Sensors **2015**, *15*, 20825-20844; doi:10.3390/s150820825

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Efficient Implementation of a Symbol Timing Estimator for Broadband PLC

Francisco Nombela †, Enrique García †, Raúl Mateos † and Álvaro Hernández †,*

Electronics Department, University of Alcalá, Campus Universitario s/n, Alcalá de Henares, Madrid 28805, Spain; E-Mails: francisco.nombela@depeca.uah.es (F.N.); enrique.garcia@depeca.uah.es (E.G.); raul@depeca.uah.es (R.M.)

† These authors contributed equally to this work.

* Author to whom correspondence should be addressed; E-Mail: alvaro.hernandez@uah.es; Tel.: +34-91-885-6583; Fax: +34-91-885-6591.

Academic Editor: Vittorio M. N. Passaro

Received: 26 June 2015 / Accepted: 14 August 2015 / Published: 21 August 2015

Abstract: Broadband Power Line Communications (PLC) have taken advantage of the research advances in multi-carrier modulations to mitigate frequency selective fading, and their adoption opens up a myriad of applications in the field of sensory and automation systems, multimedia connectivity or smart spaces. Nonetheless, the use of these multi-carrier modulations, such as Wavelet-OFDM, requires a highly accurate symbol timing estimation for reliably recovering of transmitted data. Furthermore, the PLC channel presents some particularities that prevent the direct use of previous synchronization algorithms proposed in wireless communication systems. Therefore more research effort should be involved in the design and implementation of novel and robust synchronization algorithms for PLC, thus enabling real-time synchronization. This paper proposes a symbol timing estimator for broadband PLC based on cross-correlation with multilevel complementary sequences or Zadoff-Chu sequences and its efficient implementation in a FPGA; the obtained results show a 90% of success rate in symbol timing estimation for a certain PLC channel model and a reduced resource consumption for its implementation in a Xilinx Kyntex FPGA.

Keywords: Power-Line Communications; symbol timing estimation; complementary sequences; Zadoff-Chu sequences; FPGA-based architecture

1. Introduction

In recent years Power-Line Communications (PLC) have emerged as a consolidated broadband standard for data transmission [1], taking advantage of the mains already installed in most indoor environments (public buildings, homes, industrial factories, *etc.*). Apart from facilitating the deployment of devices since there is no need for additional cabling, this approach provides feasible solutions to different kinds of applications, such as sensory and automation systems [2–4], distributed systems [4,5], smart spaces [6,7], or even industrial networks [8,9], where it is necessary to have available a broadband communication among the elements that integrate the system.

Nevertheless, PLC-based systems still have some details that can be improved in order to achieve a better performance. One of them is the implemented medium access technique, where most of the previous works have been focused on Multi-Carrier Modulation (MCM) [10,11], not only in PLC but also in other standards [12] like Long Term Evolution (LTE) [13]. MCM allows the performance and the spectrum efficiency to be enhanced by dividing the available bandwidth into subchannels, which data are transmitted through. Different MCM techniques have been successfully proposed for PLC, depending on how the subchannel division is carried out. Some of the most relevant are those based on Discrete Trigonometric Transform (DTT) [14], or those based on filter banks (Filter-Bank Multi-Carrier, FBMC) [10,15]. In any case, no matter the considered medium access technique, all the approaches require a suitable and reliable synchronization method between transmitters and receivers, in order to be capable of achieving the expected performance, thus avoiding inter-symbol interference (ISI) and inter-carrier interference (ICI).

The synchronization issue has been widely considered in previous works for multi-carrier techniques, as Discrete Modulation Tone (DMT) [16], FBMC [17], or Orthogonal Frequency Division Multiplexing (OFDM) [18,19] for wireless communications.

Nevertheless, for broadband PLC there is a reduced number of works that deal with synchronization in such a scenario, where some features from the PLC channel, such as selective frequency fading, channel length, and noise models should be considered [20]. Most of the research done for PLC is focused on the OFDM implementation proposed in the IEEE 1901–2010 standard [21], but not on the FBMC physical layer approach. In those works, the auto-correlation metric is typically used, which consists on transmitting several repeated symbols and correlating the consecutive symbols in the receiver [18,22]. This metric is useful for wireless communications as it increases the robustness to Doppler shifts, but it suffers from the specific conditions of PLC channel. Furthermore, previous PLC synchronization works consider the maximum peak of the auto-correlation metric as the beginning of a symbol [23,24]. This cannot be assumed in practice due to multipath and therefore the first arriving path can be of lower magnitude than a multipath component [25]. This is the reason why extra research efforts should be focused on the proposal and development of suitable synchronization algorithms for MCM in PLC communications.

Generally speaking, it is relevant to note that many previous proposals, not only for synchronization but also for multi-carrier medium access techniques, imply a challenge from a real-time implementation and sensory point of view [26,27]. They often handle high data rates, requiring intensive and parallel signal processing and a certain connection to digital converters and sensors. Accordingly, the possibility of providing a feasible real-time architecture for the implementation of the proposed synchronization algorithm actually becomes significant. Recent Field-Programmable Gate Array (FPGA) devices already play a key role in the implementation of this kind of systems [28]. They allow the design of highly parallel and flexible architectures for signal processing at high frequencies (in the range of MHz).

This work presents a novel algorithm to estimate the synchronization delay, suitable for a FBMC transmultiplexer used in PLC communications. The proposed synchronization is based on cross correlation, where the received signal is correlated with the transmitted one. Unlike the common approaches based on auto-correlation techniques, the proposal makes possible to improve performance and robustness of communications in the PLC channel. Furthermore, a FPGA-based architecture is also described for the real-time implementation of the proposed algorithm, optimizing terms as resource consumption and operation frequencies, and increasing resource reutilization. The rest of the manuscript is organized as follows: Section 2 reviews the medium access technique considered for PLC synchronization, whereas Section 3 explains the proposed synchronization algorithm; Section 4 describes the hardware architecture proposed for the implementation of the synchronization algorithm; Section 5 shows some experimental results that validate the design; and, finally, conclusions are discussed in Section 5.

2. Wavelet-OFDM Approach

Wavelet-OFDM is a medium access technique that can be efficiently implemented by means of the Discrete Cosine Transform (DCT), also known as a technique based on Cosine-Modulated Filter Bank (CMFB). According to the IEEE 1901–2010 standard [21], in narrowband communications through mains, Wavelet-OFDM modulation is considered robust against selective frequency fading and narrowband noise, thus providing a better use of the available bandwidth since no guard intervals are required, unlike OFDM. Furthermore, the IEEE 1901–2010 standard defines frequencies at which PLC systems can transmit. For that purpose, M = 512 carriers are distributed in the baseband version, for the range from 0 to 31.25 MHz, although only those in the range 1.8–28 MHz can be actually used. Even in this range, a mask has to be applied to filter frequencies related to amateur radio, so, finally, a set of 360 carriers is available for information transmission.

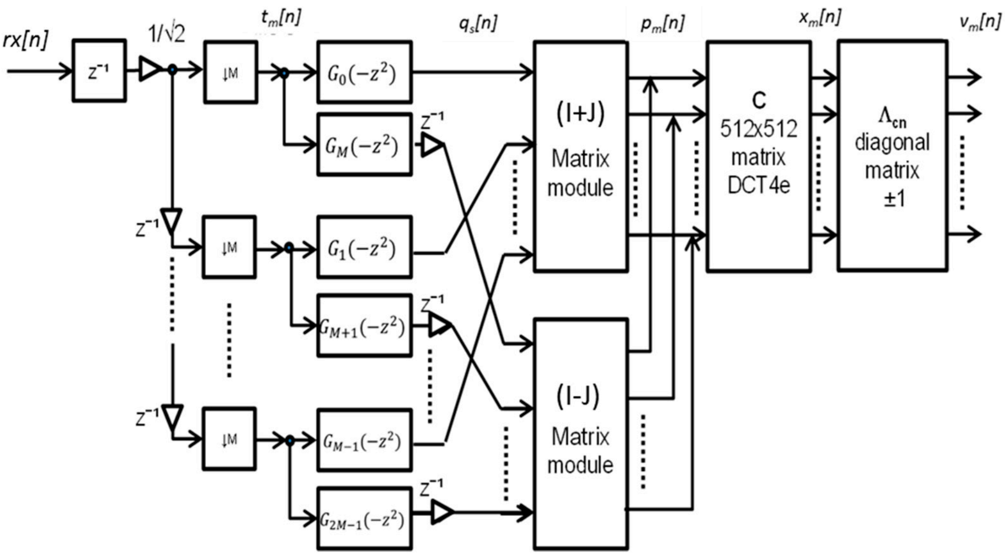

Figure 1. General block diagram of the used Wavelet-OFDM receiver.

Figure 1 shows the block diagram of a possible efficient implementation of the filter bank at the Wavelet-OFDM receiver [28], based on the DCT-4e. Basically, the processing consists of a first deserialization of the received signal $rx[n]$ to obtain the M different subchannels $t_m[n]$. These M subchannels are processed by pairs of filters $G_s(z^{-1})$, where $s = 0, 1, \ldots, S-1$ with $S = 2 \cdot M$, to obtain the intermediate signals $q_s[n]$. Afterwards, these signals $q_s[n]$ are linearly operated by matrices $(\mathbf{I} + \mathbf{J})$ and $(\mathbf{I} - \mathbf{J})$ and added, so the DCT-input signals $p_m[n]$ are obtained. The DCT module processes the M subchannels to provide the signals $x_m[n]$, which, after multiplied by the diagonal matrix $\mathbf{\Lambda_{cn}}$, provide the output subchannels $v_m[n]$.

This efficient implementation can be expressed in a matricial way for the synthesis bank $\mathbf{f}(z)$ as Equation (2):

$$\mathbf{f}(z) = [F_0(z)\ F_1(z)\ \ldots F_{M-1}(z)]^T \tag{1}$$

$$\mathbf{f}(z) = \frac{1}{\sqrt{2}} \cdot [\mathbf{g_o}(z^{2M}) \quad z^{-M}\mathbf{g_1}(z^{2M})] \cdot \begin{bmatrix} (\mathbf{I}+\mathbf{J}) \\ (\mathbf{I}-\mathbf{J}) \end{bmatrix} \cdot \mathbf{C_{4e}} \cdot \mathbf{\Lambda}_{cn} \tag{2}$$

And for the analysis bank $\mathbf{h}(z) = \mathbf{f}(z^{-1})$ as Equation (3):

$$\mathbf{h}(z) = \frac{1}{\sqrt{2}} \cdot \mathbf{\Lambda}_{cn} \cdot \mathbf{C_{4e}} \cdot [(\mathbf{I}+\mathbf{J}) \quad (\mathbf{I}-\mathbf{J})] \cdot \begin{bmatrix} \mathbf{g}_o(z^{2M}) \\ z^{-M}\mathbf{g}_1(z^{2M}) \end{bmatrix} \tag{3}$$

where $\mathbf{g_0}(z)$ is a diagonal matrix, whose diagonal elements are $[G_0(-z), G_1(-z), \ldots, G_{M-1}(-z)]$ and $\mathbf{g_1}(z)$ is also a diagonal matrix whose diagonal elements are $[G_M(-z), G_{M+1}(-z), \ldots, G_{2M-1}(-z)]$, where $G_M(-z)$, $0 \leq m \leq 2 \cdot M - 1$, is the prototype filter in the discrete domain; $\mathbf{C_{4e}}$ is the DCT-4e matrix, whose elements are:

$$[\mathbf{C_{4e}}]_{k,l} = \sqrt{\frac{2}{M}} \cdot \cos\left(\left(k + \frac{1}{2}\right)\frac{\pi}{M} \cdot \left(l + \frac{1}{2}\right)\right), 0 \leq k \leq M-1, 0 \leq l \leq 2M-1 \tag{4}$$

$\mathbf{\Lambda_{cn}}$ is a $M \times M$ diagonal matrix, whose i-th element is:

$$[\mathbf{\Lambda}_{cn}]_{i,i} = \sqrt{2} \cdot \cos\left(\frac{\pi}{2}\left(i + \frac{1}{2}\right)\right) \cdot \cos(\theta_c), \theta_c = \{0, \pi\} \tag{5}$$

and $\mathbf{I}$ denotes a $M \times M$ identity matrix, whereas $\mathbf{J}$ denotes the counter-identity matrix.

In the case of a bank with perfect reconstruction, then $R(z) \cdot \mathbf{f}(z) = \mathbf{h}(z) \cdot R(z)$, where $R(z)$ is the input to the synthesis bank. Nonetheless, ideal synchronization estimation is assumed to achieve a perfect reconstruction filter bank, otherwise the reconstructed signal is affected by ISI and ICI, making unfeasible to recover the transmitted data at the receiver side. For that reason it is significant to obtain the design and implementation of an efficient synchronization algorithm, able to accurately estimate the symbol timing.

3. Proposed Synchronization Algorithm

3.1. Multi-Level Complementary Sequences and Zadoff-Chu Sequences

The synchronization algorithm proposed here for a Wavelet-OFDM receiver is based on the use of multi-level complementary sequences or Zadoff-Chu sequences as pilot signals, since they provide suitable correlation properties and a flexible length, which allow the proposal to be adapted to the number of available subcarriers for transmission. Both types of sequences have been already used in numerous sensory systems described in previous works due to these correlation properties [29,30].

Multi-level complementary sequences present ideal correlation properties, so that a set of K sequences $s_{j,i}$, $0 \leq j$ and $i \leq K-1$ of length L is a complementary set of sequences (CSS) if the sum of their auto-correlation functions $Cs_{j,i}$ is a Kronecker delta [31], according to Equation (6):

$$\sum_{i=0}^{K-1} C_{s_{j,i}}[n] = \eta \cdot \delta[n];\ 0 \leq j \leq K-1;\ \eta \in \mathbb{R} - \{0\} \tag{6}$$

Two CSS are not correlated if the sum of the aperiodic cross-correlation functions $C_{s_{j,i}s_{j',i'}}$, between the sequences $s_{j,i}$ and $s_{j',i'}$ from both sets, is zero for any correlation shifting Equation (7):

$$\sum_{i=0}^{K-1} C_{s_{j,i}s_{j',i'}}[n] = 0;\ 0 \leq n \leq L-1;\ 0 \leq j \neq j' \leq K-1 \tag{7}$$

where K is the maximum number of uncorrelated CSS and equal to the number of sequences in any set. In a similar way to complementary sequences, Zadoff-Chu sequences present other properties that become relevant when performing cross-correlations. The Zadoff-Chu sequences are non-binary codes with a constant module. They have a null correlation function between a certain Zadoff-Chu sequence s_q[n] and a circularly shifted version of it, $s_q[n + \Delta]$, except when they are aligned ($\Delta = 0$), where L is the length of the mentioned sequence s_q and $0 \leq q \leq L-1$ is the number of spreading sequences with low cross-correlation values according to Equation (8):

$$\sum_{n=0}^{L-1} s_q[n]s_q^*[n+\Delta] = \begin{cases} 1\ \Delta = 0 \\ 0\ \Delta \neq 0 \end{cases} \tag{8}$$

where s_q^* is the complex conjugate of s_q. This property allows a correct estimation of the temporal synchronization, as well as the estimation of the channel impulse response.

By selecting a prime number for the length L, the number of sequences that provide a minimum value in the cross correlation is equal to $L - 1$ [32]. Furthermore, if a non-prime length is required, it is possible to generate another sequence by means of truncation or cyclic expansion.

Finally, other property to be remarked is that the Discrete Fourier Transform (DFT) of a Zadoff-Chu sequence s_q is another cyclically shifted sequence, thus implying that they can be generated both in time and frequency domains. This feature is relevant since processing often requires working in the frequency domain, so the correlation properties of these sequences are not lost in those cases.

3.2. Description of the Algorithm

Figure 2 shows a block diagram of the proposed synchronization algorithm, as well as the channel estimation and equalization modules. The proposed pilot-based symbol timing estimation uses cross-correlation techniques for PLC channel as it provides a better performance than the auto-correlation metric [33,34], and due to the negligible Doppler effect in PLC channels.

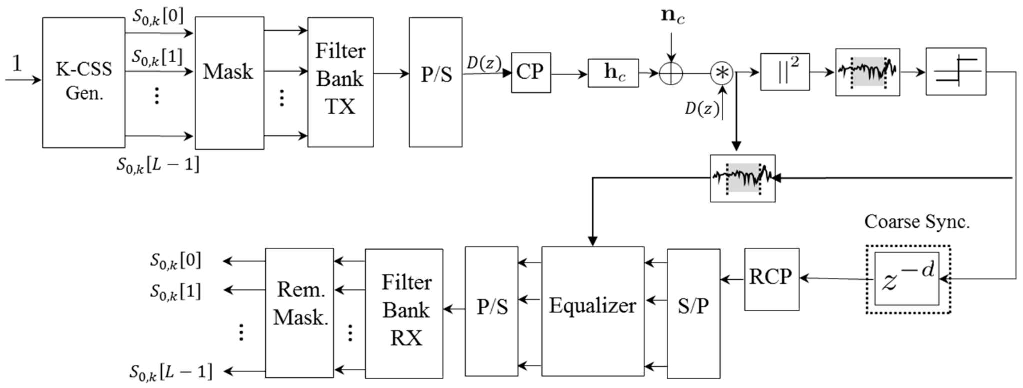

Figure 2. General block diagram of the signal processing modules involved in the PLC link.

The ideal correlation features provided by multi-level complementary sequences and Zadoff-Chu sequences allow the improved detection of the first arrival tap, in order to develop a more robust synchronization algorithm. For representation purposes, a CSS generator of length *L* has been considered in Figure 2 and shown as *K-CSS Gen*. Nonetheless, Zadoff-Chu sequences can also be used for synchronization purposes.

Given a set of multi-level complementary sequences in the Z-domain:

$$\mathbf{S}_0(z) = \{S_{0,0}(z), S_{0,1}(z), \dots, S_{0,K-1}(z)\}$$

where:

$$S_{0,k}(z) = s_{0,k}[0] + s_{0,k}[1] \cdot z^{-1} + \cdots + s_{0,k}[L-1] \cdot z^{-L+1}$$

the transmission of a set of *K* symbols with *M* samples is proposed, each one formed by a multi-level complementary sequence $S_{0,k}(z)$ of length *L* ($L \leq M$ and equal to the number of available subchannels). This number is determined by a transmission mask defined in the IEEE 1901–2010 standard [21]. The sequence bit $s_{0,k}[n]$ is assigned to the k-th symbol and to the n-th subchannel available in the synthesis bank, $0 \leq \mathrm{n} \leq L-1$. In those subchannels that are not available for transmission due to PLC standard regulation, the bit 0 is assigned, so $M - L$ subchannels will not be used.

In this paper we use a $K = 2$ multilevel CSS as a pilot waveform, thus transmitting 2 symbols ($2 \cdot M$ samples) for synchronization purposes. On the other hand, in the case of the complex-valued Zadoff-Chu sequences, the bits are allocated in a similar way to the multi-level CSS, but the first symbol of *M* samples to be sent contains the real part of the sequence, whereas the second symbol contains the imaginary one.

The input sequence bits, after being allocated in the appropriate subcarrier by the mask block, are processed in parallel by the synthesis bank **f**(*z*) (*Filter Bank TX* in Figure 2), to shape the spectrum as designed in the prototype filters $G_m(-z)$. Later the output from **f**(*z*) is converted into a serial datastream by using a parallel-to-serial converter (*P/S* in Figure 2). Consequently, the output signal, after the *P/S* block is equal to Equation (9):

$$D(z) = \sum_{k=0}^{K-1} \sum_{\substack{n=0 \\ n \in \chi}}^{L-1} s_{0,k}[n] \cdot F_\chi(z^M) \cdot z^{-(kM+\chi)} \tag{9}$$

where χ is the subset of available subcarriers. After the appropriate codes have been used as a pilot signal, the last samples of the modulated symbol are repeated at the beginning of it, as shown in Figure 3. This constitutes the so-called Cyclic Prefix (*CP* in Figure 2), which allows ISI and ICI to be efficiently mitigated in the frequency domain when the channel delay spread is shorter than the CP length L_{cp}. Although it is theoretically not necessary to use CP to avoid ISI and ICI in Wavelet-OFDM, its use efficiently reduces them by using a frequency domain equalizer with one complex multiplier per subcarrier [35].

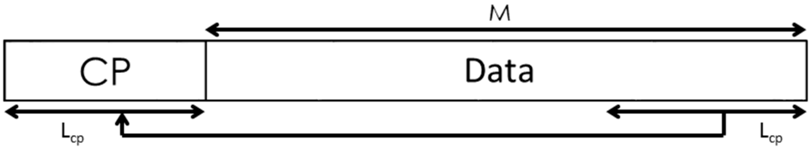

Figure 3. Block diagram of the CP insertion with length L_{cp} in a symbol of length *M*.

Assuming no CP is used and there is an ideal channel ($\mathbf{h}_c = 1$ in Figure 2) and no channel noise ($\mathbf{n}_c = 0$), the correlation of the transmitted pilot at the receiver input is equal to:

$$D(z)\cdot D(z^{-1}) = \sum_{k=0}^{K-1}\sum_{\substack{n=0\\ n\in\chi}}^{L-1} s_{0,k}[n]\cdot F_\chi(z^M)\cdot z^{-(kM+\chi)}\,.\,\sum_{k=0}^{K-1}\sum_{\substack{n=0\\ n\in\chi}}^{L-1} s_{0,k}[n]\cdot F_\chi(z^{-M})\ \cdot z^{(kM+\chi)} \tag{10}$$

Assuming a good filter design, the correlation of adjacent subcarriers can be neglected, reducing the previous expression Equation (10) to Equation (11), where only one side of the correlation function is considered:

$$\begin{aligned} D(z)\cdot D(z^{-1}) &= \sum_{k=0}^{K-1}\sum_{\substack{n=0\\ n\in\chi}}^{L-1}\left(s_{0,k}[n]\right)^2\cdot F_\chi(z^M)\cdot F_\chi(z^{-M}) \\ &+\sum_{k=0}^{K-1}\sum_{k'=k}^{K-1}\sum_{\substack{n=0\\ n\in\chi}}^{L-2} s_{0,k}[n]\cdot s_{0,k'}[n+1]\cdot F_\chi(z^M)\cdot F_{\chi+1}(z^{-M})\cdot z^{-M(k'-k)-1} \end{aligned} \tag{11}$$

The first term in Equation (11) corresponds to the auto-correlation of the spreading sequences, affected by the auto-correlation of the filter bank, and the second term represents the cross-correlation of adjacent subcarrier filters and sequence bits.

In the case of having a real PLC channel and different noises affecting the channel [20], Equation (11) turns out a cross-correlation between the received signal *R*(*z*) and *D*(*z*), where $R(z) = D(z)\cdot\mathbf{h}_c + \mathbf{n}_c$ and expressed as Equation (12):

$$\begin{aligned} R(z)\cdot D(z^{-1}) &= \mathbf{h}_c\cdot\sum_{k=0}^{K-1}\sum_{\substack{n=0\\ n\in\chi}}^{L-1}\left(s_{0,k}[n]\right)^2\cdot F_\chi(z^M)\cdot F_\chi(z^{-M}) + D(z^{-1})\cdot\mathbf{n}_c + \mathbf{h}_c \\ &\cdot\sum_{k=0}^{K-1}\sum_{k'=k}^{K-1}\sum_{\substack{n=0\\ n\in\chi}}^{L-2} s_{0,k}[n]\cdot s_{0,k'}[n+1]\cdot F_\chi(z^M)\cdot F_{\chi+1}(z^{-M})\cdot z^{-M(k'-k)-1} \end{aligned} \tag{12}$$

Therefore, the ideal correlation properties of multilevel complementary sequences are degraded by the correlation of the subcarrier filters, which is only ideal when all the subcarriers of the filter bank are used for transmission, and by the channel noise $\mathbf{n}_c$. Also, note that the CP insertion does not degrade the correlation properties of the spreading sequences. The same applies when using Zadoff-Chu sequences.

After the CP insertion, the signal goes through the PLC channel $\mathbf{h}_c$ and the channel noise $\mathbf{n}_c$, as depicted in Figure 2. Hereinafter, the Tonello PLC channel model proposed in [36] has been considered for simulation, as well as the three different types of noises modelled in [20] for PLC. The Tonello channel model shows the harsh conditions of the PLC channel as the first tap does not necessarily present the highest amplitude, being challenging to estimate the first arriving path. Figure 4 shows the module $|h_c|^2$ of a realization of the Tonello PLC channel model sampled at 62.5 MHz, where the strong multipath and long channel duration can be observed.

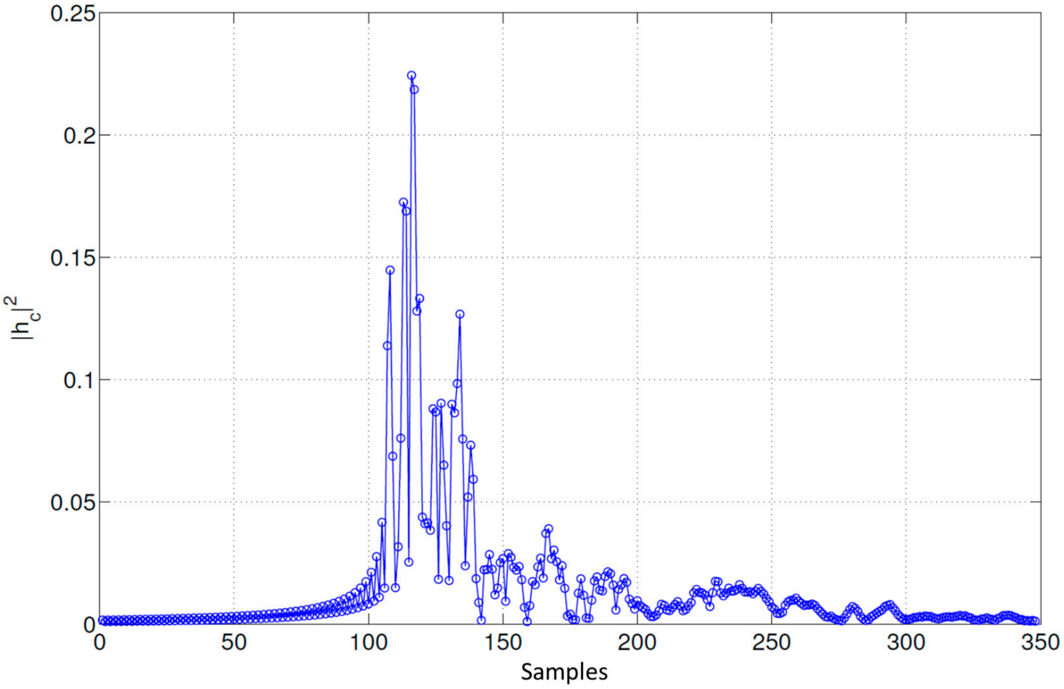

Figure 4. Module $|h_c|^2$ of the Tonello PLC channel impulse response.

In order to estimate the first arriving path, the cross-correlation derived in Equation (12) is squared and the maximum correlation peak is used to apply a window of 40 samples; within this window, the first sample higher than a certain static threshold is considered as the first tap of the channel impulse response.

After estimating and compensating the synchronization delay, *Coarse Sync* in Figure 2, the CP is removed at the receiver (*RCP* block in Figure 2). Note that if the symbol timing estimation error is lower than the CP length L_{cp}, this error appears as a phase rotation for a frequency-domain equalizer, which can be efficiently compensated by using only one complex multiplier per subcarrier. For that purpose, the windowed correlation can be reused for channel estimation, thus minimizing the required computational load. Finally, the equalized signal is demodulated by using the analysis filter bank (*Filter Bank RX* in Figure 2), recovering the transmitted baseband symbols after removing the emission mask (*Rem. Mask* block in Figure 2).

4. Proposed Architecture

Some previous works have already dealt with the implementation of the Wavelet-OFDM emitter and receiver [28], even describing in detail the efficient architecture proposed for the filter banks. Because of that, this section is focused on the definition and design of an efficient architecture for the implementation of the proposed synchronization algorithm. Furthermore, it is important to remark that the proposal has been developed on a Xilinx XC7K325T FPGA, whose internal architecture, datawidth and resource availability determine certain design decisions, especially those related to the fixed-point representation of the involved signals. This fixed-point representation has been defined hereinafter by a format Q(α·β), where α is the global number of bits and β is the number of fractional bits.

Figure 5 shows the block diagram of the synchronization proposal, according to the algorithm description carried out before. This diagram can be divided into four main modules: a correlator, a squaring module, a maximum detector, and a windowing and thresholding module. The global input for the synchronization architecture are the $2 \cdot M$ received samples r[n], obtained after discarding the cyclic prefix CP, whereas the final output is the estimated synchronization delay. Note that the input at the correlator has been parallelized in sets of 16 samples. This parallelism degree has been fixed in order not to significantly increase the resource consumption (especially multipliers), and, although it implies a latency in the global system operation, it is still possible to achieve real-time performance.

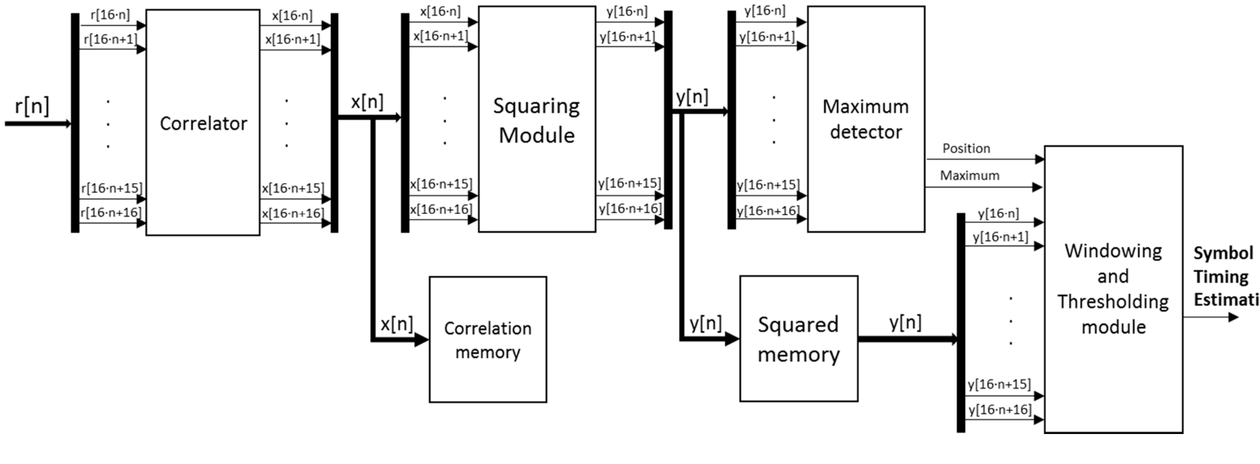

Figure 5. General block diagram of the architecture proposed for the implementation of the synchronization algorithm.

The critical issue about the design is to achieve real-time operation without discarding data at the reception, so the system should estimate the synchronization delay as fast as possible. Keeping this in mind, the most significant bottleneck is the correlation module, where a larger number of clock cycles are required to compute the corresponding correlation value x[n]. The approach based on computing the correlation function x[n] by means of sliding windows over the input data r[n] has been rejected [27], since the computational load becomes unfeasible for a real-time implementation. On the other hand, an approach based on non-overlapped windows over the input data r[n] has been considered here. At this moment, it is important to recall the importance of the cyclic prefix to perform this kind of correlation, in order not to discard a part of the information and degrade the correlation maximum value x[n], so only one block of $2 \cdot M$ samples, *i.e.*, two symbols of r[n], is processed to estimate the symbol timing.

The correlation module between the signal *r*[n] and the transmitted preamble *d*[n] involves the FFT and IFFT modules, which require most of the multipliers available in the FPGA. The proposed correlation module is based on the continuous acquisition at the receiver, signal *rx*[n], and the storing in an input buffer with a length of $2(M + L_{cp})$, where *M* is the length of every packet and L_{cp} the cyclic prefix length (see Figure 6). Afterwards, it is possible to discard the cyclic prefix and obtain the input of the correlation module *r*[n] with length $2 \cdot M$ samples. Note that the transmitted preamble *d*[n] consists of, either a pair of multi-level CSS, each one allocated in a different packet, or a Zadoff-Chu sequence. In this last case, the real and the imaginary parts are transmitted separately, so two data packets are still required for the preamble *d*[n].

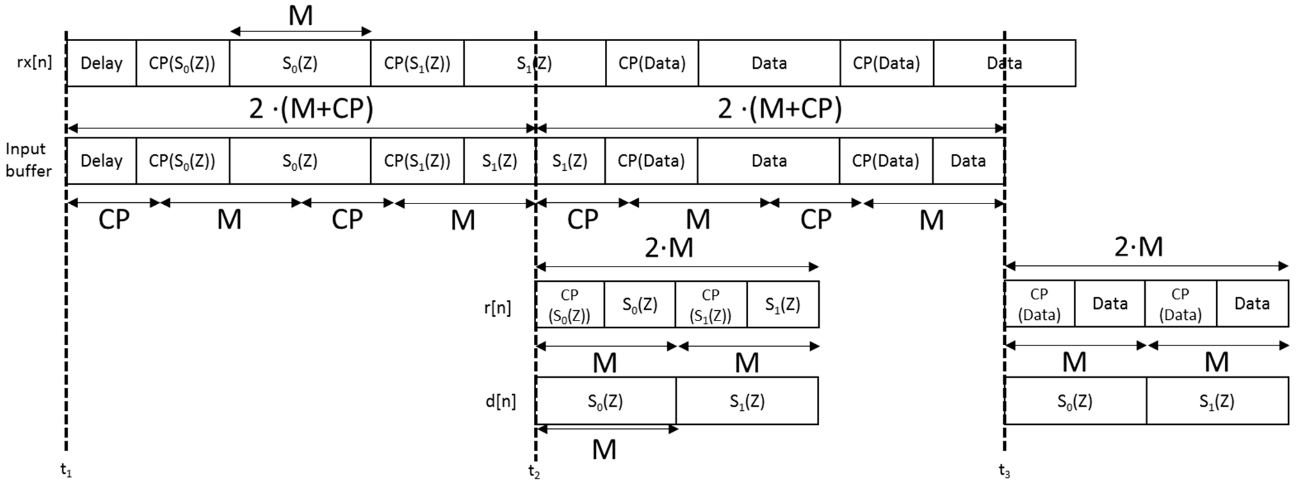

Figure 6. Scheme of the CP removal process, and the corresponding input for the correlation module.

After removing the CP, the obtained signal *r*[*n*] should be correlated with the preamble *d*[*n*] modulated by the synthesis bank. It is assumed a constant preamble, and it only changes whether a different sequence is transmitted. This correlation can be expressed in the frequency domain as Equation (13):

$$x[n] = IDFT\{DFT\{r[n]\} \cdot DFT\{d[n]\}\} \tag{13}$$

where *x*[*n*] is the correlation output; DTF and IDFT mean the direct and inverse Discrete Fourier Transform, respectively; *r*[*n*] is the input buffer after CP removal; and *d*[*n*] is the transmitted preamble, both with a length of 2 *M*.

The fact of implementing both the DFT and the IDFT significantly increases the multiplier consumption in the design. Taking into account the DFT properties, it can be concluded that it is not necessary to implement an IDFT, since it is possible to reutilize the DFT to obtain the time-domain correlation. Simplifying Equation (13):

$$X[k] = DFT\{r[n]\} \cdot DFT\{d[n]\} = R[k] \cdot D[k] \tag{14}$$

Since the IDFT is similar to DFT, with different sign in the exponential factor and different scaling constant, the correlation process can be rewritten as Equation (15):

$$x[n] = IDFT\{X[k]\} = \frac{1}{N} \cdot (DFT\{X^*[k]\})^* \tag{15}$$

where N is the length of the signal to be correlated; and $X[k]^*$ is the complex conjugate of $X[k]$. If it is assumed that the signals $r[n]$ and $d[n]$ are real-valued, it is possible to discard the imaginary part at the correlation output $x[n]$, so:

$$x[n] = \frac{1}{N} \cdot (DFT\{X^*[k]\})^* = \frac{1}{N} \cdot (DFT\{X^*[k]\}) \tag{16}$$

Taking into account Equation (14), it can be obtained Equation (17):

$$x[n] = DFT\left\{\frac{1}{N} \cdot (R[k] \cdot D[k])^*\right\} \tag{17}$$

Figure 7 shows the block diagram proposed for the implementation of the correlation module, according to the optimizations described previously. It can be observed how the 1024-point FFT is reutilized to compute the time-domain correlation.

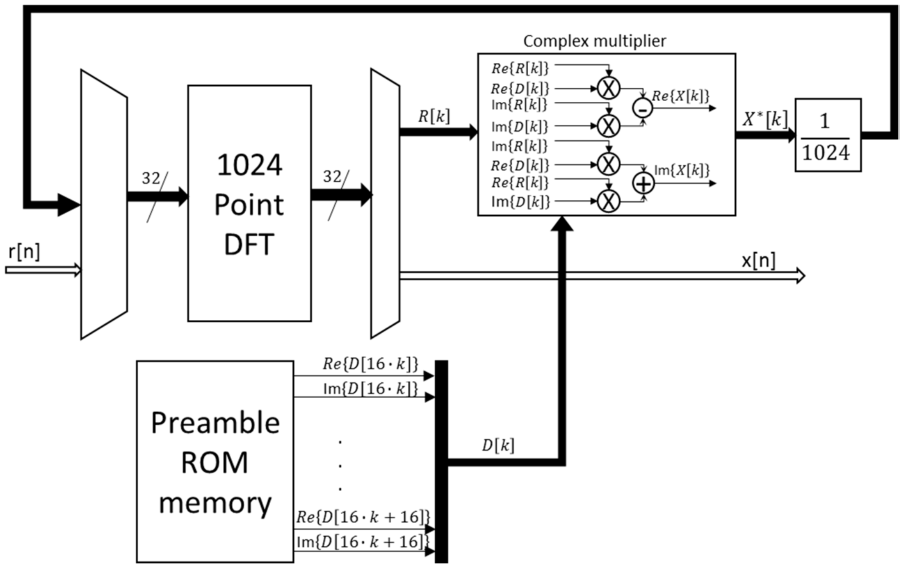

Figure 7. General block diagram of the architecture proposed for the implementation of the correlation module.

At the first stage, the system carries out the DFT for the input signal $r[n]$ with a length of $2 \cdot M$ samples. The DFT block has 32 parallel inputs, half for the real part of the input samples and the other half for the imaginary part. Since the received signal $r[n]$ is real, the imaginary inputs are null at this first stage. Not only the input data but also the outputs are represented in fixed point, with a format Q(18.8).

At the second stage, the complex multiplication between the output data from the DFT of the input signal $r[n]$ and the output data from the DFT of the preamble $d[n]$ is carried out. As has been already mentioned before, since the preamble is constant, its DFT can be computed off-line and stored in the corresponding Preamble ROM memory shown in Figure 7. The available multiplication cells, DSP48E1, are 25 × 18 bits, so, in order to use all the available data span, the stored DFT data for the preamble $d[n]$ have a format Q(25.15). The output of the multipliers is divided by a scaling factor $1/N$ and truncated to the DFT format Q(18.8). Finally, in the last stage, the DFT block is involved again to compute the time-domain correlation, thus providing the output signal $x[n]$ in the format Q(18.8) as well.

The correlation memory block in Figure 5 has been included to store the resulting correlation $x[n]$, so it can be used later in the channel estimation and the computation of the equalizer coefficients. When the correlation module starts to provide valid output data after an initial latency, it is capable of generating 16 18-bit samples $x[n]$ every clock cycle until the final length of 1024 samples. The squaring module consists of 16 multipliers that compute in parallel the squaring of signal $x[n]$ in a format Q(36.16). This squared signal $y[n]$ is truncated to the most significant 18 bits, Q(18.0), since the higher a correlation maximum is, the easier its detection becomes. The squared correlation function $y[n]$ is also stored in a squared memory block to be used in the windowing and thresholding module, whereas the squared values $y[n]$ are also processed by the maximum detector to determine the exact position of the correlation peak. Both the correlation and the squared memory blocks have a size of 64 × 288 bits, corresponding to a set of 16 samples with 18 bits each. This makes possible to read/write a whole set of 16 samples in a single memory access.

The detection of the maximum correlation value consists of a successive and pipelined comparison to determine the maximum value among every set of 16 samples $y[n]$ coming from the squaring module. Every clock cycle, a new set of 16 samples $y[n]$ is inserted, and, by comparing in pairs at every clock cycle, the local maximum is obtained after four clock cycles. A last clock cycle is dedicated to determine the global maximum for the global length of 1024 samples, by comparing the local maximum values from every set of 16 samples. Figure 8 shows the scheme for the implementation of this maximum detector. Since the correlation length is 1024 samples and 16 new samples are inserted every clock cycle, 64 clock cycles plus a latency of five cycles are required to obtain the maximum correlation value and determine its position in the squared memory block.

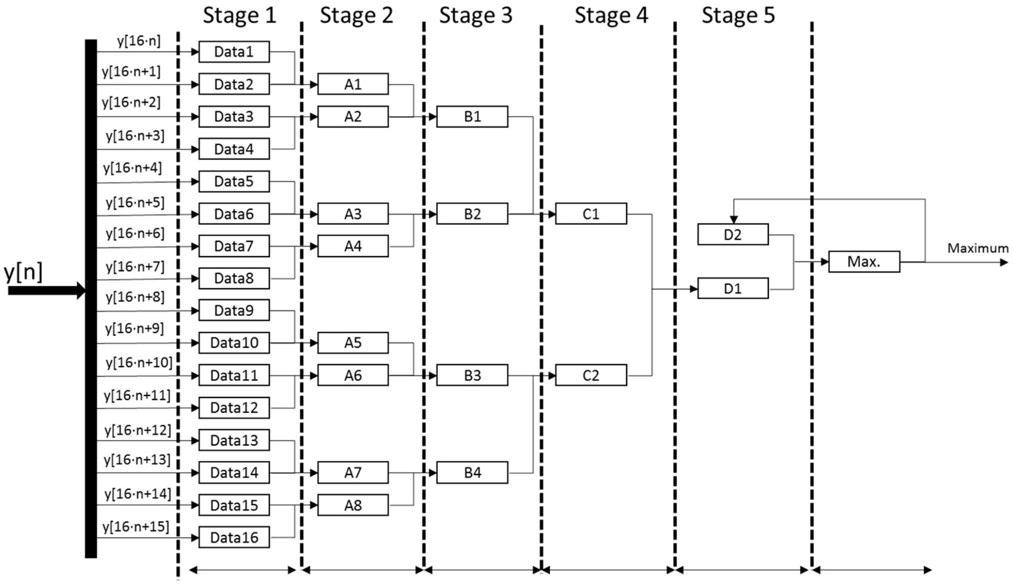

Figure 8. General scheme proposed for the implementation of the detector of the maximum correlation values.

Finally, the windowing and thresholding module is in charge of searching the first sample $y[n]$ over a certain threshold before the detected maximum value, since that sample $y[n]$ is considered as the start of the PLC channel. The windowing is implemented by reading the squared memory block at the position where the detected correlation peak is stored. Since every memory access provides 16 samples $y[n]$ and three reading accesses are carried out, the length of the final window varies from 32 to 47 samples, depending on the exact position of the detected maximum value in the set of 16 samples $y[n]$.

The start of the PLC channel is searched in this window, and it is defined by the first value over a certain threshold. This threshold is experimentally fixed at the 25% of the detected maximum squared value $y[n]$ and it should be updated for every PLC channel condition to effectively estimate the first arriving path. Figure 9 shows the block diagram of the windowing and thresholding module, where the input signals are the detected maximum squared value $y[n]$ and its corresponding location in the squared memory block. This module provides the symbol timing estimation.

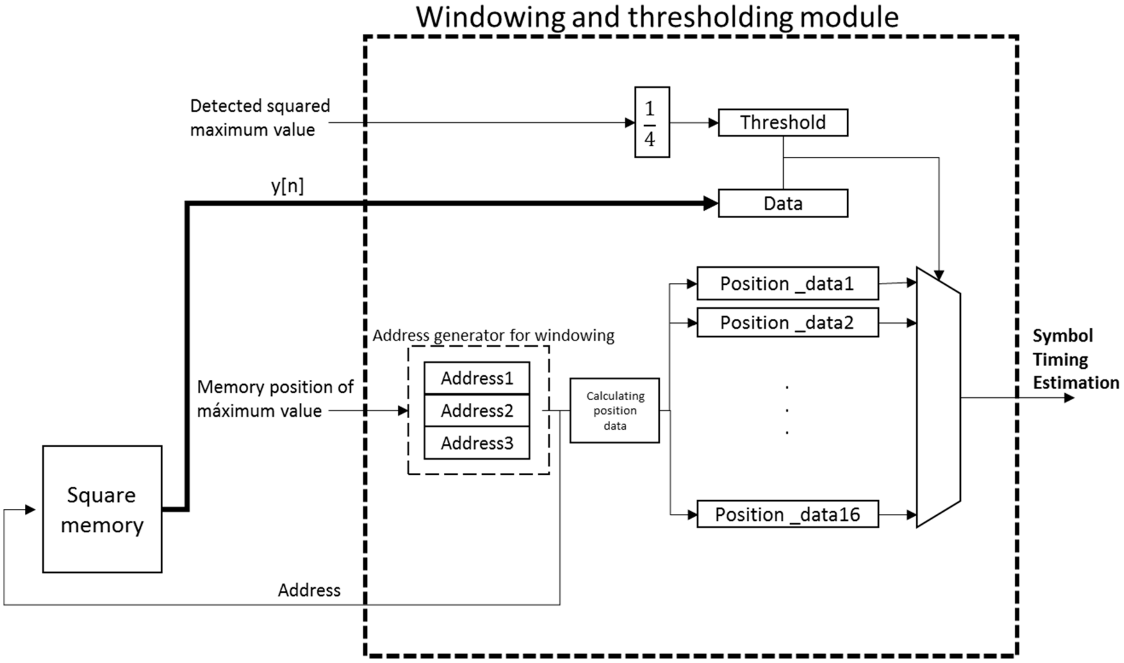
Windowing and thresholding module
Detected squared maximum value
1/4
Threshold
y[n]
Data
Position _data1
Position _data2
Address generator for windowing
Address1
Address2
Address3
Memory position of maximum value
Calculating position data
Symbol Timing Estimation
Square memory
Position _data16
Address

Figure 9. Block diagram of the proposed windowing and thresholding module.

5. Experimental Results

The synchronization algorithm proposed previously has been implemented in a KC705 platform [37] by Xilinx Inc. (San Jose, CA, USA), which is based on a Kyntex XC7K325T FPGA [38]. Regarding the resource consumption, Table 1 shows the figures for the main logic elements available in the device. Generally speaking, the reduced utilization percentage can be observed. Furthermore, Table 2 details this resource consumption for every module involved in the proposed design. In this case, the most of the required resources are dedicated to the correlation module, where it is necessary to implement a FFT block in charge of parallelizing the input data. This minimizes the latency cycles required to obtain every output sample.

Table 1. Resource consumption of the synchronization algorithm implementation in a Xilinx XC7K325T FPGA.

	Proposed Design	Utilization Percentage
Flip-Flops	27,046	6%
RAMB	68	7%
LUTs	21,207	10%
DSP48E1	236	28%

Table 2. Detailed resource consumption for the main modules of the design in a Xilinx XC7K325T FPGA.

	Flip-Flops	**RAMB**	**LUTs**	**DSP48E1**
Global Design	27,046	68	21,207	236
Correlation Module	26,388 (97.57%)	64 (94.12%)	20,587 (97.08%)	220 (93.22%)
Squaring Module	12 (0.04%)	0 (0.00%)	10 (0.05%)	16 (6.78%)
Maximum Detection	325 (1.20%)	0 (0.00%)	343 (1.62%)	0 (0.00%)
Thresholding Module	14 (0.05%)	0 (0.00%)	177 (0.65%)	0 (0.00%)

The proposed design works at a clock frequency of f_{CLK} = 50 MHz, thus requiring 1024 input data every 64 clock cycles (16 18-bit samples every clock cycle), with a latency of 580 cycles. In this case, it is assumed that the sampling frequency at the receiver is f_S = 50 Msps in order to fill up a buffer with a length $2(M + L_{cp})$, for $M = 512$ subchannels and cyclic prefix length $L_{cp} = 400$. In a parallel way, the synchronization delay is estimated for the previous buffer, providing a final estimation after a latency of 580 cycles for a clock frequency of f_{CLK} = 50 MHz. This implies a long enough interval in order to synchronize the receiver and not to discard input samples at the reception. Since the latency may still increase without any drawback for the system operation, the resource consumption in the correlation module could be minimized by parallelizing even more the data input.

Another aspect that has been analyzed is the quantization error due the fixed-point representation used in FPGA-based designs. This study has been focused on the two modules where the most intensive computation is carried out: the correlation and the squaring modules. Note that the maximum detection module and the thresholding module imply comparisons and assignments, but not arithmetical operations where quantization errors could have influence on them. Table 3 shows the maximum relative error and the averaged relative error for the input signal $r[n]$, for the correlation output $x[n]$, and for the squared output $y[n]$, obtained by performing 1000 simulations with different channel models according to [25,36], for a SNR = 10 dB. Due to the fact that all the fractional bits are discarded at the output $y[n]$ of the squaring module, quantization errors become more relevant. Nevertheless, this signal $y[n]$ is only used further to detect maximum values in correlation, so the global performance of the system is not degraded by this decision.

Table 3. Relative quantization errors for the main signals in the proposed design.

	Maximum Relative Error	**Averaged Relative Error**
Input $r[n]$, Q(18.8)	0.055%	0.052%
Correlation output $x[n]$, Q(18.8)	0.132%	0.117%
Squared output $y[n]$, Q(18.0)	7.98%	4.72%

Finally, to validate the proposed synchronization algorithm and its FPGA-based implementation, some simulation results have been performed. The study has been carried out for two different PLC channel model parameters, where signal-noise ratios (SNR) from −5 dB to 30 dB, in steps of 5 dB, have been configured. For every configuration, 1000 simulations have been performed. Regarding the PLC channel model, it is based on the Tonello PLC channel model [36], where two parameter configurations have been considered: the channel B with more favorable parameters [36] and channel A, more complex and obtained from [25]. Table 4 describes the main parameters for both models.

Table 4. Parameters for the PLC channel model considered here [25,36].

	Channel B	**Channel A**
Maximum length	300 m	800 m
Attenuation dependent of the frequency a_0	10^{-5}	0.3×10^{-2}
Attenuation dependent of the frequency a_1	10^{-9}	4×10^{-2}
Poisson arrival time intensity	0.667 m^{-1}	0.2 m^{-1}
Channel length	4 µs	5.56 µs
Stop frequency	31.25 MHz	31.25 MHz

Furthermore, four different types of noise have been evaluated: synchronous impulsive noise, asynchronous impulsive noise, background noise, and narrowband noise; whose power levels are defined in [20]. Figure 10 plots the Root Mean Squared Error (RMSE) for the synchronization delay estimation with respect to the SNR level, taking into account both channel models A and B. It can be observed that the difference between the floating-point algorithm and the fixed-point version implemented in the FPGA device is negligible. Furthermore, it can be verified how the RMSE for the model A is higher than the one for the model B, due to the complexity of this type of PLC channel. In order to compare the proposal with other approaches, Figure 10 also shows the RMSE obtained for the auto-correlation method [22]. Note that its RMSE values are higher than those achieved by the proposed algorithm for both channel models A and B, so the proposed synchronization algorithm improves the performance provided by auto-correlation methods. For every configuration, the RMSE value presents a fluctuation below one sample, which is due to the statistical estimation carried out (1000 simulations for every SNR value). The figures have been obtained by means of multilevel complementary sequences with a length $L = 360$ bits.

On the other hand, Figure 11 shows the cumulative distribution function (CDF) of the absolute error in the synchronization delay estimation for different SNR values and using multilevel complementary sequences. Note that the absolute error is considered as the absolute difference in samples between the estimated synchronization delay and the real delay. The CDF is plotted for both channel models A and B, and not only for the proposed algorithm based on cross correlation but also for the auto-correlation approach. Furthermore, the proposed synchronization algorithm based on cross correlation is evaluated in its corresponding floating- and fixed-point representations. In general terms, the proposed algorithm achieves a better performance than the auto-correlation one, especially for the more complex channel model A regardless the SNR value. In the case of the simpler channel model B, the auto-correlation approach can provide a higher rate of ideal estimation of the synchronization delay, but in case of not achieving a perfect estimation, the errors in the delay estimation are greater than the proposed cross-correlation algorithm. Also, the FPGA-based architecture proposed for the fixed-point implementation provides negligible differences with the floating-point version (note that both plots are almost overlapped in Figure 11). Taking into account the analyzed SNR values, it can be observed the immunity to noise of the proposal, since the performance is not degraded as the noise level increases.

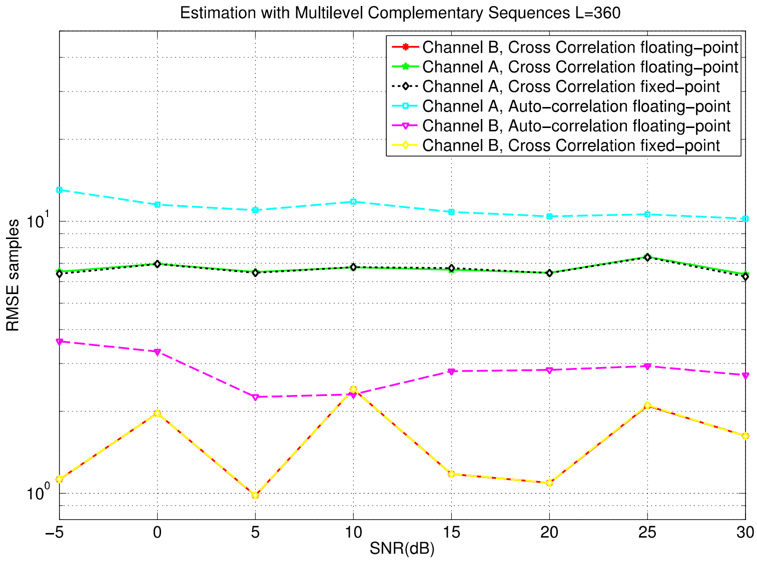

Figure 10. RMSE in the synchronization delay estimation for both channels, and for a floating- and a fixed-point representation and the auto-correlation method.

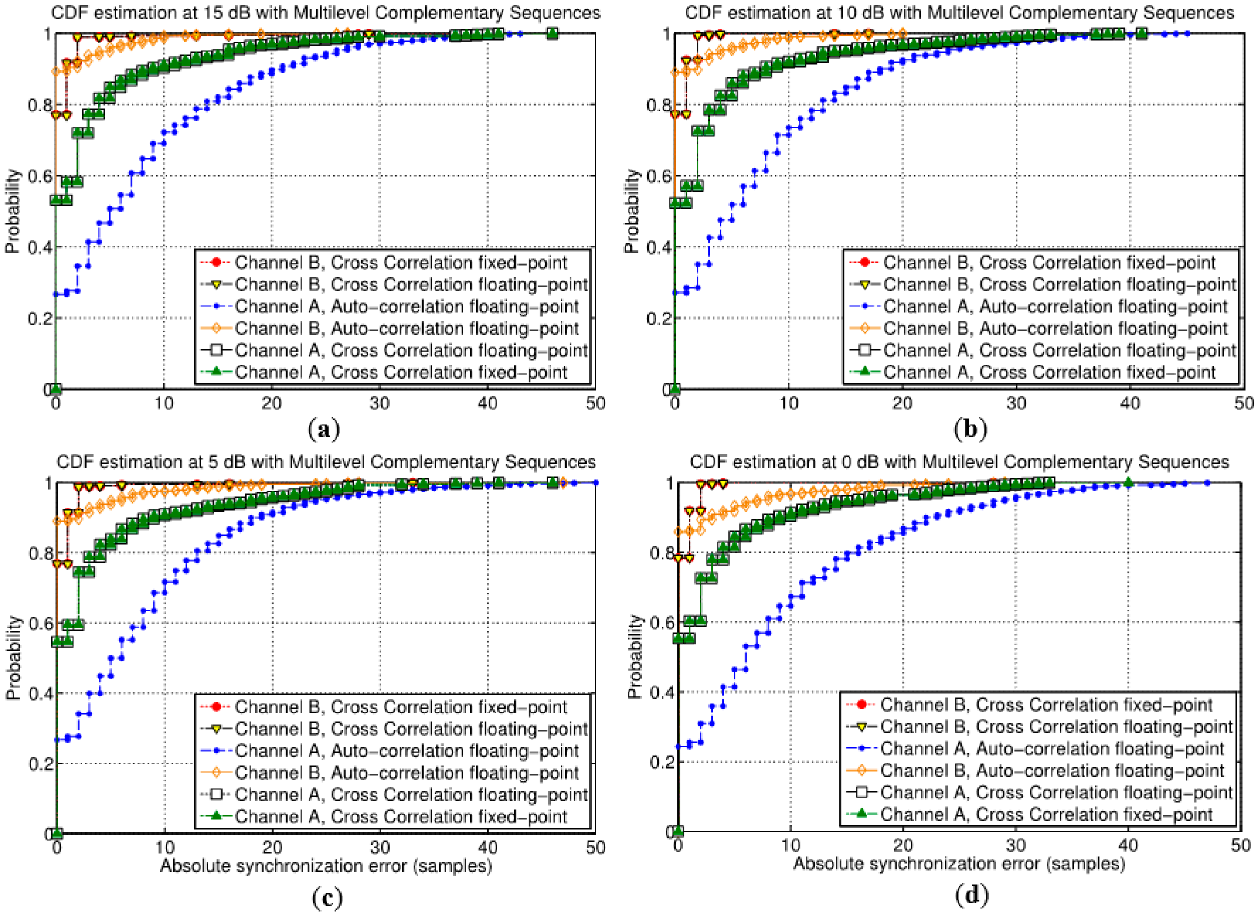

Figure 11. Cumulative distribution function (CDF) of the absolute error in the synchronization delay estimation with different SNR values, (**a**) 15 dB, (**b**) 10 dB, (**c**) 5 dB, (**d**) 0 dB, for both models A and B, for the auto-correlation method and for the floating- and fixed-point versions of the proposed algorithm.

6. Conclusions

The PLC channel shows several particularities, such as severe multipath or non-Gaussian noises in the mains, which prevent the direct use of traditional synchronization techniques in wireless communication systems. In this paper a symbol timing estimator based on cross-correlation is proposed, as well as its efficient hardware implementation in a FPGA device, where the use of multilevel complementary sequences or Zadoff-Chu sequences as pilot symbols provides robustness to the synchronization algorithm. Although the Cyclic Prefix is not needed theoretically for broadband PLC with Wavelet-OFDM as medium access technique, it allows the decrease of the required resource consumption at expense of reducing bandwidth efficiency.

To minimize the number of multipliers involved in the correlation stage, the DFT block has been reused in order to avoid an additional IDFT block, and a pipelined architecture has been designed to minimize the system latency and to maximize data throughput, whereas using the minimum number of hardware resources in the FPGA. The quantization errors in the proposed symbol timing algorithm have been also compared between a fixed-point representation and a float-point representation, showing negligible values in the worst case. The proposed synchronization algorithm is capable of estimating the first arriving path without any errors in the 90% for the Tonello PLC channel model B.

Acknowledgments

This work has been funded by the Spanish Ministry of Economy and Competitiveness (DISSECT-SOC project, ref. TEC2012-38058-C03-03, and LORIS project, ref. TIN2012-38080-C04-01) and by the University of Alcala (iPULSE project, ref. CCG2014/EXP-084, US-PHONE project, ref. CCG2014/EXP-077, and Post-Doc UAH Grants).

Author Contributions

Francisco Nombela and Enrique García designed the algorithm proposal; Francisco Nombela, Álvaro Hernández y Raúl Mateos designed the implementation architecture; and the paper was written by all the authors.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Dostert, K. *Powerline Communications*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 2001.
2. Cavdar, I.H. A solution to remote detection of illegal electricity usage via power line communications. *IEEE Trans. Power Deliv*. **2004**, *19*, 896–900.
3. Son, Y.-S.; Pulkkinen, T.; Moon, K.D.; Kim, C. Home energy management system based on power line communication. *IEEE Trans. Consum. Electron.* **2010**, *56*, 1380–1386.

4. Bumiller, G.; Lampe, L.; Hrasnica, H. Power line communication networks for large-scale control and automation systems. *IEEE Commun. Mag.* **2010**, *48*, 106–113.
5. Mohammadi, M.; Lampe, L.; Lok, M.; Mirabbasi, S.; Mirvakili, M.; Rosales, R.; van Veen, P. Measurement study and transmission for in-vehicle power line communication. In Proceedings of the IEEE International Symposium on Power Line Communications and Its Applications (ISPLC), Dresden, Germany, 29 March–1 April 2009; pp. 73–78.
6. Tanaka, M. High frequency noise power spectrum, impedance and transmission loss of power line in Japan on intrabuilding power line communications. *IEEE Trans. Consum. Electron.* **1988**, *34*, 321–326.
7. Lin, Y.-J.; Latchman, H.A.; Lee, M.; Katar, S. A power line communication network infrastructure for the smart home. *IEEE Wirel. Commun.* **2002**, *9*, 104–111.
8. Sendin, A.; Berganza, I.; Arzuaga, A.; Osorio, X.; Urrutia, I.; Angueira, P. Enhanced operation of electricity distribution grids through smart metering PLC network monitoring, analysis and grid conditioning. *Energies* **2013**, *6*, 539–556.
9. Sendin, A.; Peña, I.; Angueira, P. Strategies for power line communications smart metering network deployment. *Energies* **2014**, *7*, 2377–2420.
10. Farhang-Boroujeny, B. OFDM *versus* filter bank multicarrier. *IEEE Signal Process. Mag.* **2011**, *28*, 92–112.
11. Cruz-Roldán, F.; Blanco-Velasco, M. Joint Use of DFT Filter Banks and modulated Transmultiplexers for multicarrier Communications. *Signal Process.* **2011**, *91*, 1622–1635.
12. Shi, Z.; Wu, Z.; Yin, Z.; Cheng, Q. Novel Spectrum Sensing Algorithms for OFDM Cognitive Radio Networks. *Sensors* **2015**, *15*, 13966–13993.
13. Oudah, A.A.; Rahman, T.A.; Seman, N. Resource Element-Level Computations for Long Term Evolution Networks. In Proceedings of the International Conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia, 3–5 July 2012; pp. 904–908.
14. Hwang, T.; Yang, C.; Wu, G.; Li, S.; Li, G. OFDM and its wireless applications: A survey. *IEEE Trans. Veh. Technol.* **2009**, *58*, 1673–1694.
15. Farhang-Boroujeny, B. Filter bank multicarrier modulation: A waveform candidate for 5G and beyond. *Adv. Electr. Eng.* **2014**, doi:10.1155/2014/482805.
16. Pollet, T.; Peeters, M. Synchronization with DMT modulation. *IEEE Commun. Mag.* **1999**, *37*, 80–86.
17. Stitz, T.H.; Ihalainen, T.; Viholainen, A. Pilot-based synchronization and equalization in filter bank multicarrier communications. *EURASIP J. Adv. Signal. Process.* **2010**, doi:10.1155/2010/741429.
18. Schmild, T.M.; Cox, D.C. Robust frequency and timing synchronization for OFDM. *IEEE Trans. Commun.* **1997**, *45*, 1613–1621.
19. Williams, C.; Beach, M.A.; McLaughlin, S. Robust OFDM timing synchronization. *Electron. Lett.* **2005**, *41*, 751–752.
20. Cortés, J.A.; Díez, L.; Canete, F.J.; Sánchez-Martínez, J.J. Analysis of the indoor broadband power-line noise scenario. *IEEE Trans. Electromagn. Compat.* **2010**, *52*, 849–858.

21. *IEEE Std 1901–2010. IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications*; IEEE: New York, NY, USA, 2010; pp. 1–1586.
22. Minn, H.; Bhargava, V.K.; Ben Letaief, K. A robust timing and frequency synchronization for OFDM systems. *IEEE Trans. Wirel. Commun.* **2003**, *2*, 822–839.
23. Chen, C.; Huang, Y.; Wang, Y.; Yung, C.; Zeng, X. A robust frame synchronization scheme for broadband power-line communication. In Proceedings of the International Conference on ASIC, Xiamen, China, 25–28 October 2011; pp. 212–215.
24. Marques, C.A.G.; de Campos, F.P.V.; Oliveira, T.R.; Menezes, A.S; Ribeiro, M.V. Analysis of a hybrid OFDM synchronization algorithm for power line communication. In Proceedings of the IEEE International Symposium on Power Line Communications and Its Applications, Rio de Janeiro, Brazil, 28–31 March 2010; pp. 44–49.
25. Tlich, M.; Pagani, P.; Avril, G.; Gauthier, F.; Zeddam, A.; Kartit, A.; Isson, O.; Tonello, A.; Pecile, F.; D'Alessandro, S.; *et al*. Deliverable D3.2: PLC Channel Characterization and Modelling. Available online: http://www.ict-omega.eu/fileadmin/documents/deliverables/Omega_D3.2_v1.2.pdf (accessed on 20 July 2015).
26. Mefenza, M.; Bobda, C. FPGA implementation of subcarrier index modulation OFDM transceiver. In Proceedings of the Parallel and distributed Processing Symposium Workshops and PhD Forum (IPDPSW), Cambridge, MA, USA, 20–24 May 2013; pp. 268–272.
27. Berg, V.; Doré, J.-B.; Noguet, D. A flexible radio transceiver for TVWS based on FBMC. *Microprocess. Microsyst.* **2014**, *38*, 743–753.
28. Poudereux, P.; Mateos, R.; Hernández, A.; Cruz-Roldán, F.; Oses, D. FPGA-based implementation of a filter bank-based transmultiplexer for multicarrier communications. In Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, Brazil, 16–19 September 2014; pp. 1–6.
29. Villadangos, J.M.; Ureña, J.; García, J.J.; Mazo, M.; Hernández, A.; Jiménez, A.; Ruiz, F.D.; de Marziani, C. Measuring time-of-flight in an ultrasonic LPS system using generalized cross-correlation. *Sensors* **2011**, *11*, 10326–10342.
30. Gutiérrez-Fernández, C.; Jiménez, A.; Martín-Arguedas, C.J.; Ureña, J.; Hernández, A. A novel encoded excitation scheme in a phased array for the improving data acquisition rate. *Sensors* **2014**, *14*, 549–563.
31. García, E.; Ureña, J.; García, J.J. Generation and correlation architectures of multilevel complementary sets of sequences. *IEEE Trans. Signal Process.* **2013**, *61*, 6333–6343.
32. Chu, D.C. Polyphase codes with good periodic correlation properties. *IEEE Trans. Inf. Theory* **1972**, *18*, 531–532.
33. Fort, A.; Weijers, J.-M.; Derudder, V.; Eberle, W. A performance and complexity comparison of auto-correlation and cross-correlation for OFDM burst synchronization. In Proceedings of the 2003 IEEE International Conference on Acoustics, Speech and Signal Processing, Hong Kong, China, 6–10 April 2003; pp. 341–344.
34. Nombela, F.; García, E.; Ureña, J.; Hernández, Á.; Poudereux, P. Robust synchronization for broadband PLC based on Wavelet-OFDM. In Proceedings of the 2015 IEEE Emerging Technology and Factory Automation (ETFA), Luxembourg City, Luxembourg, 8–11 September 2015; pp. 1–7.

35. Muquet, B.; Wang, Z.; Giannakis, G.B.; de Courville, M.; Duhamel, P. Cyclic Prefixing or Zero Padding for Wireless Multicarrier Transmissions? *IEEE Trans. Commun.* **2002**, *50*, 2136–2148.
36. Tonello, A.M.; Versolatto, F.; Béjar, B.; Zazo, Z. A fitting algorithm for random modeling the PLC channel. *IEEE Trans. Power Deliv.* **2012**, *27*, 1477–1484.
37. Xilinx Inc. KC705 Evaluation Board for the Kintex-7 FPGA. Available onlone: http://www.xilinx.com/support/documentation/boards_and_kits/kc705/ug810_KC705_Eval_Bd.pdf (accessed on 23 July 2015).
38. Xilinx Inc. 7 Series FPGA Overview. Available online: http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf (accessed on 23 July 2015).