

ON THE IMPLEMENTATION OF A TRANSMITTED-REFERENCE UWB RECEIVER

Mario R. Casu[†], Giuseppe Durisi[‡] and Sergio Benedetto[†]

[†] CERCOM-Politecnico di Torino, Department of Electronics
C.so Duca degli Abruzzi, 24, I-10129, Torino, Italy

[‡] Istituto Superiore Mario Boella
Via Pier Carlo Boggio, 61, I-10138 Torino, Italy
email: mario.casu@polito.it, durisi@ismb.it, sergio.benedetto@polito.it

ABSTRACT

In this paper¹ 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. Issues, challenges and possible design solutions are discussed. In particular, an analysis of the trade-off between performance and complexity in an integrated circuit implementation is given.

1. INTRODUCTION

Ultra Wide Band (UWB) systems using short pulse modulation are nowadays considered a viable solution for short-range mobile communications. 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, at the best of our knowledge, there are not fully integrated solutions commercially available. Only few recent publications have the complete integration of a UWB transceiver in a CMOS technology as final goal ([2] and [3]). However, these works are limited in scope by the assumptions on the channel model or by the high power consumption of the analyzed receivers. Research in the field of low-complexity receivers is limited today to the system-level analysis and publications describing detailed implementation of the overall receiver are still lacking.

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 [4] is discussed. The receiver is assumed to operate on a UWB indoor channel, modelled as in [5].
- 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. 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 demodulation algorithms are presented. 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

In case of perfect channel estimation and synchronization (coherent reception), absence of intersymbol and multiuser interference, it is well known [6] 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, the high number of multipath replicas of an indoor UWB channel (up to 100 according to [5]) makes this structure extremely power-consuming, even under the assumption of perfect channel state information at the receiver. Furthermore, additional power consumption is required by the estimation step, a non-trivial task in such a dense multipath environment.

A different approach 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”, [7]. 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. Among these receivers, it is worth mentioning, for their inherent architectural simplicity, the differential [8], the transmitted-reference (TR) [4] and energy detector [9].

In this work we focus on the TR receiver, with modulation format as described in [4]. 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

¹This work has been partially sponsored by MIUR (Italian Ministry of Education and Research) under the projects CERCOM and PRIMO.

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.

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. The fully analog receiver correlates the incoming data with an internal analog template, and samples the correlation result. A hard decision consists of 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 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

4.1 Bandwidth and channel model

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. 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 we present in the following sections are completely parametric and can be adapted to other bandwidths.

The test-vectors of the channel response are built using the IEEE 802.15.3a model [5]. 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. A single user scenario is considered, assuming that some form of MAC protocol (TDMA or CSMA) is employed to limit multiple access interference.

4.2 Synchronization

4.2.1 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).

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 criterium (for example 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 [10]. However, the strategy described in [10], 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, sub-optimal with respect to the one presented in [10], 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.

4.2.2 Symbol-level synchronization

We assume that each frame contains N_f samples, where N_f is obtained by multiplying the frame length T_f by the sampling frequency f_s , and that the number of 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 $S_{bit}(k), k \in [1, N_f]$ and the starting sample k_{max} are computed as follows:

$$S_{bit}(k) = \sum_{j=1}^{N_p} \left| \sum_{i=k}^{k+N_w-1} r[i] \cdot r[i + jN_f] \right| \quad k = 1, 2, \dots, N_f \quad (2)$$

$$k_{max} = \arg \left(\max_{k \in [1, N_f]} (S_k) \right), \quad (3)$$

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. 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. Of course, increasing N_p comes at the cost of increasing acquisition time and/or hardware complexity, as shown later on.

4.2.3 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.

4.3 Demodulation

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 $\mathbf{r} = [\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N_r}]$, 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 $\mathbf{d} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_{N_d}]$, containing the discrete samples of the data part of the transmitted block.

Fig. 1 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.

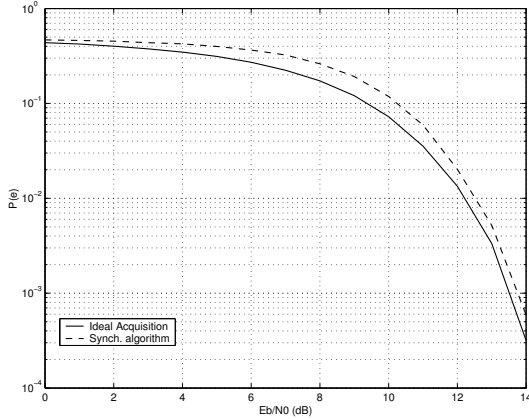


Figure 1: Error probability: ideal bit-level acquisition compared to the algorithm described in Eq. (2) with $N_p = 1$.

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 because of the very high bandwidth. Some proposals ([2], [11]) suggest the use of a bank of N_{ad} parallel A/D converters, each one with sampling frequency f_s/N_{ad} and delay between two consecutive converters of $\delta T = 1/f_s$. 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, like for low data-rate applications, the number of parallel A/D converters, 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 Fig. 2 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 Fig. 2. If more than one window are necessary, as for the previously described synchronization algorithms, the shift-register size can be easily extended.

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 are possible ranging from parallel to sequential.

5.1.1 Parallel Implementation

We assume to have sufficient resources for executing the algorithm of equation (2) in one clock cycle (or few clock cycles, in case of

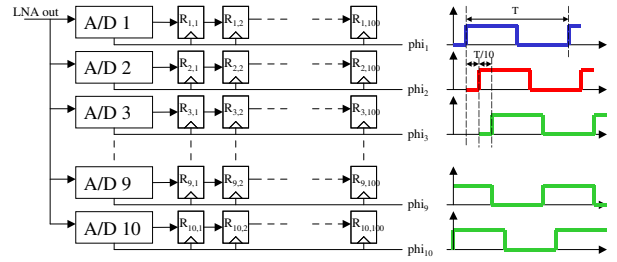


Figure 2: A possible arrangement of parallel A/D for the case $N_f = 1000$.

Table 1: Complexity for the parallel implementation according to (2) and after loop simplification.

Resource type	Number of resources according to (2)	Number of resources after loop simpl.
MEM	$(N_p + 1)N_f + N_w - 1$	$(N_p + 1)N_f + N_w - 1$
MUL	$N_w N_p N_f$	$N_p(N_f + N_w - 1)$
ADD	$N_w N_p N_f$	$N_p(\lfloor N_f + N_w - 2/2 \rfloor + \lfloor N_f/2 \rfloor + N_f)$

pipelining). The complexity requirements obtained by simply unrolling the loops are reported in Table 1, second column, 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. The minimum number of different products is not difficult to compute and is shown in Table 1, third column. 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 3, a simplified case with $N_f = 6$, $N_w = 5$, $N_p = 1$ ($MUL=6+5-1=10$), 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

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

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

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

which is consistent with the figure. If $N_p > 1$ the number of adders has to be multiplied by this factor. Table 1, third column summarizes the overall complexity of this implementation. After the evaluation of the correlations, the algorithm proceeds with the search of the maximum. The parallel architecture is simply a tree of comparators.

5.1.2 Sequential Implementation

A possible sequential implementation takes advantage of the previous observation concerning the reuse of products and sums. We can

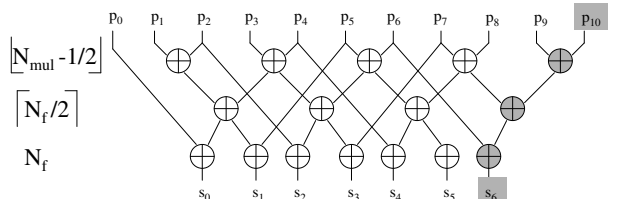


Figure 3: Example of parallel implementation.

rewrite for convenience the estimates $S_{bit}(k)$ as follows

$$\sum_{j=1}^{N_p} \left| \sum_{i=k}^{k+N_w-1} r[i] \cdot r[i+jN_f] \right| = \sum_{j=1}^{N_p} \left| \sum_{i=k}^{k+N_w-1} p(i, j) \right| = \sum_{j=1}^{N_p} |s(k, j)| \quad (4)$$

and then write a recursive formula

$$s(k, j) = s(k-1, j) - p(k-1, j) + p(k+N_w-1, j) \quad (5)$$

with $s(0, j)$ computed as in (4). It is clear that if eq. (5) 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. Table 2 summarizes the complexity of this simple implementation.

Table 2: 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 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.

5.1.3 Execution Time Considerations

The first important point to understand is if the synchronization algorithm needs realtime execution. 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 , $i = 0, \dots, MUL - 1$, (see Figure 3) 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.

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” criterion.

5.1.4 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 the strategy used for computing the correlation is, the execution time limit under the requirement of not losing any data is $(N_p + 2)N_f T_s \leq N_p N_{corr} T_{ck}$,

where N_{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 $N_{corr} = 75$. 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

The demodulation consists of two phases, template calculation and computation of the correlation between received signal and template. Also in this case, parallel and sequential solutions can be envisaged. For space constraint, the description of the demodulator is not included in the paper. The interested reader is referred to [12].

6. CONCLUSIONS

In this paper we discussed the implementation challenges for an UWB TR 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. The architectures described have been implemented and verified on a FPGA equipped board. An application specific integrated circuit (ASIC) version on an advanced CMOS 0.13 μm technology is currently under development.

REFERENCES

- [1] IEEE 802.15 WPAN Low rate Alternative PHY Task Group 4a (TG4a), [Online]: www.ieee803.org/15/pub/TG4a.html
- [2] M. S.-W. Chen, “Ultra Wide-band Baseband Design and Implementation,” 2002 M.Sc. Univ. of California at Berkeley.
- [3] R. Blazquez, et alii, “Digital architecture for an ultra-wideband radio receiver,” in *Proc. VTC Fall*, Orlando, FA, USA, Oct. 2003, vol. 2, pp 1303-1307.
- [4] S. Franz and U. Mitra, “On optimal data detection for uwb transmitted reference system,” in *Proc. GLOBECOM*, S. Francisco, CA, USA, 1-5, Dec. 2003, vol. 2, pp. 764-768.
- [5] J. Foerster (editor), “Channel modeling sub-committee report final,” IEEE P802.15 02/490r1 SG3a, Feb 2002.
- [6] J. G. Proakis, M. Salehi, *Communication Systems Engineering*. Upper Saddle River, New Jersey: Prentice Hall, 2002.
- [7] G. Durisi and S. Benedetto, “Comparison between coherent and non-coherent receivers for uwb communications,” *EURASIP journal on applied signal processing - UWB state of the art*, Dec. 2004.
- [8] M. Ho, S. Somayazulu, J. Foerster, and S. Roy, “A differential detector for an ultra-wideband communications system,” in *Proc. VTC Spring*, Birmingham, AL, USA, May 2002, vol. 4, pp. 1896-1900.
- [9] Y. Souilmi and R. Knopp, “On the achievable rates of ultra-wideband ppm with non-coherent detection in multipath environments,” in *Proc. Int. Conf. Comm. ICC*, vol. 5, Anchorage, USA, May 2003, pp. 3530-3534.
- [10] C. Carbonelli, and U. Mengali, “Timing recovery for uwb signals,” in *Proc. GLOBECOM*, Dallas, USA, Nov. 2004, vol. 1, pp. 61-65.
- [11] Helal D, Rouzet P. “STMicroelectronics proposal for IEEE 802.15.3a Alt PHY”, Mar. 2003, [Online]: <http://grouper.ieee.org/groups/802/15/pub/2003/Mar03>.
- [12] M. R. Casu, G. Durisi, “Implementation Aspects of a Transmitted-Reference UWB Receiver”, accepted for *Wireless Communications and Mobile Computing Journal*, special issue on *UWB Communications*, Jan. 2005.