

A VHDL-AMS Simulation Environment for an UWB Impulse Radio Transceiver

*Original*

A VHDL-AMS Simulation Environment for an UWB Impulse Radio Transceiver / Casu, MARIO ROBERTO; Crepaldi, Marco; Graziano, Mariagrazia. - In: IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. I, REGULAR PAPERS. - ISSN 1549-8328. - STAMPA. - 55:(2008), pp. 1368-1381. [10.1109/TCSI.2008.916402]

*Availability:*

This version is available at: 11583/1838779 since:

*Publisher:*

IEEE

*Published*

DOI:10.1109/TCSI.2008.916402

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# A VHDL-AMS Simulation Environment for an UWB Impulse Radio Transceiver

Mario R. Casu, *Member, IEEE*, Marco Crepaldi, and Mariagrazia Graziano, *Member, IEEE*

**Abstract**—Ultra-wide-band (UWB) communication based on the impulse radio paradigm is becoming increasingly popular. According to the IEEE 802.15 WPAN Low Rate Alternative PHY Task Group 4a, UWB will play a major role in localization applications, due to the high time resolution of UWB signals which allow accurate indirect measurements of distance between transceivers. Key for the successful implementation of UWB transceivers is the level of integration that will be reached, for which a simulation environment that helps take appropriate design decisions is crucial. Owing to this motivation, in this paper we propose a multiresolution UWB simulation environment based on the VHDL-AMS hardware description language, along with a proper methodology which helps tackle the complexity of designing a mixed-signal UWB system-on-chip. We applied the methodology and used the simulation environment for the specification and design of an UWB transceiver based on the *energy detection* principle. As a by-product, simulation results show the effectiveness of UWB in the so-called ranging application, that is the accurate evaluation of the distance between a couple of transceivers using the *two-way-ranging* method.

**Index Terms**—Mixed-signal integrated circuits, ultra-wide-band (UWB) communications, VHDL-AMS.

## I. INTRODUCTION

ACCORDING to the definition of the Federal Communications Commission (FCC) an ultra-wide-band (UWB) signal is characterized by a bandwidth of minimum 500 MHz or by a fractional bandwidth of at least 20%, regardless of the type of modulation or system of transmission [1]. In 2002, FCC released the spectrum between 3.1 and 10.6 GHz for unlicensed use with UWB signals, provided that severe average and peak power constraints are respected. The broad UWB definition and the large free spectrum led to many different proposals of more or less conventional modulation strategies [like wide-band orthogonal frequency-division multiplexing (OFDM) and code-division multiple access (CDMA)], which are envisaged for applications like wireless personal area networks (WPANs), (see, for example, the former work of the IEEE 802.15 WPAN High Rate Alternative PHY Task Group 3a [2] and of the current WiMedia Alliance [3]). However, UWB is more commonly referred to as a “baseband” or “impulse-based” communication technology, because one of the possibility to exploit such a large bandwidth is to directly send fast rise-time and short duration carrier-free pulses to a wide-band antenna. This is certainly not new because the origins root back to the

early 60’s work on time-domain electromagnetics [4], but the recent release of the spectrum renovated the interest in this fascinating field of wireless transmission. This paper thus deals with the simulation of UWB circuits and systems under this more accepted significance, that is of short time support, on the order of one nanosecond, baseband signals.

One of the most attractive features of UWB is the locationing capability, enabled by the possibility of isolating the first echo of the signal received through a multipath channel. The large bandwidth is the key for such accurate time-domain resolution, which translates into an accurate distance measurement [5]. UWB transceivers with locationing capabilities may open the way to a number of applications within the WPAN field, like logistics (package tracking), security (localizing persons in controlled areas), medical applications (monitoring of patients), search-and-rescue functions (communications with fire fighters), control of home appliances. An IEEE standardization group is currently working toward an alternative physical layer of the 802.15.4 standard with the aim of enabling high-precision localization (on the order of 1 m) [6].

Adoption of a new technology will depend primarily on keeping unit cost per device and power consumption as low as possible. The complete integration of UWB transmitter and receiver functions in the same system-on-chip (SoC), possibly using a standard CMOS technology, is then crucial. On the one hand, pulse-based UWB systems can be simpler than classic narrowband transceivers, because continuous wave carriers are not used, then making the SoC design somewhat easier. On the other hand, narrowband consolidated techniques are not suited to this case, as the design has to deal mostly with time-domain signal representations rather than frequency-domain. On top of that, the mixed-signal nature of the SoC, due to the coexistence of digital, analog and RF parts, makes the design process a nontrivial task. We thus believe that there is a need for a simulation environment with the following characteristics.

- It has to be flexible enough to allow both rapid assessment of system-level choices and accurate evaluation of circuit-level alternatives.
- The interaction of the two levels (system and circuit) must be explicitly brought to light in such a way that the impact of changes at the lower level are captured in the behavior of the higher level simulation.

We call *Multiresolution* a simulator with these features. In the digital design domain, hardware description languages (HDL) like VHDL and Verilog have been the key to enable this type of multiresolution simulations, thus paving the way to the design of extremely complex integrated circuits. Their extension to the analog and mixed-signal domain, viz. VHDL-AMS and Verilog-A, is recognized as the enabling instrument for taming the complexity of SoCs that feature analog, digital and RF

Manuscript received June 30, 2006; revised December 23, 2006, and June 6, 2007. This paper was recommended by Associate Editor B. Zhao.

The authors are with the Department of Electronics, Politecnico di Torino, I-10129 Torino, Italy (e-mail: mario.casu@polito.it; marco.crepaldi@polito.it; mariagrazia.graziano@polito.it).

Digital Object Identifier 10.1109/TCSI.2008.916402

parts. Although nowadays a number of commercial tools may be helpful each for a different part of the entire design flow (e.g., Matlab and SystemVue for system-level, ADS for RF design), we believe that VHDL-AMS and Verilog-A, as currently implemented in commercial programs (like Mentor ADMS) offer some important advantages. We value in particular the possibility of using a *single* tool for all design phases, from top to transistor-level simulations, then avoiding burdensome language translations. Moreover, when mixed-signal circuits are at stake, a significant portion of the design is digital. Sharing a single language (VHDL/Verilog and their analog extensions) in a heterogeneous team including system, analog, RF and digital circuit designers, is a significant benefit that helps improve efficiency and productivity.

We adopted VHDL-AMS as the hardware description language for our simulation environment. In this paper we show how this approach has been applied to the conception and simulation of an entire UWB transceiver. We first describe in this paper the architecture of an *energy detection* transceiver. Then we show functional simulation results and, as a case of study, simulations of the *two-way-ranging* (TWR) application, that is the evaluation of distance of two transceivers based on the time-of-flight (TOF) of a UWB signal traveling forth and back between these devices [7]. The strength of the methodology is evidenced by the quality of the information drawn from the simulations and by the most important by-product, that is the effectiveness of UWB in the accurate evaluation of distance.

The paper is so organized. We review the recent literature on UWB circuit and system design in Section II-A and in the Analog HDL field in Section II-B. Then Section III describes the transceiver architecture. The simulation environment is discussed in Sections IV while Sections V–VII report functional and TWR results. Final remarks, conclusions and discussion about our forthcoming work in the field are in Section VIII.

## II. RELATED WORK

### A. UWB Circuits

An impulse-based UWB transceiver consists of a transmitter section (TX) made of a pulse generator driven by a modulator, and of a receiver section (RX) whose complexity varies depending on the type of communication adopted. The TX part is conceptually simple. For WPAN applications like the ones envisaged by the IEEE 802.15.4a Task Group, the modulation is a simple 2-PPM with a random binary phase ( $\pm 1$ ) for the sake of smoothing the signal spectrum [6]. Several papers report different solutions for the generation of pulses and for positioning them on two halves of a symbol period, for 2-PPM modulation. We do not review here circuits based on off-the-shelf discrete components (but see [8] for an example) because this work focuses on fully integrated solutions. The transmitter reported in [9] is a low-power low-cost solution in a standard 0.18- $\mu\text{m}$  CMOS technology, based on a triangular pulse of defined bandwidth whose central frequency, needed to accommodate the signal in the FCC unlicensed spectrum, is obtained by multiplication with a local oscillator. A different approach is followed in [10] and [11] where the pulse is directly built in the allocated bandwidth, without up-conversion, by combining

baseband digital pulses. Another possibility to avoid local oscillators consists in shaping the signal by exploiting nonlinearities of MOS transistors and an *RLC* network [12].

The RX architecture depends on the type of modulation and the strategy adopted for demodulation. We can broadly classify UWB receivers in *coherent* and *noncoherent*. The former correlate the incoming UWB pulses with an internally generated template, ideally matched to the channel impulse response. Unfortunately, the channel response estimation involves high sampling rates and intensive signal processing, which is hardly compatible with low power consumption. Remarkable examples of CMOS coherent-receivers are [13], [14]. The noncoherent ones do not make any attempt to correlate the received signal with local coherent replicas, and in the simplest case only detect the presence of pulse energy in the allocated bandwidth [15], [16]. Noncoherent “energy detection” receivers are less complex and thus more suited to a single-chip implementation. Complexity is traded with performance in this case, as about 3 dB of signal-to-noise ratio (SNR) are lost for a given bit-error rate (BER) compared to coherent solutions. Recently published results show viable SoC solutions for energy detection receivers [10], [12]. Therefore, in the following sections we will describe and simulate the architecture of an UWB transceiver whose RX part is based on the energy collection criterion.

### B. VHDL-AMS

The complexity of telecommunications SoCs urges toward efficient means for the co-design of digital, analog and mixed-signal blocks. System-level constraints must be propagated in the design flow down to the lower levels in order to trim circuit parameters. Conversely, the impact on system behavior of transistor-level nonidealities, and the consequent limitations and constraints, must be accounted for since the very first stages of high-level design. Traditional design and simulation methods deceive these objectives, as, on the one hand, coarse system-level descriptions fail in assuring the accuracy needed for the design of analog and mixed-signal circuits, on the other hand, using transistor-level simulations for the evaluation of performance of an entire system is impractical because of the unacceptably long simulation time.

The VHDL-AMS, a superset of VHDL, has been conceived for modeling not only analog and mixed-signal circuits but also mixed-technology systems [17]–[19]. It supports the use of digital constructs together with electrical quantities, differential equations and algebraic constraints. In addition, it allows the hardware description with different levels of abstraction, then making viable a top-down design methodology in which a preliminary behavioral description of the components allows a coarse functionality test of the system, while a progressive refinement defines the real circuit performance. Such a flexibility allows the designer to understand the tradeoff between accuracy and CPU time, to translate the system constraints on the specific circuit-level parameters, and to evaluate the impact of circuit nonidealities reflected on the system behavior.

A few works document the use of VHDL-AMS as an effective tool for the efficient design of complex systems using a top-down methodology. Basic functionality tests using a behavioral VHDL-AMS description were used in [20] showing the feasibility of a full transceiver circuit simulation in which a realistic communication channel is emulated. In [21], RF blocks

of a differential quadrature phase-shift keying (DQPSK) transceiver and a channel model were implemented adding white Gaussian noise in a behavioral VHDL-AMS description and achieving BER results very close to theoretical models. Other works demonstrate the possibility to annotate transistor-level simulation results or silicon prototype measurements back to the behavioral VHDL-AMS circuit model. The aim is to shorten system-level simulations during the design verification phase. In particular a methodology for the design of RF circuits in VHDL-AMS starting from flexible specifications and assuring an accurate description of noise and nonlinearity effects was proposed in [22]: Simulation results of behavioral model and transistor-level circuit are compared and show acceptable accordance. In [23] the real behavior of a PLL was modeled using VHDL-AMS adding jitter: The phase noise simulated spectrum was in good agreement with measured results. A similar approach was followed in [24] for the modelization of a WCDMA transceiver: The behavioral model reduces simulation time while including accurate parameters measured on a prototype. A top-down design methodology, validated by measurements, was proposed in [25] for the design of a delta-sigma modulator. After a coarse description of components, various nonidealities like jitter, thermal and  $1/f$  noise, and capacitor mismatch were added to the models. From system-level simulations, thus, the specifications for the modulator design are derived and measurements confirmed both model accuracy and methodology effectiveness. In [26], finally, a Bluetooth transceiver was first modeled using a simple behavioral description. Then, thanks to the higher level specifications, few blocks were detailed down to the transistor-level. The behavioral model was then refined so as to match the transistor-level model; the BER was estimated to verify the effectiveness of the multiresolution description.

In this work, as discussed in details in Section IV, we use the top-down methodology and show the interactions between system and circuit levels, that is, how the former allows to enucleate circuit design specifications, and how the latter characteristics impact system performance.

### III. UWB TRANSCEIVER ARCHITECTURE

As previously stated, the objective of the present work is to provide a simulation environment for an UWB integrated transceiver, whose RX part is based on energy detection. In the following we summarize the receiver's operation on a 2-PPM modulated train of pulses which, also in accordance with the IEEE standardization task group, seems the most appropriate for low data-rate WPAN localization applications [6].

The received signal  $r(t)$  is

$$r(t) = \sum_{j=-\infty}^{+\infty} s\left(t - jT_s - \frac{a_j T_s}{2} - \tau\right) + n(t) \quad (1)$$

where  $s(t)$  is the channel response to an isolated UWB pulse emitted from the transmitter. The response shape is totally unknown to the receiver, except for its duration that is limited within the channel maximum delay spread which, for indoor UWB channels, is on the order of 100 ns. The  $a_j$  terms are statistically independent binary (0,1) data with identical probability distribution. The repetition period of  $a_j$  data is  $T_s$ , also

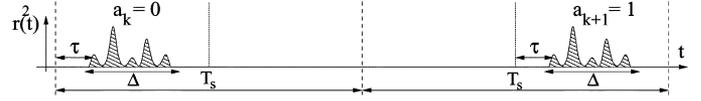


Fig. 1. Definition of symbols for 2-PPM modulation.

called symbol interval. Finally,  $n(t)$  represents an additive white Gaussian noise (AWGN) with two-sided noise spectral density  $N_0/2$ . If the  $a_j$  were all zero, the signal component in (1) would be the repetition of  $s(t)$  at the instants  $jT_s + \tau$ , where  $\tau$  represents the time offset between the transmit and receive clocks. So, a pulse would always appear at the beginning of a symbol interval. Vice-versa, as the  $a_j$  are either 0 or 1, the pulse will start either at the beginning or at the midpoint of the interval, depending on  $a_j$ .

In order to decide whether a 0 or a 1 was transmitted, the receiver computes two *energies* by integrating the squared value of  $r(t)$  in the first and the second half of  $T_s$ , respectively. For the  $k$ th transmitted symbols, the two integrals are

$$z_k^{(0)} = \int_{kT_s + \tau}^{kT_s + \tau + \Delta} r^2(t) dt \quad (2)$$

$$z_k^{(1)} = \int_{(k+1/2)T_s + \tau}^{(k+1/2)T_s + \tau + \Delta} r^2(t) dt \quad (3)$$

where  $\Delta$  is the integration time whose value is on the order of the channel spread. Fig. 1 explains the meaning of all symbols. The decision about the value of the  $k$ th data is taken by comparing the two energies. Formally stated, the receiver computes an estimate  $\hat{a}_k$  according to the rule

$$\hat{a}_k = \begin{cases} 0, & \text{if } z_k^{(0)} > z_k^{(1)} \\ 1, & \text{otherwise.} \end{cases} \quad (4)$$

We refer the interested reader to [27] for a more formal and detailed analysis of a PPM energy detection receiver.

The previously described decision rule requires clock synchronization, that is the receiver should know the exact value of  $\tau$ . Before proceeding with demodulation, the receiver acquires such timing by means of periodically repeated training sequences [16] or in a blind fashion [28]. We stick to the first paradigm as suggested by the draft standard of IEEE 802.15.4a which indicates a specific preamble to be transmitted before data with the sake of aiding the synchronization process [6]. The timing acquisition typically consists of two phases: A preliminary “coarse” synchronization whose aim is to acquire timing with an accuracy level considered sufficient for the subsequent data demodulation, and a “fine” synchronization which refines the accuracy to the level required for ranging estimation. The precision in estimating  $\tau$  is critical for ranging, as the quality of such estimation affects the evaluation of distance between a pair of transceivers.

We developed a simple but effective algorithm which coarsely estimates timing during the reception of a short nonmodulated preamble, and which refines timing when subsequent modulated data are being received. The first process evaluates the fraction of the signal energy contained in  $M$  time windows of duration  $T_w$  smaller than the symbol timing  $T_s$  and selects the one which contains maximum energy. Windows are separated in time of  $t_c$

and partially overlapped ( $t_c < T_w$ ). In formulas, the estimated coarse  $\tau$  is

$$\hat{\tau}_c = m t_c \Big|_{m \in 0:M-1} \max \int_{m t_c + m T_s}^{m t_c + m T_s + T_w} r^2(t) dt. \quad (5)$$

The integration is calculated only once  $\forall m \in [0, M - 1]$ , that is once per pulse. Since the step time  $t_c$  is smaller than the integration time, this is the only way to employ a **single integrator** which works over  $M$  pulses. Another possibility could be that of using a **bank of  $M$  integrators** working in parallel and calculating  $M$  energy values per pulse as proposed by Stoica *et al.* in [10]. Using  $M$  integrators in parallel would allow accumulating and averaging more energy, over the same period of time, than a single integrator, thus leading to a significant accuracy improvement. As a balance between higher accuracy and lower hardware cost, we privileged the second to the detriment of the first.

The second process for the fine estimation is similar. A linear search around  $\hat{\tau}_c$  with a search step finer than  $t_c$  is executed. The goal is again finding the maximum energy in a window of length  $T_w$ . A difference is that received pulses are modulated because this finer process takes place after preamble reception. Since pulses lay in the first or in the second half of  $T_s$ , two corresponding integrals are calculated and added. In formulas, the estimated fine  $\tau$  is

$$\hat{\tau}_f = n t_f \Big|_{n \in 0:N-1} \max \left\{ \int_{\hat{\tau}_c - (n-N/2)t_f + nT_s}^{\hat{\tau}_c - (n-N/2)t_f + nT_s + T_w} r^2(t) dt + \int_{T_s/2 + \hat{\tau}_c - (n-N/2)t_f + nT_s}^{T_s/2 + \hat{\tau}_c - (n-N/2)t_f + nT_s + T_w} r^2(t) dt \right\} \quad (6)$$

where  $t_f < t_c$  is the finer search step. Once again,  $N$  pulses are needed to complete the entire fine synchronization process because only one integrator is employed. Since the pulse is located in one out of two PPM locations and it is not possible to know *a priori* which one will be, one of the two integrals will inevitably integrate only noise. In case the preamble is longer, at least  $M + N$  symbols, the two processes may both take place over the nonmodulated preamble. This favorable condition would allow a better accuracy because only one of the two integrals, the one computed over signal energy and not just noise, will be calculated. Numerical results reported in Section VI-C, which refer to ranging simulations obtained with both long and short preambles, confirm this guess.

Parameters  $M$  and  $N$  depend on the level of accuracy needed and on  $T_w$  and  $T_s$  values. In all our simulations we set  $T_s = 200$  ns, value which avoids intersymbol interference (ISI), the channel spread being around 100 ns, and  $T_w = 30$  ns, which is a good compromise: Increasing  $T_w$  over this value will collect more signal noise than additional signal energy. Although the optimum value depends on the channel characteristics, it is worth adding that variations on the order of 5–10 ns have marginal effects on the receiver performance [27]. We thus selected a value that demonstrated being appropriate for the majority of the channel models we employed. It has to be remarked also that this value is a parameter of the simulation environment that can be suitably changed before execution.

Synchronization and demodulation algorithms work under the assumption that a valid signal is being received. Therefore, a very first phase takes place before synchronization which consists in sampling the channel energy from time to time in order to evaluate whether a preamble is being transmitted. This phase is split into two subphases, *noise estimation* (NE) and *preamble sense* (PS). The first estimates the AWGN noise energy while the second checks if the received energy exceeds a threshold established by the former NE subphase, condition which corresponds to a preamble detection. In formulas, the noise energy is calculated as an average  $E_n$  of  $K_n$  energy measurements over the symbol timing as such

$$E_n = \frac{1}{K_n} \sum_{k=0}^{K_n-1} \int_{2kT_s}^{(2k+1)T_s} n^2(t) dt. \quad (7)$$

Since only one integrate-and-dump (I&D) unit is used, measurements are taken in only one  $T_s$  interval over two consecutive ones, as defined by the  $2k$  shift in the integration limits. Once  $E_n$  has been set (NE), the same operation is performed over the signal, calculating  $K_s$  different energy measurements (PS). Preamble is considered detected if there are at least  $M_s$  measurements over  $E_n$ .  $M_s$  lays between 1 and  $K_s$ . In Sections V and VI we reported only the results obtained with  $M_s = 1$  and  $K_s = K_n = 3$  for sake of brevity. Augmenting  $K_n$ ,  $K_s$  and  $M_s$  on the one hand reduces the so-called probability of *false alarm*, that is the probability of incorrectly detecting a preamble while noise is being received, on the other hand reduces the time available for coarse synchronization as up to  $2K_s$  preamble symbols may have been lost in NE/PS.

We want to remark here that these algorithms in no way should be taken as optimal and others may be more effective. The very reason why we chose them was barely their simplicity and easiness of implementation. It is worth adding that the simulation environment can be promptly modified to incorporate new algorithms.<sup>1</sup>

The equations of the energy collection receiver are mapped onto some of the blocks of the architecture outlined in Fig. 2. The energy integrals of previous equations are computed by the *squarer* (i.e.,  $()^2$  in figure) and I&D units activated by the *synchronizer* block which provides the correct timing signals. The latter block basically consists of a set of clock phases of minimum distance  $t_f$ . The correct phase fed by the Synchronizer to the I&D is defined by the *demodulation and data processing* block which in turn receives its input from the I&D through the *analog-to-digital converter* (ADC). The ADC block is not critical from the timing perspective, because the pulse repetition rate is on the order of  $1/T_s = 5$  MHz, nor from the accuracy point of view, because only 5 bits suffice as shown later on in Section V.

The receiver branch of the transceiver consists also of a low-noise amplifier (LNA) and a variable gain amplifier (VGA) preceding the squarer. The VGA adapts the signal to the input range of the ADC. Its gain, controlled in steps using a digital-to-analog converter within the block called *automatic gain control* (AGC),

<sup>1</sup>Presently we are working on an ameliorated version of NE/PS assuming a different preamble compatible with the already mentioned 802.15.4a. Preliminary results not discussed in this paper showed better figures.

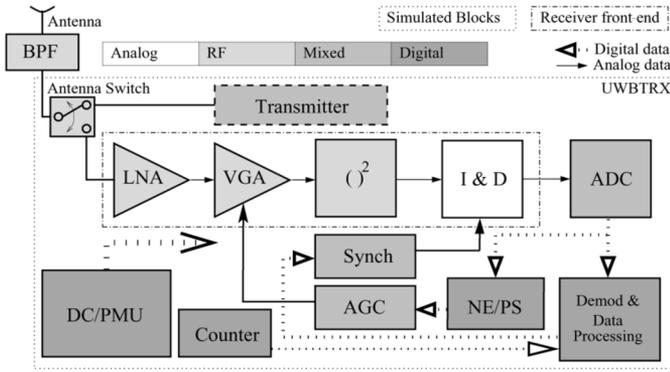


Fig. 2. Architecture of the UWB transceiver.

is calculated using the energy measurements of the NE/PS unit and with a pre-computed look-up table (LUT).

The transmitter contains a pulse generator and a modulator which formats transmitted data according to a packet structure made of a nonmodulated sequence of pulses, i.e., the preamble, followed by the modulated payload. Transmitter and receiver share the same antenna through a switch.

The *counter* block is used for the ranging operation which will be discussed later in Section VI. The *digital controller and power management* units (DC/PMU) are in charge of scheduling the various operation phases and of shutting down unused blocks in order to save power.

Two blocks in Fig. 2 have not been used in our simulations: the external *bandpass filter* (BPF) and the *UWB antenna*. Their effect is already taken into account in the channel model employed which is of public domain [29].

Fig. 2 highlights by a gray shading coloring the digital, analog, mixed-signal and radio-frequency blocks. The possibility to co-simulate all blocks within a single environment is of paramount importance when the SoC designer needs to evaluate their reciprocal impact. Therefore, using VHDL-AMS as a common hardware description language for the whole UWB transceiver is the key for the development of a successful and efficient simulation tool.

#### IV. MULTIREOLUTION VHDL-AMS ENVIRONMENT

The flexible characteristics of VHDL-AMS described in Section II make this language optimally suited for the creation of a single simulation environment. As previously stated, commercial tools like ADMS by Mentor Graphics allow to co-simulate not only VHDL and VHDL-AMS constructs together but also Spice-like netlists within a unique environment, opening the way to a design methodology based on a top-down approach. The possibility to hierarchically describe an architecture allowed us to organize this methodology in three steps as graphically schematized in Fig. 3.

1) *Phase I*: In this phase, described in details in [30], the UWB transceiver was behaviorally modeled, the system-level functionality was tested, and the coherence with another high-level description language (Matlab) was checked. In particular, both the Matlab and the VHDL-AMS versions were designed so that the BER could be evaluated varying the SNR at the receiver input. In order to ensure coherence of the simulations, both BER curves were compared with an analytic model of the energy detection receiver [27]. Perfect timing acquisition

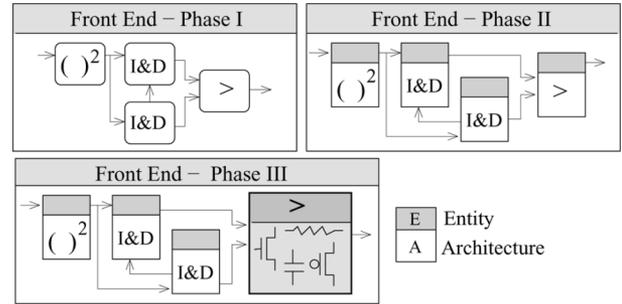


Fig. 3. Design flow phases using VHDL-AMS and ADMS.

was supposed for this analysis (ideal synchronizer). A simple VHDL-AMS code which implements the fundamental receiver front-end blocks used for BER testing purposes is shown in the following:

```
SQUARING_UNIT : input_squared == input ** 2;
INT_AND_DUMP0 :
if integrate = '0' use input_squared == int_0'Dot;
else int_0 == 0.0; end use;
INT_AND_DUMP1 :
if integrate = '1' use input_squared == int_1'Dot;
else int_1 == 0.0; end use;
SAMPLE_AND_HOLD :
process(integrate)
begin
if integrate'event and integrate = '0'then
int_held_1 <= int_1;
else if integrate'event and integrate = '1'then
int_held_0 <= int_0; end if;
end process;
COMPARATOR:
process(int_held0,int_held1)
begin
if int_held1 > int_held0 then digital_output <= '0';
else digital_output <= '1'; end if;
end process;
```

At this stage, the level of abstraction is similar to Matlab. The analog *input*, modeled as a *quantity*, is squared and *input\_squared* is obtained. Then, the integration is performed using two behavioral I&D processes (INTEGRATE\_AND\_DUMP0 and INTEGRATE\_AND\_DUMP1) for "0" and "1" phases respectively: The control signal *integrate* forces integration and dumping. Then, after being converted from *quantities* to *signals* within the SAMPLE\_AND\_HOLD process, the two energies are compared using another appropriate process labeled as COMPARATOR. Fig. 3, *Phase I*, shows the concept which this level of abstraction relies on. Here, the complete receiver front-end has a unique entity and the architecture includes all the behavioral equations used for modeling the analog and RF units.

The experiments made at this level resulted in perfectly overlapped BER curves, thus not showing any loss in accuracy in the use of VHDL-AMS with respect to Matlab [30].

Although we focused on a specific receiver, the characteristics of *Phase I*, namely extreme rapidity of blocks behavior description and system simulation, might be well suited to a fast architectural exploration of other impulse-radio alternatives. For instance, coherent receivers, like the ones described in [13] and [14], that perform coherent correlations in the digital domain could be easily simulated: The squaring and I&D units could be removed, the ADC connected directly to the VGA, and the demodulation and data processing block modified with the proper equations. One can also decide to change the modulation or the format of preamble and payload simply by modifying the transmitter parameters. The channel model can be also varied (multipath, fading, path loss, bandwidth for 802.15.4a and also for 802.15.3a [2]).

2) *Phase II*: After verification of consistency with Matlab, we built the entire architecture in VHDL-AMS, including synchronization, PS and AD conversion, with the sufficient details for a complete simulation. However, internally, every block was still described using abstract VHDL-AMS statements that include only a subset of all possible nonidealities, the goal being to demonstrate the functionality of the entire transceiver rather than pinpointing specific effects. In particular, we modeled those effects that have a relevant impact on the system-level performance, like quantization of the ADC and of the DAC controlling the AGC, various offsets, as well as the nonlinear effect of the possible saturation in all stages of the receiver branch. The coupled effects of quantization and noise, interference and offset on the bit-error rate performance have been studied, as shown later on in Section V.

In this *architectural phase* of our work the accuracy is high enough to make relevant system-level choices like, for instance, the ADC quantization bits. Reducing the abstraction would be helpful for the circuit-level design but would severely hamper upper level simulations by dramatically increasing the CPU time. In order to extend the *Phase I* testbed to this full receiver description, a functional partition based on the identification of the main building blocks is needed. This can be obtained by creating appropriate entities and architectures from the processes given in the previous listing. The graphical representation of Fig. 3, *Phase II*, shows this partition, in which each receiver block is modeled with a proper component with its own *entity* and *architecture*. A practical example is reported in the following code together with the description of an ADC converter:

```
architecture behavior of SQUARING_UNIT is
quantity vin across input to electrical_ref;
quantity vout across i through output to electrical_ref;
begin
vout == vin * *2;
end behavior;

architecture behavior of INT_AND_DUMP is
quantity vin across input to electrical_ref;
quantity vout across i through output to electrical_ref;
begin
ID : if control = '1' use vin == vout'Dot;
else vout == 0.0; end use;
```

```
end behavior;

architecture behavior of SAMPLE_AND_HOLD is
quantity vin across iin through input to electrical_ref;
quantity vout across i through output to electrical_ref;
begin
SH : if sample = '1' use vout == vin; iout == iin;
else vout'Dot == 0.0; iout == 0.0; end use;
end behavior;

architecture behavior of ADC is
quantity vin across input to electrical_ref;
quantity vref across vrefp to electrical_ref;
begin
process(convert, reset)
variable count : integer; variable space : REAL;
begin
if (reset = '1') then count := 0; D_OUT <= (others ==> '0');
elsif convert'event and convert = '0' then count := 0;
for i in 1 to ADC_2nbit - 1 loop
if (vin >= ADC_quant(i)*vref) then count := count + 1;
end if;
end loop conversion;
D_OUT <= conv_std_logic_vector(count, ADC_nbit);
else count := 0;
end if;
end process;
end behavior;
```

In this example, every “architecture” implements one of the front-end building blocks in Fig. 2 (entities, terminals, and quantities declarations have been omitted). At a higher hierarchy level, the connection among the blocks is done instantiating them as components together with the digital back-end architecture, resulting in the creation of the full simulation environment. At this stage, the description can be used, for e.g., extending *Phase I* testbeds to test quantization effects on BER varying the bit number of the ADC (see Section V for further details). Concerning this point, in the above example, number of quantization bits `ADC_nbit` and quantization steps vector `ADC_quant` are parametric and can be easily adapted according to high-level constraints.

The results of the system-level simulations reported in Sections V–VI have been obtained at this stage of the work.

3) *Phase III*: This last phase faces circuit-level design, once the final architectural details have been decided through extensive simulations in *Phase II*. This part of the work requires VHDL-AMS descriptions as close to hardware as possible. The language semantics is rich enough for this purpose but the CPU cost is relevant, especially if the entire architecture has to be simulated with the aim of capturing the low-level effects impingement on the system-level. In addition, modeling in VHDL-AMS transistor-level analog and mixed-signal circuits might be burdensome.

The simulation cost can be then tackled by a suitable *substitute-and-play* approach. In practice, the majority of the blocks are still described according to the *Phase II* model, while some of them are substituted with a circuit-level representation. The modularity of the simulation environment allows the replacement of single blocks without any modification, provided that input/output terminals are electrically compatible. Thanks to the flexibility offered by ADMS, one can directly import its design at transistor-level into the global simulation environment without having to tailor a model on it. By intelligently applying the *substitute-and-play* approach outlined above, the same environment as of *Phase II* can be recycled, permitting the use of the same *entity*, i.e., the I/O interface, and different *architectures*, i.e., internal descriptions. As a result, the behavioral architecture of a single block as of *Phase II* can be substituted with its spice-like transistor-level counterpart (schematic or extracted from a layout view) in *Phase III*, leaving the upper level description unaffected (see Fig. 3). After the replacement one can still use system-level testbenches and evaluate the impact of a single block on the entire system, by comparing *Phase II* and *Phase III* results.

The impact of a single block on the entire system is better caught if the substitution is done by replacing one unit at a time. Replacing many blocks together makes difficult the process of discerning the cause-effect relationship between unit-level issues and system-level behavior, leaving aside the increase of simulation time. Depending on the circuit complexity, the replacement of at most two blocks may be possible. In any case, the choice is left to the experience of the designer. Whether it is required to simulate together many blocks it is possible to “backannotate” the already substituted transistor-level models to VHDL-AMS to save CPU time. This operation is achieved by extracting the most relevant electrical parameters of the unit for calibrating the corresponding *Phase II* AMS model. However, whether the system level effects caused by *Phase III* substitution are negligible it is recommended to consider only the *Phase II* models for evident reasons.

Since the main focus of this work is on the methodology applied to the simulation environment, rather than on pure design, we will limit the description of *Phase III* results to one fundamental block, the LNA. In particular, Sections V–VI discuss functional simulations as of *Phase II*, while Section VII revisits the previous results under the *Phase III* perspective.

## V. FUNCTIONAL SIMULATIONS

Accurate simulations of the UWB transceiver and the consequent evaluation of performance require the use of a realistic channel model. The IEEE 802.15.4a Group released a channel impulse response model and the code for use in Matlab [6]. It consists of a modified Saleh–Valenzuela multipath model with the addition of a frequency dependent path loss. It was then crucial for our work to incorporate such model in VHDL-AMS. The simplest way was to precompute a relevant number of channel realizations with Matlab, oversample them with a sampling period on the order of the VHDL-AMS simulation step, and save them in a suitable data-base. When the simulation is run, the pulse generation consists then in a digital trigger which activates a file reading procedure and assigns the read pulse samples to a VHDL-AMS quantity. This signal is the output

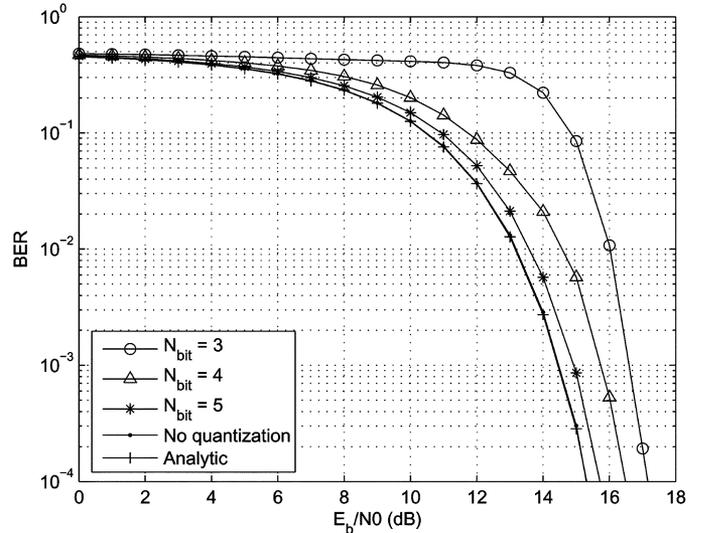


Fig. 4. BER curves varying ADC quantization bits.

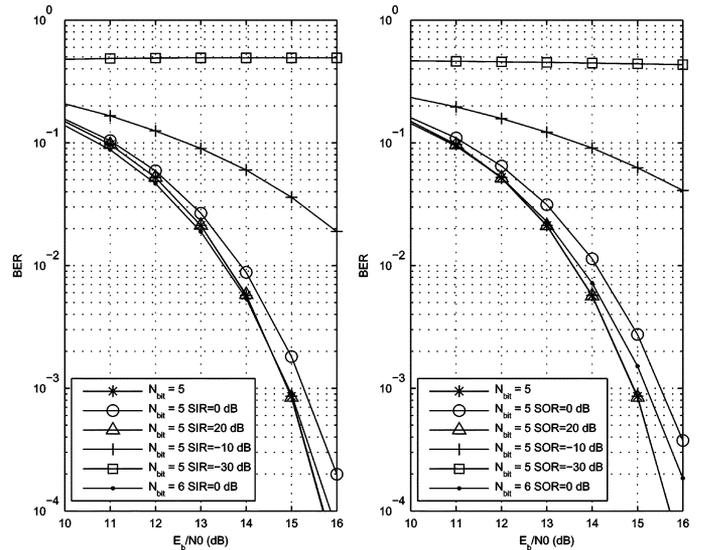


Fig. 5. BER curves varying ADC quantization bits, SIR and SOR.

of the transmitter. The pulse is normalized so as to have power spectral characteristics compatible with the FCC limitations [1], that is  $-41.3$  dBm/MHz in the bandwidth 3.1–10.6 GHz. We actually reduced the UWB bandwidth between 3.1 and 5 GHz in our simulations according to [6] which splits the entire bandwidth in a lower band, the one we are using, and an upper band. The resulting signal is then attenuated to take the path loss into account. Additional losses derived from a proper link calculation or different bandwidth of operation may be accounted for in simulations: Our environment is provided with several parameters defined in a VHDL-AMS package for this purpose. Finally, AWGN, which accounts for thermal noise ( $KT$ ) and front-end noise figure ( $NF$ ), as well as a number of narrowband interferers can be added to the received signal. Interference from other UWB impulse-radio users are not considered, assuming a time-division multiple access (TDMA). The UWB antenna model is ideal and does not introduce gain nor phase shifts. However, the modular

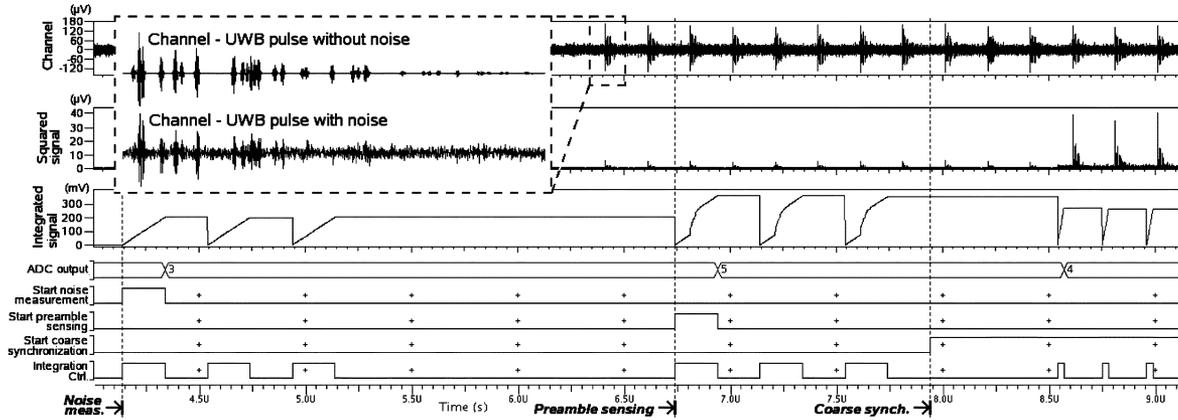


Fig. 6. NE and PS waveform.

architecture and the VHDL-AMS features allow its impromptu substitution with a more realistic model without substantial impact on the simulation chain.

The inset in Fig. 6 illustrates an example of received UWB pulse with and without added AWGN noise. The echoes of the first UWB pulse due to the multipath channel are evident.

The received signal passes through the switch, LNA and VGA and I&D units. Then it is converted to a digital format by an ADC parametrized in terms of quantization bits. The number of bits used is a critical parameter. Our environment described in Section IV can be used to define a good compromise between complexity and accuracy. The curves in Fig. 4 represent the BER, that is the fraction of correctly demodulated bits over the received ones, obtained varying the SNR. Interferers were not considered in these simulations. One of the curves has been obtained without quantization, that is with the ensemble of receiver operations described in Section III executed with floating-point precision. This ideal result has been obtained with the VHDL-AMS behavioral description developed in *Phase I*, according to the terminology of Section IV. The curve labeled as “analytic” was obtained using a closed-form equation reported by Carbonelli *et al.* [27]: The perfect overlap of the curve without quantization with the analytic one confirms the coherence of the simulator. The other curves, labeled with  $N_{bit} = 3 \div 5$ , have been obtained with the simulation environment as of *Phase II* and with the hypothesis of perfect synchronization between transmitter and receiver. The comparison between such an ideal case and a practical one of imperfect synchronization, faced in [27], shows that the loss is contained within 1 dB of SNR for a given BER. From the simulations it is clear that the 5 bits curve is close enough to the ideal curve without quantization. However, the effective resolution depends on other effects, like the presence of narrowband interferers and of various offsets [31], [32]. Following the approach depicted in [13], we considered only the WLAN’s interference at 2.4 and 5 GHz. As for the offsets, we considered only the most important one at the squarer input in Fig. 2. Fig. 5 reports simulated BER curves obtained varying SNR as well as the signal-to-interference ratio (SIR) and the signal-to-offset ratio (SOR). The latter is set calculating the amount of offset that produces an energy value, i.e., at the integrator’s output in Fig. 2, which is  $10^{SOR/10}$  times lower than the signal energy. In order to decouple the various effects, we supposed the integrator’s

output perfectly matched to the ADC input dynamics (ideal automatic gain control). For reference, the 5 bits curve of Fig. 4 was plotted again. Results with narrowband interference and offsets show that 5 bits are still sufficient for SIR or SOR equal to or higher than 20 dB. SIR of 0 dB are tolerable with 6 bits, while the same value of SOR leads to a loss of 0.5 dB of SNR at BER around  $10^{-4}$ . Overall, the receiver stops working properly at SIR or SOR lower than  $-10$  dB, that is 10% of the ADC input range, thus calling for both a careful design to limit offset and to filter out interference in order not to trespass such bound. For all the following experiments aimed at the architectural validation we used a 5-bits ADC, assuming the nonideal effects are kept properly under control.

Once the ADC precision is chosen, in order not to lose accuracy and to fully exploit the entire dynamic range, it is necessary to adapt the output of the I&D unit to the input range of the ADC. This nontrivial task is accomplished by the AGC unit which sets the correct gain during the preamble sensing phase, as described in Section III.

In Fig. 6 the *NE* and *PS* simulation results are shown. In the “Channel” waveform, only noise is present until the signal preamble is received at  $6.5 \mu\text{s}$ . The VGA initially amplifies the LNA output with a default gain and feeds the squarer input. Its output (“Squared signal” in figure), is integrated by the I&D unit (“Integrated signal”). The “ADC output” allows the *NE/PS* and the DC to digitally elaborate the data and to generate the control signals reported at the bottom of the figure. In particular, the “Start Noise Measurement” pulse enables the energy measurement described in (7): The “Integration Ctrl” signal activates iteratively ( $K_n = 3$ ) the I&D for a  $T_s = 0.2 \mu\text{s}$  integration window starting at  $4.1 \mu\text{s}$ .

The “Start Preamble Sensing” signal, asserted in this case at  $6.7 \mu\text{s}$  in Fig. 6, works as a start strobe for the *PS* phase, an operation similar to previous energy evaluation. The preamble is considered present if  $M_s$  energy measurements are above the average noise energy  $E_n$ . In this example,  $M_s = 1$ . As a consequence of this decision the coarse synchronization phase begins, strobed by the “Start Coarse Synchronization” signal at  $7.9 \mu\text{s}$  in figure. In the meanwhile, the difference between the average energy measured in the *NE* phase and the maximum energy detected during the *PS* is used by the AGC to set the optimal “VGA gain,” so that the I&D output matches the ADC input dynamics. The simplified code describing this AGC action is in the

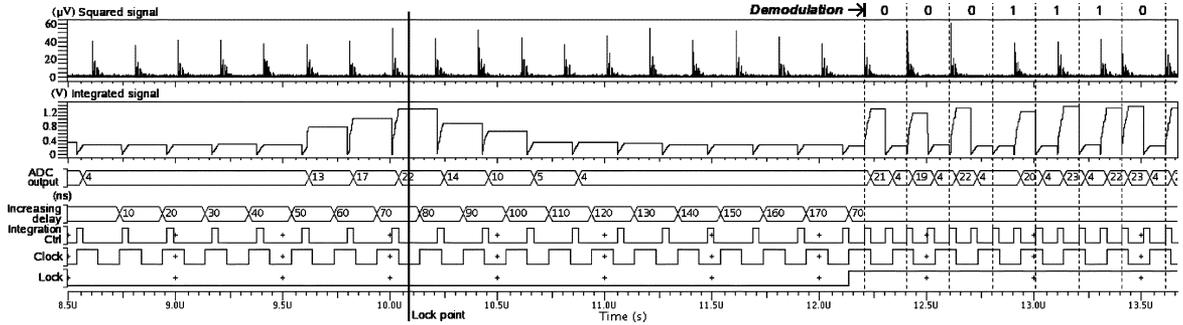


Fig. 7. Coarse synchronization and demodulation waveforms.

following:

```

type datagains is array (0 to Nbit - 1) of std_logic_vector(Nbit -
1 downto 0);

constant DATAGAIN : datagains :=
("00000", "11111", "10111", "...", "00110");

process(clk, reset)

variable gain_ctrl : std_logic_vector(Nbit - 1 downto 0);

begin

if reset = '1' then gain_ctrl := (others => '0');

elsif clk'event and clk = '1' then

gain_ctrl := DATAGAIN(energy_difference); end if;

gain_control <= gain_ctrl;

end process;

```

DATAGAIN is a LUT indexed by a value proportional to the energy difference. The LUT output `gain_ctrl`, converted by the DAC, sets the VGA gain. In Fig. 6, the new gain is set at  $8.5 \mu\text{s}$ . From this time on, the corresponding VGA output and squared signal peak-to-peak values are larger, meaning that the new gain is higher than the default one.

In Fig. 7 the *coarse synchronization* simulation results are detailed. The timing window is partially superposed to the NE/PS one in Fig. 6. This new phase begins at  $8.5 \mu\text{s}$  with the first pulse of the “Integration Ctrl” signal which is initially synchronous to the “Clock” and whose high level corresponds to an integration window of  $T_w = 30 \text{ ns}$ . The output of the I&D, “Integrated signal,” is held until the next clock edge, thus allowing the ADC to convert this value into a digital word of 5 bits. The next integration starts from the clock edge with an additional delay of  $t_c = 10 \text{ ns}$  with respect to the previous one, so that the energy is captured in a window adjacent to the preceding one and partially overlapped (20 ns is the overlap time). This task is performed iteratively until the whole energy contained in a pulse repetition period  $T_s$  has been analyzed slice by slice. The maximum among the measured energies is detected in this case approximately at  $10 \mu\text{s}$ , and corresponds to an integration window shift of 70 ns with respect to the clock rising edge (see the “Increasing delay” waveform). Once the synchronization timing is achieved, a level-triggered “Lock” signal is asserted at  $12.2 \mu\text{s}$ , event which gives rise to the *fine* synchronization and the *demodulation* phases. We did not report waveforms detailing the former, since it is not conceptually different from the coarse synchronization depicted above, and focused on the *demodulation* process, corresponding to (2)–(4). The “Integration Ctrl” signal in Fig. 7, shifted of the right amount of delay with respect to

the clock edge by the synchronization phase, is enabled twice within the pulse repetition period: The “Integrated Signal” is now used to decide whether the UWB pulse was sent in the first half of the period (so that a ‘0’ is detected), or in the second one (that is, the symbol was a ‘1’). Note, for example, the difference between the two symbols sent in the  $13.2\text{--}13.4 \mu\text{s}$  slot and in the  $13.4\text{--}13.6 \mu\text{s}$  one. According to (4), the former is a ‘1’ as  $z_k^{(0)} < z_k^{(1)}$ , that is, after the ADC conversion,  $4 < 22$  as shown by the digital waveform “ADC Output” in figure; on the contrary the latter is a ‘1’ since  $z_{k+1}^{(0)} > z_{k+1}^{(1)}$ , that is  $23 > 4$ .

## VI. TWR SIMULATION RESULTS

The evaluation of the distance between a pair of wireless radios is called *ranging*. Since wireless signals travel at speed of light, ranging can be indirectly obtained from a measure of TOF [7]. The accuracy of this measurement increases with the bandwidth of the signals involved, since a large bandwidth in the frequency domain corresponds to a short duration in the time domain. An ideal pulse  $\delta(t)$  has infinite spectrum occupation. UWB signals are thus the foremost candidates for wireless applications where ranging is required.

Based on this reasoning, we present the results of our VHDL-AMS simulations of a ranging case in this section, organized in three parts: In the first one, the *Link Budget* we adopted in our simulations is given. In the second one, we concentrate on the description of the so called *TWR* operation. Finally, the last part reports numerical results and comments.

### A. Link Budget

Given a transmitted power and bit-rate  $R_b$ , a path loss model, a SNR  $E_b/N_0$  required at the receiver for a given BER and the noise figure of the receiver, the link budget allows to calculate the minimum power required at the receiver (sensitivity) and the maximum distance between transmitter and receiver which satisfies this constraint. It is common practice to use a path loss  $\propto 1/d^n = 1/d^2$ , though the IEEE 802.15.4a suggests a  $n$  coefficient which varies depending on the channel model [6]. However, for line-of-sight (LOS) channels,  $n < 2$  so that calculating the link budget with  $n = 2$  is a conservative approximation.<sup>2</sup> Table I reports the budget used in our simulations. The geometric center frequency  $f'_c$  is calculated from the characteristics of the employed pulse shape, whose spectrum occupation at  $-10 \text{ dB}$  is 3.1–5 GHz. The average TX power is obtained

<sup>2</sup>It has to be remarked that the distance measurement through TOF evaluation is meaningful only in LOS channels, in which the first echo of the transmitted signal arrives at the receiver prior to any other echoes created via reflections and without passing through obstacles.

TABLE I  
LINK BUDGET—MAXIMUM DISTANCE  $d_{max} = 28$  m

Parameter	Value
Peak payload bit rate ( $R_b$ )	5 Mbit/s
Average TX power ( $P_T$ )	-8.5 dBm
TX antenna gain ( $G_T$ )	0 dBi
Geometric center frequency ( $f_c$ )	3.94 GHz
Path loss @ 1 meter ( $L_1$ )	44.34 dB
Additional path loss ( $L_2$ )	28.94 dB
RX antenna gain ( $G_R$ )	0 dBi
RX power ( $P_R = P_T + G_T + G_R - L_1 - L_2$ )	-81.78 dBm
Average noise power per bit ( $P_N$ )	-107.01 dBm
RX noise figure ( $NF$ )	7 dB
Minimum $E_b/N_0$ for $BER = 10^{-3}$ ( $S$ )	16 dB
Implementation loss ( $I$ )	1 dB
Link margin ( $M = P_R - P_N - NF - S - I$ )	1.22 dB
Min. RX Sensitivity ( $PG + P_N + NF + S, PG = 0dB$ )	-84.01 dBm

calculating the average power associated to an UWB pulse in this band according to the FCC regulations for indoor systems [1], and the minimum  $E_b/N_0$  is such that the associated BER is  $10^{-3}$  for a bit rate of 5 Mbit/s. From the budget analysis, the maximum distance allowed is 28 m. We did not consider the effect of channel coding nor of processing gain ( $PG = 0$  dB in table) which might extend the operating distance.

### B. TWR Scheme

The TWR scheme is based on the bidirectional data exchange among two devices (from here on  $A$  and  $B$ ) and aims at determining the TOF. Informally, TWR can be explained as follows:  $A$  sends a *Request* packet to  $B$ , which in turn sends a *Reply* packet containing a preliminary ranging information  $B$  has calculated. Finally, the distance is calculated by system  $A$  using the information contained in the replied packet and some additional computation. In the argumentation that follows a formal interpretation of this ranging scheme is given. We assumed a packet similar to the one used by IEEE 802.15.4 devices, whose structure is outlined in Fig. 8. We considered two possible preamble lengths. In the **Short preamble** case, coarse and fine synchronizations are just as in Section III where it was shown that the finer lock-point is searched during data demodulation. In the **Long preamble** case, the preamble sequence is long enough to allow a single synchronization phase. In practice, the same algorithm described before for coarse synchronization is used with a finer accuracy, precisely the same used in fine synchronization,  $t_c = t_f = 1$  ns. In both length cases, the preamble sequence of unmodulated symbols is used by the receiver to achieve symbol synchronization; the following start-of-frame delimiter (SFD) is used for frame synchronization; a frame length (FL) field indicates the data length; the Payload contains the transmitted information. From the ranging perspective, Request packets have the only purpose of allowing  $B$  to synchronize and so to define a proper pair  $\hat{\tau}_{c|B}$  and  $\hat{\tau}_{f|B}$ . In Reply packets, both coarse and fine synchronization indexes are transmitted within the payload (labeled in Fig. 8 as CI and FI, respectively). In the short preamble case  $\hat{\tau}_{c|B}$  index represents tenth of nanoseconds while  $\hat{\tau}_{f|B}$  are nanoseconds; for the long preamble case,  $\hat{\tau}_{c|B}$  index is in nanoseconds and  $\hat{\tau}_{f|B}$  field is not used.

As clear from the discussion above, TWR is merely an application of synchronization. Therefore, no more hardware than

Request

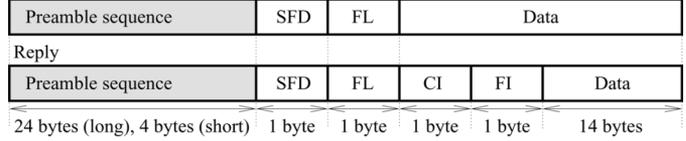


Fig. 8. Packet structure: Preamble length can be 24 bytes (long case) or 4 bytes (short case).

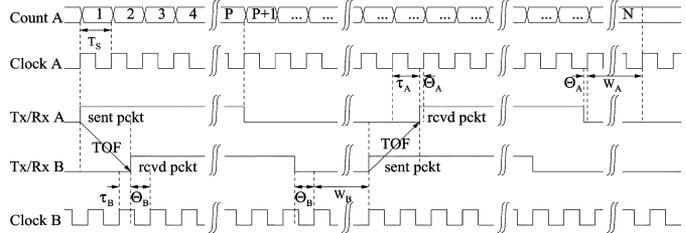


Fig. 9. TWR scheme.

what was previously described in Sections III and V is necessary for this operation.

Let us now describe the sequence of operations necessary for the TWR scheme with the help of Fig. 9.  $A$  sends a request packet to  $B$  and starts running an internal counter (“Count  $A$ ” in figure) synchronous with the symbol transmission rate  $1/T_s = 5$  MHz. After a TOF, the packet is received by  $B$ . The distance between the leading edge of the packet sent by  $A$  and the first positive edge of  $B$ 's clock is  $\Theta_B$ , while the opposite is  $\tau_B$  which corresponds to the  $\tau$  definition of Fig. 1. The latter corresponds to the sum of coarse and fine synchronization values computed by  $B$  over the received packet, i.e.,  $\tau_B = \hat{\tau}_{c|B} + \hat{\tau}_{f|B}$ , while the former is given by  $\Theta_B = T_s - \tau_B$ . After complete synchronization and packet reception, and after a fixed waiting time  $W_B$ ,  $B$  sends a reply packet whose payload contains the  $\tau_B$  information. After TOF,  $A$  receives the packet. The same previous definitions for  $B$  hold for the  $A$  case, i.e.,  $\tau_A = \hat{\tau}_{c|A} + \hat{\tau}_{f|A}$  and  $\Theta_A = T_s - \tau_A$ . After complete reception and waiting time  $W_A$  the counter is halted. The elapsed time is  $N$  clock cycles, or  $NT_s$  s.

By simple inspection of Fig. 9, we can write

$$NT_s = T_{OF} + PT_s + \Theta_B + W_B + T_{OF} + PT_s + \Theta_A + W_A \quad (8)$$

where  $P$  is the packet length in clock cycles. In case the distance between the two transceivers is zero,  $T_{OF} = 0$ , the sum of displacements  $\Theta_A + \Theta_B$  equals a clock cycle  $T_s$ , and  $NT_s = N_{os}T_s = 2PT_s + W_B + W_A + T_s$ .  $N_{os}$  is thus the offset count reached by  $A$ 's counter in this particular case and is a constant, all involved parameters being known upfront. As a result, (8) can be rewritten as follows:

$$NT_s = 2T_{OF} + (N_{os} - 1)T_s + \Theta_B + \Theta_A. \quad (9)$$

Finally the TOF can be evaluated as such

$$T_{OF} = \frac{(N - N_{os})T_s - (T_s - (\hat{\tau}_A + \hat{\tau}_B))}{2} \quad (10)$$

where we used previous relations between  $\Theta$  and  $\tau$ . The distance estimation is  $d = cT_{OF}$  where  $c$  is the speed of light.

### C. Simulation Results

The simulations have been carried out for channel models CM1 and CM3 which correspond to LOS residential and office scenarios, respectively [29]. Gaussian derivatives as UWB pulses have been shown to match the FCC recommendations for the 3.1–10.6-GHz spectrum. Only for simulation purposes, in this work a gaussian 22-nd derivative with  $\sigma = 182$  ps was employed. This pulse fits the FCC mask for the lower band between 3.1 and 5 GHz without the need of an external filter. For the practical case of more realistic waveforms, a filter may be necessary to comply with regulations. The received signal is first calculated by means of the convolution of the transmitted pulse with CM1 and CM3 channel impulse responses using Matlab and then imported within the VHDL-AMS environment. The TWR data exchange has been repeated 10 times to allow some statistical calculation over the obtained ranging estimations.

As for the channel, a VHDL-AMS description which takes delay and path loss into account like in [29] has been implemented. Here, a short part of it is shown:

```
constant DISTANCE : real := 10.0;
constant DELAY_CH : real := DISTANCE/3.0e8;
quantity dev_a across term_a to electrical_ref;
quantity dev_b across term_b to electrical_ref;
begin
dev_b == (dev_a'Delayed(DELAY_CH));
end model;
```

In this example, the two systems access the “channel” entity using terminals `term_a` and `term_b` and the *across* quantities defined over them allow to delay pulses of a given  $T_{OF}$ . It is sufficient to apply the above `'DELAYED` operator to the constant `DELAY_CH` calculated through speed of light and `DISTANCE`: Here, quantity `dev_b` is delayed with respect to `dev_a`. In the actual implementation, not reported here for space reasons, the code is enriched with the path loss implementation and includes also the support for bidirectional communication required for TWR. Though [29] refers to different path loss parameters depending on whether the channel at stake is LOS or not (NLOS), for all simulations, only LOS models have been used (see previous note 2). As specified in [29] the slow fading (shadowing) has not been included in our simulations.

Two kinds of VHDL-AMS descriptions called *Ideal* and *Non-Ideal* have been used: In the *Ideal* case quantization and other nonidealities have been excluded, following the *Phase I* paradigm described in Section IV. Furthermore, there is no need of any AGC, since saturation or minimum resolution are not accounted for. For these reasons, the ideal case results should be taken as an upper bound of TWR performance, limited by our algorithmic choices and noise only and not by implementation details. In the *NonIdeal* case, we followed the *Phase II* approach and so the receiver includes quantization and various saturation effects. Here, the analog-to-digital converter feeds the digital comparator, and the necessary digital and mixed signal blocks of the AGC have been included (such as DAC and LUT). Simulations with short and long preambles have been carried out in

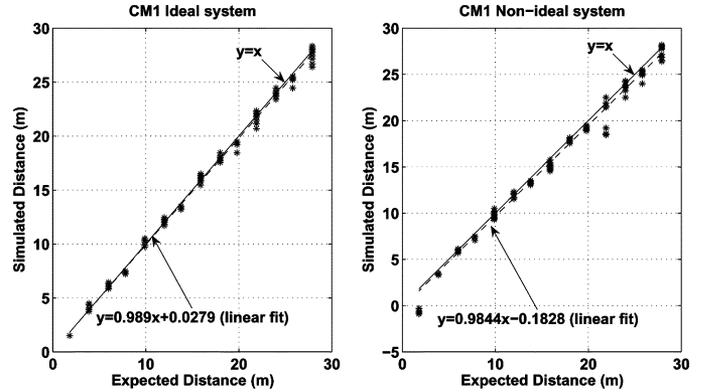


Fig. 10. Simulated versus expected distance, CM1, ideal, and nonideal systems.

both ideal and nonideal cases. Table II summarizes all the results obtained with channel models CM1 and CM3. The results for CM1, only short preamble, ideal and nonideal cases, are also reported in Fig. 10. The x axis is the actual distance between the UWB transceivers and the y axis is the corresponding simulated value. Error-free results would then sit along the  $y = x$  line. We fitted the data using the least squares method and obtained the dashed lines which, in both cases, are very close to the  $y = x$  line. Concerning deviations, average and maximum, all values have been reported in Table II.

Apart from the ideal case, the best results for the nonideal system have been obtained in the CM1, long preamble case, in which 60 cm was the maximum absolute error over the entire distance range, while the average absolute error varies between 15 and 60 cm. Averaging few consecutive measurements (e.g., on the order of 10) to reduce error is certainly an option not to be ignored. Worst results are for CM3, short preamble case, where the maximum absolute deviation varies between 0.3 and 4.5 m and the average lies between 0.3 and 2.15 m. This is not surprising because the parameters that affect accuracy have been tuned for the CM1 channel. The results for the long preamble case are in general better than those for the short one, as it was expected. This result is in agreement with the progress of the IEEE 802.15.4a committee [6] which is currently promoting the adoption of a specific preamble for ranging with a specific modulation and with length even longer than the long case used here.<sup>3</sup>

Based on our experience, these are the main factors which affect accuracy.

- *Integration window*: The channel spread varies with the channel model. Even though a fixed integration window might not cause significant BER degradation [27], ranging results might be instead appreciably affected when accuracies lower than one meter are required. This is the main reason why CM3 results are worse than CM1. For sake of brevity, we did not discuss results obtained varying the window length.
- *Processing gain*: The use of a longer preamble might be very useful for repeating the synchronization phases over consecutive symbols and averaging the time estimations. The noise variance will be lowered by a factor which is

<sup>3</sup>The preamble sequence is M-TOK modulated with perfect balanced ternary codes.

TABLE II  
SIMULATION RESULTS. ABS = ABSOLUTE

expected distance (m)	27.9	25.8	24	21.9	19.8	18	15.9	13.8	12	9.9	7.8	6	3.9	1.8
CM1, IDEAL SYSTEM SIMULATION RESULTS														
average distance (m)	27.67	25.25	23.97	21.74	19.28	18.00	16.02	13.38	12.05	10.10	7.43	6.12	4.08	1.50
standard deviation (m)	0.66	0.29	0.31	0.50	0.30	0.28	0.31	0.09	0.26	0.29	0.10	0.22	0.26	0.0
max abs. deviation (m)	1.50	1.35	0.60	1.20	1.35	0.45	0.60	0.60	0.45	0.60	0.60	0.45	0.60	0.30
average abs. deviation(m)	0.23	0.55	0.027	0.16	0.52	0.0	0.12	0.42	0.05	0.20	0.37	0.12	0.18	0.30
CM1, NON-IDEAL SYSTEM SIMULATION RESULTS														
average distance (m)	27.44	25.10	23.62	21.08	19.27	17.88	15.08	13.28	11.81	9.82	7.35	5.83	3.34	-0.64
standard deviation (m)	0.64	0.43	0.51	1.54	0.18	0.23	0.35	0.18	0.29	0.38	0.16	0.16	0.07	0.25
max abs. deviation (m)	0.46	0.67	0.38	0.82	0.53	0.15	0.82	0.52	0.30	0.60	0.45	0.18	0.56	2.44
average abs. deviation (m)	0.46	0.70	0.38	0.82	0.53	0.12	0.82	0.52	0.19	0.08	0.45	0.17	0.56	2.44
CM1, 192 SYMBOLS LONG PREAMBLE, IDEAL SYSTEM SIMULATION RESULTS														
average distance (m)	27.87	25.56	24.00	21.86	19.85	18.05	15.89	13.89	12.00	10.02	7.94	6.09	4.05	1.80
standard deviation (m)	0.25	0.39	0.41	0.35	0.30	0.29	0.18	0.31	0.24	0.18	0.19	0.24	0.20	0.24
max abs. deviation (m)	0.45	0.75	0.60	0.60	0.45	0.45	0.30	0.60	0.45	0.45	0.45	0.60	0.60	0.30
average abs. deviation (m)	0.03	0.24	0.0	0.05	0.05	0.05	0.02	0.09	0.0	0.12	0.14	0.09	0.15	0.0
CM1, 192 SYMBOLS LONG PREAMBLE, NON-IDEAL SYSTEM SIMULATION RESULTS														
average distance (m)	28.22	25.60	24.27	22.13	20.0	18.24	16.50	14.0	12.38	10.05	8.18	6.35	4.50	2.40
standard deviation (m)	0.29	0.39	0.17	0.33	0.27	0.20	0.0	0.28	0.28	0.22	0.18	0.17	0.0	0.0
max abs. deviation (m)	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.45	0.60	0.60	0.60	0.60
average abs. deviation (m)	0.32	0.20	0.27	0.23	0.20	0.24	0.60	0.20	0.38	0.15	0.38	0.35	0.60	0.60
CM3, IDEAL SYSTEM SIMULATION RESULTS														
average distance (m)	26.39	24.19	23.25	20.11	17.88	17.21	14.72	12.34	11.18	8.74	6.53	5.37	3.51	1.19
standard deviation (m)	1.04	1.12	0.88	0.96	0.98	0.68	0.82	0.81	0.63	0.85	0.45	0.39	0.53	0.26
max abs. deviation (m)	3.00	3.75	1.95	3.15	3.15	1.80	2.70	3.15	1.80	2.25	2.10	1.20	1.65	1.05
average abs. deviation (m)	1.51	1.61	0.75	1.79	1.92	0.79	1.18	1.46	0.82	1.16	1.27	0.63	0.40	0.61
CM3, NON-IDEAL SYSTEM SIMULATION RESULTS														
average distance (m)	26.14	24.24	22.75	20.13	19.40	16.98	14.43	11.89	11.15	8.77	6.0	4.98	1.85	1.50
standard deviation (m)	0.72	0.79	1.09	1.15	0.68	0.69	0.82	0.82	0.40	0.70	0.59	0.33	0.81	0.0
max abs. deviation (m)	3.0	3.30	3.0	4.50	1.95	2.40	2.55	3.30	1.50	2.40	2.85	1.50	2.40	0.30
average abs. deviation (m)	1.76	1.55	1.25	1.77	0.39	1.02	1.47	1.91	0.85	1.13	1.80	1.02	2.05	0.30
CM3, 192 SYMBOLS LONG PREAMBLE, IDEAL SYSTEM SIMULATION RESULTS														
average distance (m)	26.82	24.77	23.24	20.48	18.30	17.06	14.64	13.20	10.89	9.06	7.25	5.15	3.36	1.61
standard deviation (m)	0.85	0.59	0.98	1.0	1.02	0.68	0.68	0.44	0.87	0.52	0.46	0.72	0.57	0.14
max abs. deviation (m)	2.40	1.95	2.70	3.30	4.05	2.10	2.25	1.20	2.55	1.50	1.50	1.95	1.50	0.45
average abs. deviation (m)	1.08	1.04	0.77	1.43	1.50	0.95	1.26	0.60	1.11	0.84	0.56	0.86	0.54	0.20
CM3, 192 SYMBOLS LONG PREAMBLE, NON-IDEAL SYSTEM SIMULATION RESULTS														
average distance (m)	27.33	24.92	23.91	22.98	20.27	17.66	15.27	13.46	11.54	9.41	7.59	6.06	4.19	3.95
standard deviation (m)	0.73	0.67	0.21	0.24	1.01	0.39	0.41	0.30	0.67	0.56	0.30	0.36	0.047	0.21
max abs. deviation (m)	2.25	2.25	0.45	1.50	2.40	0.90	1.35	0.75	1.65	1.35	0.75	0.90	0.30	2.4
average abs. deviation (m)	0.57	0.89	0.09	1.08	0.47	0.35	0.63	0.35	0.47	0.50	0.21	0.06	0.29	2.15

proportional to the processing gain, that is the number of symbols over which the procedure is repeated, mitigating the noise effects on the captured energies. We have not exploited this opportunity in this work as it was immaterial for our purposes. Nonetheless it requires little modification to our VHDL-AMS description and we plan to use it in a forthcoming work.

— *Gain Control*: This phase plays a crucial role since whether the integrator output dynamics will be matched to the ADC input so as to maximize the energy measurement resolution depends on it. Indeed, the more the gain is not properly set, viz. the more the ADC input range is not matched, the more the receiver cannot distinguish the different energy samples obtained by shifting the integration window. The AGC loop we used works in the digital domain. As a consequence, the gain resolution is limited by both ADC and DAC quantization. This effect is evident, for instance, at distance 21.9 m in Fig. 10, nonideal case, where the error is larger than for the longer distance 24 m. At the shortest distance the error is also significantly large. The reason is that the signal strength is too high to avoid the ADC input saturation, even if the AGC sets the smallest possible gain stored in the LUT.

## VII. PHASE III RESULTS

To prove the effectiveness of the tri-phase flow outlined in Section IV, some of the BER and TWR simulations presented before and related to the second phase, have been run again. We decided to substitute the behavioral architecture of one of the fundamental blocks with a layout back-annotated Spice-level

netlist including parasitics. We used the low-noise amplifier presented in [33], fabricated in a 0.18- $\mu\text{m}$  CMOS technology, for which the post-layout netlist was made available. Its main features are the use of a frequency-controlled feedback, a high linearity ( $-2.48$  dBm, 1-dB compression point), an average noise figure of 4.4 dB,  $S_{21} = 8$  dB and a bandwidth that extends over the entire UWB spectrum. We refer the interested reader to the cited paper for further details.

Thanks to our *substitute-and-play* approach, the adaptation of the simulator environment is straightforward, only a short and “painless” VHDL source modification being needed. The entity declaration does not change because the connections to the surrounding system remain unchanged. As for the architecture, it now includes a Spice-like wrapper syntax which allows the simulator to import and simulate the netlist in the VHDL-AMS higher level hierarchy. The circuit-level simulator we used is Eldo, a version of Spice that works under the ADMS tool by Mentor Graphics.

```
*ELDO LNA Subcircuit (lna.ckt)
.subckt LNA Vin Vdd Gnd Vout
XR1 NET9 NET52 GND RNHR_RF L = 9.59E-6 W = 2E-6 M = 1.0...
.ends LNA

--Entity declaration and architecture of the
Spice LNAentity ams_LNA is
port (terminalinput, output, alim ,gnd : electrical);
end ams_LNA;

architecture Level0 of ams_LNA is
component LNAs
```

TABLE III  
TWR SIMULATIONS FOR A SINGLE DISTANCE POINT (*Phase II* vs *Phase III*, LONG PREAMBLE SEQUENCE, CM1)

expected distance (m)	27.9	25.8	24	21.9	19.8	18	15.9	13.8	12	9.9	7.8	6	3.9	1.8
<i>Phase II</i> measured. distance. (m)	28.5	26.1	24.6	22.5	19.95	18.45	16.05	14.4	12.6	9.75	7.95	6.6	4.5	2.40
<i>Phase III</i> measured. distance. (m)	30	26.1	24.6	22.5	20.25	18.45	16.05	14.4	12.6	9.75	7.95	6.6	4.5	2.10

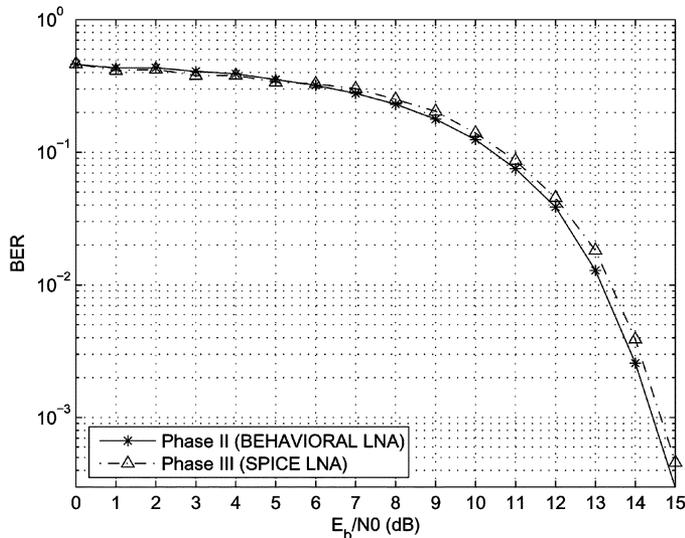


Fig. 11. BER curves obtained in *Phase II* with a VHDL-AMS behavioral LNA and in *Phase III* with a transistor-level LNA.

```

port (terminal term_in, alim, gnd, term_out:electrical);
end component;

ATTRIBUTE Eldo_device OF LNAs: COMPONENT IS Eldo_subckt;
ATTRIBUTE Eldo_subckt_name OF LNAs: COMPONENT IS "LNA";
ATTRIBUTE Eldo_file_name OF LNAs: COMPONENT IS "lna.ckt";

begin
lna0: LNAs port map (term_in=> input, alim => vdd ...
end Level0;

```

The entity and the architecture reported here represent a simplified version of the VHDL-AMS code of LNA employed in the simulations, shortened for space reasons. The Eldo instance LNAs, whose a netlist part is reported on top of the listing, is imported in the architecture. From the system point of view, the entity `ams_LNA` is identical to the *Phase II* one.

For understanding the performance losses when passing from *Phase II* to *Phase III* we ran additional BER and TWR simulations. In Fig. 11, the BER curves obtained with the behavioral and circuit-level versions of the LNA are compared. The simulation conditions are the same of Fig. 4. The 0.25-dB performance loss is imputable to a noise enhancement effect of the circuit-level LNA, owing to an equivalent bandwidth larger than its behavioral counterpart.

The two rows of data in Table III have been obtained with a single run of the TWR Request-Reply Packet exchange in the same condition for the Behavioral and Spice-level LNA (CM1, long preamble case). The estimated distance is almost always the same and so the deviation with respect to the actual distance. The fact that exactly the same numbers occurred most of the times, is not surprising, given that the distance evaluation is quantized due to the fixed one nanosecond resolution of

the  $\tau$  estimation in fine synchronization. Another reason is the pseudo-random noise repeatability in the two types of simulation. Overall, the transistor-level implementation of the LNA did not cause significant additional penalty.

## VIII. CONCLUSION

In this paper, we presented a multiresolution methodology, based on the use of VHDL-AMS, for the simulation and design of an UWB impulse-radio transceiver. We have shown how an entire mixed-signal SoC can be conceived and simulated and how the simulation environment can be used to verify the functionality, to take crucial design decisions such as the number of quantization bits, and to benchmark the performance of the transceiver using realistic channel models. In particular we have shown how the UWB transceiver can be effectively used for ranging applications in LOS links. We demonstrated how our multiresolution approach allows to evaluate the impact of circuit nonidealities described both at a high or low level, i.e., included in the VHDL-AMS description or coming from a transistor-level netlist.

In the future, we will take two different routes, as a natural evolution of the work presented herein. On the one hand, we will complete the design of the UWB impulse-radio transceiver and compare system and circuit-level simulations with measurements on a prototype. On the other hand, we will experiment with other mixed-signal circuits with the aim of extending the application of the methodology to other designs.

## REFERENCES

- [1] "Revision of Part 15 of the Commissions Rules Regarding Ultra-Wideband Transmission Systems," FCC, Washington, DC, Rep. and order, adopted, 2002.
- [2] *UWB Theory and Applications*. Oppermann, M. Hamalainen, and J. Iinatti, Eds.. Chichester, U.K.: Wiley, 2004.
- [3] "Wimedia Alliance," Wimedia, San Ramon, CA, 2008 [Online]. Available: <http://www.wimedia.org>
- [4] C. L. Bennett and G. F. Ross, "Time domain electromagnetics and its applications," *Proc. IEEE*, vol. 6, no. 2, pp. 299–318, Feb. 1978.
- [5] R. J. Fontana, "Recent system applications of short-pulse ultra-wideband (UWB) technology," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 10, pp. 2087–2104, Oct. 2004.
- [6] *Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 1: Add Alternate PHYs*, IEEE Std. 802.15.4a-2007 (Amendment to IEEE Std. 802.15.4-2006, 2007 [Online]. Available: <http://www.ieee802.org/15/pub/TG4a.html>
- [7] J.-Y. Lee and R. A. Scholtz, "Ranging in a dense multipath environment using an UWB radio link," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 10, pp. 1677–1683, Oct. 2002.
- [8] J. Han and C. Nguyen, "On the development of a compact sub-nanosecond tunable monocycle pulse transmitter for UWB applications," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 1, pp. 285–293, Jan. 2006.
- [9] J. Ryckaert, C. Desset, A. Fort, M. Badaroglu, V. De Heyn, P. Wambacq, G. van Der Plas, S. Donnay, B. Pouce, and B. Gyselinckx, "Ultra-wide-band transmitter for low-power wireless body area networks: Design and evaluation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 12, pp. 2515–2525, Dec. 2005.

- [10] L. Stoica, A. Rabbachin, H. O. Repo, T. S. Tiuraniemi, and I. Oppermann, "An ultra-wide-band system architecture for tag based wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1632–1645, Sep. 2005.
- [11] L. Smaini, C. Tinella, D. Helal, C. Stoecklin, L. Chabert, C. Devaulcelle, R. Cattenoz, N. Rinaldi, and D. Belot, "Single-chip CMOS pulse generator for UWB systems," *IEEE J. Solid-State Circuits*, vol. 41, no. 7, pp. 1551–1561, Jul. 2006.
- [12] Y. Zheng, Y. Tong, J. Yan, Y.-P. Xu, W. G. Yeoh, and F. Lin, "A low power noncoherent CMOS UWB transceiver ICs," in *Proc. IEEE RFIC*, Long Beach, CA, Jun. 2005, pp. 347–350.
- [13] I. D. O'Donnell and R. W. Borsersen, "An ultra-wide-band transceiver architecture for low power, low rate, wireless systems," *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1623–1631, Sep. 2005.
- [14] R. Blazquez, P. P. Newaskar, F. S. Lee, and A. P. Chandrakasan, "A baseband processor for impulse ultra-wide-band communications," *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1821–1828, Sep. 2005.
- [15] Y. Souilmi and R. Knopp, "On the achievable rates of ultra-wide-band PPM with noncoherent detection in multipath environments," in *Proc. IEEE ICC*, Anchorage, AK, May 2003, pp. 3530–3534.
- [16] A. Rabbachin and I. Oppermann, "Synchronization analysis for UWB systems with a low-complexity energy collection receiver," in *Proc. Joint UWB Syst. Tech.*, Kyoto, Japan, May 2004.
- [17] *VHDL Analog and Mixed-Signal Extensions*, IEEE Std. 1076.1, 1999.
- [18] E. Christen and K. Bakalar, "VHDL-AMS-A hardware description language for analog and mixed-signal applications," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 12, pp. 1263–1272, Dec. 1999.
- [19] F. Pecheux, C. Lallement, and A. Vachoux, "VHDL-AMS and Verilog-AMS as alternative hardware description languages for efficient modeling of multidiscipline systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 2, pp. 204–225, Feb. 2005.
- [20] J. Ravatin, J. Oudinot, S. Scotti, A. Le-clercq, and J. Lebrun, "Full transceiver circuit simulation using VHDL-AMS," *J. Microw. Eng.*, pp. 29–33, 2002.
- [21] E. Normark, L. Yang, C. Wakayama, P. Nikitin, and R. Shi, "VHDL-AMS behavioral modeling and simulation of a  $\pi/4$  DQPSK transceiver system," in *Proc. IEEE Int. Beh. Modeling and Simul. Conf.*, San Josè, CA, Oct. 2004.
- [22] W. Yang, H. Carter, and J. Yan, "A high level VHDL-AMS model design methodology for analog RF LNA and mixer," in *Proc. IEEE Int. Beh. Modeling and Simul. Conf.*, San Josè, CA, Oct. 2004.
- [23] A. Fakhfakh, N. Milet-Lewis, J. Tomas, and H. Levi, "Behavioral modeling of phase noise and jitter in voltage-controlled oscillators with VHDL-AMS," in *Proc. IEEE Int. Conf. Circuits Syst. Commun.*, St. Petersburg, Russia, Jun. 26–28, 2002, pp. 370–373.
- [24] Y. Joannon, V. Berouille, R. Khouri, C. Robach, S. Tedjini, and J.-L. Carbonero, "Behavioral modeling of WCDMA transceiver with VHDL-AMS language," in *Proc. IEEE Workshop Des. Diagn. Electron. Circuits Syst.*, Prague, Czech Rep, Apr. 18–21, 2006, pp. 111–116.
- [25] V. Nguyen, P. Loumeau, and J.-F. Naviner, "VHDL-AMS behavioral modelling and simulation of high-pass delta-sigma modulator," in *Proc. IEEE Int. Beh. Modeling and Simul. Conf.*, San Josè, CA, Oct. 2005, pp. 106–111.
- [26] M. Sida, R. Ahola, and D. Wallner, "Bluetooth transceiver design and simulation with VHDL-AMS," *IEEE Circuits Devices Mag.*, vol. 19, no. 2, pp. 11–14, Mar. 2003.
- [27] C. Carbonelli and U. Mengali, "M-PPM noncoherent receivers for UWB applications," *IEEE Trans. Wireless Commun.*, vol. 5, no. 8, pp. 2285–2294, Aug. 2006.
- [28] C. Carbonelli and U. Mengali, "Low complexity synchronization for UWB noncoherent receivers," in *Proc. IEEE Veh. Technol. Conf. (VTC) 2005-Spring*, Stockholm, Sweden, May 2005, pp. 1350–1354.
- [29] "IEEE 802.15.4a Channel Model—Final Report, Rev. 1," IEEE 802.14.4a Standardization Committee, Nov. 2004 [Online]. Available: <ftp://ieeewireless@ftp.802wirelessworld.com/15/04/15-04-0662-00-004a-channel-model-final-report-r1.pdf>
- [30] M. Crepaldi, M. Casu, and M. Graziano, "Energy detection UWB receiver design using a multiresolution VHDL-AMS description," in *Proc. IEEE Workshop Signal Process. Syst.*, Athens, Greece, Nov. 2–4, 2005, pp. 13–18.
- [31] C. Mishra, A. Valdes-Garcia, F. Bahmani, A. Batra, E. Sanchez-Sinencio, and J. Silva-Martinez, "Frequency planning and synthesizer architectures for multiband OFDM UWB radios," *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 12, pp. 3744–3756, Dec. 2005.
- [32] A. A. Emira, A. Valdes-Garcia, B. Xia, A. N. Mohieldin, A. Y. Valero-Lopez, S. T. Moon, C. Xin, and E. Sanchez-Sinencio, "Chameleon: A dual-mode 802.11b/bluetooth receiver system design," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 5, pp. 992–1003, May 2006.
- [33] L. M. Y. Soliman and L. Roy, "A CMOS ultra-wide-band lna utilizing a frequency-controlled feedback technique," in *Proc. IEEE Int. Conf. on UWB (ICU'05)*, Zurich, Switzerland, Sep. 2005, pp. 530–535.



**Mario R. Casu** (M'05) received the electronics engineer degree (*summa cum laude*) and the Ph.D. degree from the Politecnico di Torino, Turin, Italy, in 1998 and 2002, respectively.

He is an Assistant Professor in Department of Electronics, Politecnico di Torino. His interests encompass CAD aspects of deep-submicrometer CMOS technologies, architectural and register transfer level design of multiprocessors systems-on-chip and mixed-signal design for ultra-wide-band applications.



**Marco Crepaldi** received the D.Eng. degree in electronics engineering (*summa cum laude*) from the Politecnico di Torino, Turin, Italy, in 2005, where he is currently pursuing the Ph.D. degree.

His research interests are VHDL-AMS behavioral modeling of analog and mixed-signal circuits and ultra-wide-band communications at system and circuit level.



**Mariagrazia Graziano** (M'07) received the D.Eng. degree in electronics engineering and the Ph.D. degree in electronic and communication engineering from the Politecnico di Torino, Turin, Italy, in 1997 and 2001, respectively.

From 1999 to 2000 she was a Visiting Scholar at University of Illinois at Chicago. Since 2002 she has been an Assistant Professor at the Politecnico di Torino. Her research interests include the design of ultra-wide-band CMOS transceivers and the development of models and algorithms for the design of CMOS noise-tolerant digital and mixed-signal circuits.