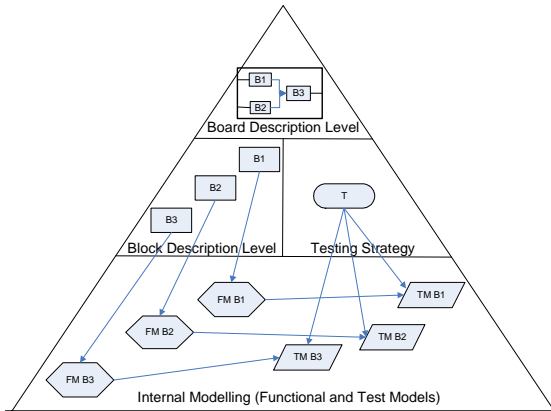
Figure 2 presents how to obtain a test report by running the test program on the maybe faulty board, using an ATE. The first information in the test report is the verdict: the board passed the tests or not. If not, for each failed test, the test report indicates the real and expected results, and the possibly faulty functional blocks. This ends the testing stage and is the departure point of the diagnosis stage. Depending on the required grain of diagnosis, one can stop there and just replace by good ones the components corresponding to faulty blocks. For more precise diagnosis, the method provides some functions that test more specifically the internal behaviour of a functional block (branch, state and path testing and coverage in an FSM context).

### 4 METHOD AND PROTOTYPE

We have implemented the ideas illustrated on Figure 1 in a prototype tool. This prototype provides a Graphical User Interface (GUI) allowing high level topological description of a mixed-signal board. In addition, the GUI includes some facilities for the choice of a testing strategy, for the description of the board-ATE connection and for the description of data (signals) flow.

A mixed-signal board is modelled as a set of connected functional blocks (each block identifies a function that can either be digital, analogue or mixed-signal). A functional block expresses a function, and does not need to correspond exactly

to one of the physical components of the board. Indeed, most of the time it involves a set of such physical components. The description of the board is thus made at two levels: the board level and the block level as illustrated by Figure 3.



**Figure 3: The architecture of the prototype**

The prototype proposes different ways for defining functional blocks. The first one is dedicated to digital functions, the second to analogue or mixed-signal functions, and the third to black box functions.

To define a digital function, the user describes the behaviour of the block using a finite state machine (FSM) formalism. In the second case, the user selects a function in a library of analogue and mixed-signal common components. Due to the specificity of maintenance testing, the prototype also allows black box functions. This kind of definition has to be used when absolutely no information on the board behaviour is available. In practice, it is therefore possible to learn from a known good board a test data set characterising the board. This description is used in a black box definition. Even though using this description reduces the power of the method at the board level, it is some time mandatory to use it.

To generate more accurate test data than that obtained just from a board behaviour specification, we use testing strategies in add of the functional description of the board. A testing strategy expresses behaviours or faults that have to be specifically targeted by the generated test data set. The prototype relies on some basic predefined testing strategies (this usefully benefits from cross-fertilisation with testing practitioners), but it also allows to describe and use its own specific testing strategy.

From an internal point of view, the different kinds of blocks (digital, analogue or mixed-signal) are treated homogeneously. They are each associated to a functional model and a test model, expressed as communicating finite state machines (cf. Figure

3). Actually, the test model is automatically inferred from both the functional model and the chosen testing strategy (excepted in the case of black box components where the functional and test models are the same, the testing strategy being already part of this model). One can view the test model as a particular instantiation of the functional model. Nevertheless, both models are needed to achieve automatic test data generation.

We define the generated board test data set as the optimized union of all blocks test data sets. A block test data set is obtained by transition coverage of its test model and full upstream and downstream propagation through other blocks functional models to primary inputs and outputs. The aim of the optimization is to reduce the size of the test data set.

The problem of test data generation is faced using constraint logic programming (CLP) and classical algorithms for finite state machines (transition coverage, state coverage, path coverage). In the prototype, test data are represented in a symbolic way, using ranges of values instead of one instantiated data. It is essential to work on a symbolic representation of data for the validity and efficiency of the approach. The instantiation of test data is only obtained at the end, on demand from the user, or automatically to derive a test program for a specific ATE.

Most of the prototype is written in C++, excepted the part concerning testing strategies that is implemented using constraint logic programming, with the ECL<sup>1</sup>PS<sup>e</sup> solver [3].

The prototype has been validated on two first industrial case studies: the “Tachy board” [4] and the “Filter Command Board“, which are mixed-signal boards. We detail the application of the method on the “Filter Command Board“ in the next section.

## 5 CASE STUDY

We first give a description of the board involved in the case study. Then, applying our method, we present the board modelling and board testing results.

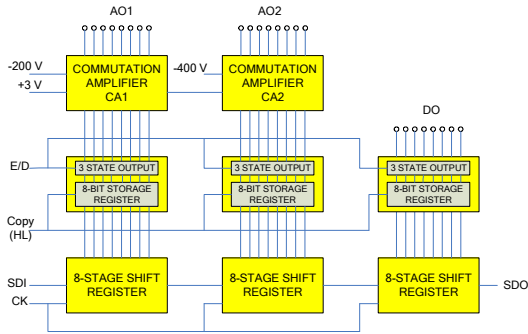
### 5.1 Board description

The “Filter Command Board“ (FCB) is a mixed-signal board (depicted in Figure 4) and is part of a telecommunication system. We describe shortly the inputs/outputs and the functionality of this board.

FCB has the following inputs/outputs signals:

- one serial digital input (SDI),

- one clock input (CK),
- one copy command (Copy),
- one transfer command (E/D),
- one serial digital output (SDO)
- eight digital outputs (DO),
- sixteen analogue outputs (AO1 and AO2).



**Figure 4: The Filter Command board**

FCB digital part is made of:

- three 8-stage shift registers serially loaded into SDI on the rising edge strobe,
- three 8-bit storage registers parallel loaded from shift registers on the rising edge of the Copy signal,
- three 3-state buffers enabling /disabling digital output data with E/D signal.

FCB mixed-signal part is made of two commutation amplifiers CA1 and CA2. Each amplifier has height digital inputs and height analogue outputs. The commutation table for each input/output is given in Table 1.

	CA1	CA2
0	3 V	3 V
1	-200 V	-400 V

**Table 1 : The commutation table**

The main function of the FCB is to command weight filters connected to the commutation amplifiers outputs AO1 and AO2. Digital outputs DO and SDO command other modules in the system.

## 5.2 Board modelling

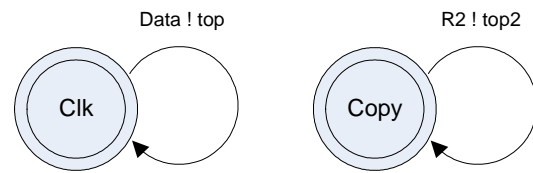
We now present the method internal representation of the board, that is the functional models corresponding to the different blocks constituting the board. We begin with the modelling of the board inputs, next the modelling of the blocks involved in the digital part of the board, the modelling of the blocks of the mixed-signal part, and finally the modelling of the board outputs. Test strategies and test models are also discussed.

### 5.2.1 Data

The modelling of data (board inputs) is particular. Data are not functional blocks of the board. However, to generate test data, we need to model them. This also allows to model characteristics of sources and generators of a specific ATE in order to produce a test program. Thus, each board input has a functional model (but no test model as it does not correspond to a board component).

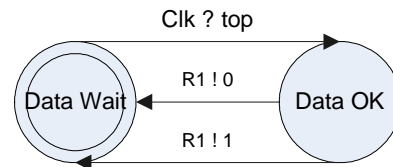
Figure 5 shows the modelling of the two input clocks (the main one and Copy), using a discrete time representation.

"A ? m" stands for reception of message m from FSM A and "A ! m" for emission of message m to FSM A.



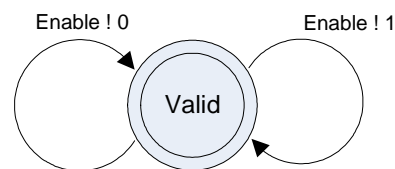
**Figure 5: The main clock signal and the copy clock signal functional models**

Figure 6 presents the modelling of the serial input data, which is cadenced by the main clock, and sends either 0 or 1.



**Figure 6: The serial input data functional model**

The transfer command modelling is shown Figure 7, and sends either 0 or 1.

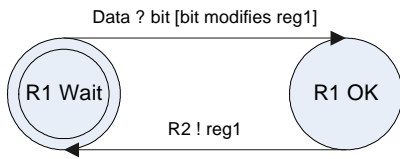


**Figure 7: The transfer command functional model**

### 5.2.2 Digital part

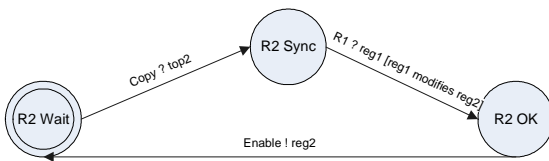
The digital part of the board is modelled by four communicating finite state machines. For simplicity, we model each serial three 8-bit buffers by a single 24-bit buffer.

The first one (R1, cf. Figure 8) is the functional model of the shift register. Its input comes from the serial input (Data) and its outputs go to the storage register (R2).



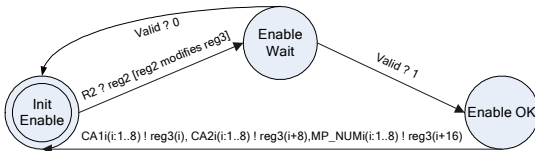
**Figure 8: The shift register functional model**

The storage register (R2, cf. Figure 9), cadenced by the Copy command, receives data from the shift register (R1) and sends it to the 3-state output buffer (Enable).



**Figure 9: The storage register functional model**

The 3-state output buffer (Enable, cf. Figure 10), receives data from the storage register (R2). It sends it to the commutation amplifiers (CA1, CA2) and to the digital measurement points (MP\_NUMi) only if the transfer command (Valid) is 1.



**Figure 10: The 3-state output buffer functional model**

### 5.2.3 Mixed-signal part

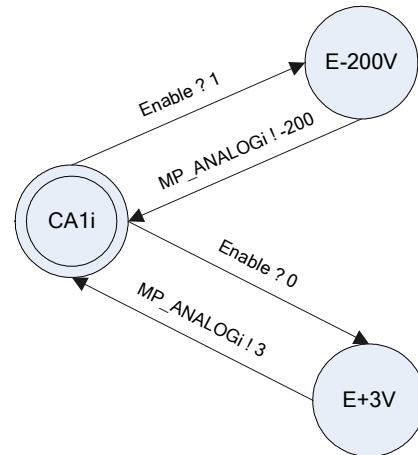
The mixed-signal part of the board is modelled by 16 instances of two communicating finite state machines (CA11..CA18, CA21..CA28).

The first one (cf. Figure 11) is the functional model of the CA1 commutation amplifier. Its input comes from the 3-state output buffer (Enable) and its output goes to the analogue measurement points (MP\_ANALOGi).

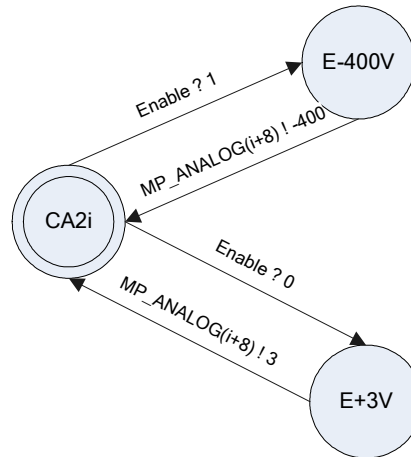
Figure 12 shows the functional model of the CA2 commutation amplifier, which behaves the same manner.

On this example, one may notice that CA1 and CA2 have only two behaviours each. Their functional models are explicitly exhaustive and thus correspond to their test models (coverage of

all kinds of behaviours, with no associated specific fault).



**Figure 11: The CA1 commutation amplifier functional (and test) model**



**Figure 12: The CA2 commutation amplifier functional (and test) model**

### 5.2.4 Outputs

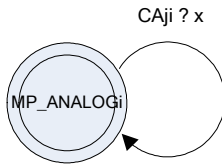
As we did previously for board inputs, we have to model the board outputs even if they are not real functional blocks. It is needed to generate test data, and it also allows to describe ATE measurement tools characteristics in order to produce a test program.

Figure 13 presents the analogue measurement point functional model. MP\_ANALOGi is just waiting for data #i coming from one commutation amplifier (CA1 or CA2).

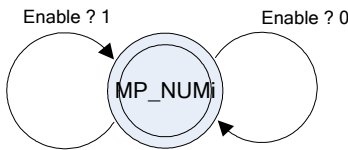
Concerning the digital part, functional models express behaviours in an implicit way, and they do not include information on specific faults to detect. Thus, we choose to add some testing strategies on the digital measurement points. Figure 14 illustrates the “stuck-at” strategy on a digital measurement point. MP\_NUMi is forced to



receive at least one 0 and at least one 1 from the 3-state output buffer (Enable).



**Figure 13: The analogue measurement point functional model**



**Figure 14: The digital measurement point functional model**

### 5.3 Board testing

We present here the test data sequences (and associated expected outputs) obtained by coverage of the different blocks of the board:

$$\{DT_i(i:1..24), DT'_i(i:1..24), DT''\}$$

with:

$$DT_i = (\text{data}=1, \text{valid}=1, \text{clock} = \text{top}, \text{copy} = \text{top2})$$

$$\text{and } R_i = (-200, X \dots X)$$

$$DT'_i = (DT_{i-1} + (\text{data}=X, \text{valid}=1, \text{clock}=\text{top}, \text{copy}=\text{top2}))$$

$$\text{and } R'_i = (X \dots X, -200/-400/1, X \dots X)$$

$$DT''_i = (\text{data}=0, \text{valid}=1, \text{clock} = \text{top}, \text{copy} = \text{top2})$$

$$\text{and } R''_i = (3, X \dots X)$$

$$DT''_i = (DT'_{i-1} + (\text{data}=X, \text{valid}=1, \text{clock} = \text{top}, \text{copy}=\text{top2}))$$

$$\text{and } R''_i = (X \dots X, 3/0, X \dots X)$$

$$DT'' = (\text{data}=X, \text{valid}=0, \text{clock} = \text{top}, \text{copy}=\text{top2})$$

$$\text{and } R'' = \ll \text{no information} \gg (\text{high-impedance state}).$$

Finally, after optimized union of all blocks test data sets, the board test data set is:

$$\text{Board test data set} = \{DT_{24}, DT'_{24}, DT''\}.$$

This test data set is made of three sequences, the first two of size 24, the third one of size 1. It covers all the possible values for the 25 outputs, and the particular case of the transfer command set to 0.

## 6 CONCLUSION AND FUTURE WORK

We have presented a method for mixed-signal board testing, designed to face the specificities of maintenance testing. Some aspects have been validated such as the functional approach at all levels, the homogeneous modelling formalism (for digital, analogue and mixed-signal functions) and the testing strategies. The prototype is already able to generate expected test data (thus completing the first part of the method, cf. Figure 1) for the boards involved in the case studies.

Further work is needed to validate our complete approach and prototype, and its real impact on diagnosis. In particular, the second part of the method (depicted in Figure 2) is not yet implemented and needs an important effort as it implies the construction of a physical board, dedicated to the application of our method on a specific ATE.

Future work includes more experimentation of the method on other types of boards, but first results are encouraging, reinforcing the idea that at least semi-automatic testing of mixed-signal boards is feasible. This kind of tool would respond to a real need in the industrial community.

### ACKNOWLEDGMENTS

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 board maintenance testing.

### REFERENCES

- [1] Bushnell Michael L., Agrawal Vishwani D. "Essentials of Electronic Testing for Digital, Memory and Mixed-signal VLSI". Springer (2000).
- [2] Burns Mark, Gordon W. Roberts "An Introduction to Mixed-Signal IC Test and Measurement". Oxford University Press (2001).
- [3] Cheadle A.M., Harvey W., Sadler A.J., Schimpf J., Shen K., Wallace M.G. "Eclipse: An introduction". Technical Report IC-Parc-03-1, IC-Parc, Imperial College London (2003).
- [4] Gilles B., Nicolas V.-A., Lemarchand L., Marcé L., Castel B. "Towards a New Modelling of Mixed Signal Boards For Maintenance Testing", Proc. of the 11th IEEE Int. Mixed-Signal Testing Workshop, IMSTW'05 (2005).