POLITECNICO DI TORINO
Repository ISTITUZIONALE

Energy Detection UWB Receiver Design using a Multi-resolution VHDL-AMS Description

(Article begins on next page)

23 September 2024

Energy Detection UWB Receiver Design using a Multi-resolution VHDL-AMS Description

Marco Crepaldi, Mario R. Casu and Mariagrazia Graziano

CERCOM, Dipartimento di Elettronica, Politecnico di Torino, Italy

Email: {mario.casu,mariagrazia.graziano}@polito.it

Abstract— **Ultra Wide Band (UWB) impulse radio systems are appealing for location-aware applications. There is a growing interest in the design of UWB transceivers with reduced complexity and power consumption. Non-coherent approaches for the design of the receiver based on energy detection schemes seem suitable to this aim and have been adopted in the project the preliminary results of which are reported in this paper. The objective is the design of a UWB receiver with a top-down methodology, starting from Matlab-like models and refining the description down to the final transistor level. This goal will be achieved with an integrated use of VHDL for the digital blocks and VHDL-AMS for the mixed-signal and analog circuits.**

Coherent results are obtained using VHDL-AMS and Matlab. However, the CPU time cost strongly depends on the description used in the VHDL-AMS models. In order to show the functionality of the UWB architecture, the receiver most critical functions are simulated showing results in good agreement with the expectations.

I. INTRODUCTION

Ultra Wide Band (UWB) systems using short pulse modulation are nowadays considered a viable solution for short-range mobile communications. For high data-rate UWB applications the OFDM-based modulation scheme has become the preferred approach. Viceversa UWB impulse radio is viewed as an interesting solution for short-range low-data rate mobile communications and in particular for *location-aware* wireless sensor networks [1].

The impulse radio UWB receiver is simpler than classic narrowband radio receivers. At low data rates, this approach seems to promise significant savings in terms of power consumption of the circuit that implements the receiver functions. However, at the best of our knowledge, there are not yet practical demonstrations of the integration of all the receiver functions, (digital, analog and radio-frequency) into a single CMOS chip. Only a few publications indicate the complete integration of a UWB transceiver in CMOS technology as a final goal [2] [3] [4]. These references, however, put severe restrictions on the propagation model. In some cases an AWGN channel is assumed, leading to a correlation receiver in which the template waveform is matched to the transmitted pulse. However, such a receiver has poor performance over a dense multipath channel. Other approaches are based on Rake receivers, whose performance is acceptable only when a large number of fingers is employed, which translates into a high power consumption. Instead of a Rake, a correlation detector may be used in which the incoming waveform is correlated with a local replica (template) of the channel response. A correlator with an ideal template is equivalent to a Rake with infinite fingers and perfect knowledge of the channel gains and delays. Unfortunately, the estimation of the channel response involves high sampling rates and intensive signal processing [7] [8], which is hardly compatible with low power consumption.

Another interesting option is a non-coherent approach [5] [6] which makes no attempts to gather information on the channel response. Its data decisions are based only on signal energy measurements. For example, assume binary pulse position modulation (2-PPM) with rate $1/T$. Symbol "zero" corresponds to transmitting a pulse in the first half of the interval (0,T) while symbol "one" corresponds to transmitting a pulse in the second half. The receiver measures the signal energies on both halves and selects the symbol with the largest energy. This energy-capture scheme appears as a valid solution in terms of complexity, cost and power consumption and for these reasons has been employed in this work. The simplicity of this approach comes at the cost of a lower immunity to interference coming from other users or other systems. The multi-user interference is barely solved using a TDMA approach which is affordable at low data-rates. As for the other interferences, out-of-band signals have to be filtered out, while in-band interference has to be limited by avoiding superposition to already assigned bandwidths.

The entry point of a typical design flow of a System-on-Chip (SoC) for wireless applications, and so also for UWB, consists in a preliminary evaluation of performance through Matlab functions or C-based programs. Then, once the achievable performance is assessed, both the analog, RF and digital parts are designed and analyzed separately through the use of different simulators (circuit and logic level). In order to assess the system performance obtained after design and before the SoC sign-off, and to compare it with the high-level estimation, a number of lengthy iterations from one simulator to another is necessary. Designers may benefit from the use of a single CAD interface. New tools are nowadays available in the market that allow the co-simulation of digital parts, described using hardware description languages like VHDL and Verilog, with analog and RF parts described using the VHDL-AMS language or Spice-like netlists [10][11]. These tools seem having the necessary flexibility to support both very accurate (e.g. transistor level) and behavioral (e.g. Matlab-like) simulations. We investigated the capabilities of one of these

commercial tools in the design and simulation of a UWB receiver employing energy detection. In particular we were interested in:

- Analyzing if Matlab and behavioral VHDL-AMS descriptions produce the same assessment of performance (e.g. bit error rate) and, in case of positive answer, what is the penalty in terms of CPU time.
- Assessing the quality of functional simulations using hardware-oriented description of blocks and system.

This paper, which summarizes our work so far, is organized as follows. In section II we report a brief survey of the use of analog hardware description languages in other contexts but with similar constraints and specifications as in our case. In section III we describe the architecture of the UWB receiver. Section IV reports the results of the high-level comparison with Matlab, while section V shows the functional simulations. Finally, conclusions are drawn in section VI.

II. USING VHDL-AMS FOR A FLEXIBLE DESIGN PROCEDURE

As the complexity of telecommunication systems rapidly grows, and product life cycle shrinks, traditional design approaches fail in following the increasing ability to manufacture large System-on-Chip. The must for filling the gap is to use a top-down design methodology not only for digital circuits, but for mixed-signal designs as well. This is mandatory especially in the light of the fact that the typical bottom-up analog design approach leads, in the case of very large circuits, to transistor-level impractical verification phases. This has been the driving force for which the VHDL has been extended to the VHDL-AMS superset language conceived for modeling mixed-signal and mixed technology systems [12]. The main advantages of this relatively new standard are its support of different levels of abstraction, from behavioral to circuit level, and the possibility to use digital constructs together with electrical quantities, differential equations and algebraic constraints. Furthermore, VHDL-AMS supports both time and frequency-domain simulation, noise modeling and various numerical forms.

This possibility to simulate both at a fine grain and at a coarser level – we call it *multi-resolution* capability – allows the user to understand the trade-off between accuracy and CPU time. The design description can start at system level using behavioral models of the blocks. High level simulations can be thus performed with the aim of testing the overall functionality, exploring different solutions and assessing the system performance. Once the architecture is stable, the blocks are progressively refined towards the final transistor level description.

A further level of flexibility is offered by one of the available commercial tools, ADVance MS^{TM} [10] (ADMS), which we are using in this work. It supports VHDL, VHDL-AMS and Spice (Eldo, Eldo-RF) languages that may coexist in the same simulation run. Therefore different abstraction levels, each assigned to a different description language, allows the designer to focus on critical blocks without missing the system environment. For instance, single analog blocks described at transistor level with a Spice netlist may be embedded in a system described at a higher level using VHDL-AMS and simulated together, provided that interfaces are coherently defined. After this step, one can, for example, go back to the behavioral description substituting the Spice block with an improved VHDL-AMS description. This one is still compatible at the terminals level but includes non-ideal effects that the previous Spice characterization reveals being of essence.

A few works document the use of VHDL-AMS as an effective simulation language for efficiently design complex systems using a top-down methodology. Many disciplines are interested in the rich features of this language, like automotive, MEMs design, and, of course, telecommunications. In [13] different combinations of VHDL-AMS and transistor-level descriptions to achieve both accuracy and simulation speed were used for the design of a Bluetooth transceiver. Simple behavioral models were adopted to validate the concept; then a few blocks were selected to refine the behavioral model so as to match the transistor level model; finally an exhaustive verification was performed on the behavioral-characterized model, with the sufficient accuracy to reliably estimate bit error rate (BER) figures in a reasonable simulation time. In [14], basic functionality tests using an abstract VHDL-AMS behavioral description were used for a full transceiver circuit simulation without the inclusion of RF blocks. In [15], RF blocks of a DQPSK transceiver and a channel model are implemented adding white gaussian noise, achieving BER results very close to theoretical models. Jitter has been introduced in [16] for modeling the real behavior of a PLL using VHDL-AMS: The phase noise simulated spectrum was in good agreement with the measured results. In [17] a methodology is proposed to design RF circuits in VHDL-AMS starting from flexible specifications and assuring an accurate description of noise and non-linearity effects.

We intend to use VHDL-AMS to aid the progressive refinement of the design of our UWB receiver through the multi-resolution engine in ADMS. This goal will be reached in 5 steps:

1) The receiver is described at a very high level and the performance in terms of BER assessed.
2) A preliminary refinement consists in identifying all blocks and their connections. Even though the architecture is defined, blocks are internally described only in a behavioral fashion.
3) The description of RF, analog and digital blocks is progressively refined including non-ideal effects (offset, delay, jitter, . . .) as well as a power evaluation.
4) The blocks are designed at the necessary level of accuracy (e.g. transistor-level for the analog and mixed parts, gate-level or RTL for digital parts) and characterized.
5) Useful information from the last step is fed to the VHDL-AMS description of blocks to tune the simulations.

At the time of writing, steps 1 and 2 are done and we report the results in this paper (section IV and V). Step 3 is in progress and the following steps are planned.

III. THE UWB RECEIVER ARCHITECTURE

As stated previously, the goal of our work, whose preliminary results are reported in this paper, is the design of a UWB noncoherent receiver based on energy detection [9]. In the following we summarize the receiver's operation on a 2-PPM modulated train of pulses. The received signal $r(t)$ is

$$r(t) = \sum_{j=-\infty}^{+\infty} s\left(t - jT_s - a_jT_s/2 - \tau\right) + n(t) \qquad (1)$$

where $s(t)$ is the channel response to an isolated UWB pulse emitted from the transmitter. It 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 $100\,\text{ns}$. The a_j are binary independent and identically distributed data symbols taken from the alphabet $(0, 1)$ and T_s is the symbol period. Finally, $n(t)$ represents a 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 τ represents the time offset between the transmit and receive clocks. So, a pulse would always appear at the beginning of a symbol interval. Viceversa, 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.

The incoming waveform is first squared and then fed to an integrate-and-dump (I&D) circuit to compute the quantities

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

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

where Δ is on the order of the channel spread. The decision device sets $\hat{a}_k = 0$ or $\hat{a}_k = 1$ according to the rule

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

The theoretical receiver's operations described above have been implemented by means of building blocks, referring to the existent literature as much as possible. However we found a lack of receiver architectures that fully integrate energy detection and ranging capabilities. Furthermore, the complexity of designing such a circuit depends on the fact that it integrates blocks of a different nature on the one hand, like RF, analog and digital parts, but, on the other hand, its functionality relies upon their strict entanglement. Finally, the necessity to lower the power consumption is mandatory in low-data-rate applications. These motivations move the authors toward aiding the receiver design, whose architecture is described in the following, with the potentiality of a multiresolution simulation, according to terms of section II.
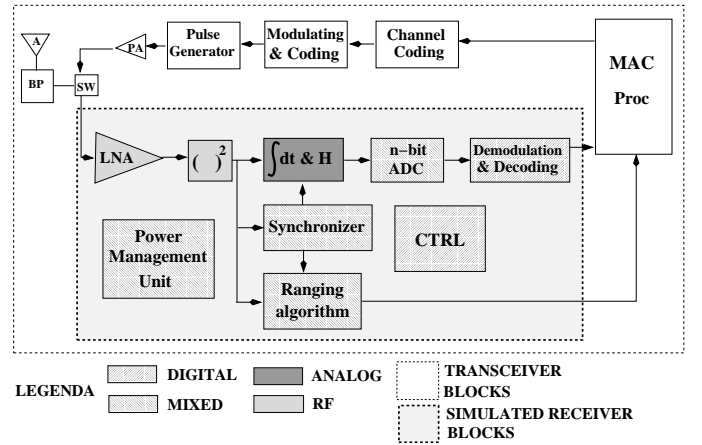

Fig. 1. Receiver structure.

The receiver building blocks are shown in figure 1 together with the transmitter and MAC level functions, which have not been implemented in this work and represent the objective of our future work aiming at the design of a complete transceiver. The receiver includes a Low-Noise Amplifier (LNA) and the blocks which realize the operation described in equations (2)-(4), viz., a square-law module, an integrate-and-dump circuit, a synchronizer whose aim is to recover the timing and which ideally sets $\tau = 0$. The A/D converter may be a simple comparator in the simple case of equation (4) if decision is taken directly on the I&D analog output values. The ranging block is a fine synchronizer which finds the first echo of the received signal and uses it, in conjunction with the MAC block, for ranging and localization. The power manager minimizes the power consumption by activating and deactivating the functional blocks. The system controller coordinates all the operations. The decoding block is used in case of multiple access and/or channel coding usage. The blocks incorporate the capability to work in stand-by and/or power-off mode, depending on the control signals issued by the power consumption management unit.

IV. COMPARISON BETWEEN MATLAB AND VHDL-AMS

Our first aim was to ensure the tool coherence with Matlab for the highest level simulation. We recall that Matlab is used in this context as a "discrete time" simulator, where all signals are represented with the least number of samples according to the Nyquist's limit. We first implemented a basic version of the receiver in Matlab and derived results of bit error rate (BER) varying the signal-to-noise ratio (SNR) at the receiver input. This was done by adding a AWGN noise, created using the Matlab random generator, to a signal compliant with the indoor UWB channel model, as defined by the IEEE 802.15.3a committe [18]. Then we implemented in VHDL-AMS the receiver shown in figure 1. The gaussian noise was created like in [15] while the channel model was the same used for the Matlab simulation. For the BER analysis we supposed perfect timing acquisition (ideal synchronizer). Moreover, the
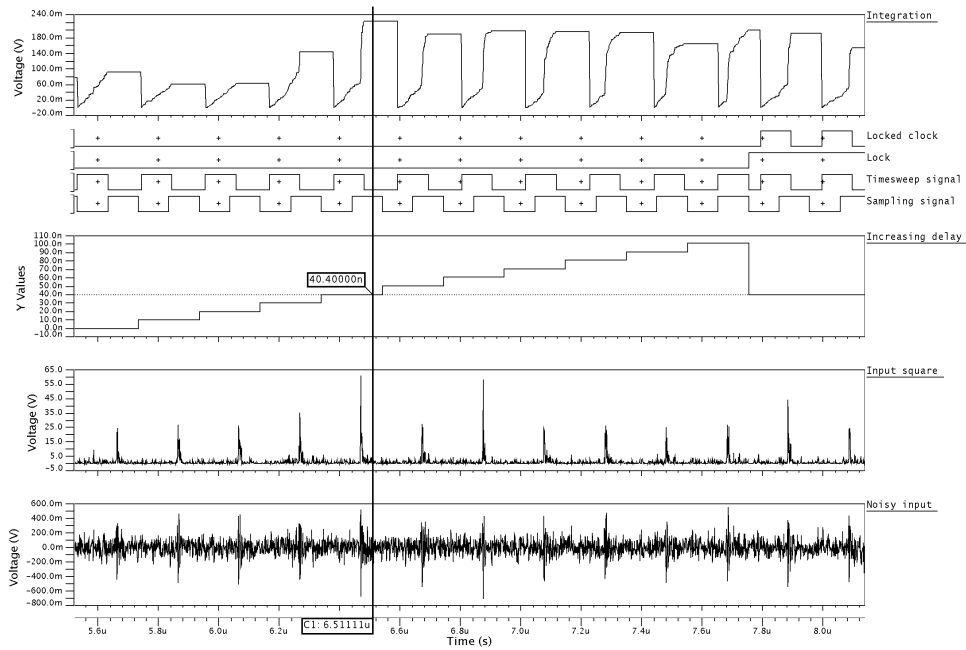
Fig. 3.   Functional simulation of the synchronization phase.

non-ideal behaviors of all blocks were disabled (offsets, non-linearities, ...). It is rather intuitive to understand that the most critical block for the simulation is the I&D one. While the other blocks implement memoryless functions, this one requires to integrate a signal in the time domain. The simulation of the VHDL-AMS receiver has been then conducted in two different ways, in the following referred to as "VHDL-AMS 1" and "VHDL-AMS 2". In the first one the integration was performed in a "continuous time" fashion. In the second one, a "discrete time" integration was done, using a suitable sampling time. Therefore, in the latter case we constrained the VHDL-AMS simulator to work in a way similar to how Matlab works. We wanted to analyze the lost in accuracy, if any, and compare the CPU times. Figure 2 compares the BER curves obtained with Matlab and ADMS (two cases, VHDL-AMS 1 and 2). A high number of simulations (on the order of 100,000) on four different channels (variable distances between TX and RX and various configurations, line-of-sight and non line-of sight) and 30 different realizations per channel have been averaged. As the graph shows, the three curves perfectly overlap. Therefore we concluded that there is no lost in accuracy using VHDL-AMS. However, the CPU times were very different. Table I reports the number of simulation runs per SNR point per hour in the three cases for a Pentium 4 running at 2.8 GHz and with 512 MB DRAM. As expected, "VHDL-AMS 2" is

TABLE I

COMPARISON BETWEEN MATLAB AND VHDL-AMS PERFORMANCE
EVALUATED AS NUMBER OF SIMULATIONS PER SNR POINT PER HOUR.

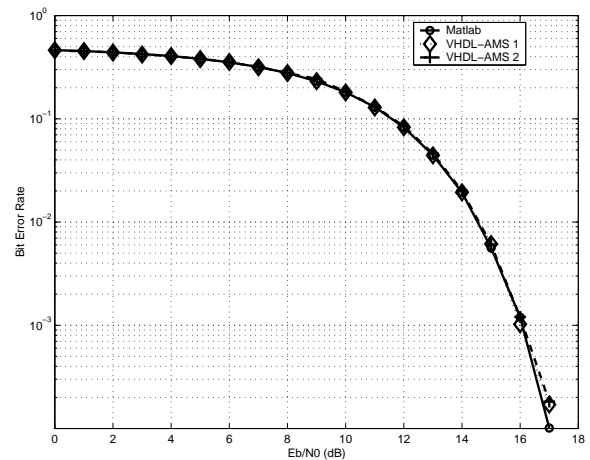| Simulator	Performance
Matlab	14100
VHDL-AMS 1	870
VHDL-AMS 2	9200

Fig. 2.   BER curves: Matlab vs. VHDL-AMS 1 and VHDL-AMS 2.

closer to the fastest Matlab implementation, while "VHDL-AMS 1" is very slow compared to the others. This leads to the conclusion that in order to obtain high-level results as fast as in the Matlab case, a "continuous time" simulation is unsuited. This is however necessary when functional simulations are needed, and this is precisely what the circuit designers need for their purposes. Fortunately, testing the functionality of blocks and system does not require extensive simulations lasting thousands of pulses.

V. FUNCTIONAL SIMULATIONS

As stated in section II we plan to complete the UWB receiver design through five refinement steps, whose the first twos are discussed in this paper, the remaining threes being the subject of current investigation. The first one has been

described in section IV. The second step consists in the definition of the blocks in figure 1, of their connections and of their functionality. At this stage a behavioral VHDL-AMS description is still employed and non-ideal effects are not included, or, if so, they are coarsely modeled.

One of the most critical functions is the *synchronization*. The synchronizer goal is to disclose the best timing so that the pulse appears at the integrator input at the beginning of a symbol interval, no matter the offset between transmitter and receiver clocks. A good synchronization, apart its need for correct timing recovery, helps improving the signal-to-noise ratio at the demodulation block input. In practice, in order not to lose significant pulse energy, integration should not start in the middle of a pulse. The solution we adopted in this work is simple, and for sure not the best one, but we consider it a valid algorithm for proving the synchronization function. It consists of the following steps:

pre-synchronize - detect when the signal appears. A synchronizer sub-block (pre-synchronizer) distinguishes between periods when the signal is absent (noise only) from periods when the signal is present.

synchronize - use a non-modulated packet preamble for the synchronization[1]; the same symbol is repeated M times and the total preamble length in time is defined a priori as $M\,T_s$. The energy of each mth symbol (from $m = 0$ to $m = M-1$) is detected and a synchronization point is chosen in the following way:

→ the total $M\,T_s$ lenght is split in M energy detection ranges of constant length T_s;

→ during each mth repeated step the signal energy is identified by integrating for a time $T_s/2$;

→ at each repetition an increasing offset t_{off}^m with respect to the initial point of each new period $m\,T_s$ delays the integration initial point, where

$$t_{off}^m = \frac{T_s/2}{M - 1} \cdot m$$

→ the t_{off}^m used for the maximum energy detected while sweeping through the whole preamble is used as the most probable synchronization point:

$$t_{synch} = t_{off}^m \Big|_{\max_{m \in 0..M-1}} \int_{t_{off}^m + m\,T_s}^{t_{off}^m + T_s/2 + m\,T_s} r^2(t)dt \quad (5)$$

The *demodulation* then starts using a clock the phase of which is locked to the t_{synch} found during the synchronization. An example of the results obtained simulating the synchronizer is reported in figure 3. In this case $M = 11$, so that m sweeps from 0 to 10; the repetition period is $T_s = 202$ns. The input with added noise and its squared value are in the bottom graphs in the figure. The *timesweep* signal is used for defining the integration range: At positive edge the integration starts (the *integration* signal rises), while at negative edge the

[1]For the packet structure, we refer the reader to the IEEE 802.15.4a committee that is revising the use of standard 802.15.4 in the context of WPAN applications [1].

integrated value stops and the last value is held (the *integration* signal is constant). In each period the *timesweep* signal is delayed by t_{off}^m. The increasing value of t_{off}^m is reported in the figure as the *increasing delay* step signal. Once the integration phase is finished, the hold phase starts and the held value is measured using a *sampling signal* (negative edge triggered). At the end of the preamble cycle the maximum measured value is detected and thus the maximum energy is found. As a consequence, the corresponding offset is defined: In this case $t_{synch} = t_{off}^m | m = 4$ as underlined in the figure. Its value is used for fixing the clock: The *lock* signal is asserted, and the *locked clock* is generated.

Once the synchronization phase is concluded, the demodulation starts using the correct *locked clock*. The decision algorithm is implemented as described in section III. Figure 4 shows the demodulation for three symbols that are in sequence 0-1-0 (k, $k+1$ and $k+2$ respectively in the graph). The logic input (*mod in*) is shifted by a clock period in the figure with respect to the *input square* so that its superposition to the demodulated signal *demod out* is made clearer.

Let's consider the case of the first symbol k: in the first half of the period (from $t = 0$ to $t = T_s/2$) the square signal is integrated, so that the energy value for a possible "zero", $z_k^{(0)}$ in figure, is held after $t = T_s/2$. During the second half of the period (from $t = T_s/2$ to $t = T_s$) the energy value for a possible "one", $z_k^{(1)}$ in figure, is detected and held after T_s. A comparison between the twos is now possible, and a decision is set using the bigger energy value. In the first symbol case, $\hat{a}_k$ is set to 0, because the pulse energy was clearly higher in the first part of the period, according to (4). Thus the *demod out* signal is set to 0 at the end of the first period according to the *mod in* shifted signal. The second symbol detection, $k+1$, works accordingly, but $\hat{a}_{k+1}$ is set to 1 as the higher energy is clearly in the second half of the period. Signal *demod out* is correctly asserted and corresponds to the *mod in* shifted signal.

The proposed architecture aims at demonstrating the synchronization feasibility. The details of the implementation have been omitted as they are the subject of our current investigation. In particular the main choices concern the digital or analog implementation of the bit decision of equation (4) and the maximum search of (5). We envisage a simpler digital implementation using a ADC after the I&D block, as shown in figure 1, but we will take a final decision on the basis of proper metrics like cost, complexity, and energy efficiency.

VI. CONCLUSION

Location-aware applications, especially for short ranges and low data-rates are gathering a growing interest by the industry and the wireless research community. UWB communications based on the transmission of short baseband pulses seems meeting the centimeter-range specifications of location accuracy. The success of such applications strongly depends on the reduction of the UWB transceivers complexity and of their power consumption. To this aim, non-coherent approaches
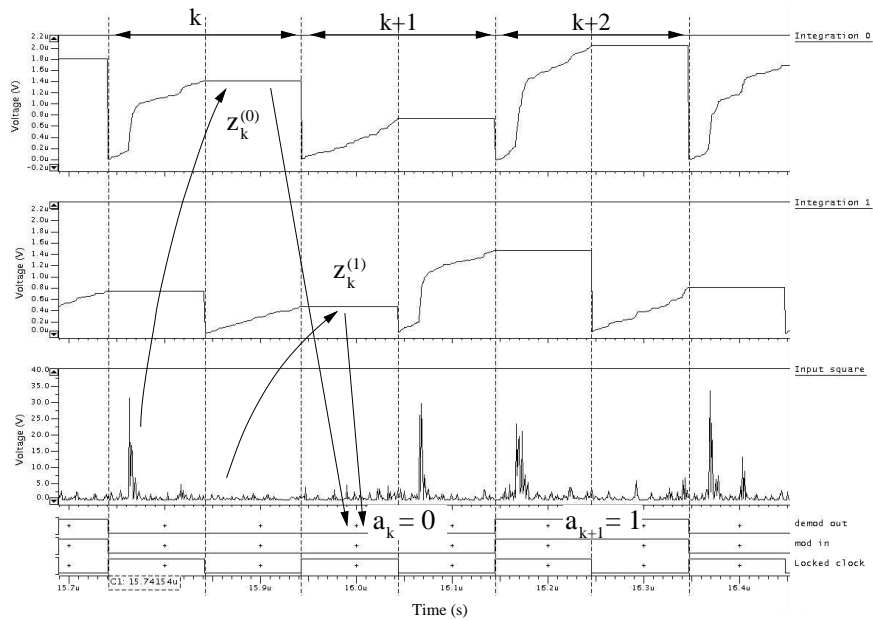
Fig. 4. Functional simulation of the demodulation phase.

based on energy detection schemes are proposed in the literature. The aim of our work, whose preliminary results are reported in this paper, is to fully design a UWB receiver based on energy detection with a top-down methodology, starting from a behavioral and converging to a transistor level description. This goal will be achieved with the aid of new tools based on the integrated use of VHDL, VHDL-AMS, and spice-based languages which allow to perform both system level simulations, normally carried out using Matlab-like languages, and accurate transistor level simulations, simply refining the basic blocks description accuracy.

We showed in this paper that coherent bit-error-rate results are obtained when comparing VHDL-AMS and Matlab simulations. Particular attention must be paid in VHDL-AMS models in order to reduce the CPU time for a given accuracy. Furthermore, we showed the functionality of the UWB architecture adopted simulating the most critical receiver functions like synchronization and demodulation. These preliminary results will be used as a starting point for the next design steps. Offset, delay, jitter, power evaluation are currently being added to the VHDL-AMS description of the RF, analog and digital blocks. The final steps will be the progressive refinement of the modeling down to the final transistor level description.

REFERENCES

[1] IEEE 802.15 WPAN Low rate Alternative PHY Task Group 4a (TG4a), [Online]: www.ieee803.org/15/pub/TG4a.html
[2] O'Donnell I, Chen MS, Wang S, Brodersen RW. "An Integrated, Low-Power, Ultra-Wideband Transceiver Architecture for Low-Rate, Indoor Wireless Systems". In *Proc. of the IEEE CAS Workshop on Wireless Communications and Networking,* Sept. 5-6, 2002.
[3] Chen MS. "Ultra Wide-band Baseband Design and Implementation". *2002 M.Sc. University of California at Berkeley,* (advisor: Brodersen RW).
[4] Blazquez R, Newaskar PP, Lee FS, Chandrakasan AP. "A Baseband Processor for Pulse Ultra-Wideband Signals". In *Proc. of the IEEE Custom Integrated Circuits Conf.,* Orlando, October 2004.
[5] Rabbachin A, Oppermann I. "Synchronization analysis for UWB systems with a low-complexity energy collection receiver". In *Proc. of Joint UWBST & IWUWBS,* Kyoto, Japan, May 2004, pp. 288 –292.
[6] Souilmi Y, Knopp R. "On the achievable rates of ultra-wideband ppm with non-coherent detection in multipath environments". In *Proc. of the International Conf. on Communications*, vol. 5, Anchorage, USA, 2003, pp. 3530–3534.
[7] Carbonelli C, Mengali U, Mitra U. "Synchronization and channel estimation for uwb signals". In *Proc. of the Global Telecommunications Conf.,* S. Francisco, USA, 1-5, Dec. 2003, vol. 2 pp. 764-768.
[8] C.Carbonelli, S.Vedantam, U.Mitra, "Sparse channel estimation with zero tap detection", In *Proc. of the IEEE International Communication Conf.*, Paris, France, June 2004.
[9] C.Carbonelli and U.Mengali, "Low Complexity Synchronization for Non-coherent UWB Receivers", in *Vehicular Technology Conf.*, Stockholm, May 30-June 1 2004.
[10] ADVance MS^{TM} Reference Manual, Mentor Graphics Inc.
[11] Virtuoso^R AMS Designer Simulator and Incisive^{TM} AMS Reference Manual, Cadence Inc.
[12] "VHDL Analog and Mixed-Signal Extensions: IEEE std. 1076.1-1999"
[13] M. Sida, R. Ahola adn D: Wallner, "Bluetooth transceiver design and simulation with VHDL-AMS", *IEEE Circuit and Devices Magazine*, vol. 19, pp.11-14, march 2003.
[14] J. Ravatin, J. Oudinot, S.Scotti, A. Le Clercq and J. Lebrun, "Full transceiver circuit simulation using VHDL-AMS", *Journal of Microwave Engineering,* pp. 29-33, May 2003.
[15] E. Normark, L. Yang, C. Wakayama, P. Nikin, and R. Shi, "VHDL-AMS Behavioral modeling and Simulation of a $\pi/4$ DQPSK transceiver System", in *Proc. of IEEE International Behavioral Modeling and Simulation Conf.*, San Josè, California, 21-22 October 2004.
[16] A. Fahkfahk, N. Milet-Lewis, J. Tomas and H. Levi, "Behavioral modeling of phase noise and jitter in voltage-controlled oscillators with VHDL-AMS", in *Proc. of IEEE International Conf. on Circuits and Systems for Communications*, St.Petersburg, Russia, June 26-28, 2002
[17] W. Yang, H. Carter and J. Yan, "A High Level VHDL-AMS model design methodology for analog RF LNA and Mixer", in *Proc. of IEEE International Behavioral Modeling and Simulation Conf.*, San Josè, California, 21-22 October 2004.
[18] J. Foerster (editor), "Channel modeling sub-committee report final," IEEE P802.15 02/490r1 SG3a, Feb 2002.