



POLITECNICO DI TORINO  
Repository ISTITUZIONALE

Implementation Aspects of a Transmitted-Reference UWB Receiver.

*Original*

Implementation Aspects of a Transmitted-Reference UWB Receiver / CASU M.R.; DURISI G.. - In: WIRELESS COMMUNICATIONS AND MOBILE COMPUTING. - ISSN 1530-8669. - STAMPA. - 5(2005), pp. 537-549. [10.1002/wcm.309]

*Availability:*

This version is available at: 11583/1399097 since:

*Publisher:*

Wiley

*Published*

DOI:10.1002/wcm.309

*Terms of use:*

openAccess

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)

# Implementation aspects of a transmitted-reference UWB receiver

Mario R. Casu<sup>1,\*†</sup> and Giuseppe Durisi<sup>2</sup>

<sup>1</sup>CERCOM-Politecnico di Torino Corso Duca degli Abruzzi 24 I-10129 Torino, Italy

<sup>2</sup>Istituto Superiore Mario Boella Via Pier Carlo Boggio 61 I-10138 Torino, Italy

## Summary

In this paper, we discuss the design issues of an ultra wide band (UWB) receiver targeting a single-chip CMOS implementation for low data-rate applications like ad hoc wireless sensor networks. A non-coherent transmitted-reference (TR) receiver is chosen because of its small complexity compared to other architectures. After a brief recapitulation of the UWB fundamentals and a short discussion on the major differences between coherent and non-coherent receivers, we discuss issues, challenges and possible design solutions. Several simulation results obtained by means of a behavioral model are presented, together with an analysis of the trade-off between performance and complexity in an integrated circuit implementation. Copyright © 2005 John Wiley & Sons, Ltd.

## 1. Introduction

Ultra wide band (UWB) systems using short pulse modulation are nowadays considered a viable solution for short-range mobile communications. This technology features some advantages that make it particularly suitable for such type of systems (especially for indoor environments). More specifically, these features are the multipath resistance, the high time resolution, which allows to limit interference problems, and finally the capability to conjugate low emitted power, high data rate, and reuse of frequency bands already assigned to other kind of services. Generally speaking, UWB systems are characterized by the emission of periodically repeated short duration pulses that replicate a basic waveform. The period between two pulses may be constant or variable, depending on the adopted modulation scheme and on the multiple access technique the system employs. Typical modulation techniques are pulse amplitude modulation (2-PAM), pulse position modulation (PPM), and on-off keying (OOK). Another characteristic of all UWB devices is the short duration

of the transmitted waveform. Due to this short time span, the spectrum of a UWB signal may extend over several gigahertz, thus overlapping part of the bands used by narrowband systems. The high time resolution due to the transmission of narrow pulses, on the order of a few nanoseconds, enables accurate localization applications (few centimeters of resolution). This makes UWB appealing for low data-rate applications like *location-aware* wireless sensor networks [1].

Another important feature of UWB technology using baseband pulse modulation is the drastic simplification of the analog front-end of the receivers. After low noise amplification, the baseband analog signal can be directly converted into a digital format, enabling the exploitation of the flexibility (programmable/reconfigurable hardware), the noise robustness, and the low power consumption of digital CMOS integrated circuits.

The above-mentioned advantages might enable the integration into a single CMOS chip of all digital, analog, and radio-frequency functions. However, to the best of our knowledge, no fully integrated solutions are commercially available. Only few recent

\*Correspondence to: Mario R. Casu, CERCOM-Politecnico di Torino Corso Duca degli Abruzzi 24 I-10129 Torino, Italy.

†E-mail: mario.casu@polito.it

publications have the complete integration of a UWB transceiver in a CMOS technology as a final goal [2–5]. However, these works are limited in scope by the assumptions on the channel model. In some cases an AWGN channel is assumed, leading to receiver concepts based on matched filters, which are expected to perform poorly on a multipath channel like the UWB indoor one. Other approaches are based on Rake receiver architectures, which show adequate performance only when equipped with a large number of fingers. As a result, the power consumption may be too high for low power applications. Research in the field of low-complexity receivers (e.g., working in absence of channel state information [6–9]) is limited today to the system-level analysis, and publications describing detailed implementation of the overall receiver are still lacking. Some aspects of the receiver are discussed in Reference [10] (low noise amplifier) and References [2,11] (wideband A/D converters with reduced output dynamic). Finally, Reference [12] contains an interesting survey on UWB transceiver challenges, especially as far as the A/D converter is concerned.

The main contributions of this paper are the following:

- A fully digital implementation of a UWB non-coherent receiver based on the transmitted-reference principle [6] is discussed. The receiver is assumed to operate on a UWB indoor channel, modeled as in Reference [13].
- A limited complexity blind synchronization algorithm is presented and its performance analyzed, assuming dense multipath channel and no multiple access interference.
- The hardware implementation of both the A/D, and the baseband part of the TR receiver is discussed. In particular, we present a wide set of implementation possibilities for both the synchronization and the demodulation operations and we rank them according to the number of resources required for the implementation.

The paper is organized as follows: in Section 2, we analyze the principal structures proposed so far in the literature for UWB receiver, briefly discussing their complexity from an implementation viewpoint. Furthermore, we present the principal characteristics of a TR receiver at the system level. In Section 3, we compare digital and analog implementation solutions for the TR receiver and in Section 4, the synchronization and tracking algorithms are presented, together

with an analysis of the precision requirements of the A/D. The hardware integration challenges and solutions are discussed in Section 5. Finally, Section 6 summarizes the achievements of this work.

## 2. Coherent versus Non-Coherent Receivers

As already mentioned, the fine delay resolution of UWB signals provide a high robustness in dense multipath environments [14]. In case of perfect channel estimation and synchronization (coherent reception), absence of intersymbol and multiuser interference, it is well known [15] that a Rake receiver is the optimal detection scheme, in the sense that it minimizes the probability of error measured at the receiver end. However, if the knowledge of the channel is not ideal but, is acquired through a suitable estimation algorithm, this structure reduces to a heuristic approximation of the optimal detection scheme. Adopting the same terminology as in Reference [16], we will refer to this class of receivers as pseudo-coherent Rake receivers.

To fully exploit the channel diversity, a pseudo-coherent Rake receiver must be able to capture and track the energy associated with a high number of multipath replicas. In Reference [13], it is shown that the number of paths to be considered to reach the 85% of the overall energy can sometimes exceed 100. In addition, the radiation and propagation processes can act on the transmitted pulse as a filter whose characteristics vary from path to path. Therefore, the received signal can be seen as a train of distorted waveforms that often show little resemblance with the transmitted pulse [17,18].

Due to complexity constraints, only a small subset of the received replicas is expected to be selected and combined, a fact that justifies the performance loss illustrated in References [14,19,20] and Reference [6] for various selection combining methods. Furthermore, the presence of pulse distortion increases the complexity of the channel estimation algorithm [21,22], a topic that has not been fully analyzed in the literature yet. In general, it can be expected that the presence of complexity constraints, for example low power consumption, will impose sub-optimal solutions for the channel estimation process, with a consequent further performance loss.

A different approach to overcome all the above-mentioned disadvantages is based on the use of techniques that do not explicitly try to estimate the

channel and, possibly, require a less power-consuming receiver architecture. This class of receivers is denoted in the literature as ‘non-coherent’ [23–27], meaning that the demodulation is performed in absence of channel state information at the receiver.

These receivers allow to capture a large amount of the transmitted energy, despite the distortions and multipath propagation experienced by the signal through the transmission over the UWB channel. They represent, however, a sub-optimal solution, if compared to coherent receivers because of the adoption of a noisy signal as a reference waveform for the demodulation process. Among these receivers, it is worth mentioning, for their inherent architectural simplicity, the differential [9], the TR [6], and energy detector [25]. For a survey on non-coherent receivers for UWB applications and a comparison in terms of performance with pseudo-coherent Rake architectures, the interested reader is referred to Reference [28] and references therein.

In this work, we focus on the TR receiver; however, all the results presented in the next sections can be easily extended to the differential case due to the similarities between the two schemes. We consider the TR modulation format described in Reference [7]. According to this model, the transmitted UWB signals are grouped in blocks of length  $N$ . Each block consists of  $N_r$  reference pulses and  $N_d$  amplitude modulated ones. The channel is assumed to be static during one block and to change independently from block to block (block fading assumption). This assumption is expected to be valid for UWB systems operating in indoor environments, as the indoor channel is characterized by a rather long coherence time. In formulas, the received signal  $r(t)$  corresponding to one transmitted block is given by:

$$r(t) = \sum_{j=0}^{N_r-1} s(t - jT_f) + \sum_{j=0}^{N_d-1} b_j s(t - (j + N_r)T_f) + n(t) \quad (1)$$

where  $\{b_j\}_{j=-\infty}^{+\infty}$  is the sequence of information bits,  $s(t)$  is the received pulse obtained convolving the transmitted pulse and the channel impulse response of the channel,  $n(t)$  is AWGN with two-sided noise spectral density  $N_0/2$  and  $T_f$  is the frame time, larger than the delay spread of the channel to avoid inter-symbol interference. At the receiver each modulated pulse is correlated with an internal template waveform, obtained through an average over the reference pulses. Finally, a hard decision is employed.

It is clear that the performance of a TR receiver increases as  $N_r$  increases because of the improved quality of the internal reference (the average filters out the white noise); on the other hand, at the same time, the data-rate decreases. While a substantial improvement is observed passing from  $N_r = 1$  to  $N_r = 2$ , a further increase of complexity does not lead to a proportional bit error rate (BER) reduction. Thus, we chose for our benchmark implementation a TR receiver whose reference sequences consist of a few symbols (typically less than four).

From the BER perspective, if a packet transmission is assumed, the DR can be thought as a TR with  $N_r = 1$ . However, the latter requires the channel to be constant over all the packet, while for the differential receiver this assumption is relaxed to two adjacent symbols.

### 3. Analog and Digital Domain Processing

The choice of the TR receiver still leaves room for a number of alternatives concerning the partitioning between analog and digital circuits. Two extreme cases are shown in Figure 1. The fully analog receiver correlates the incoming data with an internal analogical template, and samples the correlation result. A hard decision consists in a simple 1-bit A/D conversion. The fully digital scheme converts the RF signal immediately after the LNA amplification and then performs a digital correlation and decision. The analog correlator is critical as it handles a very high frequency and large bandwidth signal, while the sampling is done at the pulse repetition rate that for low power low data-rate

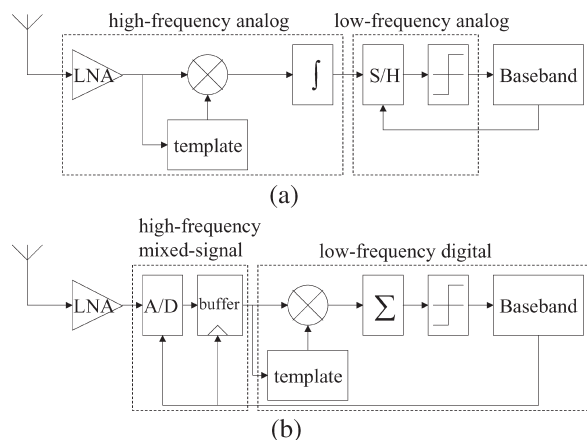


Fig 1. (a) Fully analog receiver; (b) Fully digital receiver.

applications means 0.1–1.0  $\mu\text{s}$ . The digital scheme is relatively simple from the signal processing viewpoint; on the other hand the A/D converter requires a very high sampling rate in order to meet the Nyquist criterion. Nonetheless, the analog template generation is even more difficult. In particular, analog delay lines are needed to align the template signal and the received pulse. The delay can be as long as the repetition period and, with the currently available technologies, it is not feasible to provide such kind of delays on-chip. Therefore, provided that a sufficiently fast A/D is available, the digital alternative is the most suitable for a single chip CMOS implementation. We will therefore refer to this fully digital scheme in the following. The impact of this choice on system parameters, like cost and power consumption may be relevant, especially in the light of the power consumption of fast A/D. Depending on the application constraints, like battery lifetime in wireless sensor networks other choices might be more suitable.

#### 4. Behavioral Model

In order to compare the hardware design with a reference model, we implemented a behavioral model of the receiver TR architecture as a C program. Within the C framework, synchronization and demodulation with early-gate fine tracking are fully characterized.

##### 4.1. Bandwidth

According to FCC regulation, UWB devices for indoor environments are allowed to operate in the bandwidth below 960 MHz and from 3.1 to 10.6 GHz. The first bandwidth is reserved for imaging systems, while the latter for communications and measurement devices. In this paper, we will ignore this disposition and assume to transmit the UWB pulse in the bandwidth below 960 MHz. A similar assumption was made in References [2] and [3]. More precisely, we will assume that the transmitted pulse has a bandwidth of 500 MHz, and the A/D converter samples at 1 GHz. However, the model and the hardware implementation alternatives that we present in the following sections are completely parametric and can be adapted to other bandwidths.

##### 4.2. Channel Model

The test-vectors of the channel response are built using the IEEE 802.15.3a model [13], which is based on a modification of the Saleh-Valenzuela's [32]. This

model takes into account the clustering phenomena observed in several UWB channel measurements [33]. According to Reference [13], the channel impulse response can be modeled as

$$h(t) = \sum_{l=0}^L \sum_{h=0}^H \alpha_{l,h} \delta(t - T_l - \tau_{l,h}) \quad (2)$$

where  $\{\alpha_{l,h}^{(k)}\}$  are the multipath gain coefficients,  $\{T_l^{(k)}\}$  and  $\{\tau_{l,h}^{(k)}\}$  represent the delay of the  $l$ th cluster and of the  $h$ th multipath ray relative to the  $l$ th cluster arrival time. The distribution of clusters and rays interarrival time is exponential. The average power delay profile shows a double exponential decay (for cluster average power and for rays average power in each cluster), and the fading statistics is lognormal. Finally, the sign of each multipath replica is either positive or negative, with the same probability.

The maximum delay spread of the channel is equal to 100 ns (or 100 samples). A white Gaussian noise is added to the samples with variable SNR as well as a narrowband interferer (optional). A single user scenario is considered, assuming that some form of MAC protocol (TDMA or CSMA) is employed to limit multiple access interference.

#### 4.3. Synchronization

##### 4.3.1. Introduction

The purpose of the synchronization algorithm is to recover, at the receiver, the timing information required to perform the demodulation and the detection of the transmitted information. In our case, timing recovery may be conveniently viewed as a two-part process, the first consisting of estimating the time instant representing the starting point of the pulse (*symbol-level* synchronization) and the second aiming at identifying the position of the first reference pulse in each transmitted frame (*block-level* synchronization). Therefore, our aim is to accomplish these two tasks with an algorithm which offers the best compromise between complexity and performance.

Interestingly, if one approaches the problem of timing recovery from a theoretical point of view, with the objective to derive the best strategy according to some criterion (e.g., the minimization of the Euclidean distance between transmitted and estimated signal), it turns out that the synchronization problem is strictly related to the channel estimation one. This is shown for example in Reference [31], where a least

square algorithm for the joint estimation of what we called symbol-level and block-level synchronization is presented in a slightly different context. The aforementioned algorithm assumes that training sequences are sent and achieves synchronization through an optimization over a space of dimension 2. It is worth noting that the algorithm produces as a by-product an estimation of the impulse response of the channel. The strategy described in Reference [31], albeit characterized by good performance, is of scarce utility to our purposes because of its high computational complexity.

In the next section we will, instead, propose an algorithm, suboptimal with respect to the one presented in Reference [31], which performs symbol synchronization based on heuristic considerations. Its performance is expected to be rather poor, if compared to more sophisticated approaches. However, its limited complexity makes it an interesting solution for our particular context.

#### 4.3.2. Symbol-level synchronization

We assume that each frame contains  $N_f$  samples, where  $N_f$  is obtained by dividing the frame length  $T_f$  by the sampling time, and that the number of significant samples per pulse is  $N_w$ . Our algorithm is based on the sliding correlation principle. In formulas, given the vector of received samples  $\mathbf{r} = (r[1], r[2], \dots, r[M])$ , the  $N_f$  correlation values  $S_{\text{bit}}(k), k \in [1, N_f]$  and the starting sample  $k_{\text{max}}$  are computed as follows:

$$S_{\text{bit}}(k) = \sum_{j=1}^{N_p} \left| \sum_{i=k}^{k+N_w-1} r[i] \times r[i+jN_f] \right| \quad (3)$$

$$k = 1, 2, \dots, N_f$$

$$k_{\text{max}} = \arg(\max_{k \in [1, N_f]} (S_k)) \quad (4)$$

where  $N_p$  indicates the number of pulses to be correlated with the first one. In words, for each sample of the first analyzed period, the algorithm computes an estimate of its likelihood of being a potential starting sample. Then the sample with maximum likelihood is chosen. A threshold can be set to discard frames which contain only noise. The following example in Figure 2 demonstrates this simple algorithm. Let us consider the discrete signal in Figure 2 and the stream of bits  $\{\dots, +1, -1, -1, +1, \dots\}$ . Since  $N_f = 5$  the

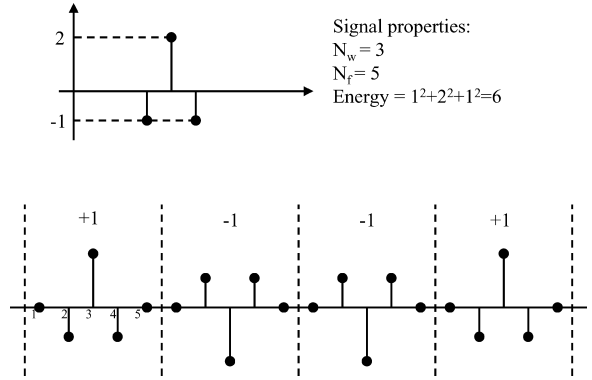


Fig. 2. Example of bit stream for symbol-level synchronization.

algorithm calculates five estimates. Suppose to start at sample number 1. Its likelihood of being the starting sample is calculated correlating the first  $N_w = 3$  samples at positions 1, 2, 3 with the next three samples at position 6, 7, 8 (that is  $N_f = 5$  samples later), and taking the absolute value:

$$(0 \cdot 0) + (-1 \cdot +1) + (2 \cdot -2)$$

$$= -5 \Rightarrow s_1 = |-5| = 5$$

In the same way, the estimate for sample 2 is calculated correlating three samples at positions 2, 3, 4 with samples 7, 8, 9:

$$(-1 \cdot +1) + (2 \cdot -2) + (-1 \cdot +1)$$

$$= -6 \Rightarrow s_2 = |-6| = 6$$

By repeating for the first five samples this procedure, the computed estimates are:

$$\mathbf{s} = (5, 6, 5, 1, 1)$$

Since the maximum is  $s_2$ , sample 2 is correctly chosen as a starting sample. The signal has been correctly acquired.

This simple algorithm works optimally for a noise-free signal. On noisy channels, its performance can be improved by not limiting the correlation to the adjacent pulse but considering also the following  $N_p - 1$  pulses ( $N_p$  indicates the number of pulses to be correlated with the first one). The trivial example above, corresponds to the case when  $N_p = 1$ . Of course, increasing  $N_p$  comes at the cost of increasing acquisition time and/or hardware

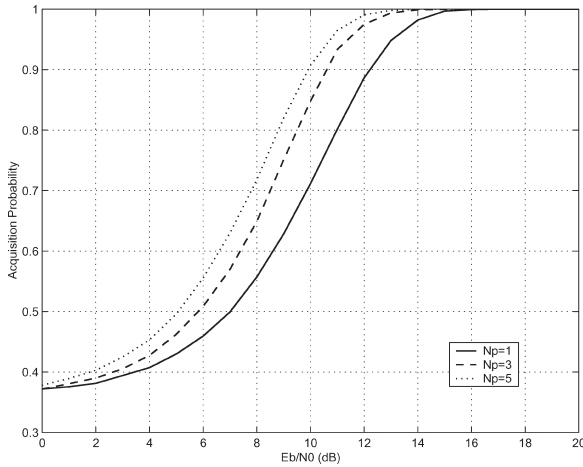


Fig. 3. Symbol-level acquisition probability. The results are averaged over  $10^6$  realizations of the input signal and channel.

complexity, as shown later on. For example, when  $N_p = 2$ , the estimate  $s_2$  becomes

$$\begin{aligned}
 s_2 &= |(-1 \cdot +1) + (2 \cdot -2) + (-1 \cdot +1)| + |(-1 \cdot -1) \\
 &\quad + (2 \cdot 2) + (-1 \cdot -1)| \\
 &= |-6| + |6| = 12
 \end{aligned}$$

Using the IEEE 802.15.3a model, we computed the performance of this simple algorithm varying  $E_b/N_0$  and channel realizations. We assume a signal acquired if the starting sample  $k_{max}$  is *at most* 50 samples (i.e., half the pulse duration  $N_w = 100$ ) early or late with respect to the ideal value, that is  $|k_{max} - k_{ideal}| \leq \frac{N_w}{2}$ . The rationale behind this simple criterion is that a large part of the energy still has to be contained in the  $N_w$  samples following the starting sample  $k_{max}$ . Figure 3 reports the simulation results in terms of acquisition probability as a function of  $E_b/N_0$  and  $N_p$ . It is clear from the curves the improvement in acquisition probability for  $N_p > 1$ , but the small difference between  $N_p = 3$  and  $N_p = 5$  suggests that the increase of complexity for  $N_p > 3$  is not justified.

### 4.3.3. Block-level synchronization

Once the signal timing at the symbol-level has been acquired, a block-level synchronization step, which reveals the position of the reference bits in the transmitted frames, is needed. This step can be performed in a blind way, employing an algorithm similar to the one described for the symbol-level case, or through the transmission of a training sequence.

## 4.4. Early-Late Demodulation and Tracking

After synchronization is achieved, the receiver demodulates the  $N_d$  data symbols contained in each block. A template vector  $\mathbf{t}$  is constructed averaging over the reference sequence  $\mathbf{r} = [r_1, r_2, \dots, r_{N_r}]$ , where with  $\mathbf{r}$  we denote the vector containing the discrete samples associated to the reference part of the transmitted block. The reference sequence can be multiplied by a reference pattern  $\mathbf{p} = [p_1, p_2, \dots, p_{N_r}]$ ,  $p_i \in \{-1, 1\}$ , if a pattern known to the receiver is employed to modulate the reference pulses. In formulas:

$$\mathbf{t} = \frac{1}{N_r} \sum_{i=1}^{N_r} p_i \mathbf{r}_i$$

The demodulated data  $d_j \in \{-1, 1\}$  are then obtained by taking

$$d_j = \text{sign}(\mathbf{t} \cdot \mathbf{d}_j^T), \quad j = 1, 2, \dots, N_d$$

with  $\mathbf{d} = [d_1, d_2, \dots, d_{N_d}]$ , containing the discrete samples of the data part of the transmitted block.

The fine tracking is implemented by correlating the data  $\mathbf{d}$  with the on-time version of the reference and with the early and late versions, that is anticipated and delayed by one sample, respectively. Then the demodulated data are retrieved from the sign of the correlation that has the maximum absolute value among

$$\begin{aligned}
 c_{\text{early}} &= \mathbf{t}_{\text{early}} \cdot \mathbf{d}_j^T \\
 c_{\text{on-time}} &= \mathbf{t}_{\text{on-time}} \cdot \mathbf{d}_j^T \\
 c_{\text{late}} &= \mathbf{t}_{\text{late}} \cdot \mathbf{d}_j^T
 \end{aligned}$$

When the early (late) correlation prevails, the starting sample for the next symbol is anticipated (delayed) by one position.

Figure 4 reports the error probability evaluated averaging over  $10^6$  realizations of the input signal and channel. Two curves are reported, the ‘No Synch’ case, where the timing is ideally acquired, and the ‘Synch’ case, where the timing is given by a previous run of the symbol-level synchronization algorithm. We suppose perfect block-level synchronization in both cases. It can be noticed that the second curve has a slightly larger error probability, but it tracks well the first one.

## 4.5. A/D Precision Requirements

We estimated the requirements of the A/D in terms of precision by means of simulations. At first, we

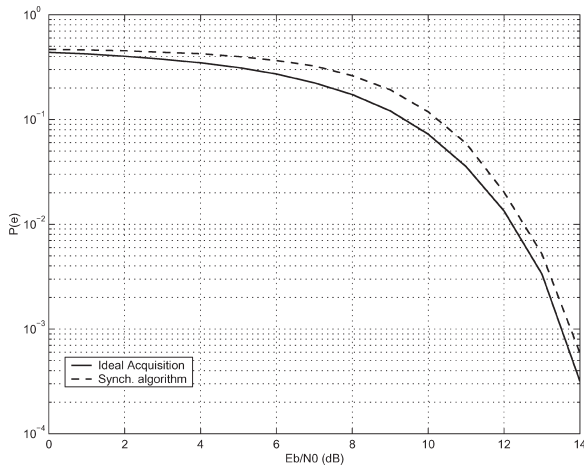


Fig. 4. Error probability: ideal bit-level acquisition compared to the algorithm described in Equation (3).

evaluated the probability density function of the signal amplitude using the IEEE 802.15.3a model, in order to select the best quantization strategy. The result is reported in Figure 5. All channel realizations in our model are normalized such that their energy is equal to one. This is equivalent to assume that the receiver employs an automatic gain control (AGC) circuit, which guarantees a constant input power to the A/D. This normalization removes from the analysis the effect of, for instance, different TX-RX distances which, in turn, would imply a different use of the A/D input range and a consequent different resolution in terms of quantization. We noticed that 99% of the area of the pdf is contained in  $[-0.5, 0.5]$ . Conse-

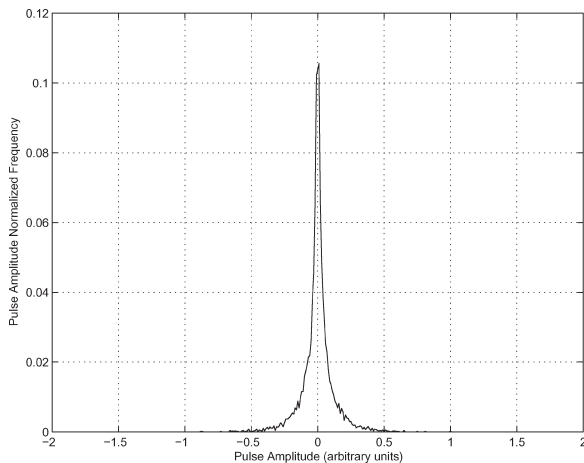


Fig. 5. Amplitude statistical distribution for the IEEE 802.15.3a channel model.

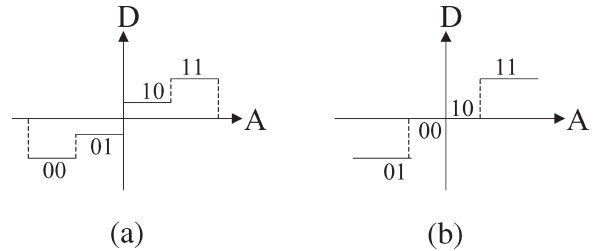


Fig. 6. Two possible quantization mappings.

quently, we set saturation thresholds to  $s_{\max} = \pm 0.5$ . We tested two quantization mappings, as shown in Figure 6. The first mapping fully uses the  $2^n$  levels for  $n$  bits but does not have a unique code for the small values around zero. The second one has one code for values around zero but, on the other hand, has  $2^n - 1$  different levels, thus not fully exploiting the binary code, because two codes represent  $\pm 0$ . The first is suited for the offset-binary representation of the numbers, while the second one for the sign-magnitude coding, as is shown in the figure. The first one is the only choice in case of  $n = 1$ . The second one suppresses the so-called 'idle noise' that leads to useless computation in the digital back-end. Thus, mapping 2 is more suited for a low power digital implementation. An example of this phenomenon is shown in Figure 7. From a performance perspective, the two methods are equivalent. In Figure 8 the results of the simulations using mapping 1 are reported. The floating point case is compared to several quantization cases from 1 to 4 bits. Similar results, not reported for brevity, have been obtained using mapping 2. Therefore, we can

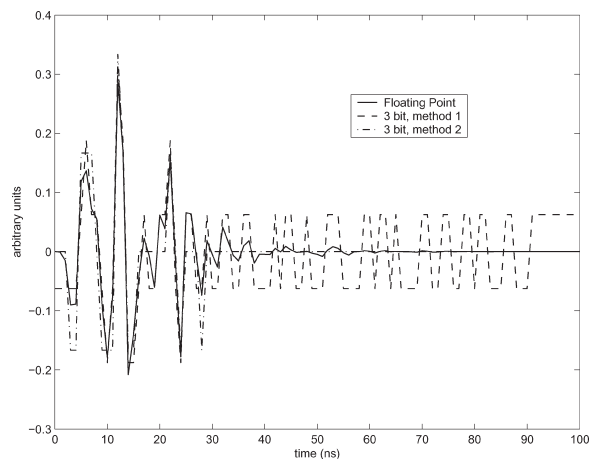


Fig. 7. Effect of the idle noise and its suppression using a proper coding.



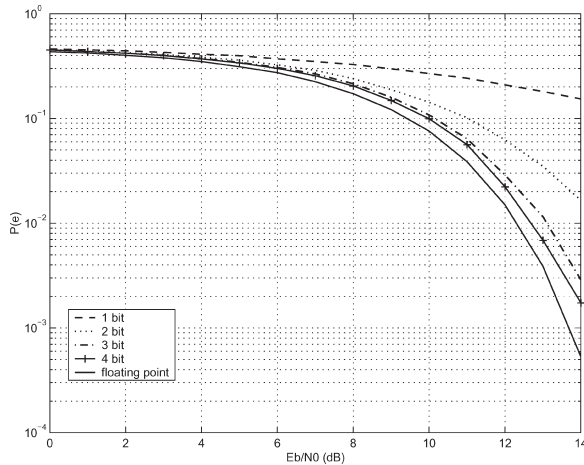


Fig. 8.  $P(e)$  parametric with  $n$  quantization bits.

conclude that none of them is preferable in terms of performance. Other parameters like ease of implementation or power consumption can be used for further discriminating between the two. Analysing the results, we notice that the 1-bit and 2-bit quantizations have poor performance. The 4-bit curve is close to the floating point one, and can be considered sufficient for implementation purposes. The fact that a small number of bits is sufficient, is relevant from the implementation perspective: the actual precision, in terms of ‘effective number of bits,’ reduces significantly for ultra-fast, high-bandwidth A/D [29] like the ones needed for the UWB system simulated here.

### 5. Hardware Integration: Challenges and Solutions

We will not discuss the integration issues of LNA and filters in a mixed-signal CMOS chip. We will focus, instead, on the baseband part of the receiver with a few hints regarding the A/D converter. The integration

of this part of the receiver is very critical; the precision requirements, as stated in Section 4, are not an issue, the main problem being the very high bandwidth. Some proposals [2,3,12,30] suggest the use of a bank of  $N_{ad}$  parallel A/D converters, each with sampling frequency  $f_s/N$  and delay between two consecutive converters of  $\delta T = 1/f_s$ . In Reference [2], each pulse repeats every  $T_{rep} = 1/f_{rep} = 1/62.5 \times 10^6 = 16$  ns, the sampling period is  $T_s = 1/f_s = 1/2 \times 10^9 = 0.5$  ns (2 GHz), therefore  $N_{ad} = 16/0.5 = 32$  parallel A/D converters.

This solution is mandatory, nonetheless it still presents difficulties:

- Even if the sampling frequency is reduced of  $1/N_{ad}$ , each A/D must have a bandwidth equal or larger than the UWB pulse bandwidth.
- The clock generation is critical, since each clock phase must maintain a stable and small relative delay. The jitter must be kept well under control.
- If the duty-cycle of the pulse is very small ( $T_{pulse} \ll T_{rep}$  and so  $N_w \ll N_f$  according to the notation used above), like for low data-rate applications, the number of parallel A/D and so the number of clock phases is too high for area/power constrained silicon implementations.

While the first two points require appropriate circuit and technology solutions, the third one can be faced by a suitable arrangement of the A/D registers. The example in Figure 9 is the case of  $N_f = 1000$ ,  $N_{ad} = 10$ . The outputs of the A/D are organized in shift-registers in order to avoid the memory input control along with its complex and slow decoder. Only one window of  $N_f$  samples is registered in the case of Figure 9. If more than one window is necessary, as for the previously described synchronization algorithms, the shift-register size can be easily extended.

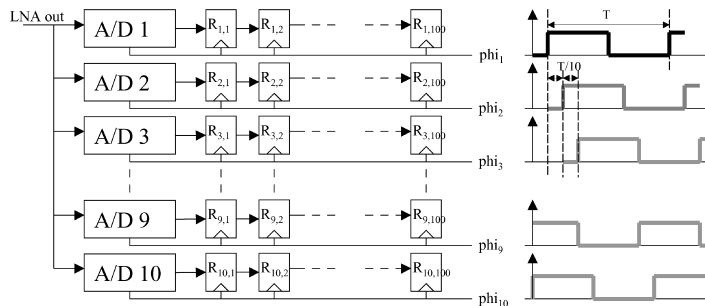


Fig. 9. A possible arrangement of parallel A/D for the case  $N_f = 1000$ .

Table I. Complexity for the parallel implementation according to Equation (3).

Resource type	Number of resources
MEM	$(N_p + 1)N_f + N_w - 1$
MUL	$N_w N_p N_f$
ADD	$N_w N_p N_f$

In the following, we will discuss the architecture of the receiver for the implementation of the synchronization and demodulation algorithms.

### 5.1. Symbol-Level Synchronization

Many implementations ranging from parallel to sequential are possible.

(1) *Parallel Implementation:* We assume to have sufficient resources for executing the algorithm of Equation (3) in one clock cycle (or few clock cycles, in case of pipelining). The complexity requirements obtained by simply unrolling the loops are reported in Table I, where MEM are memory elements, MUL stands for multipliers, and ADD for adders. The size in bits of each memory element corresponds to the number of quantization bits of the A/D. Many multiplications are unnecessarily repeated many times. For instance, for  $k = 0$  and  $k = 1$  (and with  $j = 1$ ) the sequence of  $N_w$  products are the following

$$\begin{aligned}
 k = 0: & r[0]r[N_f], r[1]r[1+N_f], r[2]r[2+N_f] \dots \\
 & r[N_w-2]r[N_w-2+N_f], r[N_w-1]r[N_w-1+N_f] \\
 k = 1: & r[1]r[1+N_f], r[2]r[2+N_f], r[3]r[3+N_f] \dots \\
 & r[N_w-1]r[N_w-1+N_f], r[N_w]r[N_w+N_f]
 \end{aligned}$$

and it is clear that  $N_w - 1$  products are common to both sequences. Therefore, the actual number of different products is much less than  $N_w N_p N_f$  and it is not difficult to compute. For  $k = 0$ , we have  $N_w$  different products. For  $k = 1$ , we have to add another product ( $r[N_w]r[N_w + N_f]$ ), for  $k = 2$  another product ( $r[N_w + 1]r[N_w + 1 + N_f]$ ), and so on until  $k = N_f$ . This must be repeated for each  $j \in [1, N_p]$ . As a result, the actual number of MUL is  $N_p(N_w + N_f - 1)$ . In a similar way, the number of ADD can be reduced by considering that many additions are repeated many times. However, it is more difficult to evaluate the exact (and minimum) number of additions. The example in Figure 10, a simplified case with  $N_f = 6$ ,  $N_w = 5$ ,

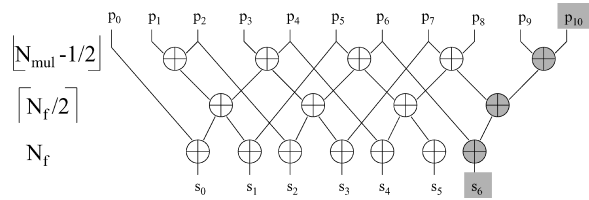


Fig. 10. Example of parallel implementation.

$N_p = 1$  ( $MUL = 6 + 5 - 1 = 10$ ), shows one possibility. Products are denoted by  $P_i$ ,  $i = 0, \dots, MUL - 1$ , while  $S_i$ ,  $i = 0, \dots, N_f - 1$  are the outputs of the correlation step. In this case, the total number of adders is

$$\lfloor N_{mul} - 1/2 \rfloor + \lceil N_f/2 \rceil + N_f = 4 + 3 + 6 = 13$$

Shadowed operators refer to the case when  $N_f = 7$  ( $N_{mul} = 11$ ). We expect a number of adders equal to

$$\lfloor N_{mul} - 1/2 \rfloor + \lceil N_f/2 \rceil + N_f = 5 + 4 + 7 = 16$$

which is consistent with the figure. If  $N_p > 1$  the number of adders has to be multiplied by this factor. The overall complexity of this implementation is summarized in Table II.

After the evaluation of the correlations, the algorithm proceeds with the search of the maximum. The parallel architecture is simply a tree of comparators, as Figure 11 shows.

(2) *Sequential Implementation:* A possible sequential implementation takes advantage of the previous observation concerning the reuse of products and sums. We can rewrite for convenience the estimates as follows

$$\begin{aligned}
 S_{bit}(k) &= \sum_{j=1}^{N_p} \left| \sum_{i=k}^{k+N_w-1} r[i] \cdot r[i + jN_f] \right| \\
 &= \sum_{j=1}^{N_p} \left| \sum_{i=k}^{k+N_w-1} p(i, j) \right| = \sum_{j=1}^{N_p} |s(k, j)|
 \end{aligned} \tag{5}$$

Table II. Actual complexity after loop simplification.

Resource type	Number of resources
MEM	$(N_p + 1)N_f + N_w - 1$
MUL	$N_p(N_f + N_w - 1)$
ADD	$N_p(\lfloor N_f + N_w - 3/2 \rfloor + \lceil N_f/2 \rceil + N_f)$

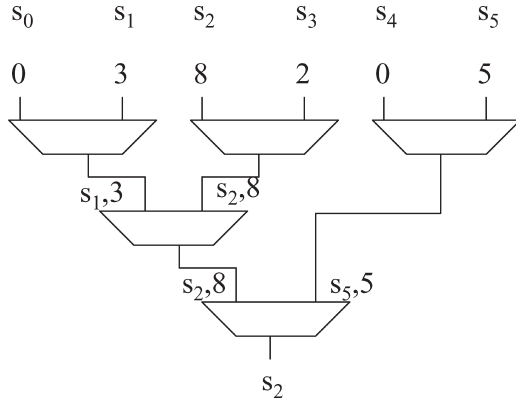


Fig. 11. Bank of parallel comparators for the maximum evaluation.

and then write a recursive formula

$$s(k,j) = s(k-1,j) - p(k-1,j) + p(k+N_w-1,j) \quad (6)$$

with  $s(0,j)$  computed as in Equation (5). It is clear that if Equation (6) is evaluated at every clock cycle, only 1 MUL, 1 ADD, and 1 SUB (subtractor) are needed, if  $N_p = 1$ . When  $N_p > 1$ , another adder is required for the outer sum  $\sum_{j=1}^{N_p}$ . Table III summarizes the complexity of this simple implementation.

The computation of the maximum is performed this time through a sequence of  $N_f$  comparisons. They are implemented using one comparator and one additional register for saving the temporary maximum and its index.

- (3) *Execution Time Considerations*: The first important point to understand is if the synchronization algorithm needs realtime execution. If not, we can certainly relax the architectural constraints, especially for the sequential case. However, since the memory size for storing incoming data is limited, we cannot save all inputs and will lose data while executing the synchronization. At the same time, we need to compute the number of lost data, so that when we pass from synchronization to tracking we still point to the correct starting sample. In

Table III. Complexity of the sequential implementation.

Resource type	Number of resources
MEM	$(N_p + 1)N_f + N_w - 1$
MUL	1
ADD	1 if $N_p = 1, 2$ if $N_p > 1$
SUB	1

the realtime case, on the other hand, while the latter potential issues are avoided, the execution speed requirement could be really stringent. The limit for the execution time in the non-realtime case depends on the design choices, while for the realtime case we can derive material limits. Suppose we start filling the input memory at time 0. Every  $T_s$  seconds a new sample is registered. The algorithm can start as soon as the products  $P_i$ ,  $i = 0, \dots, \text{MUL} - 1$ , (see Figure 10) can be computed. For the parallel case, we have to wait until time  $(N_p + 1)N_f + N_w - 1$ , when the last sample needed for the computation of the product is stored. In order not to lose data, we can add a few registers to save the last  $N_f - N_w$  samples of the last symbol used in the synchronization algorithm. These samples are not used for the computation of the  $\{P_i\}$  products, but can be needed for the following demodulation step. The parallel execution has to be completed in no more than  $(N_f - N_w)T_s$  seconds. Suppose we run the digital circuit at clock frequency  $f_{ck}$ ; thus, we need to complete the execution in  $N_{ck} = (N_f - N_w)T_s f_{ck}$  clock cycles. A numerical example may help for sensitizing to the orders of values. Suppose

- $N_w = 100$ ,  $N_f = 1000$ ,  $f_s = 1$  GHz,  $f_{ck} = 100$  MHz.

Then, we are left  $(1000 - 100) \times 10^{-9} \times 100 \times 10^6 = 90$  clock cycles, which is abundant for the parallel implementation. Therefore, we can reduce the clock frequency or the size of the additional memory. Many options can be exploited depending on the specs.

The sequential algorithm can start as soon as the first product is available, at time  $(N_f + 1)T_s$ . If the same assumption of completing the storage of the  $N_p + 2$ th bit holds, the time left for computation is  $(N_p + 1)N_f T_s$ . The clock cycles needed for this aim are about  $N_p(N_f + N_w)$ . Therefore, the following lower bound holds:

$$f_{ck} \geq \frac{N_p(N_f + N_w)}{(N_p + 1)N_f T_s} > \frac{N_p}{N_p + 1} f_s$$

For low sampling rates (i.e., less than 2 GHz) this is still feasible in CMOS, even if not straightforward. For higher rates, other solutions must be conceived, like increasing the input buffer size or relaxing the 'no data lost' criterium.

- (4) *Mixed Parallel/Sequential Architecture*: Intermediate solutions between the two extreme

parallel and sequential implementations are possible. One consists of applying sequentially, for  $N_p$  times, a parallel search limited to a correlation between two windows. In general, whatever is the strategy used for computing the correlation the execution time limit under the requirement of not losing any data is  $(N_p + 2)N_f T_s \leq N_p N_{\text{corr}} T_{\text{ck}}$ , where  $N_{\text{corr}}$  is the number of clock cycles needed for correlating two windows. For example, at 2 GHz sampling rate and 100 MHz clock frequency, with  $N_f = 1000$  and  $N_p = 4$ , the number of clock cycles for the correlation is  $N_{\text{corr}} = 75$ . This cannot be achieved with a totally sequential architecture that requires  $N_f + N_w > 1000$  cycles for each correlation. On the other hand, a parallel solution could be too costly and unnecessary fast. A mixed parallel-sequential solution is more suited for this case.

## 5.2. Demodulation and Tracking

The demodulation consists of two phases, template calculation and computation of the correlation between received signal and template. Again, parallel and sequential solutions can be implemented.

(1) *Parallel Solution:* As it was previously discussed, the template is built as an average, sample by sample, of the first  $N_r$  reference symbols in a frame. Since  $N_w$  samples are used out of the  $N_f$  ones that constitute each symbol, the number of adders for a parallel solution is  $N_w(N_r - 1)$ . The number of different products, and also of MUL for a parallel implementation, is  $N_w N_d$ . All these products have to be summed sample by sample. Thus, the number of adders is  $(N_w - 1)N_d$ . The memory requirement is simply  $(N_r + N_d)N_w$ . Table IV summarizes the resource demand, while a schematic with  $N_r = 2$  is reported in Figure 12.

If an early-late architecture is implemented, the multiply-accumulate structure highlighted in figure will be replicated three times.

(2) *Sequential Solution:* The sequential implementation consists of an adder for the template computation and a multiply-accumulate (MAC) unit for

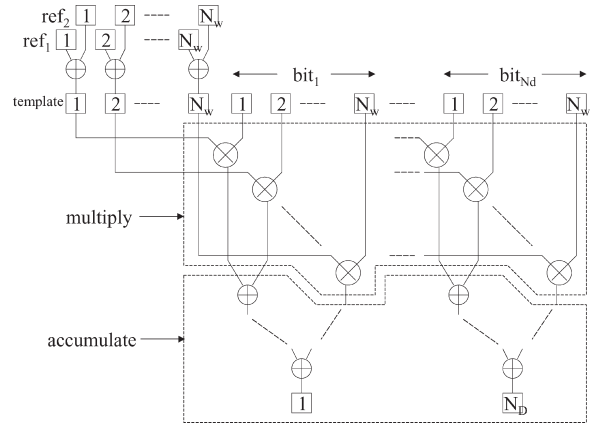


Fig. 12. Schematic of a parallel demodulator.

the correlation. For the implementation of the early-late tracking, three MACs are used. Figure 13 shows an example of this sequential structure.

(3) *Execution Time:* While the acquisition step may be executed off-line, if allowed by the system specs (at the cost of losing some data, see the previous section), the demodulation phase is necessarily realtime. As a consequence, it is easy to derive the execution time constraints. As far as the parallel execution is concerned, the demodulation starts after the last sample of the  $N_d$ -th bit is received. If we are not allowed to use additional buffers for the incoming new data, we are still left with  $(N_f - N_w)T_s$  seconds before the first sample of the new reference arrives. If  $N_{\text{ck}}$  cycles are necessary to complete the execution, the constraint for the clock frequency is the following:

$$N_{\text{ck}} \leq (N_f - N_w)T_s f_{\text{ck}}$$

For example, with  $N_f = 1000$ ,  $N_w = 100$ ,  $f_{\text{ck}} = 100$  MHz,  $f_s = 1$  GHz, we get  $N_{\text{ck}} \geq 90$  which is superabundant for the parallel implementation, meaning that there is room for further optimization.

As for the sequential solution, the number of clock cycles can be easily derived:  $N_r N_w$  for the template and  $N_d N_w$  for the demodulation. Therefore, the constraint is the following:

$$(N_r + N_d)N_w T_{\text{ck}} \leq (N_r + N_d)N_f T_s$$

or, in other words,

$$f_{\text{ck}} \geq \frac{N_w}{N_f} f_s$$

Table IV. Complexity of a parallel solution.

Resource type	Number of resources
MEM	$(N_r + N_d)N_w$
MUL	$N_w N_d$
ADD	$(N_w - 1)N_d$

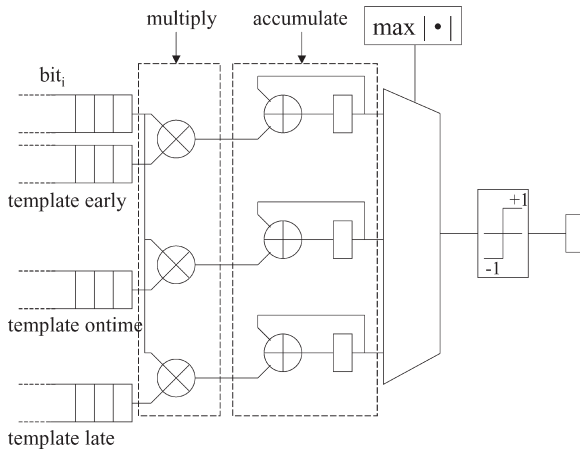


Fig. 13. Example of sequential demodulation architecture.

This value can be relatively high. It amounts to 100 MHz in the previous example, a quite relaxed value compared with  $f_s$ .

- (4) *Intermediate Solution*: Mixed parallel-sequential solutions can be a good compromise between area and clock frequency. The easiest is to duplicate the structure in Figure 13 and let it work on two different symbols, symbol  $2i$  and symbol  $2i + 1$ , or on even and odd samples of the same symbol. By  $n$ -uplicating the sequential structure, we tend to the parallel implementation. Therefore, the actual complexity lies between the two previously presented architectures.

In general, the time necessary to complete the demodulation must respect the following constraint:

$$(N_t + N_d N_{\text{dem}}) T_{\text{ck}} \leq (N_d + N_r) N_f T_s$$

where  $N_t$  is the time necessary for calculating the template and  $N_{\text{dem}}$  is the number of clock cycle necessary for demodulating one symbol. There is room left for performance optimizations and for meeting the frequency constraint. It is likely that a typical system will be then implemented with a mixed parallel-sequential architecture.

### 5.3. RTL Models and Simulations

The digital back-end of two TR receivers has been designed at the register-transfer level (RTL), that is from the A/D output to the demodulated data. The first receiver is based on a high performance parallel implementation, while the second one is characterized by a minimum area sequential implementation. Both were characterized using VHDL hardware description language and tested on the IEEE 802.15.3a channel.

We verified by simulation a perfect agreement with the performance predicted by the behavioral model written as a C program. The sequential implementation has been also synthesized and mapped on a field-programmable gate array (FPGA) device (Xilinx XCV600) and successfully tested on the field with a board equipped with that FPGA device. At present, we are working on the synthesis and Place-and-Route on a silicon 0.13  $\mu\text{m}$  technology.

## 6. Conclusions

In this paper, we discussed the implementation challenges for an UWB transmitted-reference receiver. A detailed analysis of the performance, as a function of the many system and environmental parameters has been done. The issues and challenges for an integrated circuit implementation have been discussed. An analysis of the complexity and architectural constraints for meeting the system level constraints is also proposed in the paper. The architecture described has been implemented and verified on a FPGA equipped board. An application specific integrated circuit (ASIC) version on an advanced CMOS 0.13  $\mu\text{m}$  technology is currently under development.

## Acknowledgment

This work has been sponsored by MIUR (Italian Ministry of Education and Research) under the project PRIMO.

## References

1. IEEE 802.15 WPAN Low rate Alternative PHY Task Group 4a (TG4a), [Online]: [www.ieee803.org/15/pub/TG4a.html](http://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 *Proceedings of the IEEE CAS Workshop on Wireless Communications and Networking*, September 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, Lee FS, Wentzloff DD, Newaskar PP, Powell JD, Chandrakasan AP. Digital architecture for an ultra-wideband radio receiver. In *Proceedings of VTC Fall 2003*, Orlando FA, October 2003.
5. Blazquez R, Newaskar PP, Lee FS, Chandrakasan AP. A baseband processor for pulse ultra-wideband signals. In *Proceedings of the IEEE Custom Integrated Circuits Conference*, Orlando, October 2004.
6. Choi JD, Stark WE. Performance of ultra-wideband communications with suboptimal receivers in multipath channels. *IEEE Journal on Selected Areas in Communications* 2002; 20(9): 1754–1766.

7. Franz S, Mitra U. On optimal data detection for uwb transmitted reference system. In *Proceedings of Global Telecommunications Conference GLOBECOM 2003*, S. Francisco, USA, 1–5, December 2003, vol. 2, pp. 764–768.
8. Hoctor RT, Tomlinson HW. Delay-hopped transmitted-reference RF communications. In *Proceedings of the IEEE Conference on Ultra Wideband Systems and Technologies 2002, Digest of Papers*, Baltimore, USA, 2002, pp. 265–270.
9. Ho M, Somayazulu S, Foerster J, Roy S. A differential detector for an ultra-wideband communications system. In *Proceedings of VTC Spring 2002*, vol. 4, pp. 1896–1900.
10. Bevilacqua A, Niknejad A. An ultra-wideband CMOS LNA for 3.1 to 10.6 GHz wireless receivers. In *Proceedings of the International Solid State Circuit Conference*, 2004.
11. Newaskar PP, Blazquez R, Chandrakasan AP. A/D precision requirements for an ultra-wideband radio receiver. In *Proceedings of SIPS 2002*, San Diego, CA, October 2002, pp. 270–275.
12. Orndorff AM. Transceiver Design for Ultra-Wideband Communications. 2004 M.Sc. thesis, Virginia Polytechnic Institute and State University (advisor: Buehrer RM).
13. Foerster J. Channel modeling sub-committee report final, *IEEE P802.15 02/490r1 SG3a*, February 2002.
14. Cassioli D, Win M, Valataro F, Molisch AF. Performance of low-complexity rake reception in a realistic uwb channel. In *Proceedings of the International Conference on Communications ICC*, vol. 2, New York, USA, 2002, pp. 763–767.
15. Proakis JG, Salehi M. *Communication Systems Engineering*. Prentice Hall: Upper Saddle River, New Jersey, 2002.
16. Colavolpe G, Raheli R. Noncoherent sequence detection. *IEEE Transactions on Communications* 1999; **9**(9): 1376–1385.
17. Cramer RJ, Scholtz RA, Win MZ. Evaluation of an ultra-wideband propagation channel. *IEEE Transactions on Antennas and Propagation* 2002; **50**(5): 561–570.
18. Qiu RC. A study of the ultra-wideband wireless propagation channel and optimum uwb receiver design. *IEEE Journal on Selected Areas in Communications* 2002; **20**(9): 1628–1636.
19. Mielczarek B, Wessman M, Svensson A. Performance of coherent uwb rake receivers using different channel estimators. In *Proceedings of the International Workshop on Ultra Wideband Systems (IWUWBS)*, Oulu, Finland, 2003.
20. Zhang J, Kennedy RA, Abhayapala TD. Performance of rake reception for ultra wideband signals in a lognormal-fading channel. In *Proceedings of the International Workshop on Ultra Wideband Systems (IWUWBS)*, Oulu, Finland, 2003.
21. Lottici V, D'Andrea A, Mengali U. Channel estimation for ultra-wideband communications. *IEEE Journal on Selected Areas in Communications* 2002; **20**(9): 1638–1644.
22. Carbonelli C, Mengali U, Mitra U. Synchronization and channel estimation for uwb signals. In *Proceedings of the Global Telecommunications Conference GLOBECOM 2003*, S. Francisco, USA, 1–5, December 2003, vol. 2, pp. 764–768.
23. Weisenhorn M, Hirt W. Robust noncoherent receiver exploiting UWB channel properties. In *Proceedings of Joint UWBST & IWUWBS*, Kyoto, Japan, May 2004, pp. 156–160.
24. Durisi G, Benedetto S. Performance evaluation and comparison of different modulation schemes for uwb multiaccess systems. In *Proceedings of the International Conference on Communications ICC*, vol. 3, Anchorage, USA, 2003, pp. 2187–2191.
25. Souilmi Y, Knopp R. On the achievable rates of ultra-wideband ppm with non-coherent detection in multipath environments. In *Proceedings of the International Conference on Communications ICC*, vol. 5, Anchorage, USA, 2003, pp. 3530–3534.
26. Rabbachin A, Oppermann I. Synchronization analysis for UWB systems with a low-complexity energy collection receiver. In *Proceedings of Joint UWBST & IWUWBS*, Kyoto, Japan, May 2004, pp. 288–292.
27. Stoica L, Tiuraniemi S, Repo S, Rabbachin A, Oppermann I. Low complexity UWB circuit transceiver architecture for low cost sensor tag systems. In *Proceedings of PIRMC*, 5–8 September 2004, Barcelona, Spain, Vol. 1, pp. 196–200.
28. Durisi G, Benedetto S. Comparison between coherent and non-coherent receivers for uwb communications. *EURASIP Journal on Applied Signal Processing—UWB State of the Art*, December 2004.
29. Walden RH. Analog-to-digital converter survey and analysis. *IEEE Journal on Selected Areas in Communications* 1999; **17**(4): 539–550.
30. Helal D, Rouzet P. STMicroelectronics proposal for IEEE 802.15.3a Alt PHY, [Online]: [http://grouper.ieee.org/groups/802/15/pub/2003/Mar03/03139r0P802-15\\_TG3a-STM-CFP-Presentation.ppt](http://grouper.ieee.org/groups/802/15/pub/2003/Mar03/03139r0P802-15_TG3a-STM-CFP-Presentation.ppt).
31. Carbonelli C, Mengali U. Timing recovery for uwb signals. In *Proceedings of GLOBECOM 2004*, Dallas, USA, November 2004.
32. Saleh A, Valenzuela R. A statistical model for indoor multipath propagation. *IEEE Journal on Selected Areas in Communications* 1987; **5**(2): 128–137.
33. Kunisch J, Pamp J. Measurement results and modeling aspects for the uwb radio channel. In *Proceedings of the IEEE Conference on Ultra Wideband Systems and Technologies 2002, Digest of Papers*, Baltimore, USA, 2002, pp. 19–24.

## Authors' Biographies



**Mario R. Casu** received his Electronics Engineer degree (summa cum laude) in 1998 and his Ph.D. in 2002 from the Politecnico di Torino, Italy. He is now an assistant professor at the Politecnico di Torino, Department of Electronics, and member of the VLSI laboratory and of CERCOM (center of excellence for multimedia radiocommunications). In 2001, he was with ST Microelectronics Central R&D working on SRAM development and CMOS library characterization on a partially-depleted 0.13 micron SOI technology. His research interests are in the field of digital CMOS circuits modeling and design of systems-on-chip for ultra wideband applications. He is the co-author of several papers published in international conferences proceedings and journals.



**Giuseppe Durisi** was born in Torino (Italy) in 1977. He received his Laurea degree (summa cum laude) from the Politecnico di Torino in 2001. In January 2002, he joined Istituto Superiore Mario Boella, where he works on the MIUR financed national project PRIMO. From January 2003, he has been pursuing his Ph.D at the Department of Electronics of the Politecnico di Torino, under the supervision of Professor Sergio Benedetto. In 2002, he spent 6 months as a visiting researcher at IMST, Kamp-Lintfort, Germany, working on the IST FP5 project Whyless.com. In 2004, he was a visiting student at the University of Pisa under the supervision of Professor Umberto Mengali and currently is a visiting student at the Swiss Federal Institute of Technology (ETH) under the supervision of professor Helmut Bölcskei. His fields of interest are: ultra wideband and channel coding for fading channels.