Real Time FPGA based Testbed for OFDM Development with ML synchronization

Tiago Pereira, Manuel Violas, Atílio Gameiro, Carlos Ribeiro, and João Lourenço
DETI, Instituto de Telecomunicações, University of Aveiro
Aveiro, Portugal
e-mails: targp@av.it.pt, manuelv@ua.pt, amg@ua.pt, cribeiro@av.it.pt, and jlourenco@av.it.pt

Abstract **– In this paper, we present a real-time testbed Orthogonal Frequency Division Multiplexing (OFDM) signaling scheme. The testbed is implemented in a Field-Programmable Gate Array (FPGA) through Xilinx System Generator for DSP and includes all the blocks needed for the transmission path of OFDM. Time-domain synchronization is achieved through a joint maximum likelihood (ML) symbol-time and carrier frequency offset (CFO) estimator through the redundant information contained in the cyclic prefix (CP). Results show that a rough implementation of the signal path can be implemented by using only Xilinx System Generator for DSP. This work presents a valid FPGA implementation of an OFDM receiver synchronization algorithm using a high-level design tool.**

Keywords – OFDM; FPGA; Software Defined Radio (SDR); physical layer; time-domain synchronization.

I. INTRODUCTION

Although multicarrier techniques can be traced back to 1966 [1], the first commercial application of OFDM occurred only in 1995 with the Digital Audio Broadcasting (DAB) standard [2]. OFDM is a multicarrier bandwidth efficiency scheme for digital communications where the main difference to conventional frequency division multiplexing (FDM) is that in the frequency domain the OFDM subcarriers overlap, providing spectrum efficiency. Given that OFDM implementations are carried out in the digital domain, there are a number of platforms able to implement an OFDM system suitable for software defined radio (SDR) development. SDR has been emerging within the wireless industry and can be applied to several applications. It is defined as a radio communication system where some physical (PHY) layer components that are typically implemented in analog hardware (filters, mixers, etc.) are implemented in software instead, thus allowing the user to operate the radio in different environments providing flexibility due to its reconfigurability.

The system presented was built using a high-level design tool built into Matlab's Simulink, Xilinx System Generator for DSP, providing the user with high-level abstractions of the system that can be automatically compiled into an FPGA [10].

Work presented here is divided as follows: Section II presents the literature review, the receiver critical parts and the paper's aim; Section III presents a brief description of the transceiver architecture and the received signal's ambiguities; Section IV is divided in three parts and presents the algorithm implemented for time-domain estimation and compensation; Section V presents the testbed used and how simulations were performed and Section VI explains the simulation results. Conclusions are provided in Section VII.

II. RELATED WORK

Several testbeds for OFDM systems based on SDR have been reported on literature. For example, [3] presents an OFDM modulator/demodulator with two synchronization options and two error-controlling techniques. The work in [4] uses GNU radio to transfer OFDM signals with QPSK and BPSK modulation to analyze the packet-received ratio for Quality of Service purposes. Other broadly adopted research platform is WARP [5]. The work in [6] uses this testbed to present an OFDM-based cooperative system using Alamouti's block code to study its capability versus a 2 x 1 multiple input single output system. FPGA implementations of standards 802.11a and 802.16-2004' modulators using Xilinx System Generator for DSP for high level design can be found in [7][8].

Two critical parts of the receiver are the synchronization and channel estimation subsystems. The synchronization should estimate the frame arrival time and a frequency offset between the local oscillators and RF carriers. Compensation can then be applied to the incoming signal. Unlike IEEE 802.11a/g and 802.16 (WiMax), among others, this system does not use a training sequence to achieve time-domain synchronization.

The aim of this paper is to present the implementation of an FPGA-based OFDM receiver with a ML time-domain synchronization algorithm using Xilinx System Generator for DSP.

III. THE ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING TRANSCEIVER

Figure 1 depicts the transceiver architecture of the system discussed in this paper. On the transmitter, data is generated randomly by making an inverse fast Fourier transform (IFFT) of quadrature amplitude modulated (QAM) symbol sets with 1024 subcarriers. The CP is added after the IFFT and the symbols are turned into frames. An up-conversion of 4 is performed on the interpolation block by a set of two interpolation filters: a square-root-raised-cosine and a halfband. The mixer block performs frequency translation to an intermediate frequency (IF) and is achieved by mixing the frame with a direct digital synthesizer (DDS). On the receiver side another DDS translates the IF back to baseband on the mixer block. Down-conversion and matched filtering is performed by a similar set of filters as the ones used on the transmitter by the decimator block. Once the estimations for the offsets are performed, the data forwarding control (DFC) and CFO correction blocks perform the compensations. A fast Fourier Transform (FFT) retrieves the data. Several parameters along the system are reconfigurable at user's need. Such parameters include number of symbols per frame, CP length, carrier frequency (limited by the system's frequency), modulation (QPSK, 16-QAM, 64-QAM, etc.) and the system's main clock frequency.

Preventing intersymbol interference (ISI) and preserving subcarrier orthogonality is achieved by adding a preamble of L samples to each symbol that contains the information of the last N samples of each OFDM symbol. Such symbol structure (N+L samples) maintains subcarrier orthogonality, in spite of the loss in transmission power and throughput.

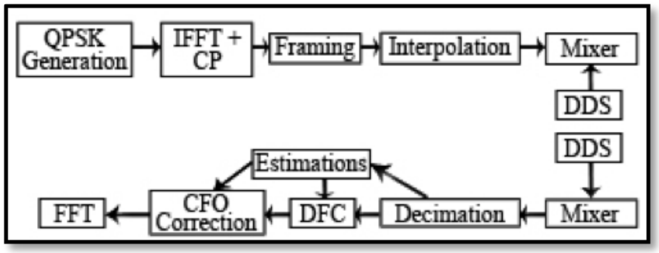

Figure 1. Baseband and IF transceiver arquitecture

CP 0	Symbol 0	CP 1	Symbol 1	● ●	CP N	Symbol N

Figure 2. OFDM Frame Format

As receiver and transmitter are physically separated, the receiver to be able to perform an efficient demodulation has to perform frame and carrier synchronization. The first operation defines the starting / ending points of the frame while the latter synchronizes the phase / frequency between transmitter and receiver. Erroneous frame detection is projected into the symbol constellation with a circular rotation, whereas the carrier frequency offset (CFO) causes all the subcarriers to shift and is projected as dispersion in the constellation points. Both ambiguities yield the received signal:

$$r(k) = s(k-\tau)e^{j2\pi\varepsilon k/N} \qquad (1)$$

where ε is the normalized CFO, τ is the unknown arrival time of a frame, s(k) is the transmitted signal, N is the number of samples per symbol and k is the sample index of each symbol ranging [0,1023].

IV. SYSTEM GENERATOR FLOW ON SYNCHRONIZATION

The following subsections present the synchronization algorithm divided in three parts.

A. Estimation of symbol arrival time and carrier frequency offset

The subsystem presented on this subsection is based on the algorithms developed by Beek [11]. The subsystem created for its purpose and adapted to the symbol pattern on Figure 2 is illustrated in Figure 3. Beek exploits the CP by correlating it with a delayed version of itself. When the repeated pattern is located, a peak is generated in order to detect the frame arrival and the phase between patterns gives the CFO.

The algorithm consists of two main branches. The top one calculates an energy term. While the bottom one calculates the correlation term required for estimating both symbol arrival time and phase offset. Equation (2) shows the calculation of the energy term and Equation (3) shows the calculation of the correlation term.

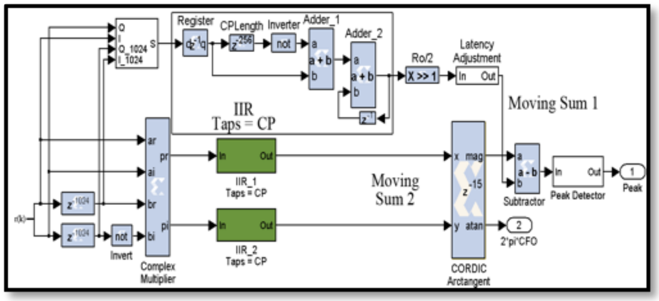

Figure 3. Estimation algorithm architecture

$$ms1 \equiv \frac{\rho}{2}\sum_{k=m}^{m+L-1} |r(k)|^2 + |r(k+N)|^2 \qquad (2)$$

$$ms2 \equiv \sum_{k=m}^{m+L-1} r(k)r^*(k+N) \qquad (3)$$

The factor ρ is the magnitude of the correlation coefficient between *r(k)* and *r(k+N)*; it depends on the signal-to-noise ratio but can be set to 1. Both moving sums were designed using infinite impulse response (IIR) filters. The complex multiplier core present on the System Generator libraries performs multiplications throughout the subsystem. In order to proceed with both estimations, two operations must be performed on the bottom branch, a complex module to create the peak when the CP correlates with its delayed version and an arctangent to calculate the angle between both IQ signals to enable CFO estimation. System Generator provides a CORDIC arctangent reference block that implements a rectangular-to-polar coordinate conversion using a CORDIC algorithm in circular vectoring mode, that given a complex-input <I,Q>, it computes a magnitude and an angle according to (4) and (5), respectively.

$$|I,Q| = \sqrt{I^2+Q^2} \qquad (4)$$

$$ang = 2\pi\varepsilon = \arctan(Q/I) \qquad (5)$$

ISBN: 978-1-61208-231-8

It is assumed that the offset between oscillators is lower than a single subcarrier and so |ε| < 1/2. Reference [13] performs a division to create the necessary peak for frame arrival detection, but such operation in hardware is more expensive and should be avoided. The only difference brought by the difference operation is how the peak is generated, since the argument to be detected will be close to 0 with a subtraction and to 1 with a division. Achieving a theoretical value of 0 when a signal is detected is not a realistic approach since the fixed-point logic used is subject to quantization errors and to contention of bit propagation along the system. The computed angle is only used when the peak is detected, ensuring the CFO is only used if the correlation is complete.

TABLE I. SYSTEM PARAMETERS

System Parameters	
Baseband frequency || Bandwidth	15.36 MHz || 10 MHz
FFT size || CP size	1024 || 256
Modulation	QPSK
Subcarrier separation	15 kHz
Symbol duration (Symbol+CP)	66.66 + 16.66 = 83.32 μs
IF sampling frequency	61.44 MHz
Oscillator frequency	7.68 MHz

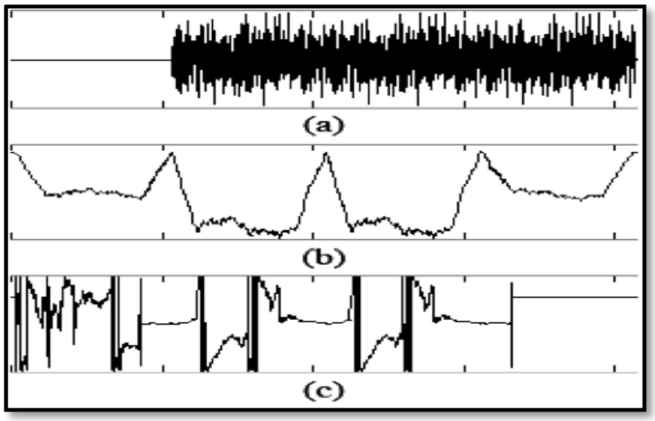

Figure 4. Estimation algorithm results for a frame with 3 OFDM symbols: (a) signal, (b) peak detection and (c) computed angle.

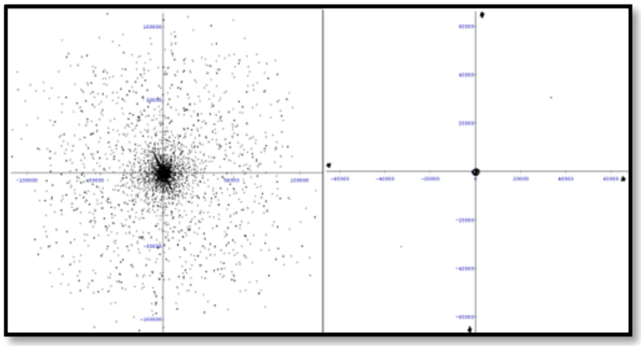

Figure 5. OFDM symbol with a 6 kHz offset between oscillators. Before compensation (left) and after compensation (right).

B. Data forwarding control

This subsystem uses the frame detection peak to process the frame in order for each symbol to be processed by the FFT. Unlike a non-deterministic simulation such as the ones ran in Simulink, a FPGA simulation doesn't have the ability to hold the information on its own while the estimations described on the previous subsection are executed. Data must be contained in a memory and forwarded when a condition is met or delayed by a constant value if the process is continuous, which is the case. The processing time required for a peak to be detected and the accurate CFO to be estimated is known, constant and introduced as a delay before the FIFOs. The peak detected on subsection *A* triggers the frame writing into the FIFOs. The CP is not needed anymore so it's not stored. The FFT will require 3*N samples to process each symbol and send it back outside. These amount of samples needs to created given that the symbols stored on the FIFOs are continuous. Reading the data stored on the FIFOs at a sampling rate four times higher as the symbols arrive creates that gap, breaking the frame back into separate symbols.

C. Carrier frequency offset correction

Correction of the CFO is achieved with a CORDIC implementing a rotate function [12]. The core rotates the vector (I,Q) by an angle ϕ yielding a new vector (I',Q') such that

$$\begin{aligned} I'(k) &= I(k)\times\cos\phi - Q(k)\times\sin\phi \\ Q'(k) &= Q(k)\times\cos\phi + I(k)\times\sin\phi \end{aligned} \tag{6}$$

where

$$\phi = {2\pi\varepsilon k}/{N} \tag{7}$$

and k is the sample index of each symbol. Taking the angle achieved at subsection *A*, the angle is first divided by N and then accumulated along each symbol nullifying the phase offset along each symbol.

V. TESTBED AND SIMULATIONS

Even though there is not a targeted standard at this point, such implementation can be adapted to several OFDM standards such as 802.11a, WiMAX, 3GPP LTE, among others, given the reconfigurability of the parameters. Xilinx

System Generator for DSP does not allow the user to replace hardware description language (HDL) completely but allows him to focus the attention on the critical parts of the design, i.e., when it comes to managing internal clocks and optimizing paths, HDL is better suited. The design was compiled through hardware co-simulation, a compilation method that allows the user to avoid HDL completely if no front-end is needed.

The targeted model for the simulation was the Xilinx ML605 development board, which contains a Virtex-6 LX240T FPGA and a 4DSP FMC150 FMC daughter card with a dual 14-bit 250 MSPS ADC and a dual 16-bit 800 MSPS DAC.

The tests were performed in a wired-channel and the system was run at a system clock of 61.44 MHz with an IF of 7.68 MHz. The results were obtained using the Xilinx ChipScope Pro tool. Because of the analog front-end present on the testbed, a wrapper must be created with Xilinx ISE Design Suite in order to connect both the system presented here and the daughter card where the DACs/ADCs are present. Table 2 shows the resources used for the full transceiver, without the wrapper.

VI. Simulation Results

In Figure 4, a rough estimation of the frame is presented, with a peak being generated at the beginning of each symbol and the respective CFO on the bottom, thus proving an accurate arrival time and CFO estimation of each symbol. It is also possible to perform a frame-based estimation instead of a symbol-based one, but no additional complexity is brought by this change.

Figure 5 proves that OFDM is sensitive to frequency offsets and even though the CORDIC corrects the phase along the symbol, it lacks the ability to compensate for the initial phase present on the oscillator.

VII. Conclusion and future work

A full baseband + IF design was presented focused on the synchronization algorithm. The work presented was performed using Xilinx System Generator for DSP, ChipScope Pro, ISE Design Suite and validated with Matlab.

It is possible to do FPGA simulations with a double floating-point precision, but not all blocks present on the System Generator libraries allow such precision and operations on floating point have a higher resource usage in hardware. Also, the front-end only allows a fixed point precision.

TABLE II. RESOURCE USAGE OF THE FULL SYSTEM

Full system resource usage for Virtex-6		
Parameter	***Used***	***%***
Slices	6662	17
Slice registers	24847	8
Slice LUTs	22059	14
Block RAMs	38	3
DSP48E	141	18

The next step is to direct the work presented here towards a 3GPP LTE MIMO-PHY receiver layer implementation with channel equalization and channel encoding algorithms.

Acknowledgment

The Portuguese projects CROWN, PTDC/EEA-TEL/115828/2009, and CelCop as well as the Mobile Network Research Group (MOBNET) from the Instituto de Telecomunicações in Aveiro supported the work presented on this paper.

References

[1] R. W. Chang, “Synthesis of band-limited orthogonal signals for multi-channel data transmission”, Bell System Technical Journal 45, 1966, pp. 1775-1796.

[2] ETS 300 401, “Radio broadcasting systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers”, ETSI, Feb. 1995.

[3] M. Majó, “Design and implementation of an OFDM-based communication system for the GNU radio platform”, Master Thesis, Dec. 2009.

[4] A. Marwanto, M. A. Sarijari, N. Fisal, S. K. S. Yusof and R. A. Rashid, “Experimental study of OFDM implementation utilizing GNU Radio and USRP – SDR”, Proc. of the IEEE 9th Malaysia International Conference on Communicatons, Dec. 2009, pp. 132-135.

[5] P. Murphy, A. Sabharwal, and B. Aazhang, “Design of WARP: a wireless open-access research platform”, Proc. of the European Signal Processing Conference, Sept. 2006, Article No. 7.

[6] P. Murphy, A. Sabharwal, and B. Aazhang, “On building a cooperative communication system: testbed implementation and first results”, EURASIP Journal on Wireless Communications and Networking, June 2009, doi:10.1155/2009/972739.

[7] J. Garcia and R. Cumplido, “On the design of an FPGA-based OFDM modulator for IEEE 802.11a”, 2nd International Conference on Electrical and Electronics Engineering”, Sept. 2005, pp. 114-117.

[8] E J. Garcia and R. Cumplido, “On the design of an FPGA-based OFDM modulator for IEEE 802.16-2004”, 2005 International Conference on Reconfigurable Computing and FPGAs, 2005, pp. 22-25.

[9] A. Goldsmith, “Wireless Communications”, Cambridge University Press, 2005.

[10] Xilinx Inc., “System Generator for DSP user guide”, http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_3/sysgen_gs.pdf, Release 13.3, Oct. 2011 [retrieved: Nov., 2012].

[11] J. van de Beek, M. Sandell, and P. O. Börjesson, “ML estimation of time and frequency offset in OFDM systems”, IEEE Transactions on Signal Processing, vol. 45, no. 7, July 1997

[12] Xilinx Inc., “DS249 LogiCORE IP CORDIC v4.0”, http://www.xilinx.com/support/documentation/ip_documentation/cordic_ds249.pdf, March 2011 [retrieved: Nov., 2012].

[13] O. Font-Bach, N. Bartzoudis, A. Pascual-Iserte, and D. L. Bueno, “A real-time MIMO-OFDM mobile WiMAX receiver: Architecture, design and FPGA implementation”, Computer Networks, 55 (16), pp. 3634-3647, 2011.

[14] Xilinx Inc., “System Generator for DSP reference guide”, http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_3/sysgen_ref.pdf, Release 13.3, Oct. 2011 [retrieved: Nov.,2012