Performance Improvement of PLC Channel Estimator and ASCET Equalizer in a FBMC Transmultiplexer Based on a Multi-Core Solution

RUBÉN NIETO, (Member, IEEE), RAÚL MATEOS, AND ÁLVARO HERNÁNDEZ, (Senior Member, IEEE)
Department of Electronics, University of Alcalá, 28805 Alcalá de Henares, Spain
Corresponding authors: Rubén Nieto (ruben.nieto@uah.es), Raúl Mateos (raul.mateos@uah.es), and Álvaro Hernández (alvaro.hernandez@uah.es)
This work was supported by the Ministerio de Ciencia, Innovación y Universidades (MICROCEBUS Project and Inercia0 Project), under Grant ref. RTI2018-095168-B-C51 and Grant ref. RTI2018-098865-B-C33.

ABSTRACT Power-Line Communication (PLC) employs multi-carrier modulations, such as Filter-Bank Multi-Carrier (FBMC), to improve communications through the PLC channel and provide an efficient use of the spectrum, thus allowing higher data rates. Since one of the main drawbacks is the noisy channel with multipath and frequency-selective fading, the receiver typically includes a channel estimator and equalizer, at the expense of increasing the computational load and complexity of that receiver and making it difficult to obtain real-time solutions. In this context, this work proposes a heterogeneous System-on-Chip (SoC) architecture for the real-time implementation of a FBMC transmultiplexer that involves the channel estimation and equalization at the reception stage. For that purpose, the performance of a multi-core approach is evaluated for both the Zynq® 7000 SoC and the Zynq® UltraScale+ (US+) devices, by using a single-, dual- and quad-core solution to perform the channel estimation and the calculation of the equalizer coefficients in the ARM processor available in the proposed architecture. Two approaches have been analysed for the necessary synchronization among cores; one based on atomic instructions and the other one on interrupts. The dual-core proposal in both Zynq® 7000 and Zynq® US+ provides a x2 acceleration compared to the single-core proposal, whereas the quad-core one in Zynq® US+ does not provide a x4 acceleration as expected, due to the timing overheads of the synchronization among cores imposed by the data dependencies existing in these tasks. Experimental results include the evaluation of the processing times for each task in the algorithm, as well as the acceleration obtained by each proposal. The proposed architecture can be easily applied to other processing algorithms that may take advantage of the parallelism provided by the multi-core approach in a more efficient way, depending on their data dependencies.

INDEX TERMS Power-line communication, channel equalizer, channel estimation, multiprocessor system-on-chip.

I. INTRODUCTION

Power-Line Communication (PLC) can be used in numerous fields, such as Smart Grids to communicate different energy systems and achieve an efficient energy distribution [1], or Internet of Things (IoT) to send data through the mains while supplying power to devices [2], [3]. Furthermore, in the automotive [4], [5], aeronautical [6] or naval sectors [7], it is used to establish the communication between the sensors and actuators deployed in a vehicle, plane or vessel, thus reducing the required wiring and its weight. In addition, the appearance of new technologies and products, such as the electric vehicle [8] or the intelligent metering systems [9], has helped consider PLC as a low-cost communication method, whereas several institutions have developed some standards for PLC in the last decade, such as [10].
Nevertheless, PLCs still present some challenges and issues that should be tackled in coming years, particularly in broadband applications. Broadband PLC (BB-PLC) provides higher transmission rates at the expense of increasing the computational complexity [11]. These high transmission rates in BB-PLC require frequencies above 1.8MHz, and up to 86MHz [12]. On the other hand, the PLC channel is another aspect that complicates its deployment. In general terms, the channel is affected by frequency selective fading, due to reflections and transmissions caused by impedance mismatches at different discontinuities of the mains; furthermore, it also increases the multipath propagation [13], [14]. It is also worth mentioning the significant level of noise with strong stationary, cyclostationary and non-stationary components, which are mainly coming from radio-frequency systems, other PLC nodes or Digital Subscriber Line (DSL) lines, among others [15]. In this context, the Tonello’s channel models consider a deterministic propagation model [12], consisting of multipath propagation, reflections due to a variety of impedances presented by the terminals of the mains, and the length of links. These Tonello’s models allow the generation of a random channel model with certain characteristics that define the variability of these channels in households or industries.

The IEEE 1901-2010 standard [10] has already considered the use of Filter-Bank Multi-Carrier (FBMC) modulations, as a way to deal with the physical layer of the link. FBMC is based on applying digital filters to improve the spectral efficiency. In addition, FBMC reduces the Inter-Carrier Interference (ICI), since the prototype filters have lower sidelobes. In [16] FBMC and Orthogonal Frequency Division Multiplexing (OFDM) have already been studied and compared for PLC, resulting in a better performance of FBMC, with higher bit rates and robustness against noise. Nevertheless, the increase of the computational load in FBMC is significant. This is particularly relevant at the receiver stage, where, besides, any modulation requires a synchronization method between transmitter and receiver, as well as an equalization method to reduce ICI.

In order to improve the robustness and reliability of PLC, channel estimation techniques based on sending pilot symbols included in the transmission frame are often used, whereas the channel estimates are obtained at the reception stage. In multi-carrier systems, these pilot symbols can be considered as a preamble to the transmission burst, or they can be distributed over time in different sub-carriers [18]. On the other hand, equalization methods are also used to improve the received signal quality, implying again a significant increase in the complexity of that receiver.

The above mentioned techniques require a high computational load, which is the reason why System-on-Chip (SoC)-based solutions become suitable for them. SoCs incorporate processors, peripherals, memory systems or configurable logic in the same die. They are complex devices, but, with the combination of the programmable logic, they provide a great versatility and flexibility, allowing different levels of parallelism to be applied to the architecture at the same time [19]. Furthermore, the existence of multi-core processors also implies another type of parallelism, making it possible to divide tasks into different cores to achieve simultaneous processing.

This work presents a mixed hardware/software (HW/SW) architecture oriented to a multi-core approach, that is capable of achieving real-time implementations of a FBMC transmultiplexer for Power-Line Communication (PLC), also including the corresponding channel estimator and an Adaptative Sine-modulated/Cosine-modulate Equalizer for Transmultiplexer (ASCET) equalizer. The main contribution of this work is this innovative and parallel solution, that overcomes common constraints when implementing PLC medium access techniques in real time. The proposal evaluates the obtained performance in detail when the heterogeneous architecture is implemented on a Xilinx Zynq® 7000 and a Zynq® UltraScale+ (US+) devices. For that purpose, a partitioning of the tasks from the algorithms involved in the FBMC receiver is presented, where the channel estimation and the equalizer coefficient calculation are carried out by the cores available in the software part. On the other hand, the rest of the FBMC receiver is implemented in specific hardware peripherals, integrated in the SoC architecture. To our best knowledge, it is not possible to find other previous works dealing with similar heterogeneous architectures for broadband communications, either in PLC or in any other standard. Nevertheless, in [22] a finite precision analysis of the whole FBMC architecture implemented only in hardware (Programmable Logic) was already proposed. In [28] a quantitative analysis in terms of Signal-Noise Ratio (SNR), Bit Error Rate (BER) and Root Mean Square Error (RMSE) for a preliminary mixed HW/SW architecture using only a single core was also provided. However, in this previous proposal [28], with only one core, the final results did not allow to fulfill the timing constraints defined by the standard, whereas SNR and BER remain comparable with previous works. Finally, a dual-core proposal for Zynq® 7000 has been presented in [31], where the resulting processing times are much shorter, actually divided by a factor x2, thus proving that a mixed HW/SW approach with multiple cores may improve performance figures significantly, while saving hardware resources. In this way, this work becomes an innovative proposal about how to achieve real-time SoC architectures in PLC systems, taking advantage of the full parallelism provided by the different elements existing in current Field-Programmable Gate Array (FPGA) devices, such as multiple cores and specific hardware peripherals designed for the application.

The rest of the manuscript is organized as follows: Section II presents the proposed FBMC transmultiplexer, including the description of the channel estimation and the calculation of coefficients for the ASCET; Section III explains the proposed HW/SW architecture; Section IV deals with the single-core approach based on the Zynq® 7000 device; Section V details a dual-core...
architecture on Zynq® 7000 and a quad-core one on Zynq® US+; Section VI shows some experimental results obtained for the evaluation of the achieved performance improvement, as well as the corresponding processing times; and, finally, conclusions are discussed in Section VII.

II. DESCRIPTION OF THE CHANNEL ESTIMATION AND EQUALIZATION

The IEEE 1901/2010 standard [10] for BB-PLC establishes the specification for a physical layer access based on windowed OFDM and on FBMC (Wavelet OFDM). In order to improve the received signal, an equalization method is included at the reception stage. In this case, an ASCET frequency domain equalizer (FEQ) has been considered in [23]. Since the standard does not actually define any particular channel estimation technique to be applied, it is possible to implement those that better fit the system requirements.

The FBMC modulation explained hereinafter has already been described mathematically in [21]–[23]. It consists of a Cosine-Modulated Filter-Bank (CMFB) at the transmission and reception stages, which uses the even type-4 Discrete Cosine Transform (DCT4e) instead of the Fast Fourier Transform (FFT) for a better energy compression. Also, the direct and inverse transformations of the DCT4e are identical, so the transmitter and the receiver actually implement similar architectures.

Fig. 1 shows the block diagram of the FBMC transmultiplexer with ASCET equalizer. The transmitter implements the CMFB; on the other hand, the reception stage involves not only the CMFB but also the Sine-Modulated Filter-Bank (SMFB), together with the ASCET equalizer. Note that the filter bank is the same in both, so that block is not replicated.

The mathematical description of the CMFB transmitter is shown in (1), where \( \Theta \) is a \( M \times M \) diagonal matrix, whose elements are \( \Theta = \cos(\theta_k) \); \( \theta_k \) is the phase angle for every subcarrier defined in [10, Table 14-10]; \( \mathbf{C}_{4e} \) is the \( M \times M \) DCT4e matrix [24], [25]; \( \mathbf{I} \) and \( \mathbf{J} \) are the \( M \times M \) identity and exchange matrices, respectively; \( \mathbf{g}_0 \) and \( \mathbf{g}_2 \) are the coefficients of the filter bank, with a size of \( M \times 2M \); and, finally, \( \mathbf{V}[k] \) and \( t[n] \) are the \( 1 \times M \) input and output signals, respectively.

\[
\mathbf{t}[n] = \mathbf{V}[k] \cdot \begin{bmatrix} \mathbf{g}_0 & \mathbf{z}^{-2} \mathbf{g}_2 \end{bmatrix} \cdot \begin{bmatrix} (\mathbf{I} - \mathbf{J}) & (\mathbf{I} + \mathbf{J}) \end{bmatrix} \cdot \mathbf{C}_{4e} \cdot \Theta
\]

With regard to the filter bank, it uses the prototype filter proposed in the IEEE 1901/2010 standard [10]. The coefficients of this filter can be obtained as described in [23, Section 2.2]. This filter \( G_T(z) \) can be expressed as (2), where only the first and third coefficients are not null.

\[
G_T(z) = g_{s,0} + g_{s,2} \cdot z^{-2}
\]

On the other hand, the description of the CMFB and SMFB at the reception stage is shown in (3) and (4), respectively. Note that they are similar to the transmission stage, where \( \mathbf{D}_{4e} \) is the \( M \times M \) even type-4 Discrete Sine Transform (DST4e) matrix [24], [25]; \( \mathbf{r}[n] \) is the \( M \times 1 \) input of the receiver, affected by the PLC channel transfer function and the corresponding noise; and the \( M \times 1 \) output signals are \( \hat{\mathbf{X}}_C[k] \) for the CMFB and \( \hat{\mathbf{X}}_D[k] \) for the SMFB. Note that the operator \( T \) indicates the matrix transposition.

\[
\hat{\mathbf{X}}_C^T[k] = \Theta \cdot \mathbf{C}_{4e} \cdot \begin{bmatrix} (\mathbf{I} + \mathbf{J}) & (\mathbf{I} - \mathbf{J}) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{g}_0 & \mathbf{z}^{-2} \mathbf{g}_2 \end{bmatrix} \cdot \mathbf{r}[n]
\]

\[
\hat{\mathbf{X}}_D^T[k] = \Theta \cdot \mathbf{D}_{4e} \cdot \begin{bmatrix} (\mathbf{I} - \mathbf{J}) & (\mathbf{I} + \mathbf{J}) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{g}_0 & \mathbf{z}^{-2} \mathbf{g}_2 \end{bmatrix} \cdot \mathbf{r}[n]
\]

A per-subcarrier equalizer is often applied, where the most common one in FBMC systems is the ASCET, based on a bank of Finite Impulse Response (FIR) filters [5], [26]. It requires two parallel filter banks, known as CMFB and SMFB, based on the DCT4e and DST4e, respectively. Every output from both filters is driven into the equalizer that performs the filtering for every subcarrier, thus obtaining the equalized signal \( \hat{\mathbf{V}}_m[k] \). The mathematical description of the equalizer is shown in (5), where \( \hat{\mathbf{X}}_C[k] \) and \( \hat{\mathbf{X}}_D[k] \) are the outputs from the CMFB and the SMFB, respectively, for \( m = 1, 2, \ldots, M \) subcarriers, and the \( L \)-ASCET coefficients are \( c_{1,m} \) and \( d_{1,m} \), which are applied to the signals \( \hat{\mathbf{X}}_C[k] \) and \( \hat{\mathbf{X}}_D[k] \), respectively. Note that \( L \) is the order of the \( L \)-ASCET equalizer.

\[
\hat{\mathbf{V}}_m[k] = \sum_{l=0}^{2L} c_{1,m} \cdot \hat{\mathbf{X}}_C[k - l] + \sum_{l=0}^{2L} d_{1,m} \cdot \hat{\mathbf{X}}_D[k - l]
\]

In order to perform the equalization, it is necessary to have the channel estimate available. However, the PLC channel presents variations over time, mainly due to the different devices connected to the power grid. This means that the channel cannot be considered as a Linear Time-Invariant (LTI) system, and it is necessary to carry out a channel estimation periodically to perform the later equalization correctly.
significantly change. Fig. 2 shows the proposed structure of the transmission stream. It consists of the preamble and the data sets of samples. The preamble consists of four sets of \( M \) samples; two of them, \( S_0 \) and \( S_1 \), are used to carry out the synchronization between the transmitter and the receiver, and the other two sets, \( C_0 \) and \( C_1 \), are dedicated to perform the channel estimation at the reception stage. After the preamble, the transmitter sends the data sets \( D_x \) with \( M \) samples each. A data set with \( M \) samples is transmitted every 8.192 \( \mu \)s, according to the IEEE 1901/2010 standard [10] for \( M = 512 \) subcarriers and a sampling frequency of 62.5 MHz. Note that in this work the transmitter and the receiver are fully synchronized and only the channel estimation is considered hereinafter. The estimation sequences should be transmitted every certain interval; actually, they should be resent before the coherence time \( t_c \) is exceeded.

In [26], a Least Squares (LS) estimator is used for the PLC channel. This estimator tries to minimize the error between the transmitted and the received signals, obtaining in this way the channel estimate. It is based on the transmission of certain sequences, considered as a preamble. The sequences should have suitable auto- and cross-correlation properties. In this case, Complementary Sets of Sequences (CSS) are used to obtain the channel estimate, and they are sent through the channel as a preamble (\( C_0 \) and \( C_1 \)) before the data sets (\( D_x \)). The LS estimator is shown in (6), obtaining the corresponding estimates \( \hat{H}_{LS} \) for each transmitted symbol. Note that it is necessary to calculate the FFT of the received symbols \( r_c[n] \), denoted as \( R_{C,x}[k] \) in (6), where \( x = \{0,1\} \) is the index of the corresponding CSS sequence. On the other hand, the FFT of the transmitted symbols \( C_i[k] \) can be computed previously offline, and not necessarily at run time.

\[
\hat{H}_{LS}[k] = \frac{R_{C,x}[k]}{C_i[k]} = R_{C_i}[k] \cdot K_{C_i} \tag{6}
\]

The ASCET coefficients are obtained from the channel estimation previously performed. For this, in [5] and [23] the calculation of coefficients for ASCET equalizers with different \( L \)-orders is detailed. It is worth noting that the number of stages in the FIR filters involved in the ASCET equalizer depends on its order \( L \), according to \( 2 \cdot L + 1 \). Consequently, the number of equalizer coefficients also depends on the order \( L \). Furthermore, since the obtained channel estimate has the same number of points as the number \( M \) of subcarriers, sometimes it is necessary to apply a linear interpolation to match the resolution required by the ASCET equalizer, still depending on the order \( L \). This interpolation is carried out according to a factor \( R \) equal to 2, 4 and 8, for the 0-, 1- and 2-ASCETs, respectively. Note that it is not considered a higher ASCET order than 2, since it has already been proved that it does not provide any relevant improvement of the system performance [5].

To clarify the structure of the channel estimation algorithm, Fig. 3 shows the block diagram of the different stages according to the order \( L \) of the \( L \)-ASCET equalizer.

The block \textit{LS Estimator} performs the FFT of the received symbols \( r_C[n] \) and the product by the inverse of the transmitted symbols \( \frac{1}{R_C[R]} \), already computed offline, as described in (6).

On the other hand, the block \textit{Mean and Thresholding} performs the arithmetic mean from both estimators \( \hat{H}_{LS} \) and the result is thresholded to mitigate the effect from the PLC channel noise on the high-amplitude frequency components. Then, depending on the \( L \) order of the ASCET equalizer, a linear interpolation is made by the factor \( R \), mentioned above. Finally, the calculation of coefficients is performed, obtaining the coefficients \( c_{i,m} \) and \( d_{i,m} \) for the ASCET equalizer, which will be applied to the CMFB and SMFB, respectively, as described in (5).

The rest of the manuscript is focused on the receiver stage, since it presents a higher complexity than the transmitter, involving a channel estimator and equalizer. Therefore, the receiver implies some challenges in the design, that will be tackled by the mixed HW/SW architecture proposed.

### III. PROPOSED HARDWARE/SOFTWARE ARCHITECTURE

In [21] a similar solution of the FBMC receiver was proposed for an architecture fully implemented in hardware (Programmable Logic - PL). This approach definitively showed the massive resource consumption required, with a utilization percentage of DSP cells around \( \approx 50 \% \) for each transmitter and receiver in a Xilinx Virtex6 (XC6VLX240T) FPGA. These figures were determinant to come up with a mixed HW/SW architecture, capable of exploiting the parallelism provided by the multiple cores existing in current FPGA devices. Taking into account the aforementioned description of the channel estimation and equalization, the block diagram of the mixed HW/SW architecture considered hereinafter is shown in Fig. 4. It includes the CMFB-based transmitter and receiver, as well as the channel estimation and the coefficient calculation for the ASCET equalizer at the reception stage.
It should be noted that the receiver contains two filter banks in parallel, (CMFB and SMFB). Finally, the FBMC transmitter includes the block Estimation Sequences, where the estimation frames, \( C_0 \) and \( C_1 \), are stored ready for transmission as pilot symbols, in order to obtain the channel estimate.

The blocks to be implemented in the hardware part (PL) and those to be included in the software part (Processing System - PS) have been selected according to the time restrictions imposed by the IEEE 1901/2010 standard [10]. This partitioning is designed to reduce the resource consumption in the PL, trying to re-utilize resources over time, with special interest in the FFT module required by the channel estimation since it consumes a significant number of them. In this case, the same criteria as in [28] have been applied:

- Firstly, the reduction of the logic resources involved in the HW implementation. The channel estimation requires the implementation of a FFT to obtain \( R_C[k] \). Since the FFT is a highly demanding module, a software approach for that task may save a significant amount of logic resources.

- Secondly, those blocks in the algorithm with less demanding timing constraints can be computed by the processors, whereas the others are implemented in the programmable logic to meet the timing constraints. Note that the channel estimation should be run every 600\( \mu \)s roughly, to avoid exceeding the channel coherence time, whereas a data set is received every 8,192\( \mu \)s, according to the IEEE 1901/2010 standard [10]. This implies that the channel estimation and the corresponding coefficient calculation for the equalizer can be carried out in parallel to the FBMC filter bank, thus making a software implementation of these two tasks feasible.

Having a mixed HW/SW architecture, where there is an exchange of information between both parts, requires to establish a mechanism to do it in an efficient way. For that purpose, the Direct Memory Access (DMA) controller is dedicated to exchange data between the PL and the PS. Additionally, the Accelerator Coherency Port (ACP) connected to the DMA controller is used to keep cache consistency between PL and PS. In this case, when the channel estimation frame is received, a DMA transfer is performed to provide the data to the PS. When the \( L \)-ASCET coefficients have been calculated, another DMA transfer is performed in order to make these coefficients available in the PL for the equalizer.

Fig. 4 depicts the reception stage divided into two parts that can operate in parallel: the filter banks (CMFB and SMFB), implemented in the PL; and the blocks dedicated to the channel estimation, interpolation and coefficient calculation, computed in the PS. From left to right, the FBMC transmitter processes the input signal \( V_m[k] \), and generates the output \( w_m[n] \). This output is serialized and the preamble is inserted before exceeding the coherence time \( t_{CPLC} \) to carry out the channel estimation at the receiver. An Analog Front-End is also used to connect the FBMC digital processing modules to the analog communication channel. The signal \( r[n] \) received from the channel is deserialized and
separated into the received estimation sequences \( r_{C_n}[n] \) and the \( M \) samples \( r_{w_n}[n] \), which will be processed by the FBMC receiver. Note that the channel estimation, interpolation and coefficient calculation must be performed before the signals \( \hat{X}_{c_n}[k] \) and \( \hat{X}_{d_n}[k] \) are available at the input of the L-ASCET equalizer, in order to implement the equalization correctly. Finally, the result of the equalization \( \hat{V}_{m}[k] \) is obtained.

As shown in [28], the equalizer coefficients should be available before the data are obtained at the output of the filter banks CMFB and SMFB, in order to properly implement the corresponding equalization. Therefore, the latency of these banks determines a temporal limitation for the blocks implemented in the PS. This latency is 55 \( \mu \)s, coming from 550 cycles with a clock period of 10 ns. It should be noted that this PL implementation has been analyzed previously in [22], so this work is focused on the software part of the architecture implemented in the PS and the study of the multicore approach.

The Xilinx SoC devices integrate an FPGA with ARM processors. In particular, some Zynq® 7000 family devices include up to two processing cores and the Zynq® US+ family up to four cores. This allows the definition of mixed HW/SW architectures where the tasks are shared between the different parts of the architecture. The heterogeneous architecture, proposed in this article, has been particularized for a single core, a dual core and a quad processor, based on two Xilinx FPGA families: Zynq® 7000 and Zynq® US+, integrated in the platforms ZC706 and ZCU102, respectively. Both approaches are tackled in detail hereinafter.

IV. SINGLE-CORE APPROACH

After describing the algorithms involved in the FBMC transmitter and receiver, a single-core architecture has implemented the proposal previously described in a mathematical way. The single-core approach is based on the Zynq® 7000 device (ZC706 development platform), that integrates a dual-core ARM Cortex™ A9 processor, configured with a clock frequency of 800 MHz. A set of software optimizations has been carried out, mainly related to the employ of fixed-point arithmetic to speed up the calculation in the processor. In addition, the NEON Vector Unit provided by the ARM Cortex™ A9 is applied to the FFT calculation as well. The results for this proposal are presented in [28]. Due to the complexity of the algorithm and the required computational load, it is not possible to implement the whole processing in the PS in a time shorter than the 55 \( \mu \)s of latency determined by the CMFB and SMFB filter banks on the hardware side. Furthermore, in [28] the results for three equalization orders (0-, 1- and 2-ASCET) are also shown. Consequently, the processing times obtained in the software part by this single-core approach are longer than the latency of 55 \( \mu \)s introduced by the filter bank at the reception stage, although an alternative solution could be inserting some intermediate memories in the PL to delay and equalize the data sets correctly when the coefficients are already available. However, this alternative implies an extra logic resource consumption.

Since the timing constraints are not achieved with the single-core approach, an analysis of a multi-core proposal is studied hereinafter in order to speed up the software processing, thus meeting the timing constraints determined by the hardware part and guaranteeing the real-time operation of the PLC receiver.

V. MULTI-CORE APPROACH

A dual-core implementation has been proposed on Zynq® 7000, taking advantage of all the available cores in the device. Furthermore, the use of the devices existing in the Zynq® US+ family makes it possible to develop a quad-core solution. Both architectures and their corresponding performances have been evaluated and compared below.

In order to make the multi-core proposal feasible, it is necessary to establish an inter-core synchronization method, so they can notify each other that a certain point in the algorithm has been reached. This allows the data provided at the output of certain blocks to be validated and then used at the input of the following ones. It is worth noting that these points are necessary since the datapath sections implemented by each core are not independent, and the blocks need samples coming from several paths. Besides, since data must be available for all the cores, it is necessary to establish a memory region shared by all of them. In this case, the On-Chip Memory (OCM), available in the Zynq® 7000 and the Zynq® US+ families, has been selected for that purpose. It has a reduced size (256 KB), but large enough to store the variables required in the proposal.

A. DUAL-CORE PROPOSAL

The multi-core proposal requires the partitioning of tasks among the available cores. According to the algorithm, to carry out the channel estimation, the receiver acquires the preamble \( C_0 \) and \( C_1 \) with both CSS sequences of \( M \) samples. For every sequence, a \( M \)-point FFT is performed, followed by the LS estimation and the average of both estimates. As shown in Fig. 3, the FFT and the initial estimation can be tackled in parallel, since each CSS sequence is managed independently until the average of the channel estimates is obtained. Furthermore, the CSS sequences can be divided into smaller subsets, so two or four cores, depending on the case under study, can cope with the different stages of the processing in a partial way.

With regard to the inter-core synchronization method in the dual-core proposal with Zynq® 7000, two approaches have been considered. The first one consists in using software interrupts to define a notification mechanism among cores; in this case the Software Generated Interrupts (SGI) from the Zynq-7000 architecture [29] are used by each core. The second one is based on the atomic instruction set (LDREX and STREX), provided by ARM architecture [30], to define mutual exclusion variables (mutex) that allow the running program to be monitored. In the first case, each core gen-
erates a software interrupt addressed to the other core. Note that every core remains waiting in an infinite loop, until it receives the interruption from the other core that enables it to proceed with the following task. In order to reduce the interrupt service latency, another option is to apply a mutex-based approach using the set of atomic instructions. These instructions support atomic updates with exclusive monitors that track exclusive memory accesses. The mutexes must be stored in a shared memory region that supports exclusive accesses. Both cores work in parallel and one remains locked when it reaches the synchronization point, until the other unlocks the corresponding mutex.

In [31] both inter-core synchronization methods have been described and evaluated to determine their performance. Fig. 5 shows the block diagram of the algorithm divided into two cores, showing the inter-core synchronization points along the processing. It is worth mentioning that the inter-core synchronization time required by the approach based on the atomic instruction set is slightly shorter. It is only possible to use atomic instructions when the mutual exclusion variables are located in a non-cacheable external memory, but this condition adds an extra latency when accessing them. Furthermore, it is a blocking solution, preventing the CPU from running any other task until the mutual exclusion variable is unlocked.

The processing times obtained for the dual-core approach are shorter than those from the single-core implementation, achieving an approximated speed-up of x2 for the different 1-ASCET orders. However, the 0-ASCET is the only one that reaches a processing time in the same range as the filter banks latency of 55 μs. With regard to the other orders (1- and 2-ASCET), although their software processing times are reduced in the dual-core approach, they are still longer than the hardware limit of 55 μs, not allowing a real-time operation of the full PLC receiver.

**B. QUAD-CORE PROPOSAL**

The Zynq® US+ family has an enhanced architecture compared to the Zynq® 7000 one, including an ARM Cortex™ A53 processor with up to four cores. Furthermore, the OCM memory has eight exclusive monitors, making it easier to work with atomic instructions. In this way, the inter-core synchronization issue can still be tackled either with atomic instructions or with interrupts, as in Zynq® 7000.

The Generic Interrupt Controller (GIC) from the Zynq® US+ family includes Inter-Processor Interrupt channels (IPI), thus defining mechanisms for message and response exchanges between cores. Briefly, the IPI works as follows: after the destination core receives an interrupt from another core through the IPI, the destination processor handles it, and, if necessary, it can access a message buffer where some information from the interrupt source can be found. The protocol for these messages is not previously defined and depends on the application. The IPI channels can be routed to the system processors (Application Processing Unit MPCore), the real-time processors (RPU), to the Power Management Unit (PMU), or to the desired application that may be running in any core or even in a soft processor in the PL. More detailed information on these dedicated channels can be found in [32].

In this way, the Inter-Processor Interrupt (IPI) is dedicated to carry out the inter-core synchronization mechanism. To generate an IPI interrupt, the sender writes a ‘1’ in the bit of the corresponding receiver in the TRIG register. That receiver is interrupted with the incoming interrupt in the Interrupt Status Register (ISR), where the active bit in the status register corresponds to the channel generating this interrupt. Afterwards, the receiver generates an interrupt to the GIC, which is processed as any other by the core. The handler for this interrupt should access the ISR register and, in addition, check if the sender wrote any message in the corresponding sending buffer.

In order to synchronize the four cores, the IPI processor interrupt channels explained above are involved. Channels 7 to 10 are assigned to cores 0 to 3, respectively. Every interrupt sender writes a message in the buffer and generates the interrupt to the desired receiver. The receiver acknowledges the interrupt and reads the message, which updates the status variable and runs the corresponding task of the algorithm.

On the other hand, in the quad-core approach it is necessary to analyse the scalability of the algorithm to be implemented in software shown above, in order to use the four cores optimally. In this way, the dual-core proposal has been extended to the quad-core, as can be observed in Fig. 6. The grey-shaded area is an initial transient period when all the cores notify to core 0 that they have booted correctly. Furthermore, the red star represents the reception of the channel estimation frames, ready to be processed by the software part. As shown in Fig. 6, the number of inter-core synchronization points in the system is significantly higher, since more partitions among the four cores are needed in the algorithm. Nevertheless, the FFT is not divided, since it has been verified that the time required for the FFT calculation partitioned among the available cores is similar to the one obtained whether it is not partitioned. This also discards any extra time dedicated to inter-core synchronization in the FFT.

**VI. EXPERIMENTAL RESULTS**

Taking into account the previous considerations, the algorithm for the channel estimation and the equalizer coefficients calculation has been coded for the Zynq® 7000 and Zynq® US+ families, on the ZC706 (XC7Z045 FFG900) and ZCU102 (XCZU9EG-2FFVB1156) development...
FIGURE 6. Block diagram of the channel estimation and calculation of equalizer coefficients implemented in software for the quad-core version, denoting the inter-core synchronization points.

platforms, respectively. It is worth noting that the results shown for the Zynq® 7000 have been compiled from previous works [28] and [31], and included here for comparison’s sake, showing the performance improvement that Zynq® US+ can provide.

Based on the previous functional description of the algorithm, its C-coded implementation has been evaluated for the ARM Cortex™ A9 and ARM Cortex™ A53 available in the Zynq® 7000 and Zynq® US+, respectively. The compiler optimization options have been configured at the highest level (-O3) for the implementation. Likewise, the clock frequency of the processors has been set at the maximum values allowed by the platforms, 800 MHz and 1200 MHz for ZC706 and ZCU102, respectively. Finally, the number of sub-carriers in the FBMC transmultiplexer has been set at $M = 512$.

Table 1 shows the processing times (average value and standard deviation) for the software implementation (channel estimation and equalizer coefficient calculation), as well as the speed-up achieved for the single-core [28] and dual-core approaches [31] in the Zynq® 7000 family, depending on the order $L$ of the ASCET and using the inter-core synchronization method based on atomic instructions (this one is better than the one based on interrupts for this device, as was mentioned before). These results are the average and standard deviation for a thousand realizations of the algorithm. It is possible to check that the dual-core proposal obtains a speed-up of $x2$ or higher for the different equalizer orders $L$. Nevertheless, both approaches are not capable of getting below the 55 µs hardware latency from the filter banks (CMFB and SMFB), thus making impossible to work in real-time without inserting intermediate memories to perform the equalization when its coefficients are already available. The processing time for each of the stages is also very important, as the dual-core proposal reduces that time in half compared to the single-core proposal. However, the dual-core proposal has a slight penalty due to the inter-core synchronization of the cores.

Similarly, the performance obtained for the Zynq® US+ family is shown in Table 2, where the average running time and the standard deviation for the different orders $L$ of the equalizer are shown, over a thousand runs. Due to the fact that the Zynq® US+ architecture is slightly different and the IPI is used to synchronize the cores, this solution already achieves better performances for the single-core and dual-core implementations with respect to the Zynq® 7000 proposal. The speed-up in the case of a dual-core approach is roughly $x1.46$, $x1.76$ and $x1.80$ for the 0-, 1- and 2-ASCET, respectively with regard to the single-core proposal. Nevertheless, for the quad-core approach, there are more inter-core synchronization points, which implies a higher time overhead. This is the reason why, although the algorithm has

| Table 1. Processing times and speed-up obtained for the three types of $L$-ASCET equalizers implemented in Zynq® 7000 for the single-core and dual-core proposals. |
|---------------------------------|-----------------|-----------------|------------------|
|                                | Single-Core     | Dual-Core       |                  |
|                                | Total Time      | Average (µs)    | Std. Dev. (µs)   |
| 512 Points                     | 208.93          | 88.12           | 0.151            |
| FFT                            | 60.25           | 26.65           | 0.123            |
| LS Estimation                  | 15.26           | 15.26           | 0.019            |
| Channel Mean                   | 4.42/75         | 4.42/75         | 0.003            |
| Interpolation                  | 11.05           | 10.65           | 0.001            |
| Coefficients Calculate         | 50.19           | 30.19           | 127.53           |
| Std. Dev.                      | 0.008           | 80.27           | 0.133            |
| Speed-up (related to single-core) | 1.99           | 1.99            | 1.99             |
| 1-ASCET                        | 220.95          | 46.03           | 1.72             |
| Total Time                     | 180.61          | 46.03           | 1.72             |
| Average (µs)                   | 90.33           | 46.03           | 1.72             |
| Std. Dev.                      | 0.072           | 46.03           | 1.72             |
| 2-ASCET                        | 232.29          | 30.19           | 127.53           |
| Average (µs)                   | 15.26           | 50.19           | 80.27            |
| Std. Dev.                      | 0.123           | 50.19           | 80.27            |

VOLUME 8, 2020
TABLE 2. Processing times and speed-up obtained for the three types of L-ASCET equalizers implemented in Zynq® US+ for the single-core, dual-core and quad-core approaches using IPI.

| L-ASCET | Single-Core | Dual-Core | Quad-Core |
|--------|-------------|-----------|-----------|
| 0-ASCET | Average (µs) | 43.500 | 85.744 | 152.476 |
|        | Std. Dev.    | 0.215 | 0.239 | 0.591 |
| 1-ASCET | Average (µs) | 6.797 | 6.797 | 6.797 |
|        | Std. Dev.    | 0.020 | 0.020 | 0.020 |
| 2-ASCET | Average (µs) | 7.81 | 7.81 | 7.81 |
|        | Std. Dev.    | 0.008 | 0.008 | 0.008 |
| Interpolation | Average (µs) | 6.300 | 6.300 | 6.300 |
|        | Std. Dev.    | 0.001 | 0.001 | 0.001 |
| Coefficients Calculate | Average (µs) | 14.044 | 14.044 | 14.044 |
|        | Std. Dev.    | 0.007 | 0.007 | 0.007 |

Speed-up (related to single-core): 1.456, 1.761, 1.801

Been further parallelized by partitioning it into four cores, the acceleration is lower than expected, particularly for the 0-ASCET. In this case, the speed-up is around x1.18, as a consequence of the trade-off between the inter-core synchronization overhead and the algorithm computing time. This reduces the performance improvement, still slightly better than for the case of dual core. In the case of the 1-ASCET with the quad-core approach, the acceleration is x1.83; and the higher acceleration is obtained for the 2-ASCET, being x2.29.

Fig. 7 shows graphically the achieved processing times. It is possible to verify that each stage of the algorithm reduces its time by half for the dual-core proposal, however the penalty due to the inter-core synchronization is higher than the Zynq® 7000 one, due to the use of the IPI. This effect is more significant for the quad-core proposal where the Channel Mean, Interpolation and Coefficients Calculate stages reduce their time up to half compared to the dual-core proposal, but the penalty due to the inter-core synchronization increases as the number of cores does. This aspect is reflected in Amdahl’s Law, due to the inter-core synchronization overhead and the algorithm computing time. This reduces the performance improvement, still slightly better than for the case of dual core. In the case of the 1-ASCET with the quad-core approach, the acceleration is x1.83; and the higher acceleration is obtained for the 2-ASCET, being x2.29.

Summing it up, the quad-core approach achieves the equalization of the data samples with the appropriate coefficients for the 0-ASCET and 1-ASCET equalizers in real time, since the software processing time is below the hardware limit of 55 µs. Furthermore, the number of data sets that are processed with the inadequate coefficients for the 2-ASCET equalizer is only two, thus minimizing the consumption of the necessary integrated memories to delay and carry out the equalization of those two data sets with M samples correctly. Comparing these results with the 25 data sets with M samples that are not processed with the correct equalizer coefficients in the single-core approach, it implies a considerable reduction, saving a significant amount of memory in the target architecture.

Unfortunately, the quad-core proposal does not provide a proportional acceleration for all the equalizer orders. As it can be observed in Fig. 7, the inter-core synchronization for the 0- and 1-ASCET cases is the operation that consumes more time (20 µs approximately). The increase in the inter-core synchronization points causes this time penalty. In addition, the IPI provides an ideal infrastructure for multi-core applications, although, the write and read accesses to/from the buffers increase the time to process each interrupt. According
TABLE 4. Comparison of previous works related to the proposed architecture, in terms of the modulation scheme, the communication standard, the target device, the clock frequency, the transmission rate in Msamples/s (Ms/s), and the type of implementation.

| Reference | [35] | [36] | [21] | Zynq® 7000 Proposal | Zynq® US+ Proposal |
|-----------|------|------|------|---------------------|---------------------|
| Modulation scheme | 5G | 5G | PLC | FBMC with ASCET Equalizer | PLC |
| Target device | Zynq® 7000 | Zynq® 7000 | Virtex6 | Zynq® 7000 | Zynq® US+ |
| Clock frequency | 220 MHz | 100 MHz | 100 MHz | 100 MHz (PL) | 100 MHz (PL) |
| Transmission rate | 220 Ms/s | 192 Ms/s | 62.5 Ms/s | 62.5 Ms/s | 62.5 Ms/s |
| Type of implementation | Hardware | Hardware | Hardware | Hardware/Software | Hardware/Software |

to Amdahl’s Law, the maximum theoretical speed up that it is possible to reach in this application is roughly $x_2$ [34], instead of the expected $x_4$ speed-up, mainly due to the synchronization overhead. For the dual- and quad-core proposals there are stages of the algorithm that cannot be parallelized, such as the FFT calculation, the LS estimation or the inter-core synchronization. Nevertheless, the proposed architecture can provide valuable and improvements (with greater speed-up) if other applications with different data dependencies and less inter-core synchronization points are considered.

The possibility of using the mutex-based inter-core synchronization method, with atomic instructions, has also been evaluated. In this case, Zynq® US+ provides up to eight exclusive monitors in the OCM memory for using atomic instructions. The obtained total processing times are shown in Table 3. Note that some figures have not been detailed, since they are the same as before, and the only difference is in the inter-core synchronization operation.

It should be noted that the dual-core approach based on mutex reduces the total time in 3 µs, whereas the quad-core approach does it in 6 µs, with respect to the solutions based on the IPI, thus providing a better acceleration. Finally, the standard deviation obtained using mutex is slightly higher due to the competition that exists among the different cores to take over the resource.

The speed-up in the case of a dual-core approach based on mutex reduces the total time in 3 µs, whereas the quad-core approach is in 6 µs, with respect to the solutions based on the IPI, thus providing a better acceleration. Finally, the standard deviation obtained using mutex is slightly higher due to the competition that exists among the different cores to take over the resource.

The speed-up in the case of a dual-core approach based on mutex reduces the total time in 3 µs, whereas the quad-core approach is in 6 µs, with respect to the solutions based on the IPI, thus providing a better acceleration. Finally, the standard deviation obtained using mutex is slightly higher due to the competition that exists among the different cores to take over the resource.

The resource consumption of the architecture for the Zynq® 7000 (ZC706 platform) and Zynq® US+ (ZCU102 platform) is shown in Fig. 8. Note that the number of available resources (shown in the figure legend) for the ZCU102 platform is slightly higher than for the ZC706 platform, which makes the corresponding utilization percentage slightly lower. In the case of Zynq® 7000, taking into account the FBMC transmitter (Tx-FBMC) and the receiver (Rx-FBMC), the resource consumption of BRAM blocks is 53 %, 17 % for DSP48e1 cells, 3 % for Flip-Flops, and 14 % for Look-Up Tables. Most of these resources are the result of the implementation of the DCT4e and DST4e at the FBMC transmitter and receiver. This mixed HW/SW architecture allows saving the necessary resources for the implementation of the M-points FFT in the LS estimator, which requires mainly 31 BRAM memory blocks (around 3 % for ZC706) and 36 DSP48e1 cells (4 % for ZC706) that are limited in the FPGA. On the other hand, the energy consumption of the mixed HW/SW architecture is 2.946 W on Zynq® 7000 and 4.737 W on Zynq® US+. Note that the obtained power consumption is an estimate from the synthesis tool.

The last aspect to consider is the comparison of the proposal with other previous works focused on mixed HW/SW implementations using a SoC. This requires to be dealt with carefully, as these previous implementations are only hardware architectures, not involving any software part, and the mathematical definition of the modulation is not exactly the same as the FBMC studied here. In [35] a Filter-Bank Multi-Carrier with Offset Quadrature Amplitude Modulation (FBMC/OQAM) hardware implementation is proposed, which provides a complexity reduction in terms
of computational and memory resources in a SoC Xilinx Zynq® 7000 device. In [36] a reconfigurable FPGA-based Frequency Spreading FBMC (FS-FBMC) baseband modulator is proposed, whose design is resource-efficient, due to hardware virtualization. Finally, in [21] an architecture based on DCT-4e is presented to implement a FBMC system, as well as a comparative analysis, in terms of resource consumption, performance, and precision. It can be verified that, in general terms, the results presented here are in the same range as other previous works, always taking into account the variations about the SoC technology involved in the implementations of the modulation carried out in the aforementioned works. Table 4 shows a further comparison of the proposals for Zynq® 7000 and Zynq® US+ with those related works. Note that references [35] and [36] provide a hardware implementation for the 5G mobile communication standard, in which the sampling rate is higher than in PLC. On the other hand, aspects such as the modulation scheme involved, the device used, the transmission rate or the type of implementation have also been taken into account.

VII. CONCLUSION

This work has presented a heterogeneous SoC architecture based on multiple cores that has allowed a significant performance improvement in the calculation of the channel estimation and the equalizer coefficients for a FBMC transmultiplexer in PLC, using a Zynq® 7000 and Zynq® US+ device. Taking into account the timing constraints from the application, it was proved that the dual-core approach satisfies the timing constraints for the 0- and 1-ASCET equalizers, but not for the 2-ASCET one. In this context, the HW/SW architecture proposed here exploits the parallelism of up to four cores to achieve the real-time requirements. For that purpose, two types of inter-core synchronization methods have been studied; one based on atomic instructions (mutex) and the other one on interrupts, where this one presents slightly longer latencies but provides a scalable solution for more cores based on the IPI unit. Experimental results for the dual-core proposal present a x2 acceleration compared to the single-core proposal, whereas the quad-core solution does not provide a x4 acceleration as expected. This limitation is actually coming from the inter-core synchronization overheads, since the involved tasks (the channel estimator and the calculation of the equalizer coefficients) have some data dependencies that generate a significant number of inter-core synchronization points. Nevertheless, the presented multicore architecture can be easily applied to other processing algorithms, which may have less data dependencies, thus better exploiting the parallelism and the acceleration provided by the proposal.

REFERENCES

[1] S.-G. Yoon, S. Jang, Y.-H. Kim, and S. Bahk, “Opportunistic routing for smart grid with power line communication access networks,” IEEE Trans. Smart Grid, vol. 5, no. 1, pp. 303–311, Jan. 2014.
[2] K. Rabie, A. M. Tonello, N. Al-Dhahir, J. Song, and A. Sendin, “IEEE access special section editorial: Advances in power line communication and its applications,” IEEE Access, vol. 7, pp. 133371–133374, 2019.
[3] G. Lopez, J. Matanza, D. De La Vega, M. Castro, A. Arrinda, J. I. Moreno, and A. Sendin, “The role of power line communications in the smart grid revisited: Applications, challenges, and research initiatives,” IEEE Access, vol. 7, pp. 117346–117368, 2019.
[4] L. Zhang, H. Ma, D. Shi, P. Wang, G. Cai, and X. Liu, “Reliability oriented modeling and analysis of vehicular power line communication for vehicle to grid (V2G) information exchange system,” IEEE Access, vol. 5, pp. 12449–12457, 2017.
[5] F. A. Pinto-Benel and F. Cruz-Roldan, “2-ASCET for broadband multi-carrier transmission over in-home and in-vehicle power line networks,” in Proc. IEEE 18th Int. Conf. Intell. Transp. Syst., Sep. 2015, pp. 1351–1356.
[6] M. Brison, M. Bensetti, R. De Lacerda, F. V. D. Santos, and C. Taurand, “Optimized PLC power transfer on avionic DC power lines: Coupling circuit and lightning protection,” in Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC), 2017, pp. 1–6.
[7] L. Pan, J. Wang, and J. Wang, “Research on control system of intelligent shore-to-ship power connecting pile based on PLC technology,” in Proc. Chin. Control Decis. Conf. (CCDC), Jun. 2018, pp. 1120–1125.
[8] A. Pittolo, M. De Piante, F. Versolatto, and A. M. Tonello, “In-vehicle power line communication: Differences and similarities among the in-car and the in-ship scenarios,” IEEE Veh. Technol. Mag., vol. 11, no. 2, pp. 43–51, Jun. 2016.
[9] M. Choi, S. Ju, and Y. Lim, “Design of integrated meter reading system based on power-line communication,” in Proc. IEEE Int. Symp. Power Line Commun. Appl., Apr. 2008, pp. 1–5.
[10] IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications, IEEE Standard 1900-2010, New York, NY, USA, 2010.
[11] Z. Yin, S. Dou, H. Bai, and Y. Hou, “Light-weighted security access scheme of broadband power line communications for multi-source information collection,” in Proc. IEEE 3rd Int. Conf. Netw., Electron. Autom. Control Conf. (ITWEC), Mar. 2019, pp. 1–4.
[12] A. M. Tonello and A. Pittolo, “Considerations on narrowband and broadband power line communication for smart grids,” in Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm), Nov. 2015, pp. 13–18.
[13] S. Laksir, A. Chaoub, and A. Tamaatou, “Impulsive noise reduction techniques in power line communication: A survey and recent trends,” in Proc. Int. Conf. Sel. Topics Mobile Wireless Netw. (MoWinT), Jun. 2018, pp. 13–20.
[14] W. Zhu, X. Zhu, E. Lim, and Y. Huang, “State-of-art power line communications channel modelling,” Procedia Comput. Sci., vol. 17, pp. 563–570, Jan. 2013.
[15] S. Galli, K. J. Kerpeh, H. Mariotte, and F. Moulin, “PLC-to-DLSH interleaver: Statistical model and impact on VDSL2, vectoring, and G.Fast,” IEEE J. Sel. Areas Commun., vol. 34, no. 7, pp. 1992–2005, Jul. 2016.
[16] R. Bale, T. S. Seetharamaiah, Y. S. Reddy, A. Dubey, and T. Panigrahi, “OFDM vs FBMC: A comparative analysis for broadband-PLC,” in Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS), Jan. 2018, pp. 480–483.
[17] S. Ma, X. Pan, G.-H. Yang, and T.-S. Ng, “Blind symbol synchronization based on cyclic prefix for OFDM systems,” IEEE Trans. Veh. Technol., vol. 58, no. 4, pp. 1746–1751, May 2009.
[18] S. Moya, M. Hadad, P. Donato, M. Funes, and D. Carrica, “Channel estimation and equalization of broadband PLC systems—Part 2: Comparison between single carrier and OFDM approaches,” in Proc. IEEE Biennial Congr. Argentina (ARGENC0N), Jun. 2016, pp. 1–6.
[19] F. Renzini, C.ucci, D. Rossi, E. F. Scarselli, and R. Canegallo, “A fully programmable eFPGA-augmented SoC for smart power applications,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 2, pp. 489–501, Feb. 2020.
[20] B. Farhang-Boroujeni, “Filter bank multicarrier modulation: A wave-form candidate for 5G and beyond,” Adv. Electr. Eng., vol. 2014, pp. 1–26, Dec. 2014. [Online]. Available: https://www.hindawi.com/journals/ace/2014/482805/
[21] P. Poudereux, A. Hernandez, F. Cruz-Roldan, and R. Mateos, “FPGA-based architecture for medium access techniques in broadband PLC,” IEEE Access, vol. 6, pp. 9534–9542, 2018.
[22] R. Nieto, R. Mateos, and A. Hernandez, “Finite precision analysis of FPGA-based architecture for FBMC transmultiplexers in broadband PLC,” in Proc. Conf. Design Circuits Integr. Syst. (DCIS), Nov. 2018, pp. 1–6.
[23] F. Cruz-Roldán, F. A. Pinto-Benel, J. D. O. del Campo, and M. Blanco-Velasco, “A wavelet OFDM receiver for baseband power line communications,” *J. Franklin Inst.*, vol. 353, no. 7, pp. 1654–1671, May 2016.

[24] V. Sánchez, P. García, A. M. Peinado, J. C. Segura, and A. J. Rubio, “Diagonalizing properties of the discrete cosine transforms,” *IEEE Trans. Signal Process.*, vol. 43, no. 11, pp. 2631–2641, 1995.

[25] S. A. Martucci, “Symmetric convolution and the discrete sine and cosine transforms,” *IEEE Trans. Signal Process.*, vol. 42, no. 5, pp. 1038–1051, May 1994.

[26] F. Nombela, E. García, and A. Hernandez, “Preliminary study between frequency-domain equalizer and 0-ASCET for PLC,” in *Proc. IEEE 17th Int. Conf. Ubiquitous Wireless Broadband (ICUWB)*, Sep. 2017, pp. 1–5.

[27] F. Canete, J. Cortes, L. Diez, and J. Entrambasaguas, “A channel model proposal for indoor power line communications,” *IEEE Commun. Mag.*, vol. 49, no. 12, pp. 166–174, Dec. 2011.

[28] R. Nieto, R. Mateos, and A. Hernandez, “HW/SW architecture for a broadband power-line communication system with LS channel estimator and ASCET equalizer,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 6740–6749, Nov. 2020.

[29] Zynq-7000 SoC. Technical Reference. UG585 (1.12.2), Xilinx, San Jose, CA, USA, 2018.

[30] ARM Synchronization Primitives. DHT0008A (ID081709), ARM Ltd., Cambridge, U.K., 2009.

[31] R. Nieto, R. Mateos, A. Hernandez, and E. Diaz, “Dual-core architecture for PLC channel estimator and ASCET equalizer in a FBMC transmultiplexer,” in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 1342–1345.

[32] Zynq UltraScale+ Device Technical Reference Manual (UG1085), Xilinx, San Jose, CA, USA, 2019.

[33] M. D. Hill and M. R. Marty, “Amdahl’s law in the multicore era,” *Computer*, vol. 41, no. 7, pp. 33–38, 2008.

[34] M. A. N. Al-hayanni, F. Xia, A. Rafiev, A. Romanovsky, R. Shafik, and A. Yakovlev, “Amdahls law in the context of heterogeneous many-core systems a survey,” *IET Comput. Digit. Techn.*, vol. 14, no. 4, pp. 133–148, 2020.

[35] J. Nadal, C. Abdel Nour, and A. Baghdadi, “Low-complexity pipelined architecture for FBMC/OQAM transmitters,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 1, pp. 19–23, Jan. 2016.

[36] M. L. Ferreira and J. C. Ferreira, “Flexible and dynamically reconfigurable FPGA-based FS-FBMC baseband modulator,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2018, pp. 1–5.

RUBÉN NIETO (Member, IEEE) received the B.S. degree in technologies of telecommunications engineering, the M.Sc. degree in telecommunication engineering, and the Ph.D. degree from the University of Alcalá (UAH), Spain, in 2015, 2016, and 2020, respectively, where he is currently pursuing the Post Ph.D. degree. His current research interests include embedded systems, multiprocessor SoC, and power line communications.

RAÚL MATEOS received the M.Sc. degree in telecommunication engineering from the Technical University of Madrid, in 1999, and the Ph.D. degree from the Universidad de Alcalá (UAH), Spain, in 2006. He is currently an Associate Professor with the Department of Electronics, UAH. His research interests include high-performance SoC architectures and EDA tools for system-level performance analysis.

ÁLVARO HERNÁNDEZ (Senior Member, IEEE) received the Ph.D. degree from the Universidad de Alcalá (UAH), Spain, and Blaise Pascal University, France, in 2003. He is currently a Professor of Digital Systems and Electronic Design with the Department of Electronics, UAH. His research interests include multisensor integration, electronic systems for mobile robots, and digital and embedded systems.