Packet Detection, Frequency Synchronization, and Channel Estimation/Equalization in OFDM-Based SDR Receivers

M.M. Ebrahimi, S.A. Bassam, M. Helaoui, and F.M. Ghannouchi
iRadio Laboratory, Electrical Engineering Department, University of Calgary, Calgary, Alberta, Canada, T2N 1N4.

***Abstract* — This paper demonstrates the development and implementation of packet detection, frequency synchronization and channel estimation/equalization blocks of an OFDM-based standards, such as WLAN and WiMAX, receivers for SDR applications. This paper goes step by step through the implementation and prototyping of an OFDM-based receiver using the MATLAB/Simulink and system generator. The receiver is implemented in DSP/FPGA Lyrtech platform and the performance of the solution is evaluated using the WLAN signal.**

***Index Terms* — OFDM, SDR, packet detection, frequency synchronization, channel estimation and equalization.**

I. Introduction

Orthogonal frequency division multiplexing **(**OFDM) is one of the most popular schemes in communication systems due to its resistance to multipath fading, robustness to intersymbol-interference (ISI), simple channel equalization, and ease of implementation using Fast Fourier Transform (FFT) [1]. The scheme generates orthogonal sub-carriers; therefore, implementing a precise frequency synchronization algorithm is very critical.

In this paper, theory and implementation of packet detection, frequency synchronization, frequency offset compensation, and channel estimation/equalization of OFDM-based receivers for SDR reconfigurable receivers is comprehensively presented. The design is based on Xilinx FPGA using both Simulink and VHDL for development of different blocks of the receiver. Using both Simulink's Xilinx library and VHDL-based blocks to build the receiver, helps to optimize the implementation in terms of complexity and resource. This work implements most of the required blocks of OFDM-based SDR receiver and provides a rapid OFDM SDR prototyping.

The paper is organized as follows; section II overviews the OFDM scheme and the block diagram of the OFDM-based receivers. Section III presents the packet detection and frequency synchronization blocks of the receiver. Channel estimation and equalization of the OFDM-based receivers are described in section IV. Finally, section V demonstrates the development and implementation of the described receiver blocks on a DSP/FPGA platform and concludes with measurement results.

II. Overview of OFDM Receivers

A. OFDM Scheme

The OFDM technique has been used in recent wideband communication standards like IEEE 802.11 [2] and 802.16 [3] standards. OFDM is a special case of frequency division multiplexing (FDM) scheme where the channel spacing between the sub-channels is chosen to ensure orthogonality between the sub-carriers.

In general, OFDM-based standards are mostly developed for packet switch communication networks, which are built on burst transmission. Each burst consists of preamble (or training sequence), signal, and data. The preamble itself consists of two sections: a short training sequence (STS) and a long training sequence (LTS). For example in IEEE 802.11 a [2], the STS is composed of 10 repeated 16-sample blocks and the LTS consists of two repeated 64-sample blocks plus a guard interval. In the OFDM-based receivers, the STS is usually used for the packet detection and frequency synchronization and LTS is used for fine packet tuning and channel estimation. The remaining sections of the burst consist of the signaling and data.

Fig. 1 shows an instance of an OFDM burst and guidelines on how to use various sections of the burst for each step of the synchronization [2].

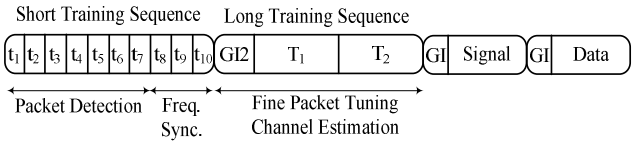

Fig. 1. Structure of IEEE802.11a signal burst.

B. OFDM Receivers

The block diagram of an OFDM receiver is shown in Fig. 2 [4]. After RF to IF down-conversion and analog-to-digital converter (ADC), the preamble section of the digital baseband burst is used to synchronize the receiver and to estimate the frequency offset. Applying FFT on the synchronized signal, the LTS is used to estimate and equalize the frequency response of the wireless fading channel. Finally, signal demodulation and decoding reconstruct the original transmitted data.

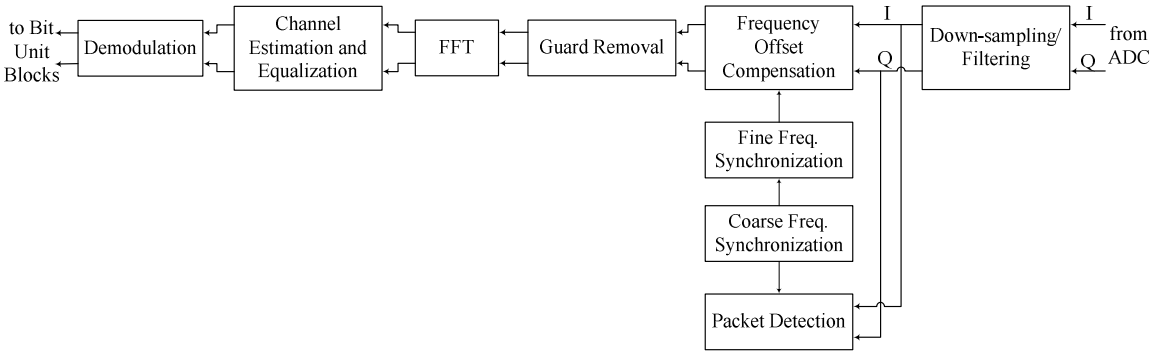

Fig. 2. Symbol processing blocks of an OFDM receiver.

III. Packet Detection and Frequency Synchronization

Synchronization in OFDM-based receivers consists of three major steps: i) packet detection, ii) frequency synchronization and iii) fine packet tuning, which also referred as fine frequency synchronization. In the following subsections, we describe these steps and explain the methods used for their implementation.

A. Packet detection

The packet detection shows the approximate starting of the frame. This approach is based on delay and correlation algorithm presented in [4][5]. In this algorithm, $c_{rr}^{i}(n)$ is the correlation between the received signal and its delayed version and $p_{rr}(n)$ is the energy of the received signal. They are defined as follows:

$$c_{rr}^{i}(n) = \sum_{k=0}^{L-1} r(n+k)r^{*}(n+k+iD) \tag{1}$$

$$\begin{aligned} p_{rr}(n) &= \sum_{k=0}^{L-1} r(n+k+D)r^{*}(n+k+D) \\ &= \sum_{k=0}^{L-1} |r(n+k+D)|^{2} \end{aligned} \tag{2}$$

where *r(n)* is the received signal, *D* is the length of one period of the preamble, *L* is the length of summation, which should be at least twice the value of *D*, and *i=1,2,…10* where 10 is number of repetition of the short preamble in one OFDM symbol. For example, in IEEE 802.11a *D* is 16 symbols and *L* is in the range of 32 to 160 symbols.

The packet detection is based on the ratio of the absolute value of $c_{rr}^{i}(n)$ to $p_{rr}(n)$ as follows [5]:

$$thr(n) = \frac{\left|c_{rr}^{i}(n)\right|}{p_{rr}(n)} \tag{3}$$

If the ratio in (3) passes a certain threshold, it indicates the start of the frame. The value of the threshold that determines the start of the frame depends on signal-to-noise ratio of the received signal. In (3), $p_{rr}(n)$ is used to normalize the decision statistic in order to eliminate the dependency from the absolute received signal power level.

Fig. 3 shows the relevant MATLAB/Simulink implementation of the packet detection algorithm for the OFDM system. The input in-phase and quadrature signals were used to calculate $p_{rr}(n)$ and $c_{rr}^{1}(n)$ to find the ratio. Exceeding the threshold would result in logic high level at the output of the packet detection unit. This is shown in Fig. 5 where the packet detection goes high after receiving a valid signal.

B. Frequency synchronization

The frequency synchronization consists of frequency offset estimation and compensation. Similar to packet detection, the frequency offset estimation has also been performed on the short training sequence using delay and correlation algorithms [4][6]. If $s(n)$ is the baseband input signal, the complex envelop of the transmitter's output signal, $x(n)$, can be defined as:

$$x(n) = s(n)e^{j2\pi f_t nT_s} \tag{4}$$

where f_t is the carrier frequency of the transmitted signal and T_s is the sampling period. The received signal, $r(n)$, with the carrier frequency of f_r is defined as:

$$r(n) = s(n)e^{j2\pi f_t nT_s}e^{-j2\pi f_r nT_s} = s(n)e^{j2\pi f_\Delta nT_s} \quad (5)$$

where the frequency offset between the transmitter and the receiver is equal to $f_\Delta = f_t - f_r$. Applying the correlation function in (1) on the received signal, $r(n)$, one can obtain:

$$c_{rr}^{i}(n) = \sum_{k=0}^{L-1} r(n+k)r^*(n+k+iD) = e^{j2\pi f_\Delta iDT_s}\sum_{k=0}^{L-1}|s(n)|^2 \quad (6)$$

From (6) the frequency deviation between the transmitter and the receiver can be estimated using the phase of $c_{rr}^{i}(n)$ as follows:

$$f_\Delta = f_t - f_r = 0.5 \times \left(\frac{\angle c_{rr}^{1}(n)}{2\pi DT_s} + \frac{\angle c_{rr}^{2}(n)}{4\pi DT_s}\right) \quad (7)$$

where the phase error can be calculated as:

$$\Delta\varphi = \frac{2\pi f_\Delta}{f_s} = \frac{\angle c_{rr}^{1}(n)}{2D} + \frac{\angle c_{rr}^{2}(n)}{4D} \quad (8)$$

where in WLAN standard, IEEE 802.11a, D = 16. The frequency offset estimation can be implemented using the same circuits applied for packet detection. In fact, as shown in Fig. 3, the frequency offset estimation can use the entire block of the packet detection circuit to calculate $c_{rr}^{1}(n)$ and similar circuit to find $c_{rr}^{2}(n)$ where both correlations are used in (8) to calculate the phase offset value. The estimated frequency offset and its "enable" signal are both shown in Fig. 5 where the amount of frequency offset is set to 20 KHz.

After estimating the phase offset from (8), the frequency offset compensation block, as shown in Fig. 4, would compensate for the frequency offset error between the transmitter and the receiver.

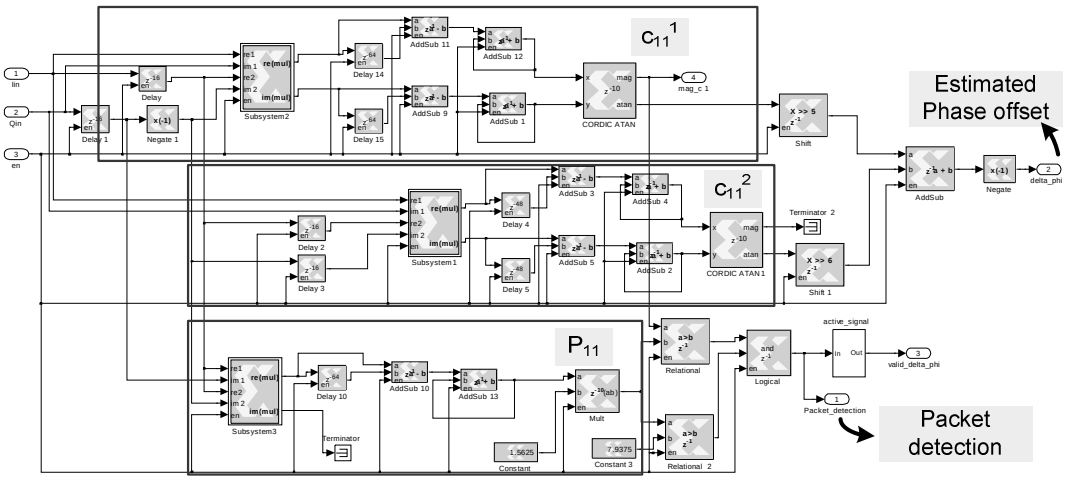

Fig. 3. MATLAB/Simulink implementation of the packet detection and frequency offset estimation units.

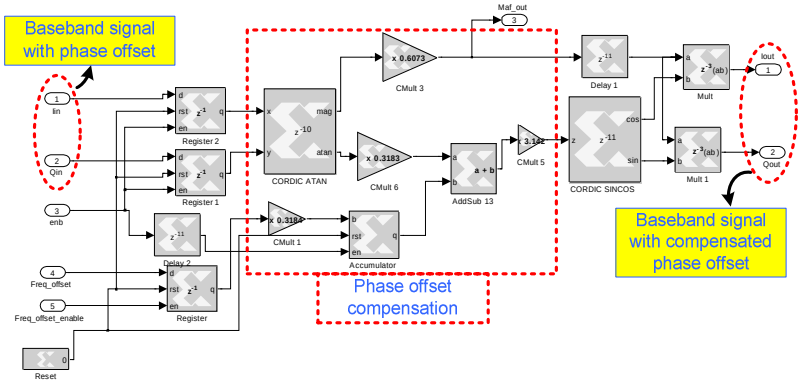

Fig. 4. MATLAB/Simulink implementation for frequency synchronization (frequency offset compensation) unit.

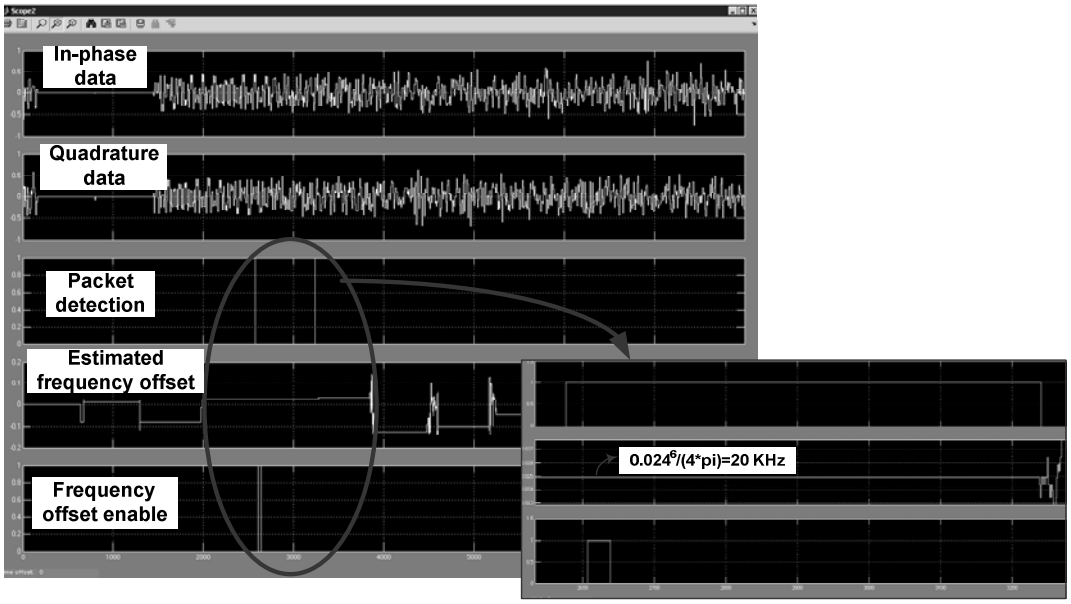

Fig. 5. Packet detection and frequency synchronization (frequency offset compensation) results of IEEE 802.11 a signal.

C. *Fine packet tuning*

Fine packet tuning is critical to avoid ISI. Using fine packet tuning after frequency offset compensation, the exact position of the beginning of the sequence is determined. This is required to make sure that in the next step, the right part of the sequence is chosen for the FFT operation to extract the data and apply the channel estimation and equalization.

The fine packet tuning is performed on the LTS by correlating the ideal LTS with the received LTS [7]. As shown in Fig. 1, there are two LTS in each frame; therefore, two peaks are expected as the result of correlation.

The MATLAB/Simulink implementation unit for the fine packet tuning is shown in Fig. 6. It consists of four FIR filters with fixed filter coefficients equal to the ideal LTS. As shown in Fig. 6, there are two peaks at the output of the correlation block, indicating the position of the two received LTS. By choosing the right threshold value, the exact starting symbol can be determined.

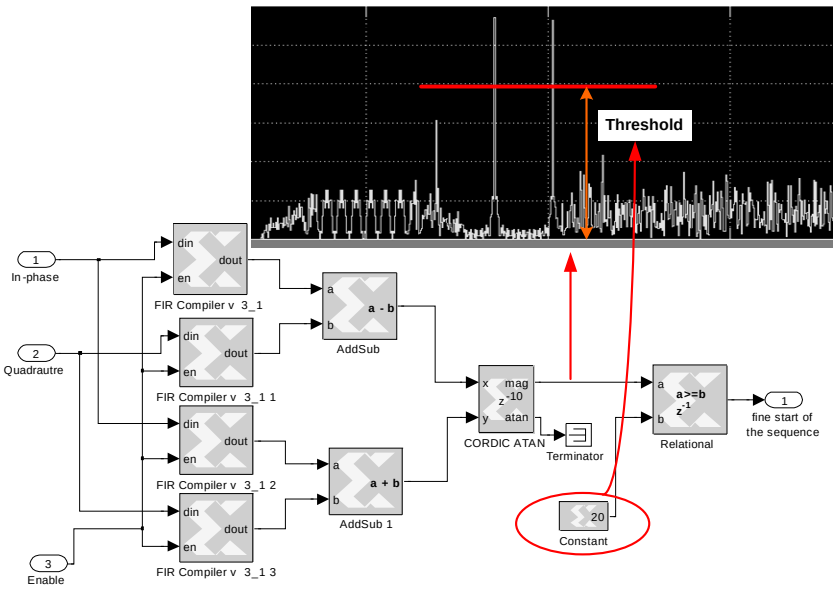

Fig. 6. MATLAB/Simulink implementation of fine packet tuning unit.

IV. Channel Estimation and Equalization

One of the advantages of OFDM signals is the simplicity of its channel estimation and equalization. In fact, the OFDM signal can be viewed as parallel narrowband signals rather than one wideband signal and therefore its channel estimation can be simplified like the one for narrowband signals [8]. After compensating for carrier frequency offsets, the FFT is applied to transform the OFDM symbols into the frequency domain. At this point, the effects of the channel should be compensated to find the magnitude and phase information of each OFDM subcarrier. There are two methods for channel estimation; it can be based on least square (LS) estimation or minimum mean-square error (MMSE) estimation [8]. Here the LS method, which is also usually referred to as zero-forcing method, is used. The method provides less complexity and is more suitable for FPGA implementation but suffers from higher level of error. The LTS will be applied to estimate the effect of the channel. If $x(n)$ is the transmitted signal in the frequency domain, $h(n)$ is the channel transfer function, $n(n)$ is the additive white Gaussian noise (AWGN) and $y(n)$is the received signal, then:

$$y(n) = x(n) * h(n) + n(n) \tag{9}$$

Equation (9) can be formulated in frequency domain as follows:

$$\bar{Y} = \bar{X}\bar{H} + \bar{N} \tag{10}$$

The least square estimation is based on minimizing the following expression [9]:

$$(\bar{Y} - \bar{X}\bar{H})^H . (\bar{Y} - \bar{X}\bar{H}) \tag{11}$$

where $(.)^H$ means conjugate transpose. Based on (10) and (11), it can be shown that the LS estimator of H can be written in the following format:

$$\hat{H}_{LS} = \bar{X}^{-1} . \bar{Y} \tag{12}$$

In OFDM-based systems the channel estimation is performed in frequency domain after FFT operator. In (12) the vector $\bar{X}$ is known at the receiver and $\bar{Y}$ is the received signal in the frequency domain.

In packet based communication such as WLAN, the channel is considered as quasi-stationary, which means that the channel does not change during the transmission of a packet. Therefore, the estimated channel response using LTS will be used to compensate for the effect of the channel for the remaining part of the burst. The channel estimation is performed and averaged over the two sections of the LTS as follows:

$$\hat{H}_{LS} = \frac{1}{2}(\bar{X}_1^{-1}\bar{Y}_1 + \bar{X}_2^{-1}\bar{Y}_2) = H + \frac{1}{2}(N_1 + N_2) \tag{13}$$

The long preamble data, which consists of $\pm\mathbf{1}$ in frequency domain, is stored locally in a memory block at the receiver. As a result, the received preamble after FFT operator can be considered as frequency response of the channel while frequency points relevant to -1 value require a 180º phase shift.

The estimated frequency response of the channel will be used to compensate for the channel frequency response of the rest of the frame. The estimated received signal after channel equalization is as follows [10]:

$$\begin{aligned} \hat{X}_{est} = \bar{Y}\frac{\hat{H}_{LS}^H}{\left|\hat{H}_{LS}\right|^2} &= \bar{X}\bar{H}\frac{\hat{H}_{LS}^H}{\left|\hat{H}_{LS}\right|^2} + \bar{N}\frac{\hat{H}_{LS}^H}{\left|\hat{H}_{LS}\right|^2} \\ &= \bar{X} + \bar{N}\frac{\hat{H}_{LS}^H}{\left|\hat{H}_{LS}\right|^2} \end{aligned} \tag{14}$$

Expression (13) is implemented using two complex multiplications and two dividers using a CORDIC divider. Fig. 7 presents the block diagram of the MATLAB/Simulink implementation of the channel equalizer.

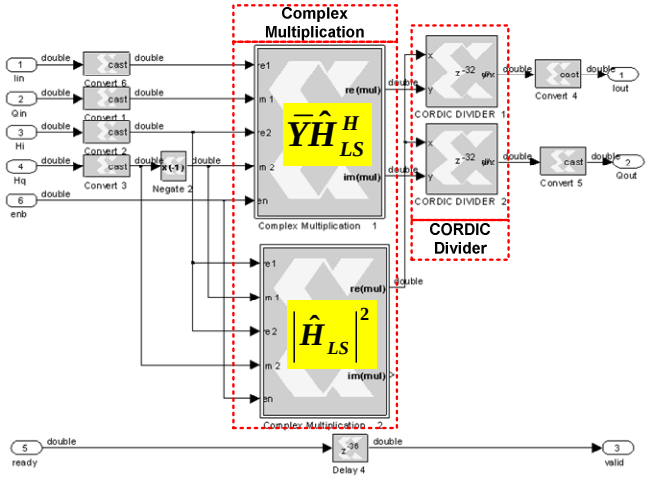

Fig. 7. MATLAB/Simulink implementation of channel equalizer

V. Description of the DSP/FPGA Platform for Prototyping

The described receiver blocks are implemented on a small form factor (SFF) SDR Development board from Lyrtech [11]. This platform is shown in Fig. 8. It is equipped with Virtex IV SX35 FPGA from Xilinx, analog to digital converter (ADS5500) and digital to analog converter (DAC5687) from Texas Instruments. MATLAB/Simulink, ISE and System Generator from

Xilinx have been used to design and implement the OFDM-based receiver.

The schematic of the OFDM-based receiver using the MATLAB/Simulink is depicted in Fig. 9. The design is based on MATLAB/Simulink using the Xilinx library and VHDL code. System Generator is used to synthesize and compile the design file to program the SDR platform and Xilinx ISE is used to analyze and synthesize the VHDL codes.

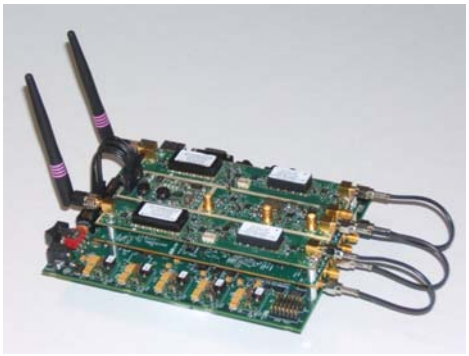

Fig. 8. Lyrtech Virtex-4 FPGA SDR Development Platform.

The following table shows the resource used to implement all the receiver's blocks shown in Fig. 9.

Table I. Resource used to implement the OFDM-based receiver for SDR application

Slices	FFs	RAM Blocks	LUTs	Embedded MULs	IO Blocks
7500	8028	11	10891	70	14

VI. Conclusion

This paper proposed a rapid prototyping of the physical layer of OFDM based-receivers for SDR applications. The implemented units of packet detection, frequency synchronization, fine packet tuning, and channel estimation/equalization were demonstrated. These receiver's units were designed and developed using MATLAB/Simulink and VHDL code, in order to reduce the complexity and minimize the required resource on the platform. The required resources for the implementation on the Xilinx FPGA VI based platform were also estimated. The performance of the receiver for all the blocks and the integrated receiver was evaluated and validated through implementation on an SDR development board from Lyrtech.

References

[1] R. V. Nee and R. Prasad, OFDM for Wireless Multimedia Communications: Artech House, 2000.

[2] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Std 802.11a-1999.

[3] Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, IEEE Std 802.16e-2005.

[4] J. D. Guffey, A. M. Wyglinski, G. J. Minden, "Agile Radio Implementation of OFDM Physical Layer for Dynamic Spectrum Access Research", GLOBECOM07 2007 , pp. 4051-4055.

[5] C. Dick, F Harris "FPGA Implementation of an OFDM PHY", 37th Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 905-909, 2003.

[6] T. M. Schmidl and D. C. Cox, "Robust Frequency and Timing Synchronization for OFDM," IEEE Trans. Comm., vol. 45, pp.1613 -1621, Dec. 1997.

[7] L. Schwoerer, "VLSI suitable Synchronization Algorithms and Architecture for IEEE 802.1la Physical Layer", ISCAS 2002.

[8] J.-J. van de Beek, O. Edfors, M. Sandell, S.K. Wilson, P. O. Borjesson, "On Channel Estimation in OFDM Systems", IEEE Vehicular Technology Conference, Vol.2, pp. 815-819, July 1995.

[9] H. Yuan , Y. Ling, H. Sun, W. Chen, "Research on Channel Estimation for OFDM Receiver Based on IEEE 802.11a", 6th IEEE International Conference on Industrial Informatics, pp. 35 – 39, July 2008.

[10] D. Tse, P. Viswanath, "Fundamentals of Wireless Communication", Cambridge University Press, 2005.

[11] Small Form Factor (SFF) SDR development platform from Lyrtech Incorporated. Product description can be found at: http://www.lyrtech.com/index.php?act=view&pv=SFF%20SDR%20development%20platforms.

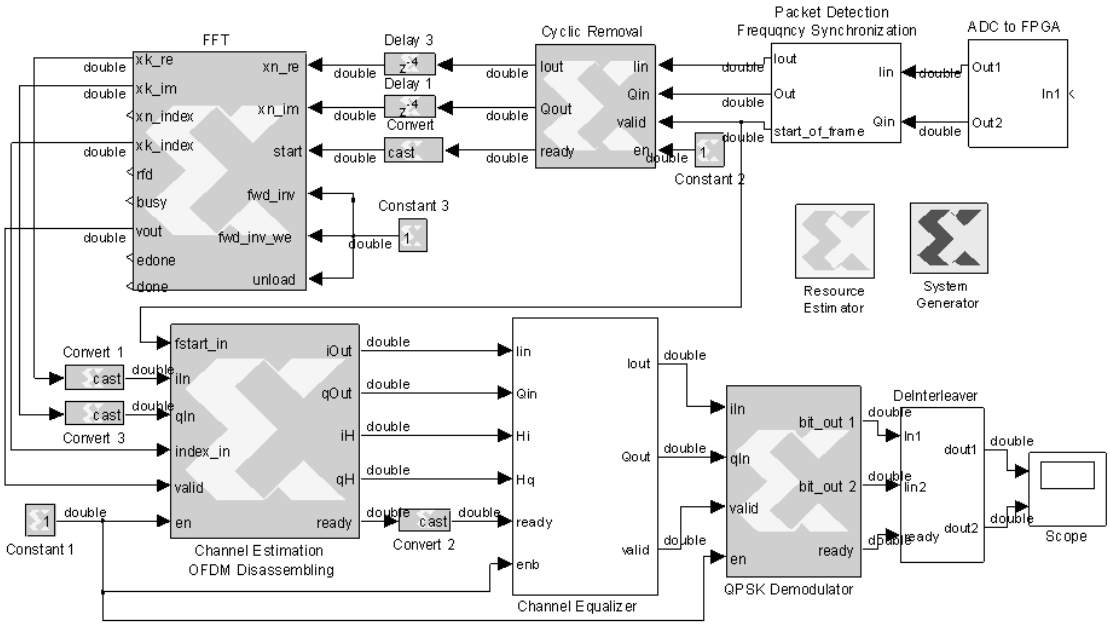

Fig. 9. MATLAB/Simulink schematic of packet detection, synchronization and channel estimation/equalization