

Research Article

System Level Modelling of RF IC in SystemC-WMS

Simone Orcioni, Mauro Ballicchia, Giorgio Biagetti, Rocco D. d'Aparo, and Massimo Conti

Dipartimento di Elettronica, Intelligenza artificiale e Telecomunicazioni, Università Politecnica delle Marche, 60131 Ancona, Italy

Correspondence should be addressed to Simone Orcioni, s.orcioni@univpm.it

Received 1 October 2007; Revised 18 February 2008; Accepted 12 April 2008

Recommended by Christoph Grimm

This paper proposes a methodology for modelling and simulation of RF systems in SystemC-WMS. Analog RF modules have been described at system level only by using their specifications. A complete Bluetooth transceiver, consisting of digital and analog blocks, has been modelled and simulated using the proposed design methodology. The developed transceiver modules have been connected to the higher levels of the Bluetooth stack described in SystemC, allowing the analysis of the performance of the Bluetooth protocol at all the different layers of the protocol stack.

Copyright © 2008 Simone Orcioni et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

In recent years, an important issue in electronic design has been the integration of complex intelligence into a single silicon integrated circuit. Today, after years of research and development in the field of silicon technology, it is possible to integrate a complex electronic system, equivalent to millions of transistors, into a single silicon chip: a system-on-chip (SoC). The design of a complex SoC is based on efficient modelling methodologies, on the development of seamless tool chains, on the creation and collection of appropriate test suites, and on the availability of libraries with easily reusable interfaces and clear documentation.

The international technology roadmap for semiconductors (ITRS) [1] and the MEDEA+ electronic design automation (EDA) roadmap [2] highlight heterogeneous systems design among the major challenges in the semiconductor business that need innovative EDA solutions. In moving to top-down synthesis paradigms, descriptions of analog mixed signal (AMS) and radio frequency (RF) parts of an SoC will be the major challenge. A global simulation and performance estimation environment needs to be developed, with high flexibility and powerful capabilities. The intellectual property (IP) reusability will be even more important than it is in the digital domain, though much more complicated to implement.

A unified design framework that encompasses all the key phases of an SoC design, including the simulation of

mechanical and simplified electrical physical models, the design and simulation of digital discrete time algorithms, and the design, simulation, and optimization of their implementation on DSP and/or FPGA, is not currently available. Such a design environment would require the ability to cosimulate systems of different natures: mechanical, analog electronic, and digital discrete time electronic systems are just a few examples.

SystemC is a consolidated design language and environment, based on C++, used for system level description of SoCs [3]. SystemC supports a wide range of models of computation and is very well suited to the design and refinement of HW/SW systems from system-level down to register-transfer-level (RTL). However, for a broad range of applications, the digital parts of an electronic system closely interact with the analog parts and thus with the continuous-time environment. Due to the complexity of these interactions, and to the dominant role that the analog parts often play in the overall system behavior, it is essential to consider these analog parts within all the stages of the design process of a mixed-signal system.

The extension of SystemC to mixed-signal applications is currently under development. The aim of this extension is the codesign and cosimulation of mechanical parts and of both analog and digital electronic circuits inside a system. To this end, the SystemC-AMS working group inside OSCI is working on the development of a new library, to be integrated in SystemC, for the description of mixed-signal

systems [4–6]. As application examples, systems in which the analog RF part is integrated within the same chip as the digital part, or a digital control of the injection system in the automotive field are presented.

Recently, the SystemC-WMS (wave mixed signal) library has also been presented [7, 8], implementing a simple SystemC extension to allow mixed-signal modelling and simulation using the concept of incident and reflected waves. This extension to SystemC toward mixed-signal systems enables the creation of a high level executable model of both digital and analog subsystems in the same simulation environment, thus permitting a fast evaluation of the performance of the complete system. SystemC-WMS is suitable to be used in a number of different applications, including, for example, the control of electronic and mechanical systems in automotive applications, or the simulation of analog RF and high-level digital blocks in various wireless network technologies, such as Bluetooth or Zigbee.

Some system level designs of heterogeneous systems have been presented in the literature. In [9], Bjørnsen et al. presented a mixed-signal simulation of two pipelined A/D converter architectures and a CMOS camera-on-a-chip. In [10], Al-junaid and Kazmierski proposed a mixed-signal model of a boost DC-DC converter and a PLL-based frequency synthesiser, while in [11] the physical model of an electric guitar and associated effects, for the implementation in a digital circuit, is reported.

Short-range wireless communications are widely used and of great commercial interest. The design of SoC including wireless transceivers is extremely complex due to the fact that design specifications of the analog and the digital parts are strictly correlated. In [12], an architecture of a ZigBee transceiver has been designed and simulated using the Agilent ADS Ptolemy simulator.

This work presents a system level design of a Bluetooth transceiver, consisting of digital and analog blocks. In Section 2, the SystemC-WMS design framework is presented. Section 3 reports the system level modelling methodology used to design RF mixed signal transceivers starting from the specifications. Section 4 shows an application example regarding the design of a Bluetooth transceiver and the simulation results of the complete system, while the conclusions are reported in Section 5.

2. ANALOG MODULE REPRESENTATION IN SYSTEMC-WMS

SystemC-WMS is a C++ class library, developed to work in conjunction with SystemC, that allows the user to model, simulate, and debug complex systems described at system level, comprising analog and mixed signal blocks, or even blocks operating on different physical domains (e.g., transducers, MEMS, mechanical actuators and systems, etc.). This was achieved by fully exploiting the computation model of SystemC, which is basically an event-driven simulation kernel, by providing a methodology to describe the functionality of analog or mixed signal modules and a flexible system to describe the interconnections between them.



FIGURE 1: Port symbols using electrical and wave quantities.

In fact, the SystemC kernel's primary function is to schedule the execution of concurrent processes, which describe the functionality of individual modules, according to the interconnections between the modules themselves. The application of this simulation paradigm to an analog system required the specification of an interface that could be used to model the inherent coupling that arises from the interaction of interconnected analog blocks.

As will be discussed shortly, the proposed interface is also able to take care of the dependencies between the modules and the channels directly connected to them, without the need of dealing with the global system topology, so that the simulation can still be carried out by the same event-driven scheduler.

2.1. Module representation based on wave exchanges

Without loss of generality, we can fix our attention to an N -port in the electrical domain, described by its port quantities v_j and i_j , with $j = 1, \dots, N$. Although these quantities suffice to describe the relations between the module and the external environment, they are not well suited to be part of an event-driven interface because there is no generic way to associate a cause/effect relationship to them.

Moreover, in the frequency domain it is customary to use representations of blocks by means of their scattering matrix [13], and the same approach can straightforwardly be extended to other domains as well. For instance, since SystemC is a time-domain simulator, the following definition of incident (a_j) and reflected (b_j) wave:

$$\begin{aligned} a_j(t) &= \frac{1}{2} \left(\frac{v_j(t)}{\sqrt{R_j + i_j(t)\sqrt{R_j}}} \right), \\ b_j(t) &= \frac{1}{2} \left(\frac{v_j(t)}{\sqrt{R_j - i_j(t)\sqrt{R_j}}} \right) \end{aligned} \quad (1)$$

can be used, so that $a_j^2(t) - b_j^2(t)$ is the instantaneous power entering port j , and R_j is a normalisation resistance, that can be assumed to be alike to the characteristic impedance of the transmission line connected to the port if we were working in the frequency domain.

This representation lends itself to the very simple and widespread interpretation of a_j as the cause and of b_j as the effect, so that analog modules thus modelled have well-defined inputs and outputs, as sketched in Figure 1 for a single port. Their activation can thus follow the normal SystemC scheduling practice of re-evaluating those blocks whose inputs have changed.

Having defined the port interface, the description of the dynamics of each analog module can be done by means of a

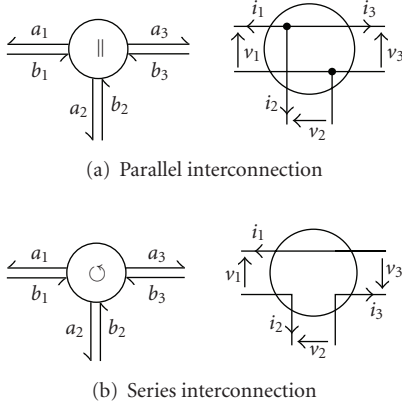


FIGURE 2: Wavechannel symbols corresponding to different port interconnections.

system of nonlinear ordinary differential equations (ODEs) of the following type:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{a}), \\ \mathbf{b} &= \mathbf{g}(\mathbf{x}, \mathbf{a}), \end{aligned} \quad (2)$$

where \mathbf{f} and \mathbf{g} are vector expressions describing the system dynamics, \mathbf{x} is the state vector, while \mathbf{a} and \mathbf{b} are input and output vectors expressed in terms of incident and reflected waves, respectively. These equations will then be solved by an embedded local ODE solver.

2.2. Wavechannels

The use of the incident/reflected wave model greatly simplifies the problem of taking into account the connection between modules, since it can be mandated that modules use incident waves as inputs and produce reflected waves as outputs. This immediately solves the problem of cascading modules, whereas the parallel or series connection can be accomplished by using a new primitive channel that dispatches waves to the modules it connects together, and permits the formulation of a generic and standard analog interface usable across a variety of domains [7].

Such channel behaves similarly to the scattering junction of WDFs [14], which are digital models of analog filters, obtained by the discretisation of individual circuit components. Our approach uses, like in the WDF theory, the ab parameters as input/output signals and implements the duties of the scattering junction in a new entity called wavechannel, complying with SystemC conventions for channels.

Wavechannels are the means by which modules described by wave quantities communicate. They can be thought of as a bunch of transmission lines connecting ports to a junction box, in which the lines are tied together, and their role is to model the scattering of waves occurring at the junction. In the current implementation of wavechannels the propagation delay can be excluded, so that their connection to instantaneous blocks may result in the production of delay-free loops.

This fact is dealt with by the standard SystemC delta cycle mechanism which, without further intervention, would just use a fixed-point algorithm to search for the solution of the instantaneous loops, provided that the embedded ODE solver does not advance its state while iterating to find the fixed point.

These delay-free loops that arise from the interconnection of modules are equivalent to algebraic relationships, either between state variables or input/output quantities, giving rise to systems that are globally described by differential algebraic equations (DAEs). It may be worth noticing that, with the fixed-point method, state variable values are not altered during delta cycles, so that only index-1 DAE systems can be solved, i.e., only those admitting a solution of the algebraic part for every value of the state variables.

Let us consider a junction between N ports, each with its own normalisation resistance R_j , and let \mathbf{v} and \mathbf{i} be the voltage and current vectors, respectively, with

$$\begin{aligned} \mathbf{A}_v \mathbf{v} &= 0, \\ \mathbf{A}_i \mathbf{i} &= 0 \end{aligned} \quad (3)$$

being a complete and minimal set of Kirchhoff's equations describing the junction (with $[\mathbf{A}_v]_{ij}, [\mathbf{A}_i]_{ij} \in \{0, \pm 1\}$). We maintain that letting

$$\mathbf{A}_x = \mathbf{A}_v \operatorname{diag} R_k, \quad \mathbf{A}_y = \mathbf{A}_i \operatorname{diag} \frac{1}{R_k} \quad (4)$$

the scattering matrix \mathbf{S} (such that $\mathbf{a} = \mathbf{S}\mathbf{b}$) can be computed from (1), (4), and (3), resulting in:

$$\mathbf{S} = \begin{bmatrix} \mathbf{A}_x \\ \mathbf{A}_y \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{A}_x \\ \mathbf{A}_y \end{bmatrix}, \quad (5)$$

where \mathbf{b} are the waves reflected by modules and thus entering the junction, and \mathbf{a} are scattered back from the junction to the modules by that means interconnected.

The above formulation can be used for any kind of junction. In the current implementation it is possible for a user to implement connections of an arbitrary type, by means of a wavechannel, simply by specifying the incidence matrix of the connection itself. To help SystemC-WMS users, the most common types of connections, such as parallel and series structures, half-bridges, and bridges, have already been implemented.

For example, from Kirchhoff's laws, a parallel connection is characterised by the equations:

$$\sum_{j=1}^N i_j = 0, \quad v_1 = v_2 = \dots = v_N, \quad (6)$$

whereas for a series wavechannel we have

$$\sum_{j=1}^N v_j = 0, \quad i_1 = i_2 = \dots = i_N \quad (7)$$

with similar equations that can easily be found for any other type of connection. The derivation of the incidence

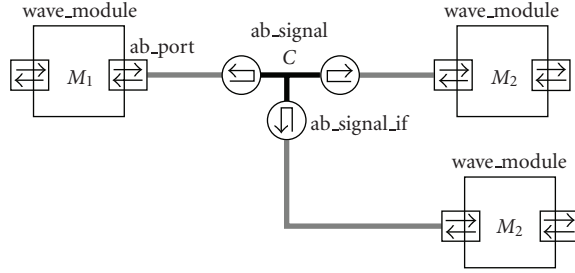


FIGURE 3: SystemC-WMS class library: sample interconnection structure.

matrix from them is then straightforward. The symbols and the electrical schematic of these wavechannels are shown in Figure 2.

It may be worth noticing here that, if $N = 1$, (6) and (7) simply imply $a_1 = \pm b_1$, and the two channel types are thus able to model the total reflection that takes place at an open circuit or at a shunt, respectively.

2.3. SystemC-WMS class library

To aid the implementation of systems by means of analog blocks described with wave quantities, a number of templates and classes have been designed, as shown in Figure 3: a new kind of port to let modules communicate via wave quantities (*ab_port*), the wavechannel that can interconnect them and that does the real computation of the scattering that occurs at junctions (*ab_signal*), and a template base class (*wave_module*) that eases the declaration of modules by taking care of handling sensitivity lists and port declarations. Moreover, a generic base class *analog_module* (not shown) can be used by noninstantaneous modules to include and access their local ODE solver.

Ports expose an interface that allows users to read (*read*) the incident wave value and to report (*write*) the reflected wave value. Of course, in the same *wave_module*, they can freely be mixed with standard SystemC ports and, in the same design, instantaneous analog modules, signal flow graph (SFG) analog modules (deriving from *analog_module*), and wave modules (deriving from *wave_module*) can be used and mixed together.

An example of declaration of a noninstantaneous wave module is given in Algorithm 1, where the template parameters denote the number of wave ports and the nature of them, with the nature specifying the underlying type of the wave variables together with their name and measurement units for output documentation purposes. After that the user only needs to implement the functions (2) defining the module behavior, as sketched in Algorithm 2.

All the rest is taken care of by the *ab_signal* the module connects to. It is a primitive channel that exploits the two-phase (evaluate/update) scheduling paradigm of SystemC, where the *write(...)* call, occurring during the evaluate phase, determines which channels are to be updated. Then during the update phase, the scattering occurring at the junction is computed, and a new delta cycle triggered if a change

```

struct example:wave_module(1, electrical), analog_module
{
    // state variable x is inherited from analog_module
    void field (double *var) const;
    void calculus ();
    SC_CTOR (example) : analog_modulee (... )
    {
        SC_THREAD (calculus);
        sensitive << activation;
    }
};

```

ALGORITHM 1: Typical declaration of a wave module in SystemC-WMS.

```

void example: :field (double *var) const
{
    double a = port->read ();
    var[0] = f(x, a);           // evaluate state change
}

void example: :calculus ()
{
    x = 0;                       // state initialization here
    while (step())               // perform an ODE solver step
        double a = port->read (); // read incident wave here
        double b = g(x, a);      // compute reflected wave
        port->write (b);         // and send it out
}
}

```

ALGORITHM 2: The structure of a typical wave module implementation.

exceeding a certain threshold is detected on any of the scattered waves. Only when no further delta cycles are scheduled, and the core SystemC scheduler advances the time are the embedded local ODE solvers (invoked by the *step()* call) permitted to update their state.

To ease the implementation of complex systems with SystemC-WMS, a number of predefined templates and classes for linear and nonlinear modules have also been provided. The full source code of the SystemC-WMS library and associated device library is freely available under an open source license from the authors' website [15].

3. MODELLING OF ANALOG RF MODULES IN SYSTEMC-WMS

The analog module representation based on (2), which has been used to model different heterogeneous analog systems, ranging from a library of simple linear systems to electrical switching power supplies [7, 8, 15], can easily be extended to better suit electrical circuit requirements in the RF domain. In particular, the representation can be modified by explicitly adding a stochastic signal, which represents the

TABLE 1: Typical RF module specifications.

Input matching	$s_{11} < \tilde{s}_{11}$
Output matching	$s_{22} < \tilde{s}_{22}$
Gain	$s_{21} \approx \tilde{s}_{21}$
Isolation	$s_{12} < \tilde{s}_{12}$
Nonlinearity	$IIP3_{dBm} < \widetilde{IIP3}_{dBm}$
Noise	$NF < \widetilde{NF}$

noise produced by the module itself or by some external interference,

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{a}, \mathbf{n}), \\ \mathbf{b} &= \mathbf{g}(\mathbf{x}, \mathbf{a}). \end{aligned} \quad (8)$$

This is now a stochastic differential equation system, that can be solved in the usual way by considering a realization of the stochastic signal $\mathbf{n}(t)$. If the system can be assumed ergodic, all of the signal features in the probability space can be estimated from time-domain solutions.

To simplify the system-level simulation of (8), some modules can be considered to be instantaneous, resulting in the following simplified representation:

$$\begin{aligned} \mathbf{y}(t) &= \mathbf{S}(\mathbf{a}(t) + \mathbf{n}(t)), \\ \mathbf{b}(t) &= \mathbf{g}(\mathbf{y}(t)), \end{aligned} \quad (9)$$

in the following, for the sake of simplicity, the explicit dependence on time will be neglected. The rationale for (9) is to simulate at system level analog RF modules, like a low noise amplifier (LNA), a mixer, or a power amplifier (PA), in the worse conditions, which happen when considering the S-parameters of the two-port to be constant in frequency, and equal to the specifications. This type of representation allows a straightforward extraction of the parameters of (9) from common RF specifications, such as those shown in Table 1, without the need of assuming any particular circuit implementation for the modules themselves.

As an example to demonstrate how the parameters of (9) can be extracted, and this model applied to specific RF circuits, amplifier and mixer modules will be analyzed in detail in the following.

3.1. Amplifier

Since the nonlinearity involving electrical quantities at the input port can be neglected for amplifiers—indeed there are usually no specifications regarding the nonlinear behavior of the input impedance—(9) can be customised as

$$\begin{aligned} y_1 &= \tilde{s}_{11}(a_1 + n_1) + \tilde{s}_{12}(a_2 + n_2), \\ y_2 &= \tilde{s}_{21}(a_1 + n_1) + \tilde{s}_{22}(a_2 + n_2), \\ b_1 &= y_1, \\ b_2 &= g_2(y_2). \end{aligned} \quad (10)$$

Figure 4 shows a schematic representation of these relationships.

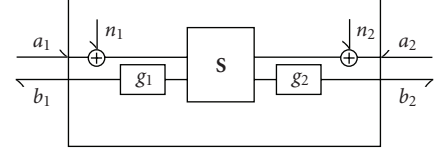


FIGURE 4: An RF module in SystemC-WMS.

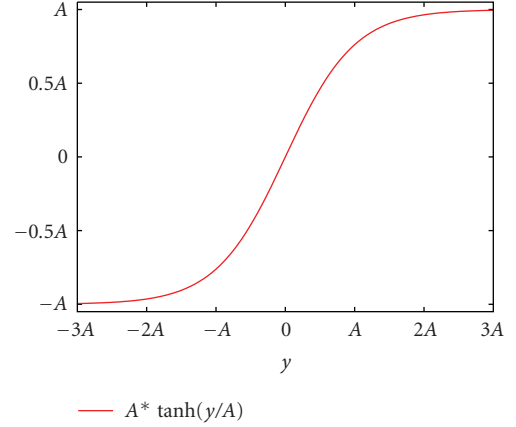


FIGURE 5: LNA output nonlinearity.

A nonlinear function that well represents the amplifier output nonlinearity, and that can easily be characterized from the specifications reported in Table 1, is the hyperbolic tangent. In the following, we will thus assume that

$$g(y) = A \cdot \tanh\left(\frac{y}{A}\right), \quad (11)$$

so that the small-signal gain is directly set by the values of the linear S-parameters, with a saturation level depending on the value of the parameter A , as can be seen in Figure 5.

3.1.1. Noise

A noisy two-port is generally modelled by adding a couple of noise generators to the input or to the output of a noise-free two-port. A convenient model for our goal is the noise representation based on the a, b parameters, as reported in [16] with output-referred noise generators, or more conveniently with input-referred noise generators, which results in

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = [\mathbf{S}] \begin{bmatrix} a_1 + n_1 \\ a_2 + n_2 \end{bmatrix}, \quad (12)$$

which, when the nonlinearity can be neglected, as it is usually done in the case of noise analysis, corresponds to (10).

A noisy two-port can be characterized, in terms of noise performance, by its noise factor F defined as [16]

$$F = \frac{(S/N)_I}{(S/N)_O} = 1 + \left(\frac{N_{AI}}{N_I}\right), \quad (13)$$

where N_{AI} is the added noise power, referred to the input, and N_I is the noise power entering the input port. The noise

factor, when expressed in decibel, is called noise figure, $NF = 10 \log(F)$.

Since the noise specification of an amplifier is usually expressed only in terms of a scalar variable (F or NF), only the parameters of one of the two noise generators can be determined. Because of this we have chosen to put $n_2 = 0$ in (12).

A noisy resistor can give a maximum noise power equal to

$$P_{\text{MAX}} = \frac{\overline{v_n^2}}{4R} = \frac{4kTRB}{4R} = kTB, \quad (14)$$

where k is the Boltzmann constant, T is the temperature, and B the bandwidth. This relationship can be applied to (13) to obtain

$$\sigma_n^2 = \overline{n_1^2} = kTB(F - 1). \quad (15)$$

This relationship lets us add a stationary white noise wave generator of variance that depends only upon the specifications reported in Table 1.

3.1.2. Nonlinearity

The nonlinearity performance analysis of an amplifier module can be performed, following [17], by approximating the nonlinearity (11) by its Taylor expansion truncated to the third order:

$$g(y) = y - \frac{y^3}{3A^2} = \alpha_1 y + \alpha_3 y^3, \quad (16)$$

which leads to the following expression for the OIP3 :

$$\widehat{\text{OIP3}} = \alpha_1 \sqrt{\frac{4}{3} \left| \frac{\alpha_1}{\alpha_3} \right|} = 2A. \quad (17)$$

We have used the hat sign in the OIP3 to underline the fact that (17) regards the amplitude of the signals, which in a, b parameters is expressed in \sqrt{W} . Since the IIP3 or OIP3 are usually expressed in terms of power, (17) must be translated from \sqrt{W} to W . It is straightforward to relate the amplitude of incident and reflected waves to the mean power available at a port,

$$\begin{aligned} P_{\text{av}} &= \frac{1}{T} \int (b^2(t) - a^2(t)) dt \\ &= \frac{1}{T} \int b^2(t) dt = \frac{\widehat{b}^2}{2} \end{aligned} \quad (18)$$

obtained for sinusoidal signals and neglecting the incident wave at the port. This relation can be rewritten as $\widehat{b} = \sqrt{2P_{\text{av}}}$, or $\widehat{\text{OIP3}} = \sqrt{2\text{OIP3}}$. Since the specifications are usually expressed in terms of IIP3_{dBm}, it is easy to find that

$$\text{OIP3}_{\text{dBm}} = \text{IIP3}_{\text{dBm}} + \text{TPG}_{\text{dB}}. \quad (19)$$

Finally, from the preceding equations, we can express the nonlinear parameter characterizing (10), in term of specifications as $A = 1/2\widehat{\text{OIP3}}$, or

$$A = \frac{1}{2} \sqrt{2 \cdot \text{OIP3}}. \quad (20)$$

TABLE 2: Amplifiers' specifications.

Parameter	LNA spec.	PA spec.	PGA spec.
\tilde{s}_{11}	-15 dB	-15 dB	—
\tilde{s}_{12}	-40 dB	-40 dB	—
\tilde{s}_{21}	18 dB	18 dB	—
\tilde{s}_{22}	-15 dB	-15 dB	—
$\widehat{\text{IIP3}}$	0 dBm	18 dBm	—
$\widetilde{\text{NF}}$	4 dB	—	34 dB

Equation (10), which can be assumed to be a model valid for every type of amplifier, can finally be rewritten with all previously extracted parameters as

$$\begin{aligned} b_1 &= \tilde{s}_{11}(a_1 + n_1) + \tilde{s}_{12}a_2, \\ y_2 &= \tilde{s}_{21}(a_1 + n_1) + \tilde{s}_{22}a_2, \\ A &= \frac{1}{2} \sqrt{2 \cdot \widehat{\text{OIP3}}}, \\ b_2 &= A \cdot \tanh\left(\frac{y_2}{A}\right), \\ \overline{n_1^2} &= kTB(\tilde{F} - 1). \end{aligned} \quad (21)$$

3.1.3. Implementation

With the facilities provided by SystemC-WMS and the equations previously reported in (21), it is straightforward to implement a high-level module to model an LNA. Its full implementation is reported in Algorithm 3, where *noise* is another library class that generates a sequence of gaussian random numbers, of variance equal to the desired noise power, and then modulates the result into the requested frequency band, so as to produce a brick wall-shaped noise spectrum with the desired properties.

The LNA so defined can then be used in constructs like the one reported in Algorithm 4, where the tags like dB or GHz that appear after several of the numerical constants are simple macros that implement the trivial scaling and log-to-linear conversions of the parameters as needed.

3.2. Mixer

The relations (8) for the mixer take the following form:

$$\begin{aligned} b_1 &= \tilde{s}_{11}(a_1 + n_1) + \tilde{s}_{1\text{LO1}}a_{\text{LO}} + \tilde{s}_{12}a_2, \\ y_2 &= K(1 + \tilde{s}_{22})[(1 + \tilde{s}_{11})(a_1 + n_1) + \tilde{s}_{1\text{LO1}}a_{\text{LO}} + \tilde{s}_{12}a_2] \\ &\quad \times \frac{a_{\text{LO}}}{A_{\text{LO}}} + \tilde{s}_{22}a_2, \\ A &= \frac{1}{2} \sqrt{2 \cdot \widehat{\text{OIP3}}}, \\ b_2 &= A \cdot \tanh\left(\frac{y_2}{A}\right), \end{aligned} \quad (22)$$

```

struct lna : wave_module <2, electrical>
{
    SC_HAS_PROCESS (amplifier_scatter);
    amplifier_scatter (sc_core :: sc_module_name name,
        scatter S11, scatter S12, scatter S21, scatter S22,
        double IIP3, double F, double f0, double band,
        double R01 = 50 ohm, double R02 = 50 ohm
    ) :
        S11(S11), S12(S12), S21(S21), S22(S22),
        A (0.5 * sqrt (2 * IIP3) * S21),
        n1 (k * T0 * band * (F - 1), f0, band)
    {
        SC_METHOD (calculus); sensitive << activation;
        // set normalization resistances on ports:
        port [0] <<= R01;
        port [1] <<= R02;
    }
private:
    void calculus ()
    {
        double t = sc_core :: sc_time_stamp (). to_seconds ();
        double a1 = port [0] -> read () + n1 (t);
        double a2 = port [1] -> read ();
        double y1 = S11 * a1 + S12 * a2;
        double y2 = S21 * a1 + S22 * a2;
        port [0] -> write (y1);
        port [1] -> write (A * tanh (y2 / A));
    }
private: /* data members */
    const double S11, S12, S21, S22;
    const electrical :: wave_type A;
    noise n1;
};
    
```

ALGORITHM 3: Definition of the LNA module in SystemC-WMS.

```

ab_signal (electrical, parallel) antenna(50 ohm);
ab_signal (electrical, parallel) amplified(50 ohm);
source <electrical> src1("RX-SIGNAL", cfg :: wave);
src1(antenna, ...); // connection of signal source
lna amp1("LNA1",
    -15 dB, -40 dB, 18 dB, -15 dB, //S-parameters
    0 dBm, 4 dB, // nonlinearity and noise specs
    2.485 GHz, 1 MHz); // band specification
amp1 (antenna, amplified); //connections of the LNA
    
```

ALGORITHM 4: Example of a typical usage of the LNA module.

The mixer has been modelled by a two-port electrical module with an SFG input a_{LO} , that is the input of the local oscillator. This signal has been modelled by a sinusoidal function:

$$a_{LO} = A_{LO} \cos(\omega_{LO}t + \phi), \quad (23)$$

where $A_{LO} = \sqrt{2P_{avLO}}$ is a function of local oscillator power. The reflected wave at the input port of the mixer b_1 is linearly

TABLE 3: TX-RX Mixer specifications.

Parameter	TX spec.	RX spec.
\tilde{s}_{11}	-15 dB	-15 dB
\tilde{s}_{12}	-90 dB	-90 dB
\tilde{s}_{21}	2 dB	12 dB
\tilde{s}_{22}	-15 dB	-15 dB
\tilde{s}_{LO1}	-86 dB	-90 dB
$\tilde{IIP3}$	15 dBm	12 dBm
\tilde{NF}	—	24 dB

related to all the inputs of the module by the following RF specifications of the mixer:

- (i) the input reflection coefficient \tilde{s}_{11} ;
- (ii) the power transfer from the output port \tilde{s}_{12} ;
- (iii) the power transfer from the local oscillator \tilde{s}_{LO} .

The relationship that characterizes y_2 has been obtained considering that the output current of the mixers based on Gilbert multiplier is approximately $i'_2(t) \approx Gv_1(t)v_{LO}(t)$. On the base of the proposed model, the normalized current $i'_{2n}(t)$ can be expressed as a function of the incident and reflected wave at the input port as follows:

$$i'_{2n}(t) = G\sqrt{R_{02}R_{01}}(a_1(t) + n_1(t) + b_1(t))a_{LO}(t). \quad (24)$$

Now, substituting the expression for b_1 in (24), we obtain

$$i'_{2n} = G\sqrt{R_{02}R_{01}}[(1 + \tilde{s}_{11})(a_1 + n_1) + \tilde{s}_{LO}a_{LO} + \tilde{s}_{12}a_2]a_{LO}. \quad (25)$$

The output port of the mixer has been, therefore, modelled by a controlled current generator that provides the normalized current of (25) with a parallel load of impedance R_L . The relationship between b_2 and a_2 of such a kind of circuit can be expressed as follows:

$$y_2 = \frac{1 + \tilde{s}_{22}}{2}i'_{2n} + \tilde{s}_{22}a_2 \quad (26)$$

with $\tilde{s}_{22} = (R_L - R_{02})/(R_L + R_{02})$. Now substituting (25) in (26), we obtain the final expression for b_2 , where the power conversion gain is provided by the factor $K = G\sqrt{R_{02}R_{01}}A_{LO}/2$.

4. BLUETOOTH TRANSCEIVER

4.1. Bluetooth RF specifications

Bluetooth [18] is a widely used wireless standard for devices that have regular charge (e.g., mobile phones) and in applications like handsfree audio and file transfer.

The Bluetooth frequency band is comprised within the industrial, scientific, and medical (ISM) band, between 2.4000 GHz and 2.4835 GHz and is subdivided into 79 channels. Three power classes of devices are defined, with

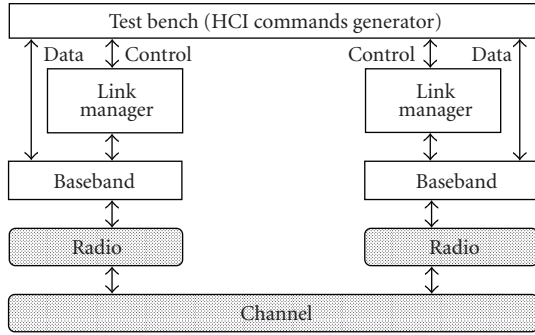


FIGURE 6: Bluetooth baseband and link layer, as modelled in [19] (in white), with the addition of the radio layer (in gray).

different output power requirements. For a class 2 device, that has a rated maximum operating distance of 10 m, the maximum output power is nominally 1 mW (0 dBm), and must always be within the range of 0.25 mW (−6 dBm) to 2.5 mW (4 dBm). The modulation employs a Gaussian frequency shift keying (GFSK) with $BT = 0.5$, a modulation index that must lie between 0.28 and 0.35, and a symbol rate of 1 MS/s. The symbol rate accuracy must be better than $20 \mu\text{s/s}$, the minimum frequency deviation must never be smaller than 115 kHz. The zero crossing error, that is the time difference between the ideal symbol period and the measured crossing time, must be less than 0.125 microsecond (1/8 of a symbol period at 1 MS/s). The transmitter initial center frequency accuracy must be 75 kHz. The Bluetooth receiver must have an actual sensitivity level of −70 dBm or better, where the actual sensitivity level is defined as the input level for which the BER is 0.001.

4.2. RF layer architecture

The Bluetooth baseband and link layers have previously been modelled in SystemC [19], without considering the analog radio layer. The heterogeneous modelling methodology allowed by SystemC-WMS, described earlier in this paper, permits the modelling and simulation of the lower layers of the Bluetooth stack including the analog radio layer, as reported in Figure 6. The possibility of performing a fast and yet detailed simulation, requiring a CPU time of less than 10 seconds for simulating the transmission, RF propagation, and reception of each bit (i.e., 1 microsecond of simulated time), allows the verification of the effect of changes in the design specifications of the analog blocks on the overall system performance at packet level.

In the rest of this section, a detailed description of the radio layer is reported. Many architectures of transmitters and receivers for the 2.4 GHz band have been proposed, some of them suitable for a single standard, some for multiple standards. The receivers can be classified according to their operation as superheterodyne, image-reject, zero-IF, or low-IF. Analogously, the transmitters can be classified as superheterodyne, direct-up, or two-step-up. A review of the state of the art in wireless transceivers can be found in [20].

The designed transceiver is composed of a transmitter and a receiver whose structures are reported in Figures 7 and 8, respectively. The low-IF receiver (RX) is derived from the designs reported in [21, 22]. The LNA amplifies the signal from the antenna. The voltage controlled oscillator (VCO) generates the in-phase (I) and quadrature-phase (Q) components for the RF to intermediate frequency (IF) down conversion performed by the mixer. A low-band filter centered at the 2 MHz IF rejects the image signal. The programmable gain amplifier (PGA) amplifies the signal to the desired level. The IF to baseband down conversion is performed digitally after the A/D conversion, whose resolution can easily be changed in the code. The specifications of the single blocks, derived from the Bluetooth RF specifications, are reported in Tables 2-3.

The direct-conversion transmitter (TX) is reported in Figure 7, and is derived from the designs reported in [22, 23]. A digital Gaussian filter is implemented using a lookup Table (LUT), the signal is integrated and then the baseband in-phase (I) and quadrature (Q) components are generated. The I and Q digital signals are converted to analog signals, low pass filters clean the signals and single side band mixers upconvert them directly to 2.4 GHz. A power amplifier amplifies the signal before feeding it to the antenna. The specifications for the single blocks are reported in Tables 2-3.

4.3. Simulation results

The complete system with transmitter and receiver was simulated with the specifications previously discussed.

The propagation of the signal from the transmitter to the receiver was modelled by a classical additive white Gaussian noise (AWGN) channel with an attenuation factor of $\lambda^2/(4\pi d^2)$, where d is the distance between transmitter and receiver and λ is the RF wavelength. The noise level introduced by the channel was set to kT_0 with $T_0 = 290$ K, that corresponds to −144 dBm/kHz.

The simulation of the transmission of 50 bits took about 8 minutes on a 3 GHz Pentium D processor. Figure 9 shows the simulation results for different signals sensed inside the digital portions of the transmitter and of the receiver during this simulation. The first signal is the digital data stream to be transmitted, at the input of the Gaussian filter, followed by the output of the gaussian filter itself. The third signal is the received signal at the output of the quadrice correlator followed again by the digital data stream obtained at the output of the receiver.

The performance of the complete system was measured by means of the normalized root mean square error (RMSE), computed as

$$\text{Normalised RMSE} = \frac{\overline{(s_i(t) - s_o(t))^2}}{\overline{(s_i(t))^2}}, \quad (27)$$

where the transmitted signal s_i is sensed at the output of the Gaussian filter, and the received signal s_o at the output of the quadrice correlator. Both signals have been synchronized and rescaled before applying (27).

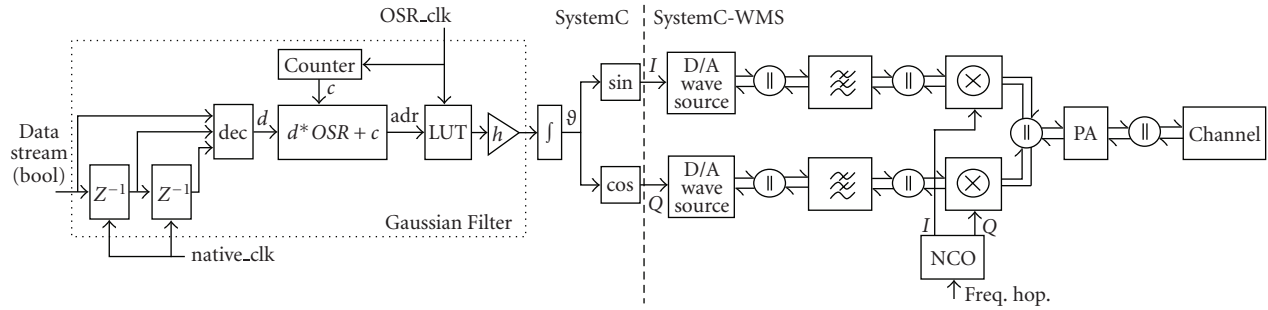


FIGURE 7: Transmitter block diagram.

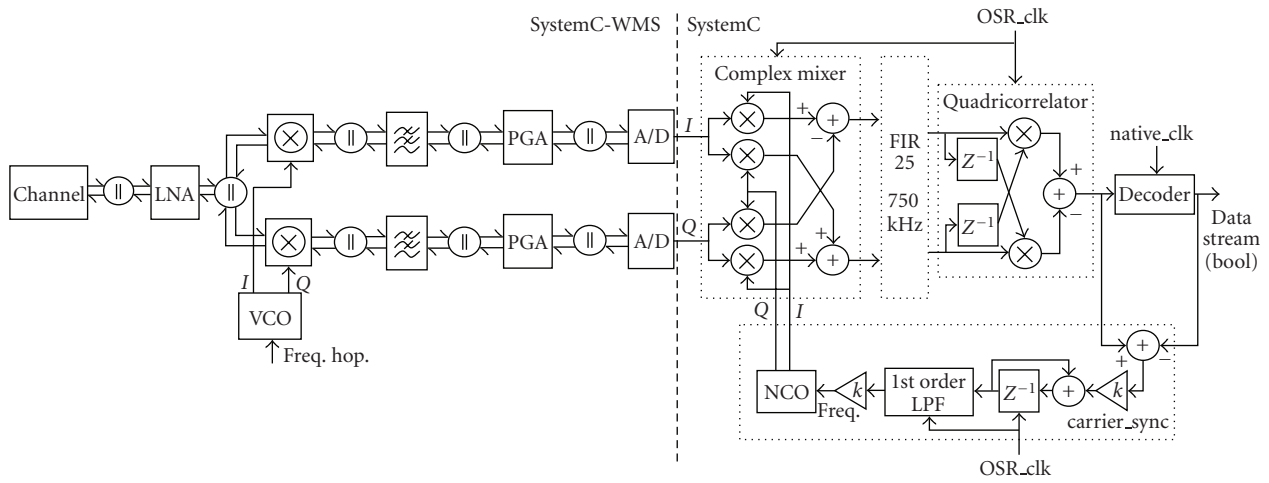


FIGURE 8: Receiver block diagram.

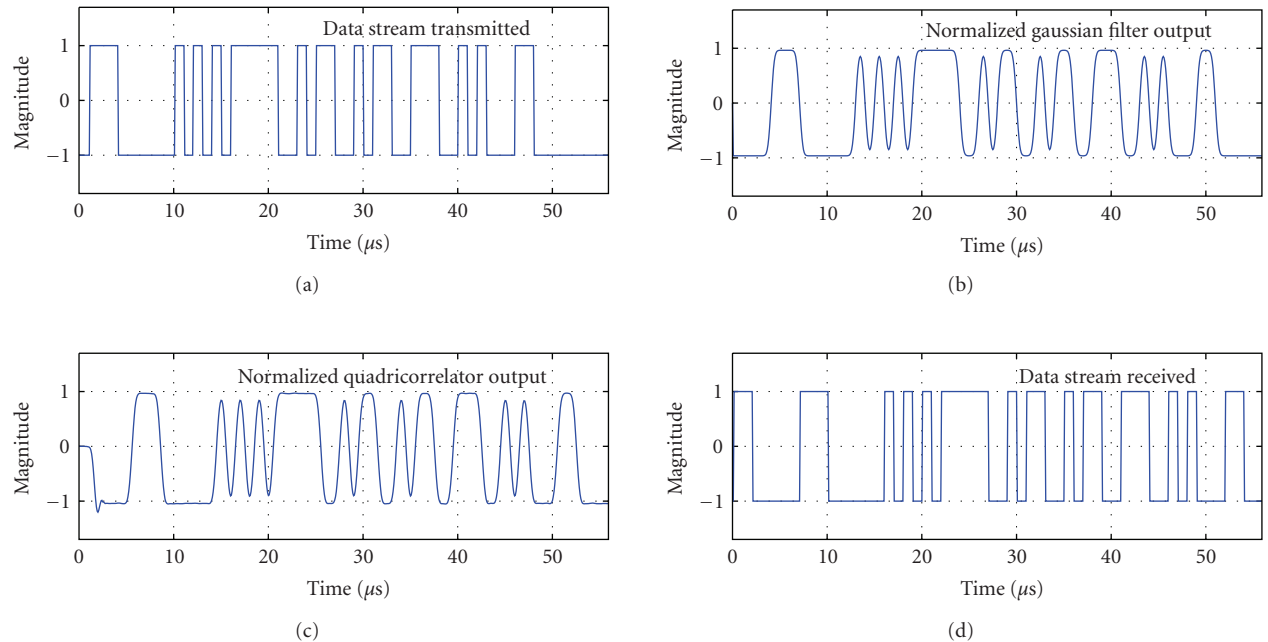


FIGURE 9: Example of transmitted and received signals inside the digital portions of the transceiver. All the magnitudes have been normalised according to the available resolution.

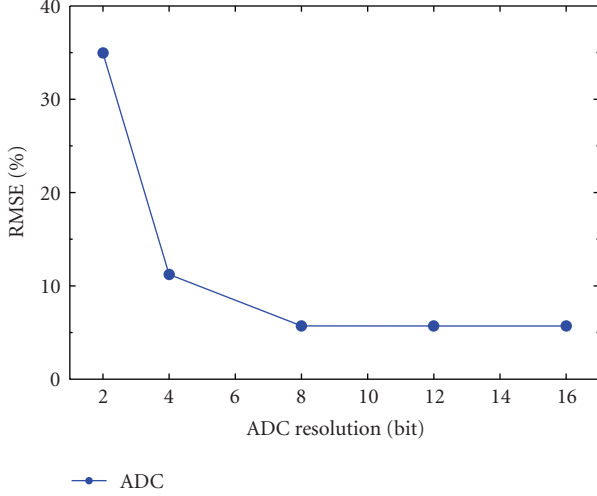


FIGURE 10: Normalized RMSE of the whole system as a function of ADC resolution.

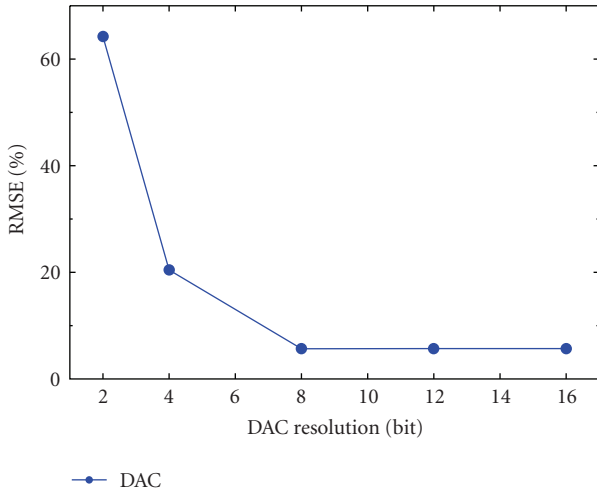


FIGURE 11: Normalized RMSE of the whole system as a function of DAC resolution.

The RMSE is affected by several factors, related to both the transmitter and the receiver, such as output-stage nonlinearity, DAC nonideality, LPF nonideality, and VCO nonideality.

Figures 10-11 show the simulated RMSE performance versus the ADC and DAC resolutions. The curve in Figure 10 was obtained by calculating the RMSE for different values of the ADC resolution, ranging from 2 to 16, while the DAC resolution was kept fixed at the maximum value. Analogously, the same sweep was performed on the DAC resolution, keeping constant the other, in order to obtain the results shown in Figure 11.

The receiver was designed in order to have a noise figure of about 12 dB [22], as can easily be verified applying Friis equation [16] to the noise figure and power gain

TABLE 4: Transducer power gain and noise figure of various receiver stages as obtained from simulation results.

Receiver	TPG	NF
LNA	17.7 dB	4 dB
MIXER	14.4 dB	24.2 dB
PGA	12 dB	34.2 dB
RX	41.1 dB	12.9 dB

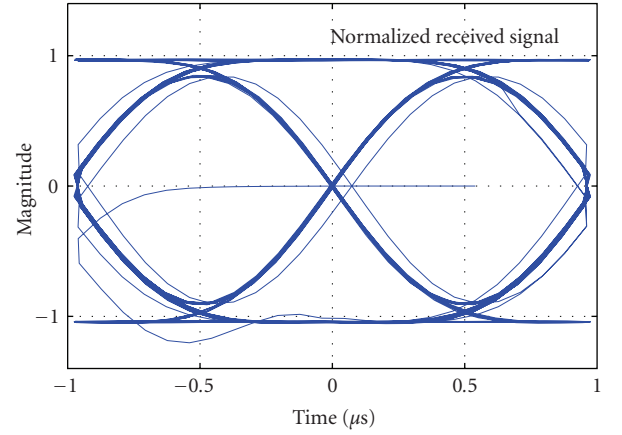


FIGURE 12: Eye-diagram measured at quadricorrelator output, with only thermal noise in the channel and noisy receiver as per the specifications. The bit time is $T_b = 1 \mu s$.

specifications given for each of the receiver components in Tables 2-3:

$$NF_{RX} = 10 \log \left(F_{lna} + \frac{F_{mixer} - 1}{TPG_{lna} A_{splitter}} + \frac{F_{pga} - 1}{TPG_{lna} A_{splitter} TPG_{mixer}} \right) = 12 \text{ dB}, \quad (28)$$

where $A_{splitter}$ is the power attenuation due to the splitter at the output of the LNA, that is of about 3 dB.

In order to experimentally evaluate the noise figure of the receiver from simulations, two transmissions of the same 50-bit sequence had been simulated, one with noise and another without noise. The power of the signal (S) and of the noise (N) at the input and at the output of each stage of the receiver was then calculated from the data provided by this two simulations, leading to the following result:

$$NF_{RX} = 10 \log \left(\frac{S_{lna}}{N_{lna}} \cdot \frac{N_{opga}}{S_{opga}} \right) = 12.9 \text{ dB}. \quad (29)$$

These simulations allowed us to also evaluate the power gain and the noise figure of each stage of the receiver. These data are reported in Table 4, and it is possible to observe a substantial agreement between them and the corresponding specifications in Tables 2-3.

Figure 12 shows the eye-diagram performance of the receiver, while Figure 13 shows the power spectrum at the receiver input, obtained by the transmission of a sequence

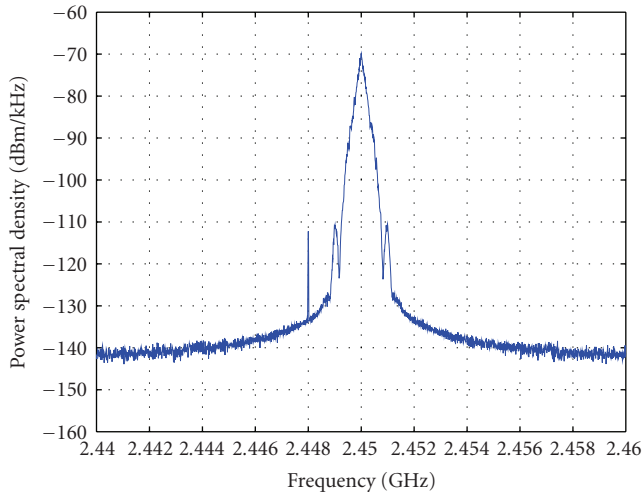


FIGURE 13: Power spectral density of the RF signal measured at the LNA input.

of 1024 random bits. The narrow peak at 2.448 GHz is the local oscillator feedback due to the imperfect mixer isolation (-90 dB) and finite reverse gain (-40 dB) of the LNA.

5. CONCLUSIONS

This work proposes an effective methodology that simplifies the modelling of the interaction between analog models belonging to heterogeneous domains, as well as model reuse. By using power waves as standard input/output signals for analog modules, these can be independently modelled and freely interconnected together in arbitrary topologies. The main peculiarity of SystemC-WMS is to allow non-linear dynamical analog models simulation seamlessly with interconnections that automatically take care of load and interconnection effects.

This paper also proposes a methodology for system level modelling of RF systems in SystemC-WMS, directly based on the high-level specifications of the RF modules. The proposed methodology was applied to the modelling of a Bluetooth transceiver, and the simulation results reported and discussed.

REFERENCES

- [1] "ITRS, 2005 Edition: Design," International Technology Roadmap for Semiconductors, December 2005.
- [2] "MEDEA Electronic Design Automation (EDA) Roadmap, 5th release," MEDEA+, September 2005.
- [3] The Open SystemC Initiative, (OSCI), "SystemC documentation," <http://www.systemc.org/>.
- [4] A. Vachoux, C. Grimm, and K. Einwich, "SystemC-AMS requirements, design objectives and rationale," in *Proceedings of the Design, Automation and Test in Europe (DATE '03)*, pp. 388–393, Munich, Germany, March 2003.
- [5] A. Vachoux, C. Grimm, and K. Einwich, "Analog and mixed signal modelling with SystemC-AMS," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '03)*, vol. 3, pp. 914–917, Bangkok, Thailand, May 2003.
- [6] OSCI Analog/Mixed-Signal Working Group, (AMSWG), "SystemC-AMS documentation," <http://www.systemc.org/apps/group-public>.
- [7] S. Orcioni, G. Biagetti, and M. Conti, "SystemC-WMS: mixed signal simulation based on wave exchanges," in *Applications of Specification and Design Languages for SoCs*, A. Vachoux, Ed., ChDL, chapter 10, pp. 171–185, Springer, Berlin, Germany, 2006.
- [8] S. Orcioni, G. Biagetti, and M. Conti, "SystemC-WMS: a wave mixed signal simulator," in *Proceedings of the Forum on Specifications & Design Languages (FDL '05)*, pp. 61–72, Lausanne, Switzerland, September 2005.
- [9] J. Bjørnsen, T. E. Bonnerud, and T. Ytterdal, "Behavioral modeling and simulation of mixed-signal System-on-a-Chip using SystemC," *Analog Integrated Circuits and Signal Processing*, vol. 34, no. 1, pp. 25–38, 2003.
- [10] H. Al-Junaid and T. Kazmierski, "Analogue and mixed-signal extension to SystemC," *IEE Proceedings: Circuits, Devices and Systems*, vol. 152, no. 6, pp. 682–690, 2005.
- [11] F. Gambini, M. Conti, S. Orcioni, F. Ripa, and M. Caldari, "Physical modelling in SystemC-WMS and real time synthesis of electric guitar effects," in *Proceedings of the 5th Workshop on Intelligent Solutions in Embedded Systems (WISES '07)*, pp. 87–100, Madrid, Spain, June 2007.
- [12] N.-J. Oh and S.-G. Lee, "Building a 2.4-GHz radio transceiver using IEEE 802.15.4," *IEEE Circuits and Devices Magazine*, vol. 21, no. 6, pp. 43–51, 2005.
- [13] K. Kurokawa, "Power waves and the scattering matrix," *IEEE Transactions on Microwave Theory and Techniques*, vol. 13, no. 2, pp. 194–202, 1965.
- [14] A. Fettweis, "Pseudo-passivity, sensitivity, and stability of wave digital filters," *IEEE Transactions on Circuits Theory*, vol. 19, no. 6, pp. 668–673, 1972.
- [15] G. Biagetti, M. Conti, and S. Orcioni, "SystemC-WMS home page," <http://www.deit.univpm.it/systemc-wms/>.
- [16] G. D. Vendelin, A. M. Pavo, and U. L. Rohde, *Microwave Circuit Design Using Linear and Nonlinear Techniques*, John Wiley & Sons, New York, NY, USA, 1990.
- [17] B. Razavi, *RF Microelectronics*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1998.
- [18] "Specification of the Bluetooth System," 1st ed., Bluetooth, December 1999.
- [19] M. Conti and D. Moretti, "System level analysis of the Bluetooth standard," in *Proceedings of the Design, Automation and Test in Europe (DATE '05)*, vol. 3, pp. 118–123, Munich, Germany, March 2005.
- [20] P.-I. Mak, S.-P. U, and R. P. Martins, "Transceiver architecture selection: review, state-of-the-art survey and case study," *IEEE Circuits and Systems Magazine*, vol. 7, no. 2, pp. 6–24, 2007.
- [21] D.-C. Chang and T.-H. Shiu, "Digital GFSK carrier synchronization," in *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS '06)*, pp. 1523–1526, Singapore, December 2006.
- [22] H. Darabi, S. Khorram, H.-M. Chien, et al., "A 2.4-GHz CMOS transceiver for Bluetooth," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 12, pp. 2016–2024, 2001.
- [23] T. A. D. Riley and M. A. Copeland, "A simplified continuous phase modulator technique," *IEEE Transactions on Circuits and Systems II*, vol. 41, no. 5, pp. 321–328, 1994.