High level modeling of signal integrity in field bus communication with SystemC-AMS

Ruomin Wang, Julien Denoulet, Sylvain Feruglio, Farouk Vallette, Patrick Garda

High Level Modeling of Signal Integrity in Field Bus Communication with SystemC-AMS

Ruomin Wang, Julien Denoulet, Sylvain Feruglio, Farouk Vallette and Patrick Garda
UPMC Univ. Paris 06, UMR 7606, Laboratoire d'Informatique de Paris 6, F-75005, Paris, France
Email: firstname.lastname@upmc.fr (e.g. ruomin.wang@upmc.fr)

***Abstract* — This paper presents a novel method for modeling the functionality of a mixed-signal system, and analyzing its signal integrity (SI) at a high-level of abstraction with SystemC-AMS. Our model includes on a unique platform a functional module and a non-functional module. The functional module represents the operative behavior of the system and the non-functional module, based on neural network techniques, displays the SI characteristics of the system. The proposed method is demonstrated by modeling field bus communication system with two nodes. We achieved an error of about 3% for the neural network based Time Data Flow (TDF) model with respect to a RLC Electrical Linear Networks (ELN) model.**

***Index Terms* — High-level modeling, signal integrity, field bus, SystemC-AMS.**

I. INTRODUCTION

Nowadays, mixed-signal systems can be found in many applications (telecommunications, multimedia, automotive, etc.). Their design has become one of the key issues for the industry and researchers. Meanwhile, with the continued down scaling of CMOS technology, the increase of clock frequencies and the decrease of power supply voltages (V_{DD}), Signal Integrity (SI) [1] has become another key issue. The effects of SI (e.g. crosstalk) may jeopardize the functionality and reliability of embedded systems. But SI issues are usually not investigated until system hardware is under test. For a more efficient detection of SI problems, it's imperative for the design process to take it into account from the start (i.e. at the virtual prototyping step). Many approaches were proposed concerning mixed signal system [2]-[4], or SI [5]-[7] modeling. However, one point seems not to be covered: modeling at a high level of abstraction the functionality of a mixed-signal system along with its SI performances.

Alassir *et al.* [8] tried to address this issue by presenting a modeling methodology using SystemC-AMS [9]. This modeling methodology allows to simulate the execution of the embedded software and its analog response on the lines of a communication bus. One limit of this approach is that the SI performances are modeled at a low-level, using the Electrical Linear Networks (ELN) continuous time Model of Computation (MoC) (i.e. RLC equivalent circuits).

In this paper, we propose a new method to address the issue of high level modeling of SI. In our method, SystemC-AMS is still used as a unique modeling environment. Furthermore, we add neural network techniques to model SI and integrate into a SystemC-AMS description. Our approach is applied to field-bus based systems, as in Fig. 1. It is meant to be quicker and more generic in order to reduce the time to market.

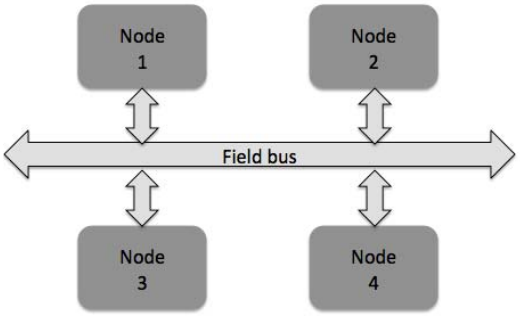

Figure 1. Field-bus based system.

The paper is divided into five sections. Section 2 briefly describes our methodology. Section 3 presents the neural network used to model the performances and how it can be implemented into the SystemC-AMS functional description of the mixed system. Section 4 gives simulation results of our platform. Section 5 discusses the potential advantages of our method. Although our methodology is generic, we will present it for didactical purposes in the specific case of an I^2C bus [10] to analyze crosstalk between adjacent bus lines and I/O influence on signal quality.

II. METHODOLOGY

In this section, we present our method to build a SystemC-AMS modeling platform able to simulate the system functionality and its SI characteristics. Considering the long-term challenge we are addressing, we chose a step-by-step research strategy. So we begin with a field bus communication system, which includes just two nodes.

The core of our work is modeling and analyzing SI issue at a high-level of abstraction. For this purpose, we propose to model the system in two kinds of blocks: one functional module and one non-functional module. Functional modules represent the digital or analog functions of the bus nodes, while non-functional modules represent the performance (SI) of the system.

A. Functional modules

The work presented in this paper is centered on an I^2C bus that connects a master and a slave node (Fig. 2.a). The master is an 8051 microcontroller, which is associated with a bus controller. The slave is a memory device, which will be accessed by the master in the course of simulation.

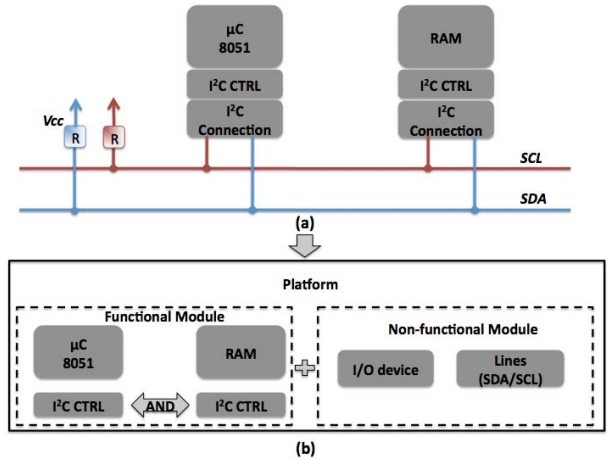

Figure 2. (a) System example. (b) General structure of our platform.

The functional modules of the master and the slave node were introduced in [8]. Although the nodes in our platform are digital, analog functions of a node (for instance a sensor) could be modeled as well in SystemC-AMS. To represent the wired-AND function [10] implemented by the I²C bus, we also added a digital model of the bus lines.

These models can be independently simulated to visualize the functionality of the nodes and the transactions on the bus.

B. *Non-functional modules*

The non-functional modules allow us to represent the analog behavior of the bus lines and the nodes' I/Os. In association with the functional modules, they show the SI performances (crosstalk between adjacent bus lines, I/O influence on signal quality) of the nodes and the bus lines.

To model SI at a high-level of abstraction with SystemC-AMS, we propose to develop this non-functional module using a neural network. With the help of a neural network, we need less information of one device and we can model SI with the Time Data Flow (TDF) MoC of SystemC-AMS. The implementation of the non-functional module is presented in the next section.

III. MODELING OF NON-FUNCTIONAL MODULE

In this section, we introduce the method to build a non-functional module using neural networks. The non-functional modules are then implemented using the TDF MoC of SystemC-AMS. Neural networks are information processing systems that are inspired by the structure and functional aspects of biological neural networks [11]. They have been recognized as a powerful tool for system modeling and design issues. So they have been used in a variety of applications [12]. They have an important property: the ability to learn from input data with or without a teacher [13]. Consequently, we can use them to model the relation between inputs and outputs even when devices formulas or equivalent circuits are unavailable [14]. This suits our high-level abstraction modeling approach of SI.

Our method is based on three stages represented in Fig. 3: firstly, we get the input/target pairs (e.g. input voltage/output voltage, input current and temperature/output current) of bus by measurement or simulation of data transmission on the bus;

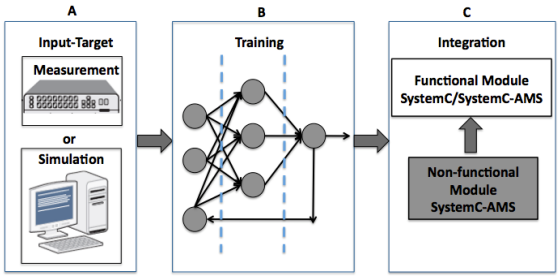

Figure 3. Non-functional module building methodology.

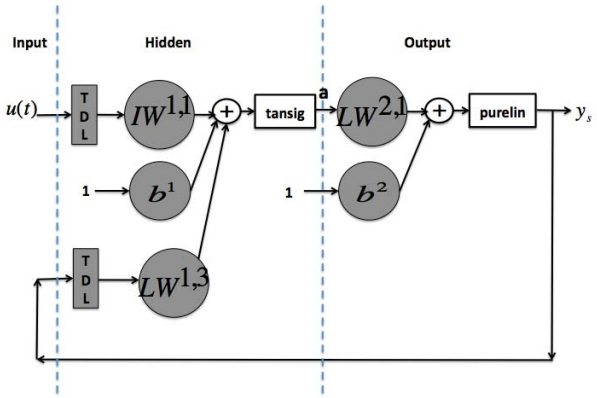

Figure 4. NARX network.

secondly, we train a neural network with these input/target pairs to model the analog behavior; thirdly, we implement this neural network into a SystemC-AMS module that we can directly instantiate in our platform.

A. *Variables Input/Target signal pairs*

In the first step, we feed (time varying) inputs to a component (e.g. lines) and sweep these inputs over their entire operating ranges [15], then monitor and store (time varying) outputs as target. These input/target signal pairs will be used to train our neural network in the following step. For our modeling platform (Fig. 2.b), the inputs are the logic levels of data emitted on the bus by the transmitter node. The targets are the voltages observed at the input of the receiver node. The values can be obtained through measurements or simulations.

B. *Training the neural netwok*

In the second step, we train a neural network to approximate the relation between inputs and targets collected in the first step. The neural network training is performed with the "Neural Network Toolbox" in MATLAB [16].

We chose a recurrent dynamic neural network in order to take into account the memory effect in the bus RLC equivalent circuit, such as a Nonlinear AutoRegressive models with eXogenous (NARX) network proposed by [17]. Fig. 4 presents its structure. It has three layers: an input layer, a hidden layer and an output layer. The defining equation is shown in (1):

$$y(t) = f(y(t-1), \ldots y(t-n), u(t-1), \ldots u(t-m)) \quad (1)$$

where ***y(t)*** is a vector containing the output signals and ***u(t)*** is a vector containing the input signals. *m* and *n* represent the number of delay steps for ***u(t)*** and ***y(t)***, respectively. The transfer function of the hidden layer is a sigmoid function (*tansig*),

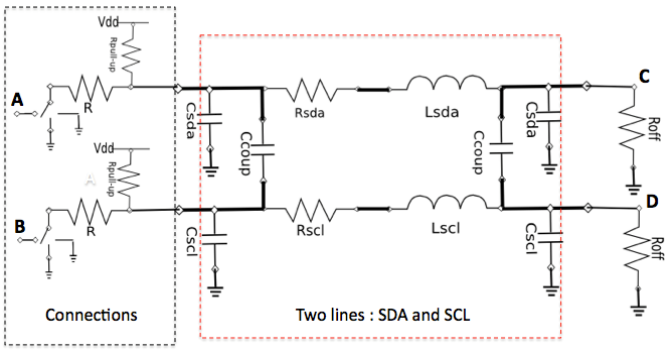

Figure 5. Simplified equivalent circuit of bus I²C.

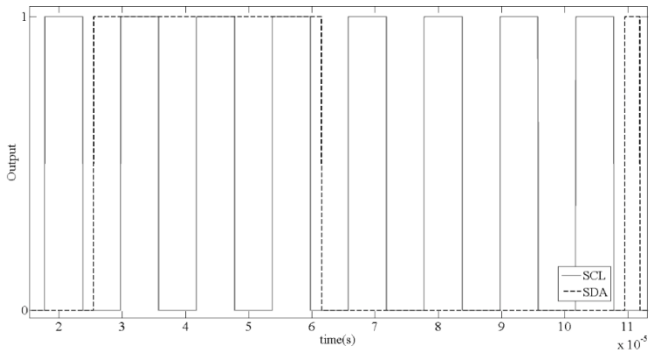

Figure 6. Results of our platform without non-functional module.

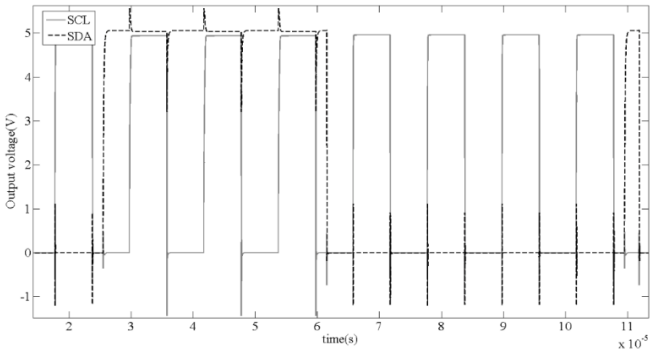

Figure 7. Results of our platform with non-functional module.

as in (2), and the transfer function of the output layer is linear (*purelin*), as in (3).

$$\text{tansig}(x) = \frac{2}{1+e^{-2\times x}} - 1 \tag{2}$$

$$\text{purelin}(x) = x. \tag{3}$$

The learning rule is scaled conjugate gradient back propagation (*trainscg*). We train the network with our input/target pairs. During training, the values of the matrices ***IW***, ***LW*** and ***b*** are updated. At the end of training, we get an approximation of the relation between inputs and outputs of bus given in (4) and (5):

$$a_i = \text{tansig}\Big(\sum_{j=1}^{k}\Big(\sum_{d=1}^{n}(u_j(t-d)\times IW_{j,i}^{1,1})\Big) + \sum_{j=1}^{l}\Big(\sum_{d=1}^{m}(y_j(t-d)\times LW_{j,i}^{1,3})\Big) + b_i^1\Big) \tag{4}$$

$$y_S = \text{purelin}\Big(\sum_{i=1}^{p}(a_i \times LW_{s,i}^{2,1}) + b_s^2\Big) \tag{5}$$

k and *l* represent the number of input and output respectively. *p* is the number of neurons in the hidden layer.

C. Model Integration

In the last step, we show how to implement the neural network in a SystemC-AMS TDF description and how to associate it with the functional module presented in Section II.

We chose the discrete time TDF MoC for its potential for simulation speed as opposed to the continuous time ELN MoC to implement the neural network in a SystemC-AMS module. We simply write equations (4) and (5) in the TDF module.

The SystemC-AMS model of the neural network can then be connected to the functional modules (also modeled in SystemC/SystemC-AMS) to represent the SI characteristics of the system. Finally, our complete model allows us to represent at a high level of abstraction the functionality of the system and also the SI performances.

IV. Simulation Results

In this section, we analyze the simulation results of our platform. These results are validated by comparing them to the simulation of the same system with Alassir's method [8].

To build our non-functional module, we can use input/target pairs obtained by circuit measurement or by simulation of an equivalent circuit with a simulation tool such as NGSPICE (Fig. 5), we chose that last option to validate our method more quickly. This circuit included two blocks: the node I/O device (including a switch and resistor to model the open-drain transistor of an I²C device) and the bus lines (SDA and SCL) [18]. For our platform, the inputs were the logic levels at points A and B. The targets were the voltages at points C and D. We got about 210 000 input/target pairs. Once the training was finished, we implemented it in a SystemC-AMS module and associated it with the functional module.

Fig. 6 presents the digital behavior of our platform by simulating only the functional modules. Fig. 7 presents the analog behavior of the I²C bus lines when we added the non-functional modules. Comparing our analog results with the results of Alassir's method, we found that our method had good accuracy. The average differences were 3.21% for SDA and 2.79% for SCL. The CPU time required during simulation for our method was about 7.945s and Alassir's method needed 6.9s (Intel® Core™ i5-660 3.33GHz, RAM 2Gbits).

V. Discussion

Simulation results show our platform achieves similar precision as Alassir and comparable simulation speed, although with a 15% timing overhead. This result seems a contradictory since our models are at a higher-level of abstraction. But a closer analysis shows our approach has an important speedup potential. As we said, we used the TDF MoC of SystemC-AMS, which computes data at a constant fixed time step. The lower the time step, the more precise the result. Our goal is to detect potential SI issues during the transmission of digital

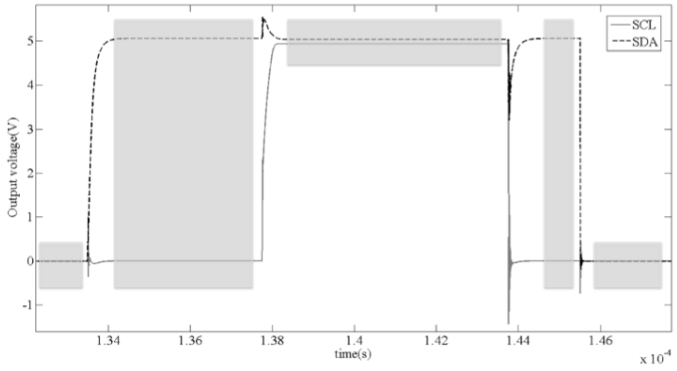

Figure 8. Zones of low interest in a clock cycle (in grey).

data (i.e. zeros and ones) on a bus. Such issues only happen during transitions on the bus lines. To precisely model what happen during these transitions, a very short time step is required (1ns compared to a SCL clock cycle of 12.2µs). Unfortunately, this same time step has to be used when both SDA and SCL signals are stable and no SI problem is expected, causing unnecessary computations. Fig. 8 shows in grey the low interest zones where a long time step could be used. It shows that 88% of the 12 201 points of a clock cycle are of low interest.

Fortunately, in its forthcoming release, SystemC-AMS will introduce Dynamic TDF [19] to allow a dynamic evolution of a TDF module time step. With such a feature, we will avoid all the unnecessary computations and achieve a potential speedup of one order of magnitude. This would make our approach faster than Alassir's method.

CONCLUSIONS

In this paper, we modeled SI in mixed-signal systems at a high level of abstraction with SystemC-AMS, as shown in Tab. 1. To our best knowledge, this is the first SI model at the TDF level of abstraction.

For this purpose, neural networks were used for the mathematical function approximation of the analog behavior in this system. The trained neural network was developed as a non-functional module and integrated into the functional module. With the help of the neural network, the non-functional module was described in the SystemC-AMS TDF. We evaluated it on a simple system through Alassir's method.

We achieved an error of about 3% for the neural network based TDF model with respect to the RLC ELN model of [8]. This validates the performances of our method. The simulation time was comparable with Alassir, although with a 15% overhead. However, the forthcoming TDF kernel will bring a speedup of an order of magnitude for our TDF model, whereas no speedup can be expected for the ELN model.

In conclusion, we introduced a new high-level model of SI, suitable for its implementation in a TDF model. This will allow a significant speedup when suitable TDF simulation kernels will be available. Also the choice of neural networks allows to model the non-linear effects.

In the future, we'll extend our approach by modeling a multi-nodes system. Furthermore, we'll develop a Transaction Level Modeling (TLM) of the digital part in SystemC. This TLM model will be at a higher level of abstraction than the current Cycle Accurate/Bit Accurate (CABA) models.

TABLE I. METHODS FOR MODELING SI

Method	***Level of abstraction***	***Object***
[5], [7]	Low	Circuit
[6]	High	Component/circuit
[8]	Mixed	Mixed-signal system
[11]	High	Component/circuit
Our method	High	Mixed-signal system

REFERENCES

[1] E. Bogatin, Signal Integrity – Simplified, Prentice Hall PTR, 2003.

[2] M. Vasilevski, F. Pecheux, N. Beilleau, H. Aboushady and K. Einwich, "Modeling and refining heterogeneous systems with SystemC-AMS: Application to WSN", in Proc. Design, Automation and Test in Europe (DATE), pp. 134 – 139, 2008.

[3] K. Kundert, H. Chang, D. Jefferies, G. Lamant, E. Malavasi and F. Sending, "Design of mixed-signal systems-on-chip", IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 19, no. 12, pp. 1561 – 1571, 2000.

[4] P. Wambacq, G. Vandersteen, P. Dobrovolny, M. Goffioul, W. Eberle, M. Badaroglu and S. Donnay, "High-level simulation and modeling tools for mixed-signal front-ends of wireless systems", Analog Circuit Design, Part I, pp. 77 – 94, Kluwer Academic Publishers, 2002.

[5] C. Labussiere-Dorgan, S.B. Dhia, E. Sicard, J. Tao, H.J. Quaresma, C. Lochot and B. Vrignon, "Modeling the electromagnetic emission of a microcontroller using a signle model", IEEE Trans. Electromagn. Compat., vol. 50, no. 1, pp. 22 – 34, 2008.

[6] S. Grivet-Talocia, P. Brenner and F.G. Canavero, "Fast macromodel-based signal integrity assessment for RF and mixed-signal modules", IEEE Symp. on Electromagn. Compat. (EMC), pp. 1 – 6, 2007.

[7] J. Loeckx and G. Gielen, "Assesment of the DPI standard for immunity simulation of integrated circuits", IEEE Trans. Electromagn. Compat.,pp. 1 – 5 , 2007.

[8] M. Alassir, J. Denoulet, O. Romain, A. Suissa and P. Garda, "Modeling field bus communications in mixed-signal embedded systems", EURASIP Journal on Embedded Systems, vol. 2008, pp. 1 – 11, 2008.

[9] OSCI Analog/Mixed-signal Working Group, SystemC-AMS extensions requirements specification.

[10] Philips Semiconductors, The I^2C-Bus Protocol Specification, 2000.

[11] Q. Zhang and K.C. Gupta, Neural Networks for RF and Microwave Design. Norwood, MA: Artech House, 2000.

[12] M.T. Hagan, H.B. Demuth and M.H. Beale, Neural Network Design, PWS Publishing Compony, 1996.

[13] V.K. Devabhaktuni, M.C.E. Yagoub, Y. Fang, J. Xu and Q. Zhang, 2001, "Neural networks for microwave modeling: model development issues and nonlinear modeling techniques", Int. Journal of RF and Microwave, John Wiley and Sons Inc., CAE 11, pp. 4 – 21.

[14] S. Haykin, NEURAL NETWORKS: A Comprehensive Foundation, Pearson Education, 1999.

[15] L. Ljung,"System identification:theory for the user",Prentice-Hall, 1987.

[16] MathWorks, MATLAB Neural Network Toolbox.

[17] S. Chen, S.A. Billings and P.M. Grant, "Non-linear system identification using neural networks", Int. J. Contr., vol. 51, no.6, pp.1191-1214, 1990.

[18] G. Vasilescu, Electronic Noise and Interfering Signals: Principles and Applications, chapter 12, Springer, New York, 2005.

[19] M. Barnasconi, K. Einwich, C. Grimm, T. Maehne and V. Vachoux, "Advancing the SystemC Analog/Mixed-Signal (AMS) Extensions: Introducing Dynamic Timed Data Flow", Open SystemC Initiative, 2011.