

POLITECNICO DI TORINO

Department of Electronics and Telecommunications
Telecommunications Engineering

Master Thesis

Design and Implementation of OFDM System on FPGA



Supervisors:

Prof. Marina Mondin (PoliTo)

Prof. Roberto Garello (PoliTo)

Prof. Amir Keyvan Khandani (uWaterloo)

Candidate:

Seyed-Ehsan Kohestani

July 2015

Summary

The *orthogonal frequency division multiplexing* (OFDM) technology provides a high transmission data rate in wireless and mobile communications where multipath fading is a severe issue in degradation of the quality. Managing feasible coherent bandwidth to overcome Inter-Symbol Interference, OFDM enhances communication performance at a relatively small bandwidth cost. The improvement can be reached by interactive proper channel estimation and compensation which needs synchronization of transmitter and receiver. A *Discrete Fourier Transform* (DFT) algorithm-based configuration simplified the digital implementation of OFDM system on field programmable gate array (FPGA) as a highly flexible solution, which provide prominent performance.

In this thesis, steps to design a base-band OFDM system with channel estimation and timing synchronization upto implemented on FPGA are studied. It is a prototype based on the *IEEE 802.11a* standard and the signals is transmitted and received using a bandwidth of 20 MHz. Focusing on the quadrature phase shift keying (QPSK) modulation, the system can achieve a throughput of 24Mbps . For the coarse estimation of timing, a modified maximum-normalized correlation scheme is investigated and implemented. Starting from theoretical study, this thesis in detail describes the system design and verification on the basis of both MATLAB simulation and hardware implementation. In the meanwhile, comparison is made between the simulation and implementation results, which verifies system performance from the system level to the register transfer level (RTL).

First of all, the entire system is modeled in MATLAB and a floating-point model is established. Then, the fixed-point model is created with the help of Xilinx's System Generator for DSP (XSG) and Simulink. Subsequently, the system is synthesized and implemented within Xilinx's Integrated Software Environment tools and targeted to Xilinx Zynq board. What is more, a hardware co-simulation is devised to reduce the processing time while calculating the BER for the fixed-point model. Some time-based standards on IEEE 802.11a are discussed and optimum implementation of on FPGA, for instance Cross-Correlation and algebraic machine, will be introduced. Besides, we will demonstrate an engineering steps for choosing the radio board and the processor software implementations.

Acknowledgements

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of this Master thesis. I am thankful for their aspiring guidance, invaluable constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project. I express my warm thanks to Prof. Amir Keyvan Khandani and CST Lab members for their support, trust and guidance at the University of Waterloo. I would also like to thank guide of Prof. Roberto Garello and Prof. Marina Modin at Politecnico di Torino and their aspiration in Communication Systems. Especially, I appreciate my family for their constant devotion and love during years of separation.

Thank you,
Author

Contents

Summary	II
Acknowledgements	IV
1 Introduction	1
1.1 Motivation	3
1.2 Methodology	4
2 Theory, Design and Simulation	5
2.1 OFDM System Architecture	5
2.2 OFDM Specifications in IEEE 802.11a Standard	10
2.2.1 Introduction of IEEE 802.11	10
2.2.2 System Design	11
2.2.3 IEEE 802.11a Standard in Time and Frequency	16
2.2.4 Origin of CFO	17
2.2.5 Impact of CFO	19
2.2.6 Synchronization	20
2.2.7 Channel Estimation and Distortion Reject	21
3 FPGA Implementations	23
3.1 System Design in System Generator	23
3.2 Hardware Introduction	26
3.2.1 FPGA Board	26
3.2.2 Radio Board	28
3.2.3 Clock Chain on FMCOMMS1	30
3.3 Expectations for CFO on FMCOMM1	32
3.4 Time Domain CFO Correction	33
3.5 Carrier Frequency Offsets	33
3.6 FPGA Architecture	33
3.7 Test Methodology	35

4 Sample Analysis and Conclusion	38
4.1 Hardware Samples and Analysis	38
4.2 Bit-Error Rate Calculation	43
4.3 FPGA Resource Consumption	44
4.4 Conclusion	45
4.4.1 Future Work	45

List of Tables

2.1	System parameters defined for the proposed OFDM system	15
2.2	Rate dependant parameters in IEEE 802.11a Standard [14]	15
4.1	Device Utilization Summary (actual values)	45

List of Figures

2.1	Input and Output of IDFT [14]	7
2.2	OFDM System Model	8
2.3	Basic baseband OFDM system	9
2.4	OSI Reference Model	11
2.5	Architecture of an OFDM system	12
2.6	Transmitter and receiver block diagram of OFDM PHY	12
2.7	The frequency allocation of IEEE 802.11a sub-carriers	14
2.8	Preamble of IEEE 802.11 [14]	17
2.9	General models of a direct conversion RF	18
2.10	OFDM performance loss due to CFO-induced ICI.[1]	19
2.11	LTS in Time Domain.	20
2.12	Time Domain CFO Estimation.	20
2.13	Schimdl and Cox Delay and Correlate Algorithm.	22
3.1	System Generator Cycle.	24
3.2	OFDM System..	24
3.3	OFDM Transmitter Block.	25
3.4	OFDM Receiver Block.	26
3.5	Auto-Correlation Block.	27
3.6	Fine Packet Detection Block.	27
3.7	Complex Division Block.	27
3.8	Division Block.	28
3.9	ZC706 Evaluation Borad Block Diagram.	29
3.10	Xilinx Zynq-7000 SoC ZC706 Evaluation Kit	29
3.11	AD-FMCOMMS1-EBZ (Radio Board)	30
3.12	AD-FMCOMMS1-EBZ Block Diagram	30
3.13	AD9548 Block Diagram	31
3.14	CVHD-950 Ultra Low Phase Noise Oscillator	32
3.15	Zynq-7000 Diagram	34
3.16	Design Block Diagram	35
3.17	Hardware set-up	36
4.1	OFDM Frame (I/Q) detected in Chipscope.	39

4.2	STS (I/Q) detected in Chipscope.	40
4.3	Auto-Correlation detected in Chipscope.	40
4.4	Cross-Correlation detected in Chipscope.	41
4.5	LTS Spectrum in Baseband chain (IF filter on 5MHz is enable)	41
4.6	LTS Spectrum- passed RF chain (IF filter on 5MHz is enable)	42
4.7	LTS Spectrum- passed RF chain (IF filter is disable)	42
4.8	Frequency Response of a perfect channel detected in Chipscope	43
4.9	Frequency Response of on air channel detected in Chipscope	43
4.10	OFDM Symbols of a 16QAM 64-byte message Coded 1/2 rate	44
4.11	OFDM Symbols of a 16QAM 64-byte message no-Coded	44

Chapter 1

Introduction

High quality of services (QoS) and reaching to high data rate communication to overcome the necessities in multimedia services, telecommunications industry is working currently toward the fifth generation (5G) wireless communication systems. The orthogonal frequency division multiplexing (OFDM), is the most promising technology to meet the requirement as a mechanism in rapid development of digital signal processing techniques. [2]

The first introduction of OFDM dated back 1960s as a parallel data transmitting scheme. There are many realization proposals although the foundations are fixed generally. The basic idea is to divide a single high rate data stream into a number of lower-rate data streams. Each of these data streams is modulated on a specific carrier, which is called subcarrier, and transmitted simultaneously. Robustness will be preserved against multipath fading effect. Moreover, spectrum efficiency is enhanced in comparison to conventional multi-carrier transmission. OFDM considered as a frequency division multiplexing (FDM) where the data stream carried by each subcarrier separated. [3]

Traditional methods used in single-carrier modulation require a number of sinusoidal subcarrier oscillators in the modulator side and multipliers and correlators in the demodulator. Introduction of Discrete Fourier Transform (DFT) until 1971 made a revolution in the complexity development. The DFT block simplified the two side processes and helped to implement the baseband in the digital manner. Since 1990s, OFDM has been employed in wideband data transmission. Applications of OFDM technology include asymmetric digital subscriber line (ADSL), high-bit-rate digital subscriber line (HDSL), and very-high-speed digital subscriber line (VDSL) in wired systems, and digital audio broadcasting (DAB), digital video broadcasting (DVB) in wireless systems. Furthermore, it has also been recognized as the basis of the wireless local area network (WLAN) standards, among which the

IEEE 802.11a standard is one of the most important ones.

Two main topics in wireless and mobile communications are high data rate and high QoS, which cause communication systems be adaptive to fast varying channel conditions and providing a steady environment to various kinds of users at a high speed of data transmission.

Due to its capabilities of providing high data rate and less sensitivity to fast channel fading, OFDM technology, in combination with other powerful techniques such as the multiple-input, multiple-output (MIMO) technique, has been the mainstream of wireless and mobile networks. It has been used in various applications, such as wireless fidelity (Wi-Fi), worldwide inter-operability for microwave access (WiMAX), and the third generation partnership project (3GPP) long term evolution (LTE).

Recent development of digital integrated circuits, the high flexibility and low complexity of digital implementation of OFDM modem has accelerated its application. In competition of the technologies, field programmable gate array (FPGA) has attracted the most attention in recent years due to its superior performance and high flexibility. As a flexible general-purpose technology, FPGA is an array of gates that can be reconfigured by the designer as a versatile design platform. It is developed based on the programmable logic devices (PLDs) and the logic cell array (LCA) concept. By providing a two-dimensional array of configurable logic blocks (CLBs) and programming the interconnection that connects the configurable resources, FPGA can implement a wide range of arithmetic and logic functions. Compared to other popular IC technologies such as application specific integrated circuits (ASICs) and digital signal processors (DSPs), FPGA has the following advantages:

Performance: Inherently parallel architecture, FPGA has the ability to overcome the speed limit of sequential execution technologies and is able to process data at a much higher speed than DSP processors and whose performance is estimated by the system clock rate. Therefore, it can achieve much higher performance in various applications that requires large arithmetic resources, such as OFDM. However, DSP processors are still developing as an alternative.

Reliability: The high isolation and high parallelism mechanism not only minimize the reliability concerns, but also reduce the deterministic hardware dedicated to every task. Besides, there are mechanism in testing and verification of the system dynamically which are developing exponentially.

Cost: Because of its re-programmable nature, FPGA is a cost-effective solution

for system development although the purchase cost are normally more than DSP processors which the architecture is fixed. It can be easily customized and reconfigured so that effectively versatile functionality can be realized using FPGA and there is no need to kick off design for each application. Normally, the products are tested and design and implement on the FPGA initially and after successful output it worth to transfer design into ASIC for mass production.

Flexibility: The most prominent functionality of FPGA is that the design can be changed rapidly in the prototype process. Recently, some other options like partial reconfiguration let the designer to look for more dynamic mechanisms. This let the manufacturers to have better performance in the time to market issues. There is also some trends like IP core programs which help the big short-cuts in the designs but it very depends on the initial cost which should be decided very carefully.

1.1 Motivation

Practically, the signal is attenuated and distorted by multipath effect in real channel transmission. Fading estimation and equalization of the channel in wireless technology is inevitable to have a reliable communication. Implementation an OFDM system on FPGA with capability of channel estimation and synchronization is the final goal of this thesis.

There are many techniques and mechanism to implement OFDM wireless communication on FPGA. In Chapter 2 the authors helped OFDM transceiver on certain topics in the receiver design, such as the synchronization, packet detection, channel estimation and equalization. Moreover, OFDM transceivers implementations on FPGA and the hardware details are described in Chapter 3. However, there are not a comprehensive work presenting a complete development of OFDM system with channel estimation and synchronization using the FPGA technology.

A top-down approach and demonstrative system performance in baseband OFDM is done in this thesis. System synchronization will also be discussed in this thesis. In addition, we focus on the design and implementation of channel estimation and equalization, while a verification at system level is performed. The detailed objectives include:

- To design, model and implement after proper simulation a baseband OFDM system including both the transmitter and the receiver, and to analyze the system performance.

- To prototype an OFDM system based on a specific wireless communication standard.
- To implement the synchronization and channel estimation system for the receiver and provide system evaluation under different channel conditions.

1.2 Methodology

It is tried to explain the theoretical concepts firstly and then to show some facts in the simulation based on extracted models. Finally, the issues is examined on hardware.

The hardware chosen is consisted a Zynq board which is an FPGA with two embedded ARM processors and the radio board which is FMCOMMS1. Details of the electronic design and hardware considerations will be described.

Chapter 2

Theory, Design and Simulation

The fundamental concepts of an OFDM design is described. A detailed block diagram is shown. A theoretical basic based band OFDM system is compared with IEEE 802.11 standard scheme. Influential parameters to are explained with guided us to have other parameters in Table 2.1. Arrangement of the IEEE 802.11a carriers demonstrated.

A short discussion of Carrier Frequency Offset is initiated and its origin and the impacts on the whole system on receiver side and a possible solution to reduce the issue is discussed.

2.1 OFDM System Architecture

Generally, an OFDM signal is defines as a summation of many OFDM standard symbols, which can considered continuous in the time domain. It can be defined as following:

$$s(t) = \sum_{k=-\infty}^{+\infty} s_k(t) \quad (2.1)$$

where $s_k(t)$ is the k-th OFDM symbol which starts at time $t = t_s$. An OFDM system is a multi-carrier transmission mechanism which the mathematical model is generalized by the summing a series of modulated subcarriers digitally. This modulation can be phase shift keying (PSK) or quadrature amplitude modulation (QAM) and transmitted in parallel. So, we can conclude:

$$s_k(t) = \begin{cases} Re \left(\sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} d_{i+\frac{N}{2}} \exp[j2\pi(f_c - \frac{i+0.5}{T})(t - t_s)] \right), & t_s \leq t < t_s + T \\ 0, & \text{otherwise} \end{cases} \quad (2.2)$$

where T is the symbol duration, N is the number of subcarriers, f_c is the signal carrier frequency on the radio frequency (RF) band, and d_i is the complex value for PSK or QAM modulated symbol. We reach I_i and Q_i being the in-phase and quadrature part of d_i , respectively.

The complex envelope of an OFDM signal given by the following equation is used as the baseband notation:

$$s_k(t) = \begin{cases} Re \left(\sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} d_{i+\frac{N}{2}} \exp[j2\pi \frac{i}{T}(t - t_s)] \right), & t_s \leq t < t_s + T \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$$

The real and imaginary parts of 2.3 are the in-phase (I) and quadrature (Q) of the baseband OFDM signal. Consequently, they are multiplied by a cosine and a sine waveform with a carrier frequency to generate the passband OFDM. At the receiver, each subcarrier is down-converted with a subcarrier of the desired frequency and supported over the symbol period. For example, the complex value for the m -th subcarrier d_m is obtained equation 2.4, where the whole signal is multiplied by the frequency of $\frac{m}{T}$, and then integrated over the symbol period T :

$$\begin{aligned} \int_{t_s}^{t_s+T} s_k(t) \exp[-j2\pi \frac{m}{T}(t - t_s)] dt &= \int_{t_s}^{t_s+T} \left(\sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} d_{i+\frac{N}{2}} \exp[j2\pi \frac{i}{T}(t - t_s)] \right) \exp[-j2\pi \frac{m}{T}(t - t_s)] dt \\ &= \sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} d_{i+\frac{N}{2}} \int_{t_s}^{t_s+T} \left(\exp[j2\pi \frac{i-m}{T}(t - t_s)] \right) dt \\ &= T d_{m+\frac{N}{2}} \end{aligned} \quad (2.4)$$

2.4 shows all the subcarriers over the integral region are zero except the desired one. The desired output for the signal demodulation, $d_{m+\frac{N}{2}}$ multiplied to a constant factor T , is exactly the integration for the m -th subcarrier. Since each subcarrier has an exact integer number of cycles within OFDM symbol duration, the orthogonality between subcarriers is guaranteed.

the mathematical model for discrete time signal is as below if the OFDM symbol is

sampled with a sampling period $\frac{T}{N}$:

$$s(n) = s_k\left(\frac{nT}{N}\right) = \sum_{i=0}^{N-1} d_i \exp(j2\pi \frac{in}{N}), n = 0, 1, \dots, N - 1 \quad (2.5)$$

This represents an inverse DFT (IDFT) for PSK or QAM symbols. See Figure 2.1.

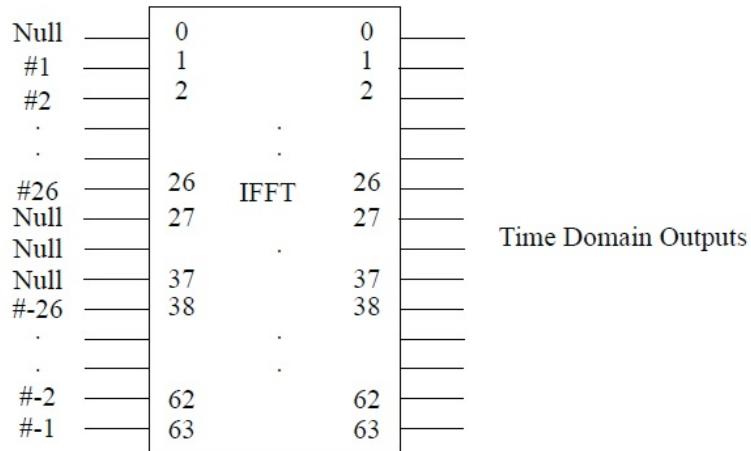


Figure 2.1. Input and Output of IDFT [14]

Figure 2.2 presents the OFDM system block diagram. The first block represents the data source. They are the bits which an application may send. In the simulation it is realized by a random and fixed bit generator. The following QPSK modulation block converts this bits into symbols, which are complex numbers. Each symbol carries several bits. The third block is the first one actually relevant for OFDM. The serial symbol stream is converted into a channel and OFDM symbol structure. In the simulation it is represented in a matrix shape where the rows are different channels and each column is an OFDM symbol. This means that each OFDM symbol, formed by c (#channels) serial symbols (complex numbers), is distributed over the c channels.

In the next step zero channels are added in order to separate well subsequent OFDM symbols. For the simulation structure zero rows are added in the middle of the matrix. The following block interprets each symbol (complex number) of an OFDM symbol as a orthogonal frequency and converts each OFDM symbol per IFFT into a vector of time discrete values of the same length. As sixth step the

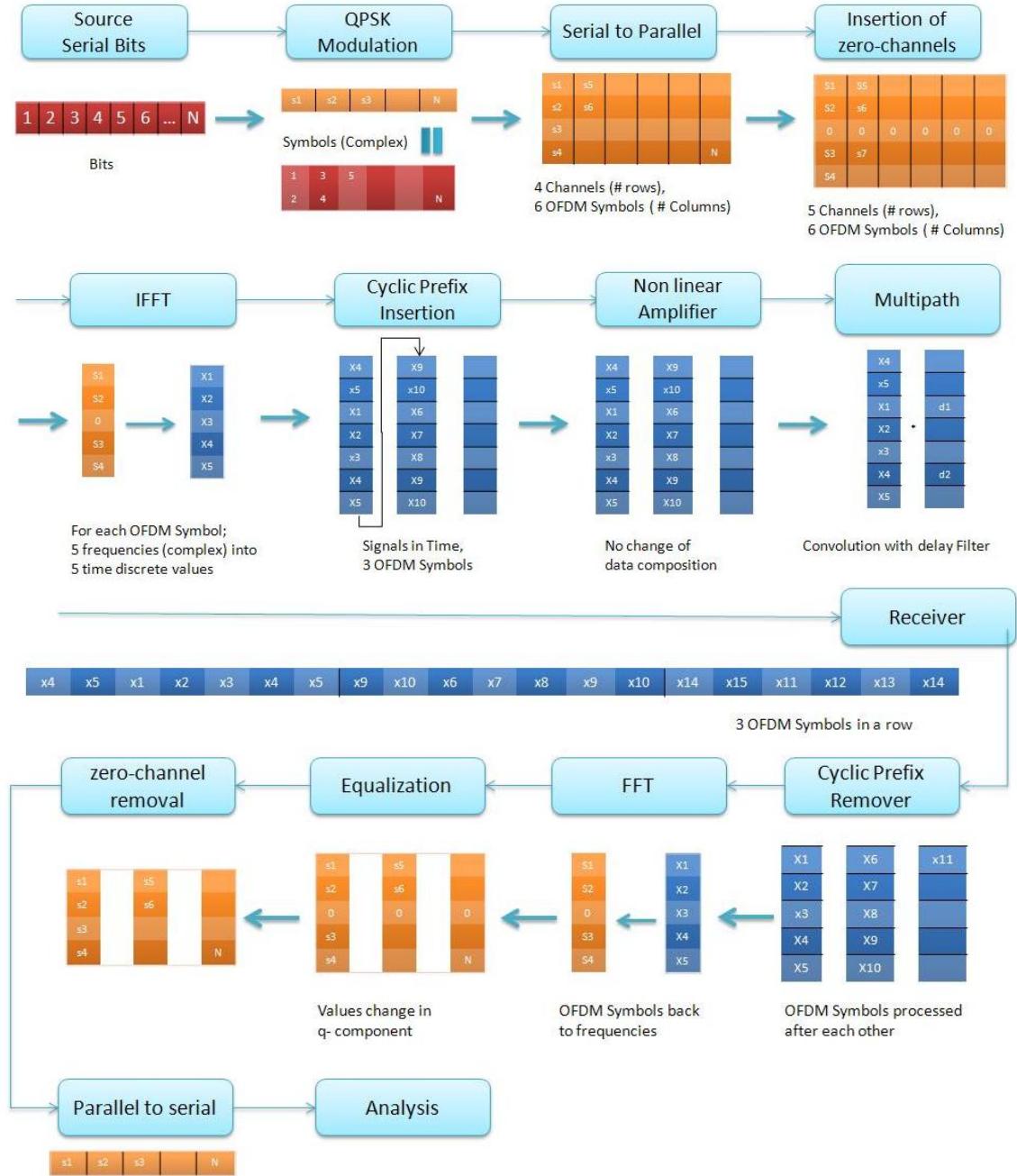


Figure 2.2. OFDM System Model

cyclic prefix insertion is done. In order to maintain orthogonality of the frequencies but prevent ISI, an amount of *guard values* are copied from the end of each OFDM symbol to its beginning. The number of rows in the simulation matrix grows by that by the number of *guard values*.

The following block is the NLA which depends on the optimization parameter β (back-off). At this point the transmitter side ends. In order to simulate the multipath a convolution is made on each OFDM symbol with the delay filter. Since the simulation is made on each OFDM symbol separated, this operation works with a memory in order simulate a serial transmission.

The receiver side is just the opposite of the transmitter. In the cyclic prefix remover the copied values are deleted and the simulation matrix size decreases. The next block performs the FFT on each OFDM symbol which reconstructs the as frequencies interpreted complex numbers.

The following equalization block tries to remove the effect of the multipath. For the simulation a multiplication with the inverted transfer function of the multipath is operated. By this, only the phase of the complex symbols changes.

Finally, the zero channels are removed and the matrix structure is reconverted to a series of symbols. The following analysis is done on the received symbols and consequently they are not demodulated into a bit-stream.

According to the above analysis, the basic architecture for a baseband OFDM system that contains the essential parts is shown in Figure 2.3.

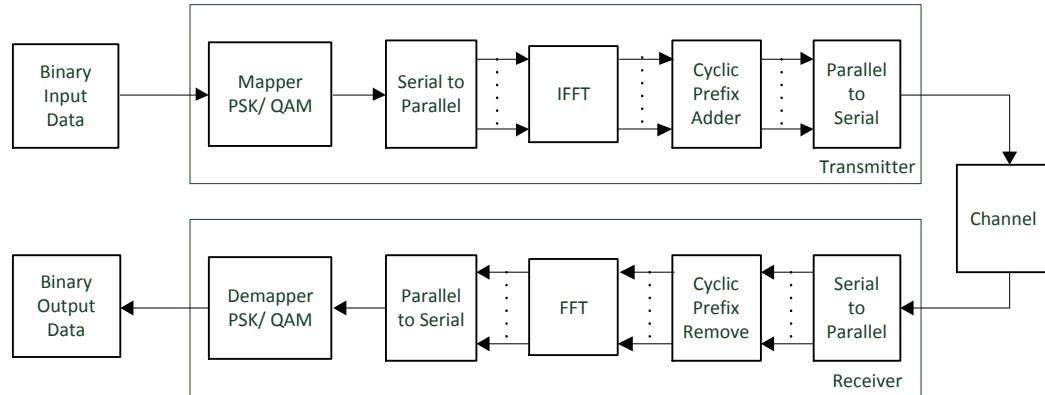


Figure 2.3. Basic baseband OFDM system

In practice, to prevent sharp transactions at the sample time boundaries, a windowing block is used for filter shaping. In conclusion, spectrum utilization is enhanced dramatically. Therefore, the baseband OFDM symbol can be written as below:

$$s_k(t) = \begin{cases} w(t - t_s) \sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} d_{i+\frac{N}{2}} \exp[j2\pi\frac{i}{T}(t - t_s - Tg)], & t_s \leq t < t_s + (1 + \beta)T_{sym} \\ 0, & \text{otherwise} \end{cases} \quad (2.6)$$

where T_g is the guard interval duration, $T_{sym} = T + T_g$ is the OFDM symbol period, symbol starting time $t_s = kT_{sym}$, and $w(t - t_s)$ is the pulse shaping window, which is usually a raised cosine filter, and β is the roll-off factor.

The OFDM mechanism is used in 802.11 and Wimax standards. [14] It has good tolerance against multipath and the receiver is easier to implement. We will see the details of implementations in practical systems.

2.2 OFDM Specifications in IEEE 802.11a Standard

2.2.1 Introduction of IEEE 802.11

The Institute of Electronic and Electrical Engineers (IEEE) has released IEEE 802.11 in June 1997. The standard defined physical and MAC layers of wireless local area networks (WLANs).

The physical layer of the original 802.11 standardized three wireless data exchange techniques:

- Infrared (IR);
- Frequency hopping spread spectrum (FHSS);
- Direct sequence spread spectrum (DSSS).

The 802.11 radio WLANs operate in the $2.4GHz$ (2.4 to $2.483GHz$) unlicensed Radio Frequency (RF) band. The maximum isotropic transmission power in this band allowed by FCC in US is $1Wt$, but 802.11 devices are usually limited to the $100mWt$ value.

The physical layer in 802.11 is split into Physical Layer Convergence Protocol (PLCP) and the Physical Medium Dependent (PMD) sub layers. The PLCP prepares/parses data units transmitted/received using various 802.11 media access techniques. The PMD performs the data transmission/reception and modulation/demodulation directly accessing air under the guidance of the PLCP. The 802.11 MAC layer to the great extend is affected by the nature of the media. For example, it implements a relatively complex for the second layer fragmentation of PDUs.

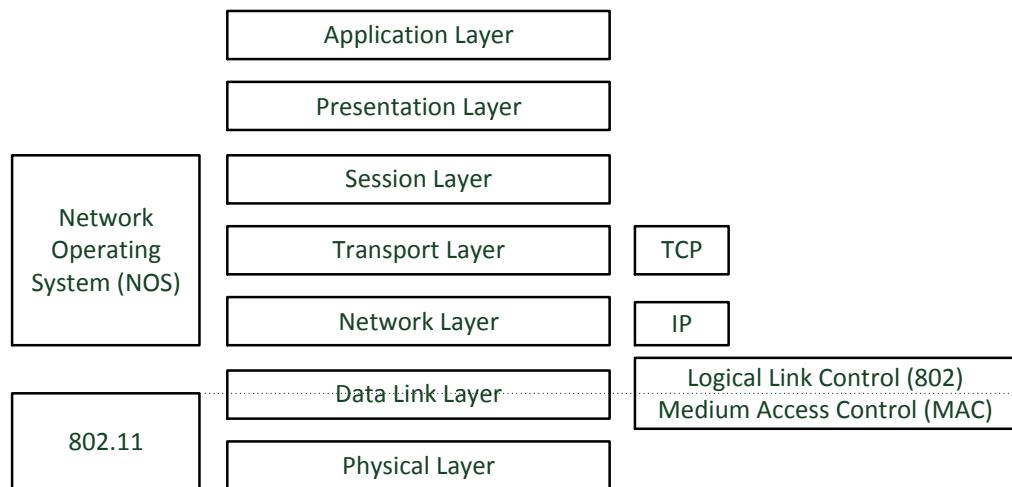


Figure 2.4. OSI Reference Model

2.2.2 System Design

In reality, having an anti-aliasing configuration, oversampling is performed before passing the digital signal to digital-to-analog converter. There are many other blocks in standards like channel coding, symbol interleaving and channel estimation. In comparison to the fundamental architecture shown in Figure 2.4, some other building blocks are added in a practical IEEE 802.11 design shown in Fig 00, marked with blue and dashed on line. At the transmitter, several "null" subcarriers or tones are reserved besides of the data subcarriers in order to perform oversampling of the transmitted signal. In this context, "null" means the symbol carried on this subcarrier has a value of zero. Besides, some other subcarriers used as pilot for channel

estimation are also inserted. The subcarriers are allocated at the input of the IFFT block to generate a phase shift. The windowing for pulse shaping is achieved after CP extension.

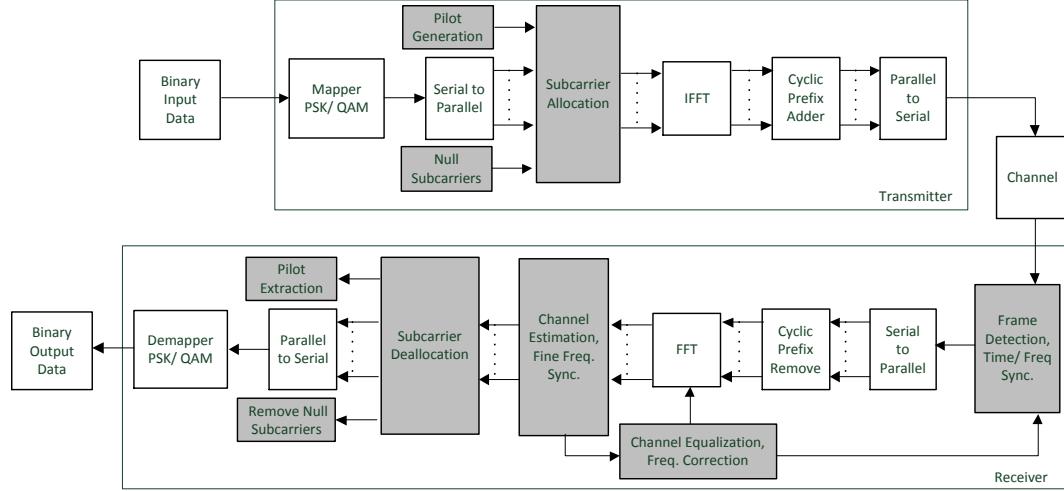


Figure 2.5. Architecture of an OFDM system

Considering the radio frequency parts, Figure 2.6 is concluded [14].

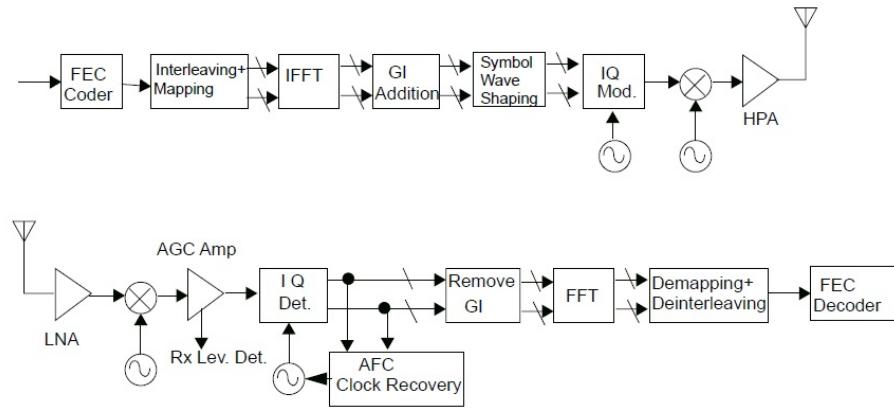


Figure 2.6. Transmitter and receiver block diagram of OFDM PHY

At the receiver side, the frame synchronization and detection for both timing and frequency is performed in the first stage. Channel estimation is performed after

the FFT block outputs the preambles in the frequency domain. The result is fed back to the FFT block for the equalization, which eliminates the effects of fading channel, while the fine synchronization for both timing and frequency is also added to further improve the system performance.

At least these basic parameters should be specified for a system design:

1. Delay spread expected for the channel (300ns)
2. Guard duration (800ns) which describes symbol duration ($4.0\mu\text{s}$)
3. Available bandwidth
4. Data rate

For indoor environment a delay spread less than 300 ns expected. We consider the guard duration 800 ns, which effectively protects the signal from ISI in the indoor environment and some of the outdoor wireless communication environments. Five times the guard duration for limiting the power and bandwidth loss is regarded for the symbol duration, and is set to $4.0 \mu\text{s}$ in our case. Hence, the OFDM symbol rate is 0.25 mega symbol per second (Mbaud).

Keep in mind, the useful OFDM symbol duration without the guard interval is $3.2 \mu\text{s}$. So, the subcarrier spacing, which is the reciprocal of the useful symbol duration, can be determined as 312.5 kHz. Assuming that there is a bandwidth of 20 MHz available, the number of subcarriers is calculated to be 64. This is exactly the same as the specification defined in IEEE 802.11a standard.

As mentioned, some tones are reserved for pilot subcarries (channel estimation), null subcarriers (realizing oversampling to avoid aliasing) and windowing (reduce the out-of-band spectral energy).

In our design we chose 48 data tones and 4 pilot subcarriers. So, 52 subcarries are occupied. Applying a raised-cosine window with roll-off factor $\beta = 0.02$ the total occupied bandwidth is

$$(1 + 0.02) \times (52 \times 312.5\text{kHz}) \approx 16.6\text{MHz} \quad (2.7)$$

To accomplish Oversampling, some zeros before and after the data vector are appended in the frequency domain as shown below.

$$\overbrace{0, 0, \dots, 0}^{1/2 \text{ appended zeros}}, \underbrace{d_{-\frac{N_d}{2}}, d_{-\frac{N_d}{2}+1}, \dots, d_{-1}}_{\text{Negative subcarriers}}, \underbrace{d_1, d_2, \dots, d_{\frac{N_d}{2}}}_{\text{Positive subcarriers}}, \overbrace{0, 0, \dots, 0}^{1/2 \text{ appended zeros}} \quad (2.8)$$

In the IEEE 802.11a transmitter, a 64 point IFFT multiplexes the orthogonal sub-carriers and the sub-carriers are renumbered as in Figure 2.7 before performing the Fourier transformation. Only 48 of them are used for data transmission and they are modulated by using BPSK, QPSK, 16-QAM or 64-QAM according to the Rate parameter. The sub-carriers P_{-21} , P_{-7} , P_7 and P_{21} are dedicated to comb-type pilot signals which are used to track the phase variations due to the time varying channel or a frequency offset error. The pilot sub-carriers are modulated by using BPSK and to prevent the generation of spectral lines, they transmit a pseudo random binary sequence generated by the same polynomial used in the scrambler.

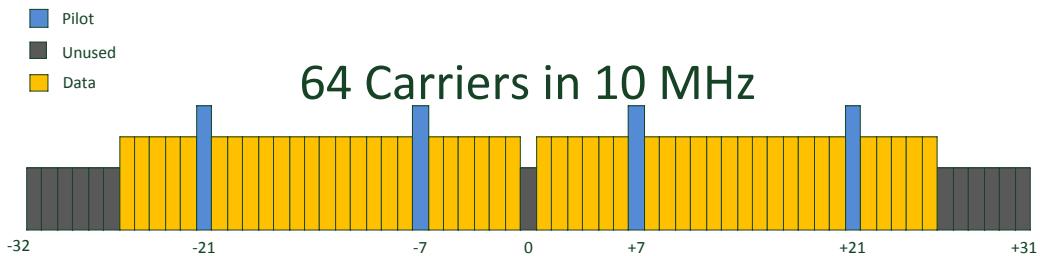


Figure 2.7. The frequency allocation of IEEE 802.11a sub-carriers

The nonzero data values are mapped onto the subcarriers around 0 Hz, and the zeros are mapped onto frequencies around sampling rate. Basically, in the BPSK modulation is applied on each subcarrier, each symbol for an individual subcarrier has one bits. The bit rate achieves without channel coding:

$$\frac{1}{4.0\mu s} \times 48 \times 1 = 12Mbps \quad (2.9)$$

The same calculation can be performed for QPSK to reach 24Mbps. But, channel coding will reduce this value. Variation of coding rates and modulation methods, In the 802.11a standard, the data rate ranges from 6 Mbps to 54 Mbps.

Table 2.2 shows the length parameter indicates the number of information bytes with different code rates:

The theoretical equation of the Bit-Error Rate for a QPSK channel is:

Parameter	Description	Value
B_w	Available channel bandwidth	$20MHz$
σ_τ	Delay spread of the channel	$< 300ns$
T_g	Guard interval duration (Cyclic Prefix)	$0.8\mu s$
T_{sym}	OFDM symbol period	$4.0\mu s$
T	Effective symbol duration (FFT period)	$3.2\mu s (= T_g - T_{sym})$
Δf	Subcarrier spacing	$312.5kHz (= 1/T)$
N_g	Number of guard samples	16
N	FFT size	$64 = B/\Delta f s$
N_d	Number of data subcarriers	48
N_p	Number of pilot subcarriers	4
N_u	Number of used subcarriers	52
B_u	Signal occupied bandwidth	$16.6MHz$
	Modulation type	$BPSK, QPSK$
R_b	Data rate without coding	$12Mbps, 24Mbps$

Table 2.1. System parameters defined for the proposed OFDM system

Data rate	Modulation	Code Rate	Coded bits per symbol	Data bits per symbol
6 Mbps	BPSK	1/2	48	24
9 Mbps	BPSK	3/4	48	36
12 Mbps	QPSK	1/2	96	48
18 Mbps	QPSK	3/4	96	72
24 Mbps	16QAM	1/2	192	96
36 Mbps	16QAM	3/4	192	144

Table 2.2. Rate dependant parameters in IEEE 802.11a Standard [14]

$$P_b(e) = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right) \quad (2.10)$$

It is discussed that the BER can be computed by considering the non-ideality which the two parameters *guard time* and *pilots* will inject into the result. The formulation would be:

$$P_b(e) = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0} \frac{T}{T + T_g} \frac{N_u}{N_u + N_p}} \right) \quad (2.11)$$

Replacing the standard value from Table 2.1 in the equation we have:

$$P_b(e) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}} 0.65\right) \quad (2.12)$$

2.2.3 IEEE 802.11a Standard in Time and Frequency

A packet of OFDM will be described here. In an OFDM frame, a preamble which carries no data is transmitted first, followed by the signal field which give some information about data and transmitted data. An OFDM frame has the general form as below:

$$s_{OFDM}(t) = s_{preamble}(t) + s_{signal}(t - T_{preamble}) + s_{data}(t - T_{preamble} - T_{signal}) \quad (2.13)$$

where

$$s_{preamble}(t) = s_{short}(t) + s_{long}(t - T_{short}) \quad (2.14)$$

As shown in Figure 2.8, the preamble starts with 10 short training symbols (STSs) from T_1 to T_{10} , followed by a guard interval (GI_2) and two long training symbols (LTSs) L_1 and L_2 . Both the short and long training sequences have an $8\mu s$ duration and the entire preamble lasts for $16\mu s$. Then, a $3.2\mu s$ signal symbol, as well as 800 ns guard interval is transmitted. This field bears some information necessary for the data symbols, such as the coding rate and length. Finally the various data symbols that carry user information are transmitted. Each data symbol has a duration of $4.0\mu s$, within which there is a 800 ns CP, as already described.

The application of STS and LTS for training are different. STS used for AGC, frame detection, coarse timing and frequency synchronization. Each symbol in this sequence has a duration of 800 ns and contains 16 samples, and is identical to one another. It will be shown in a professional system, auto-correlation will apply to this portion to perform such the operations. After the short training sequence is transmitted, a $1.6\mu s$ guard interval that contains 32 samples is introduced. The LTS is cyclically extended within this interval. Then two identical LTSs with the same duration of $4.0\mu s$ are followed. The LTS is used for fine frequency offset and channel estimation. It will be described that a cross-correlation with a stored array is done for extraction of the offset.

The data being transmitted should pass several stages and be prepared by PLCP (Physical Layer Convergence Procedure) before transmission. The preamble and the PLCP header are transmitted at 1Mbps regardless of the current data transmission speed. After the preamble the payload prepared by the MAC layer is sent to the receiver at the rate specified in the services field.

The picture below shows the OFDM packet data layout. It starts with training sequence (PLCP preamble), followed by the SIGNAL field and data. The data is followed by 6 tail bits and padding (not shown on the picture). Both the training sequence and the 24 bit SIGNAL field are transmitted at $6Mbps$ rate. The SIGNAL field tells the receiver at what rate the following data will be transmitted and indirectly defines the subcarriers' modulation technique employed. The BPSK, QPSK, 16-QAM and 64-QAM are the available choices. The SIGNAL field also delivers the length (12 bit) of the following data and includes a zero bit sequence for the data scrambler synchronization. The total training sequence and SIGNAL field transmission times add up to about $20\mu s$, which is an overhead equivalent to approximately 140 bytes transmission at the maximum transmission rate of $54Mbps$ defined by the standard.

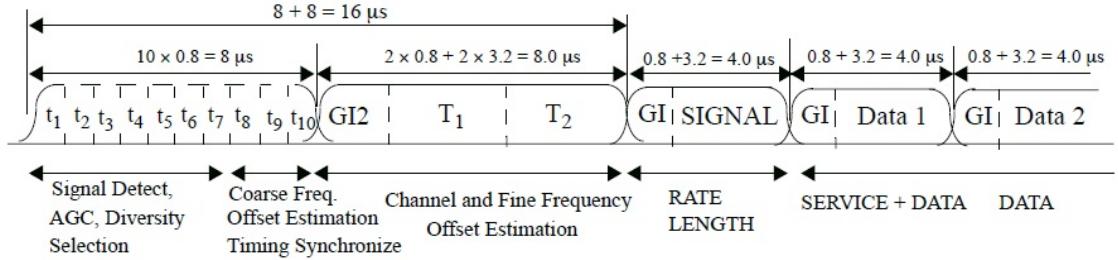


Figure 2.8. Preamble of IEEE 802.11 [14]

As we already analyzed, 52 subcarriers are used for an OFDM data symbol and pilot. Oversampling is achieved by adding 12 null subcarriers in order to eliminate aliasing which might occur during digital to analog conversion. Because FFT shift is performed, the null subcarriers with a value of zero are located in the middle of the input vector for the IFFT block. Note that dc carrier is not used to transmit data. The short and long training sequences can also be applied to this mapping rule, since they both have a length of 52 samples with frequency index from -26 to +26.

2.2.4 Origin of CFO

A simple model of a radio transmitter and receiver can depict the basis of the CFO source. [13] [11]

In Figure 2.9, ω is The carrier frequency and X_{BB} is the complex baseband signal, X_{RF} is a real-valued RF signal. These models simplified many other operations in a

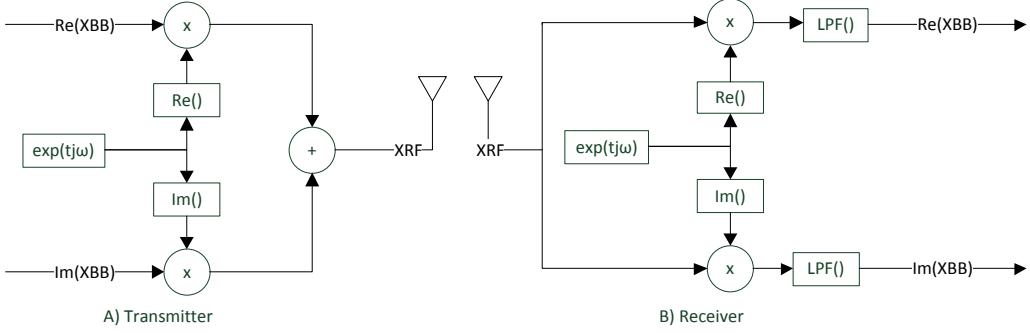


Figure 2.9. General models of a direct conversion RF

real RF transceivers although none of these affect the up/down-conversion processes as they relate to CFO.

These equations about transmit and receive processes can be written in equation (2.15):

$$\begin{aligned}
 X_{RF} &= TX(X_{BB}) \\
 &= Re(X_{BB}) \cos(\omega t) - Im(X_{BB}) \sin(\omega t) \\
 &= \frac{1}{2}(X_{BB}e^{j\omega t} + X_{BB}^*e^{-j\omega t})
 \end{aligned} \tag{2.15}$$

$$\begin{aligned}
 X_{BB} &= RX(X_{RF}, \omega) \\
 &= LPF(X_{RF}e^{j\omega t})
 \end{aligned}$$

Assume a signal S_{BB} transmitted with carrier frequency ω_S which is received with carrier frequency ω_D . we can express the received baseband signal D_{BB} in terms of the transmitted baseband signal S_{BB} and the carrier frequencies. Then:

$$\begin{aligned}
 D_{BB} &= LPF\left(\frac{(S_{BB}e^{j\omega_S t} + S_{BB}^*e^{-j\omega_S t})e^{j\omega_D t}}{2}\right) \\
 &= S_{BB}(e^{j\omega_D t})
 \end{aligned} \tag{2.16}$$

The received baseband signal is equal to the original baseband signal modulated by a complex sinusoid. In the frequency domain, this gives a received spectrum

equal to the transmitted one, only shifted away from DC by the difference in the carrier frequencies of the transmitter and receiver (i.e. $\omega_S - \omega_D$). This shift of the received signal is the baseband manifestation of carrier frequency offset.

2.2.5 Impact of CFO

There are two destructive impacts on an OFDM system. Firstly, the phase offset across subcarriers in a symbol which can be estimated and corrected in frequency domain to prevent errors in a constant rotated constellation. Some subcarriers are allocated as pilot tones which receiver can estimate phase errors.

The second effect of CFO is the degradation of orthogonality between subcarriers in receiver's FFT which causes inter-carrier interference (ICI). ICI acts as an effective SNR reduction as a result of CFO increasing.

The impact is displayed in Figure 2.10 which shows simulated OFDM system uses 10 MHz bandwidth and 64 subcarriers, 48 of which are random 16-QAM data symbols. CFO and AWGN are applied between the transmitter and receiver. The receiver model uses perfect knowledge of the CFO to correct the phase offset in each OFDM symbol, but does not implement any correction for ICI. [1]

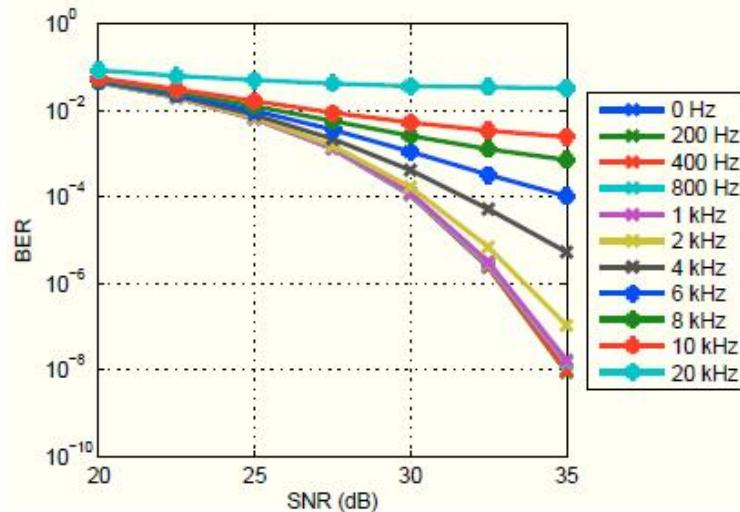


Figure 2.10. OFDM performance loss due to CFO-induced ICI.[1]

The results show that for large CFOs errors caused by ICI dominate performance, even at high SNR. It is also clear that for small CFOs performance is dominated by SNR. Specifically, for frequency offsets smaller than 1 kHz, the performance degradation due to ICI is negligible.

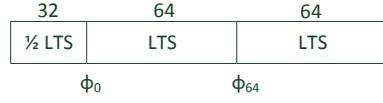


Figure 2.11. LTS in Time Domain.

Let's focus on LTS part in a OFDM symbol, Figure 2.11, for a while. In time domain, we have 160 samples in this section which creates two complete LTS symbols and a half and each LTS symbol has 64 samples.

We can say:

$$CFO \approx (\phi_{64} - \phi_0)$$

$$CFO_{EST} = \frac{f_s}{2\pi \cdot 64^2} \sum_{n=64}^{127} \phi_n - \phi_{(n-64)} \quad (2.17)$$

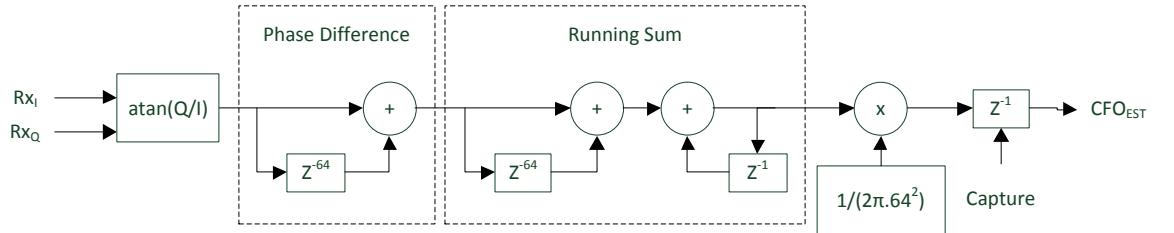


Figure 2.12. Time Domain CFO Estimation.

Which CFO_{EST} is the estimated CFO. To have such the structure in the Simulink we can arrange as Figure 2.12.

2.2.6 Synchronization

The synchronization tasks is challenging in an OFDM-based communication system. Prior to performing channel estimation equalization and demodulation, OFDM symbol timing must be detected. The receiver has no information when a packet starts, and so the first synchronization task is packet detection. Once a packet has been

detected the remaining synchronization functions include coarse and fine timing recovery and carrier recovery.

Figure 2.8 shows the structure of the IEEE 802.1la standard preamble. The 10 short preambles (t_1-t_{10}) are identical 16-sample duration sequences. The cyclic prefix ($GI2$) is a 32-sample sequence and the long preambles (T_1 and T_2) are identical 64-sample sequences. As indicated in the figure, the various fields are used for packet detection, automatic gain control (AGC), diversity selection, coarse and fine frequency offset estimation, fine symbol timing estimation and channel estimation.

The packet detector is based on the Schimdt and Cox delay and correlate algorithm employed for acquiring symbol timing commonly. [5] The algorithm, as illustrated in Figure 2.13, is basically a sliding window correlator combined with an energy detector used to normalize the decision statistic and hence guard against fluctuations of the input signal power level. [4]

The sliding window P computes a auto-correlation between the input signal and a D -sample delayed version on short preamble interval. We chose $D=16$. The second sliding window R is used to compute the received signal energy in the cross-correlation interval.

$$P(n) = \sum_{m=0}^{L-1} r_{n+m} r_{n+m+D}^* \quad (2.18)$$

$$R(n) = \sum_{m=0}^{L-1} r_{n+m+D} r_{n+m+D}^* \quad (2.19)$$

The cross-correlation $P(n)$ and auto-correlation $R(n)$ are calculated according to Equation 2.18 and Equation 2.19 respectively. The decision statistic is computed as

$$M(n) = \frac{|P(n)|^2}{R(n)^2} \quad (2.20)$$

2.2.7 Channel Estimation and Distortion Reject

Imagine the estimated channel is \hat{H} and the input information is A . So, we can expect to compensate the signal as below:

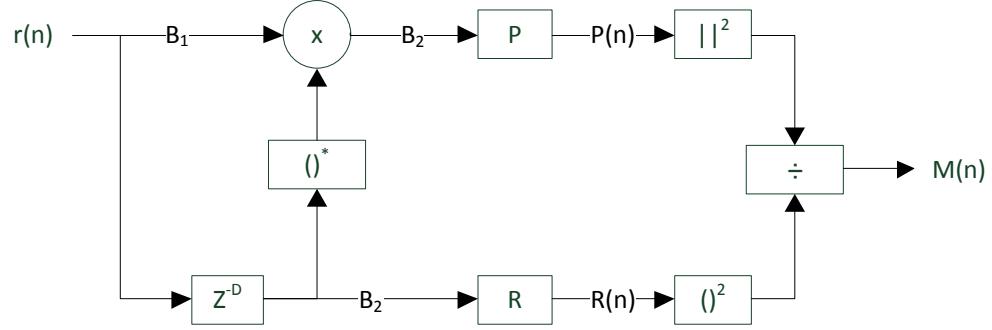


Figure 2.13. Schimdl and Cox Delay and Correlate Algorithm.

$$\begin{aligned}
 A_{\text{compensate}} &= \frac{A}{\hat{H}} \\
 &= \frac{A \cdot \hat{H}^*}{|\hat{H}|}
 \end{aligned} \tag{2.21}$$

If the estimation is done correctly and the channel is perfect without any distortion, we expect \hat{H} to be a flat shape in whole frequencies. Such the estimation is done by LTS to have the same input on all the tones at the transmitter and calculate the channel in the receiver by studying LTS.

Chapter 3

FPGA Implementations

The hardware is introduced with some details of the implementations. The main FPGA side project is done in Xilinx System Generator which is a high level alternative with standard scripting languages like VHDL and Verilog. An overview to select the radio board and the clock chain inside will be described.

Main blocks in transmitter and receiver is defined and the mechanism for packet detection is illustrated in details.

3.1 System Design in System Generator

Simulink[®] from The MathWorks[®] is a powerful graphical modeling system which allows complex systems to be designed using a block diagram methodology. Xilinx System Generator for DSP is a blockset for Simulink[®] which allows the modeling of fixed point systems which can be transformed into VHDL and targeted at an FPGA. Automatic generation of the bitstream is supported with the synthesis and implementation tools run from within the Simulink[®] environment.

The main core of an OFDM modulator and demodulator are the inverse FFT (IFFT) and FFT respectively. In 802.11a WLAN standard a 64-point transform with 52 of the subcarriers carrying user data in a BPSK, QPSK, 16-QAM or 64-QAM alphabet. The symbol rate in this system is $20MSym/s$. The OFDM symbol period is $4\mu s$, with $3.2\mu s$ of this interval occupied by the 64-point FFT symbol and the additional $0.8ps$ used for the cyclic prefix.

Figure 3.2 shows the main scheme of an OFDM system in transmitter and receiver. Another block of Channel is a Additive White Gaussian Noise which is used in simulation only.

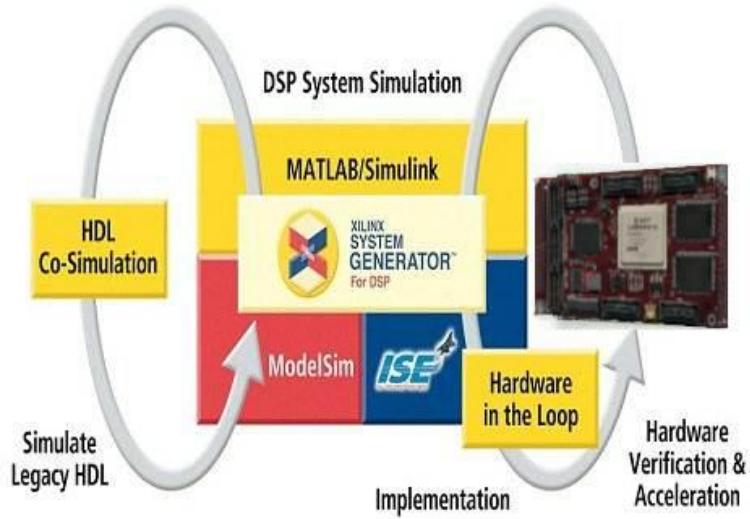


Figure 3.1. System Generator Cycle.

As you can see there are some others blocks which are necessary for system implementations. The whole system is connected to a main hard processor which is located in the FPGA. It is a ARM Cortex-A9 with maximum frequency of $666.66MHz$. EDK processor represents the main processor which connected to the OFDM block by AXI protocol.

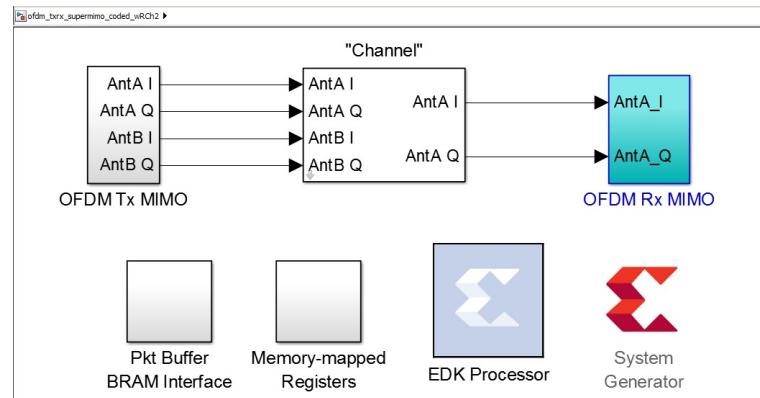


Figure 3.2. OFDM System..

Figure 3.3 illustrates the transmitter which consists of many blocks. Controlling of the time we have *TxControl* block for synchronization of the blocks. It also generate the semi-fixed preamble (LTS and STS) and the relevant Training signals for the system. *Training Data* generates the training pattern which is used to estimate the

channel frequency response. The main clock is IFFT which convert the time-based signals into frequency. In the current picture we set a 64 point IFFT although the recent design it upgraded to 256 as a result of the strategy changes. The data captured by the IFFT blocks are integrated in the *OutputBuffers*. *OutputMuxes* block chooses between two possible antenna to transmit the stream. In *PreSpin*, *Filters DACs*, some sub-blocks for soft gain and DAC preparation data are implemented. Besides, the are a generic block to rate change matter which can be activated by the processor.

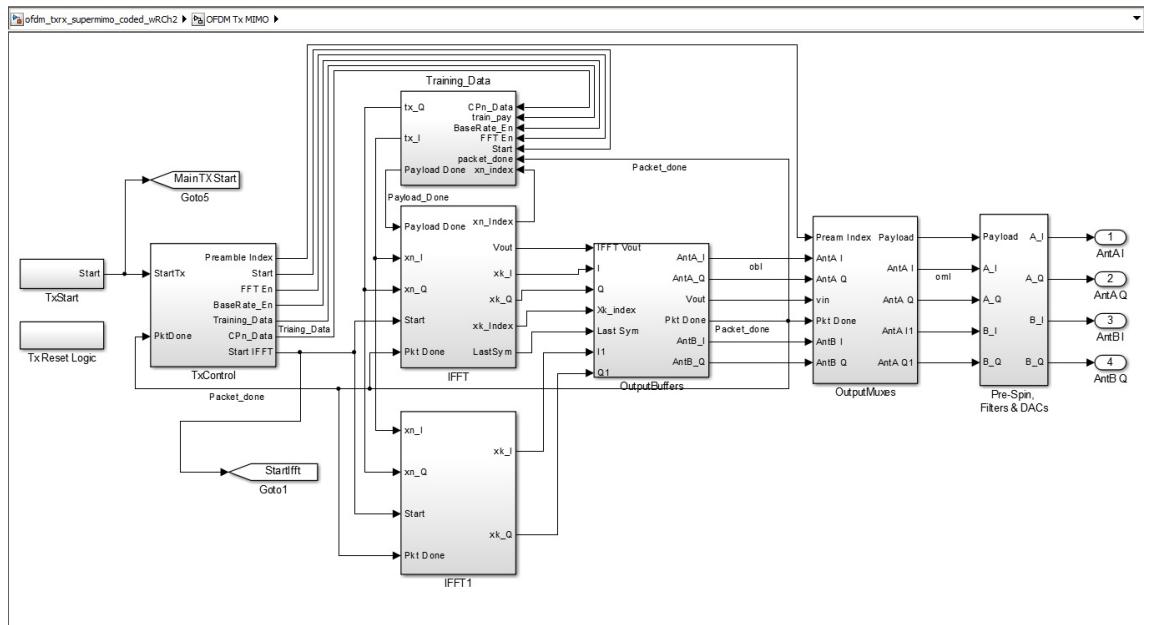


Figure 3.3. OFDM Transmitter Block.

Figure 3.4 demonstrates the receiver block with its main blocks. The input signals enter into the device in a I/Q form from the analogue board. The is *ADC inputs Antenna Selection* which we can switch between the two antennas and also the internal TX block which is reside in to the FPGA for the testing purposes. This block also adjust the input gain for the rest of the design. The frequency correction is done in the *Coarse Freq Correction* using the STS stream.

In the *Packet Detection* block an auto-correlation approach is done on the signal to detect the energy of the preamble in the beginning of STS shows in Figure 3.5. This is implemented based on the magnitude square of the both I/Q signals and comparing with a threshold after a sliding window. In other branch a multiplication

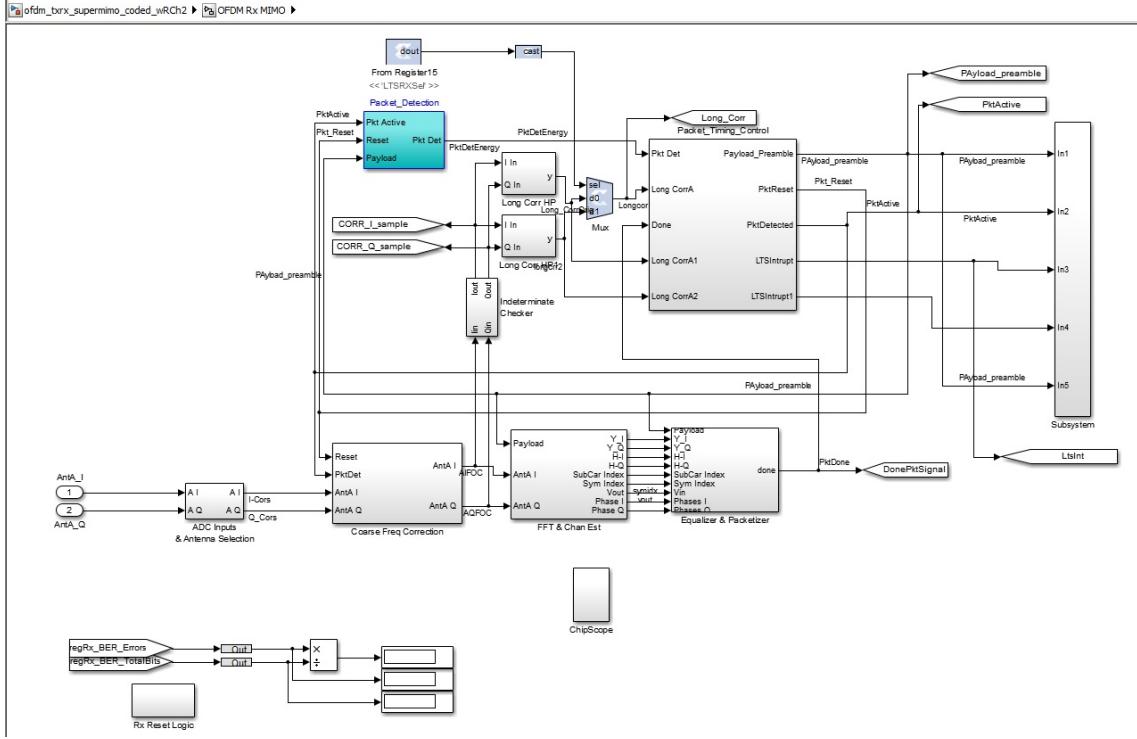


Figure 3.4. OFDM Receiver Block.

of the imaginary and real part with their 16 clock delayed version is calculated and the square magnitude in a sliding windows is detected. In *Detection Decision* we use some other threshold to be ensure if the two cross correlation peaks are detected in the right time to signal the packet detection.

As described in Section 2.2.7, the main part of the receiver is estimation of the channel which can be acquire by the LTS and training signal at the preamble. It is one of the most complex issue of the whole but the main part is shown in Figure 3.7 and ??.

3.2 Hardware Introduction

3.2.1 FPGA Board

The ZC706 evaluation board for the XC7Z045 All Programmable SoC (AP SoC) provides a hardware environment for developing and evaluating designs targeting the Zynq-7000 XC7Z045-2FFG900C AP SoC. The ZC706 evaluation board provides

3 – FPGA Implementations

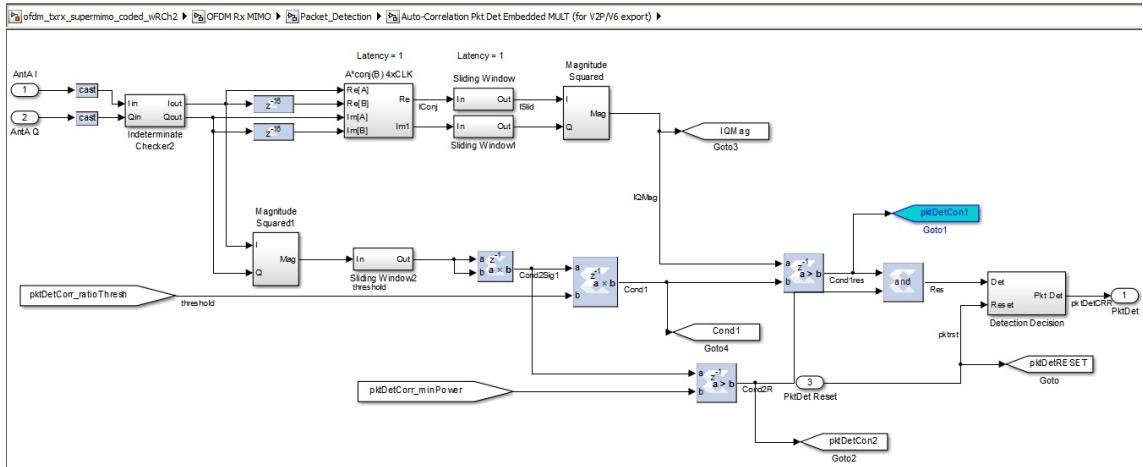


Figure 3.5. Auto-Correlation Block.

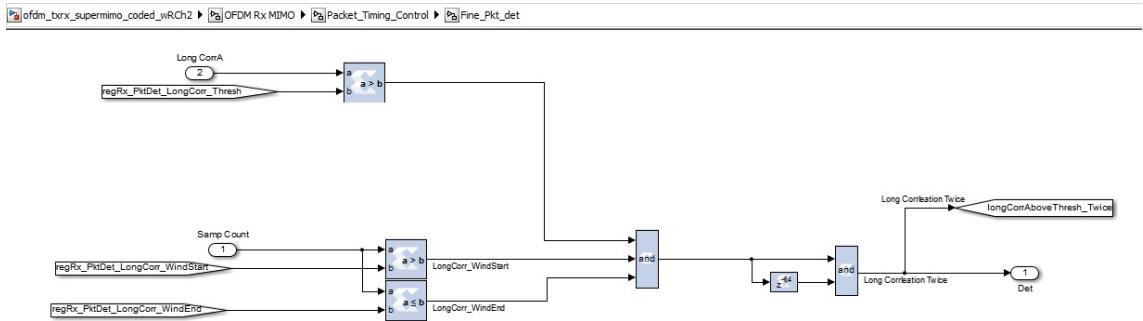


Figure 3.6. Fine Packet Detection Block.

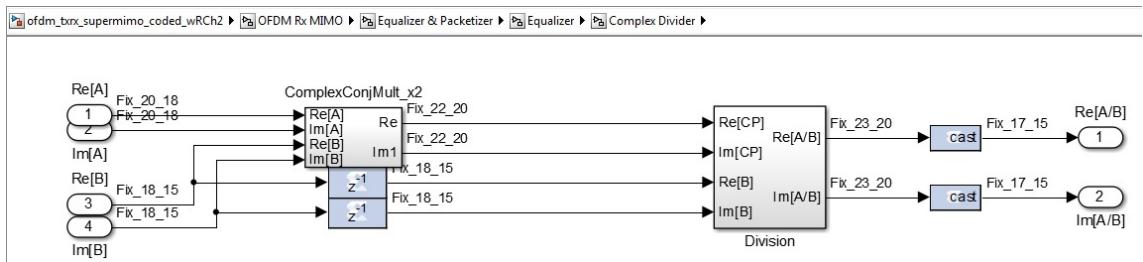


Figure 3.7. Complex Division Block.

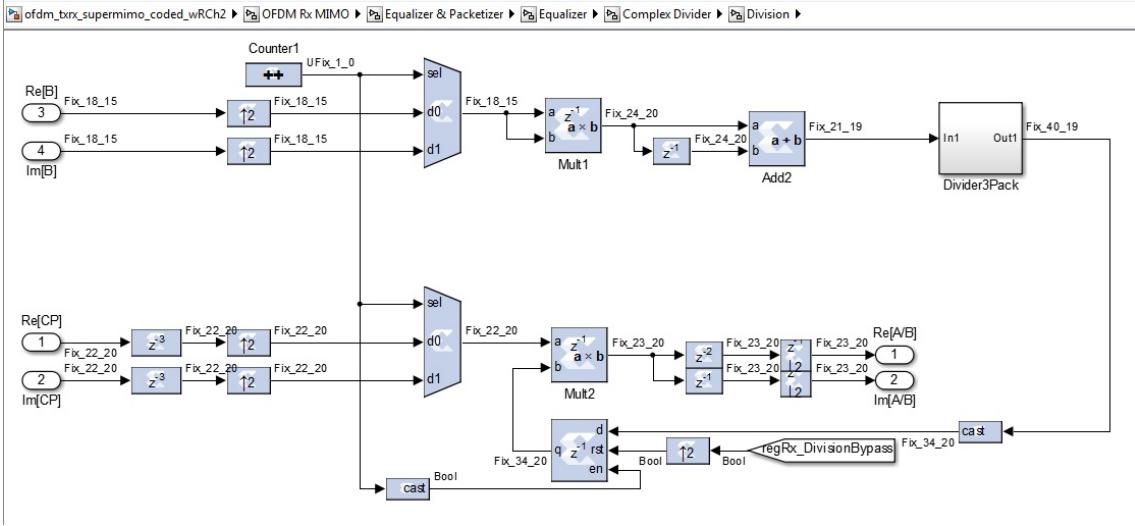


Figure 3.8. Division Block.

features common to many embedded processing systems, including DDR3 SODIMM and component memory, a four-lane PCI Express interface, an Ethernet PHY, general purpose I/O, and two UART interfaces. Other features can be supported using VITA-57 FPGA mezzanine cards (FMC) attached to the low pin count (LPC) FMC and high pin count (HPC) FMC connectors. For details of architecture see Section 3.6.

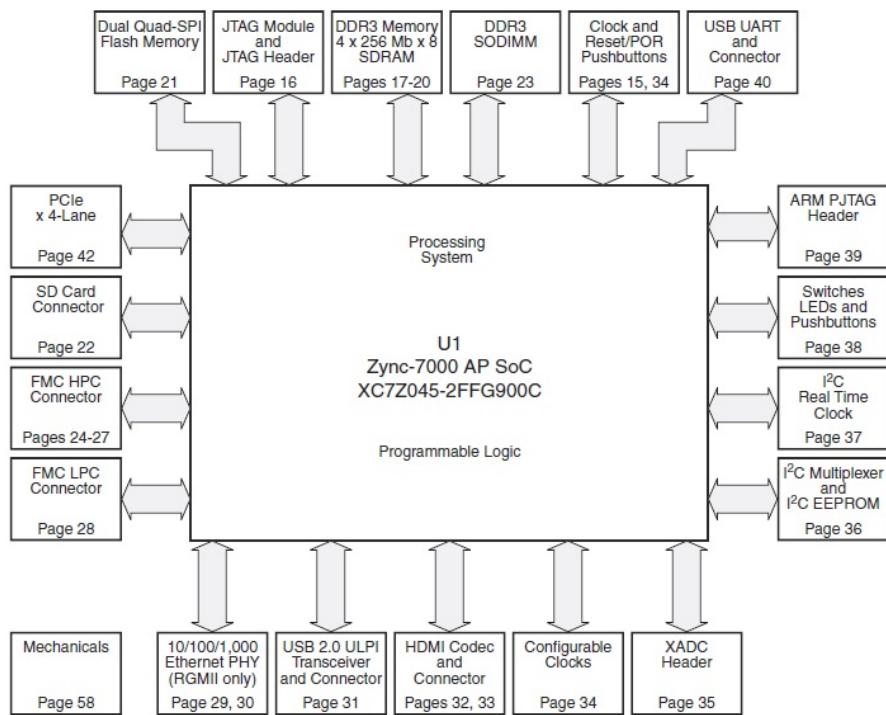
3.2.2 Radio Board

The AD-FMCOMMS1-EBZ [9] high-speed analog module is designed to showcase one of the latest generation high-speed data converters. The AD-FMCOMMS1-EBZ provides the analog front-end for a wide range of compute-intensive FPGA-based radio applications.

The AD-FMCOMMS1-EBZ enables RF applications from 400MHz to 4 GHz. The module is customizable to a wide range of frequencies by software without any hardware changes, providing options for GPS or IEEE 1588 Synchronization, and MIMO configurations. When combined with the Xilinx ZC706, AD-FMCOMMS1-EBZ enables a variety of wireless communications functions at the physical layer, from baseband to RF. With up to 4 GB of flash storage space, 512 MB of RAM, Gigabit Ethernet interface (depending on the base platform). The platform offers enough flexibility for many applications, and supports streaming data, and standard web interfaces to analyze transmitted RF data.



Figure 3.9. ZC706 Evaluation Board Block Diagram.



Note: Page numbers reference the page number of schematic 0381513.

UG954 v1 01 1002012

Figure 3.10. Xilinx Zynq-7000 SoC ZC706 Evaluation Kit

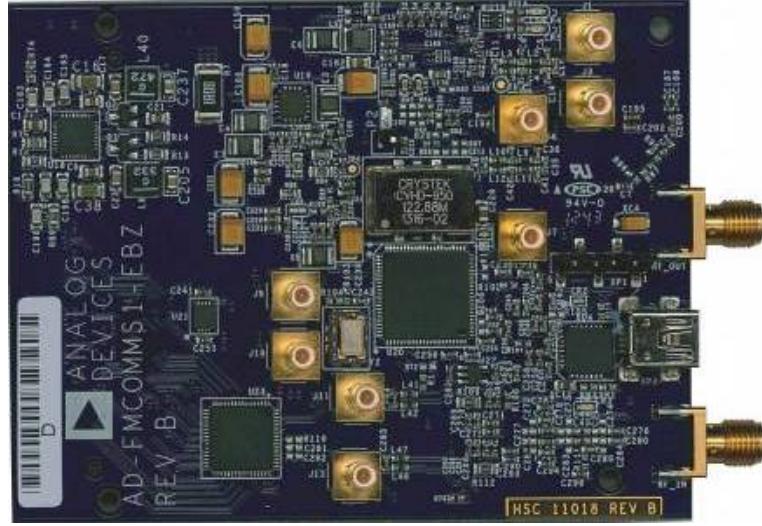


Figure 3.11. AD-FMCOMMS1-EBZ (Radio Board)

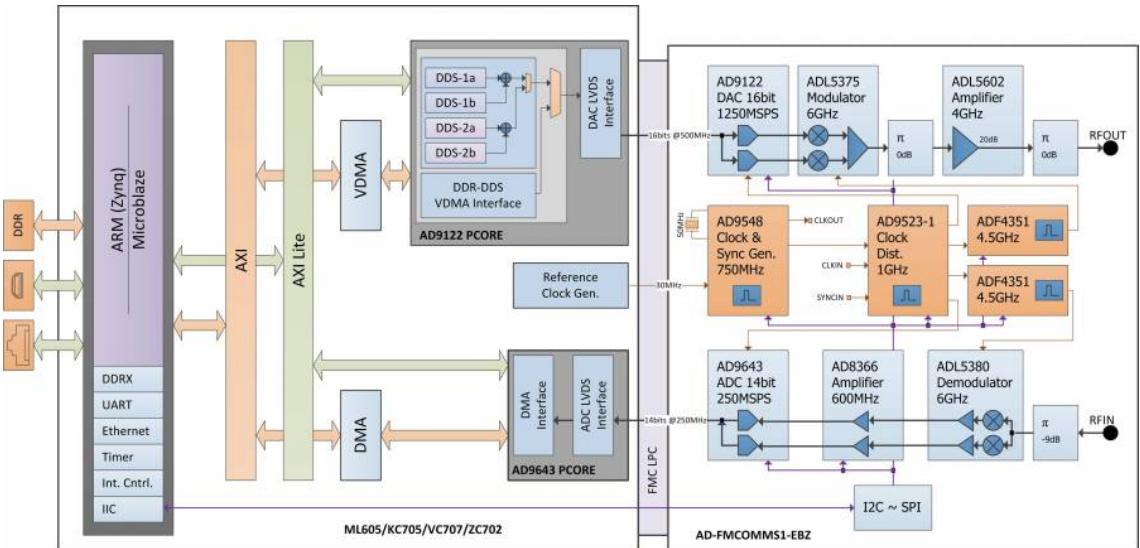


Figure 3.12. AD-FMCOMMS1-EBZ Block Diagram

3.2.3 Clock Chain on FMCOMMS1

Now, we discuss more about the clock chain and distribution mechanism on the board to find some meaningful number. As you can see in the figure, we configure the board and internal FPGA architecture to generate a 30MHz clock to the RF board. This 30MHz is just chosen because a relevant crystal mounted on the Zynq

board and the all generated clock is supposed to be in-phased with it. This 30MHz is an input for AD9548 as a clock generator/synchronizer [7] which has a very precise PLL inside to generate a 20MHz.

The AD9548 generates an output clock synchronized to one of up to four differential or eight single-ended external input references. The digital PLL allows for reduction of input time jitter or phase noise associated with the external references. The AD9548 continuously generates a clean (low jitter), valid output clock even when all references have failed by means of a digitally controlled loop and holdover circuitry. AD9548 is a very complicated device to generate 20MHz with maximum precision.

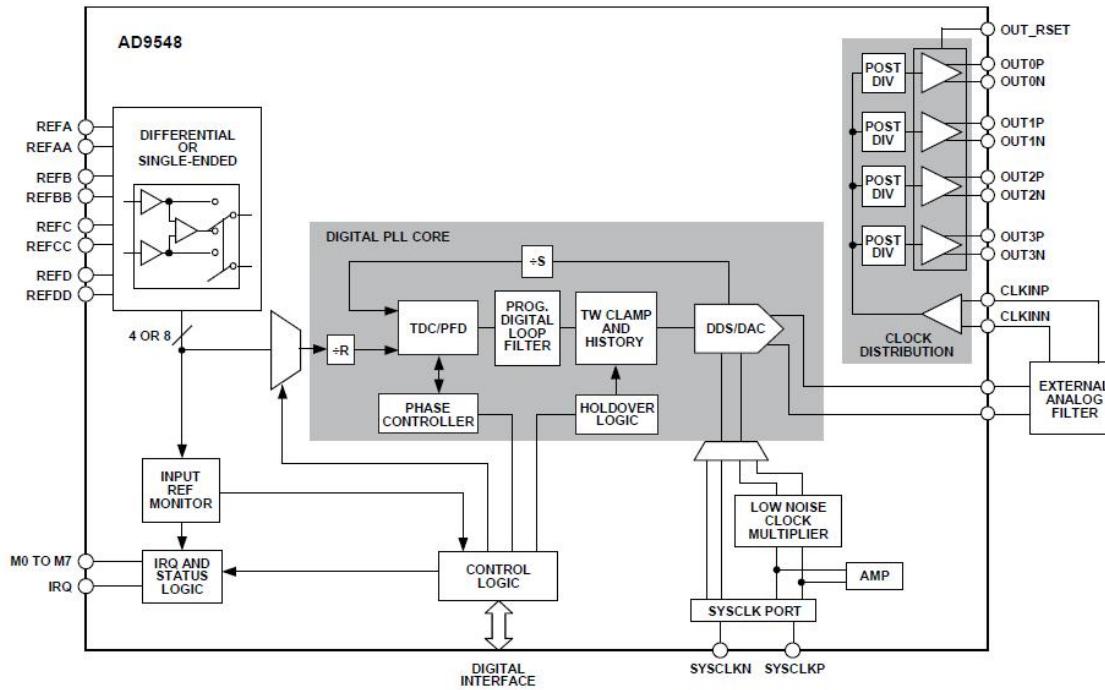


Figure 3.13. AD9548 Block Diagram

The next IC in the clock chain is AD9523-1 which is Low Jitter Clock Generator [8]. The AD9523-1 provides a low power, multi-output, clock distribution function with low jitter performance, along with an on-chip PLL and VCO with two VCO dividers. The on-chip VCO tunes from 2.94 GHz to 3.1 GHz. The AD9523-1 is defined to support the clock requirements for long term evolution (LTE) and multicarrier GSM base station designs. It relies on an external VCXO to provide the reference jitter cleanup to achieve the restrictive low phase noise requirements necessary for acceptable data converter SNR performance.

The input receivers, oscillator, and zero delay receiver provide both single-ended and differential operation. When connected to a recovered system reference clock and a VCXO, the device generates 14 low noise outputs with a range of 1 MHz to 1 GHz, and one dedicated buffered output from the input PLL (PLL1). The frequency and phase of one clock output relative to another clock output can be varied by means of a divider phase select function that serves as a jitter-free, coarse timing adjustment in increments that are equal to half the period of the signal coming out of the VCO. In our chain we have a 80MHz VCXO connected to AD9523-1. It is supposed to generated 40MHz for ADC, DAC and also the main OFDM architecture FPGA program. You can see the specification of the crystal oscillator in Figure 3.14 [6].

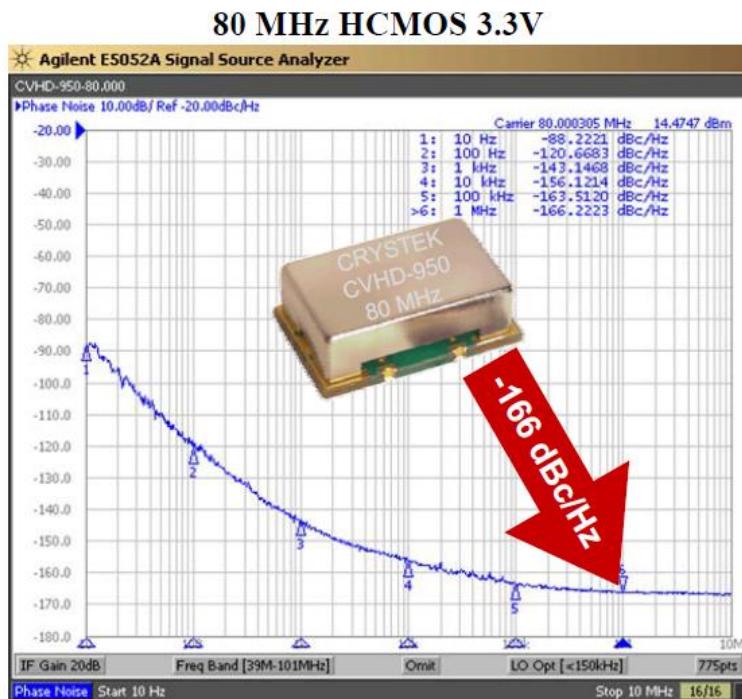


Figure 3.14. CVHD-950 Ultra Low Phase Noise Oscillator

3.3 Expectations for CFO on FMCOMM1

In implementation of an OFDM chain, we should have good understanding the range of carrier frequency offsets which can be expected on our hardware platform. Our RF board foundation is based on FMCOMMS1. As a result, the elements in term of phase -noise and CFO should be studied. The main RF frequency refrence is a Crystek CVHD-950 (VCXO). This VCXO provides a clock signal at a nominal

frequency of 80 MHz. Actual output frequency varies as a function of multiple factors, and is only specified by the manufacturer with some tolerance. The CVHD-950 is specified with a frequency tolerance of ± 4 ppm. Thus, we must design for a reference frequency of 80 ± 0.000320 MHz. Imagine our target RF carrier frequency is 2452 MHz which implies $2400 \text{ MHz} \pm 4$ ppm (or 2400 ± 0.009600 MHz).

The worst case CFO will occur when the transmit and receive nodes operate at opposite ends of this range. Thus, for operation in the 2.4 GHz band our OFDM transceiver design must be ready to handle any carrier frequency offset up to ≈ 20 kHz.

3.4 Time Domain CFO Correction

Prevention of the degradation of CFO, the receiver should estimate and correct the offset in the time domain before the FFT block. The FFT block translates the received signal into the frequency. Regarding to the variety issue of OFDM, many estimation algorithms have been proposed.

3.5 Carrier Frequency Offsets

As a result of the frequency variation between local oscillators of the transmitter and the receiver nodes that generate the carrier signals, carrier frequency offsets (CFO) is happened. It causes when the baseband signal is going to be translated to RF. The issue is understood well but the impact to overcome CFO and suppression this phenomena is always depend on the specific parameters of the given transceiver and the hardware.

The origin of the CFO effect is studied in this section. We explore in a specific scenario of OFDM and the impact on the hardware design. Both simulation and experiments will be demonstrated and the CFO estimation and compensation is described.

3.6 FPGA Architecture

Based on the Xilinx All programmable SoC architecture, the Zynq-7000 All Programmable SoCs [10] enable extensive system level differentiation, integration, and flexibility through hardware, software, and I/O programmability. Using the Zynq-7000 platform, you can design smarter systems with tightly coupled software based control and analytic with real time hardware-based processing and optimized system

interfaces.

As you can see in Figure 3.15 the foundation of Zynq-7000 is divided into two main parts. Firstly, Processor System which are two ARM processors and the fixed implemented peripherals. The rest are just the raw Programmable Logic which the main OFDM physical layer is implemented inside. We should build the gates, DSP and RAM in this region by VHDL programming or System Generator software in Matlab environment.

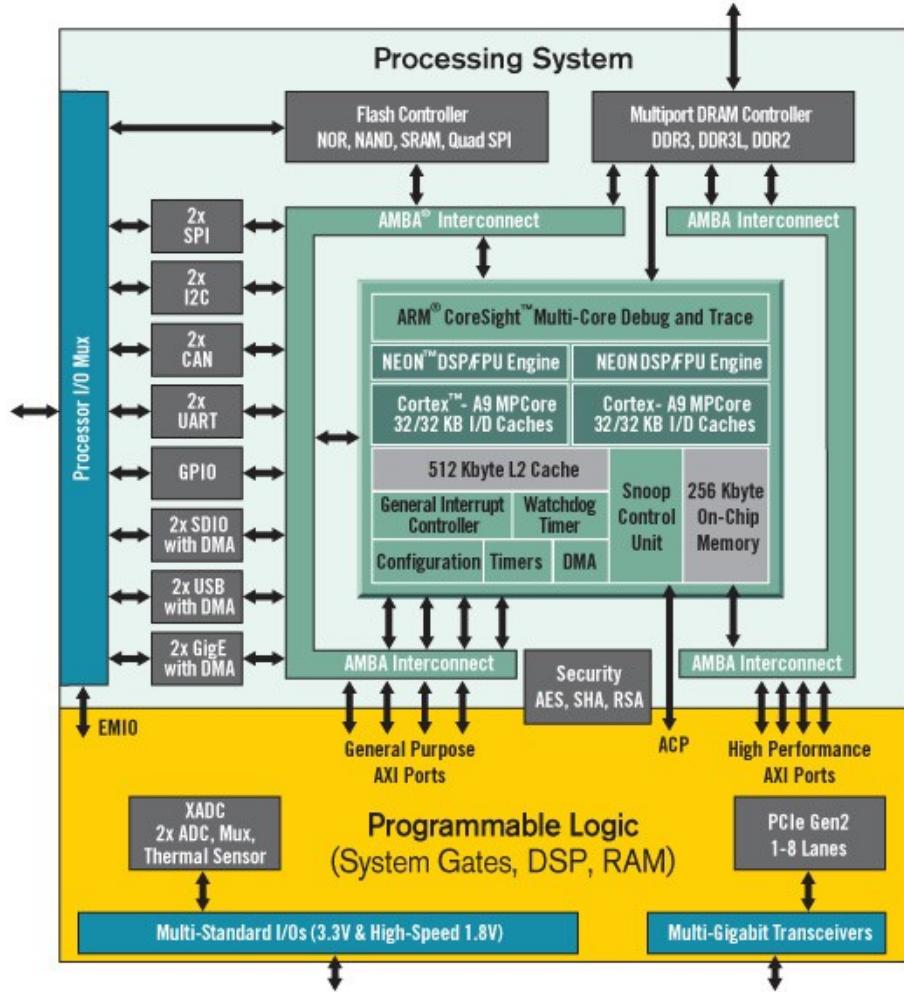


Figure 3.15. Zynq-7000 Diagram

The block diagram of the design is illustrated in Figure 3.16. We activate one of the ARM processors. In the TX chain, a PC sends a packet data via the Ethernet port to Zynq-7000. The EMAC block receives the packet and DMA it into the RX

Block RAM which is divided into 32 bank with size of $2K \times 64-bit$ which realize a Circular Buffer to relief the burst data stream enters from the asynchronous Ethernet port. As the OFDM block works in $40MHz$ and each $2K$ block reading takes maximum $50\mu s$ for the EMAC and OFDM-PHY pessimistically, the tolerance of the Ethernet stream will be $\frac{32 \times 64b}{50\mu s} = 40Mbps$ which proves good number of banks. The offset pointer of reading and writing by DMA MAC which is govern under a scatter-gather scheme and the OFDM-PHY is controlled by the ARM.

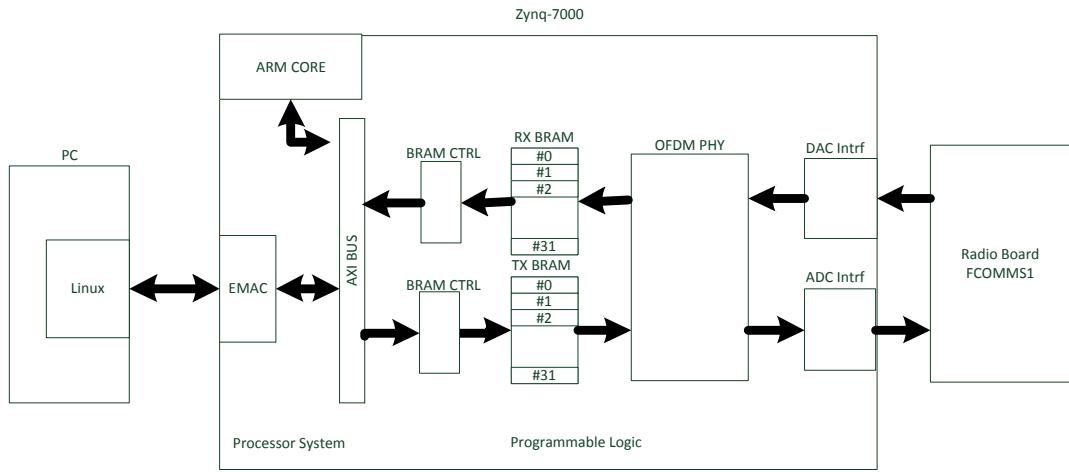


Figure 3.16. Design Block Diagram

As you can see in the block diagram the connection bridge between the Processor System and the Programmable Logic in the Zynq architecture can be AXI protocol. You can find the detail of AXI at ref..... There are some other communication protocols but in the Zynq design AXI works optimum. For easier programming issues in PC side, we used Linux Virtual Machine inside a Windows OS. It is very helpful because this configuration prevents unnecessary data exchange of the system and helps us to have a real estimation of the bit rate.

3.7 Test Methodology

The configuration set-up is consisted of two Zynq board each carrying a FCOMMS1 radio board. They are connected to two individual PC via Ethernet cables as shown

in Figure 3.17.

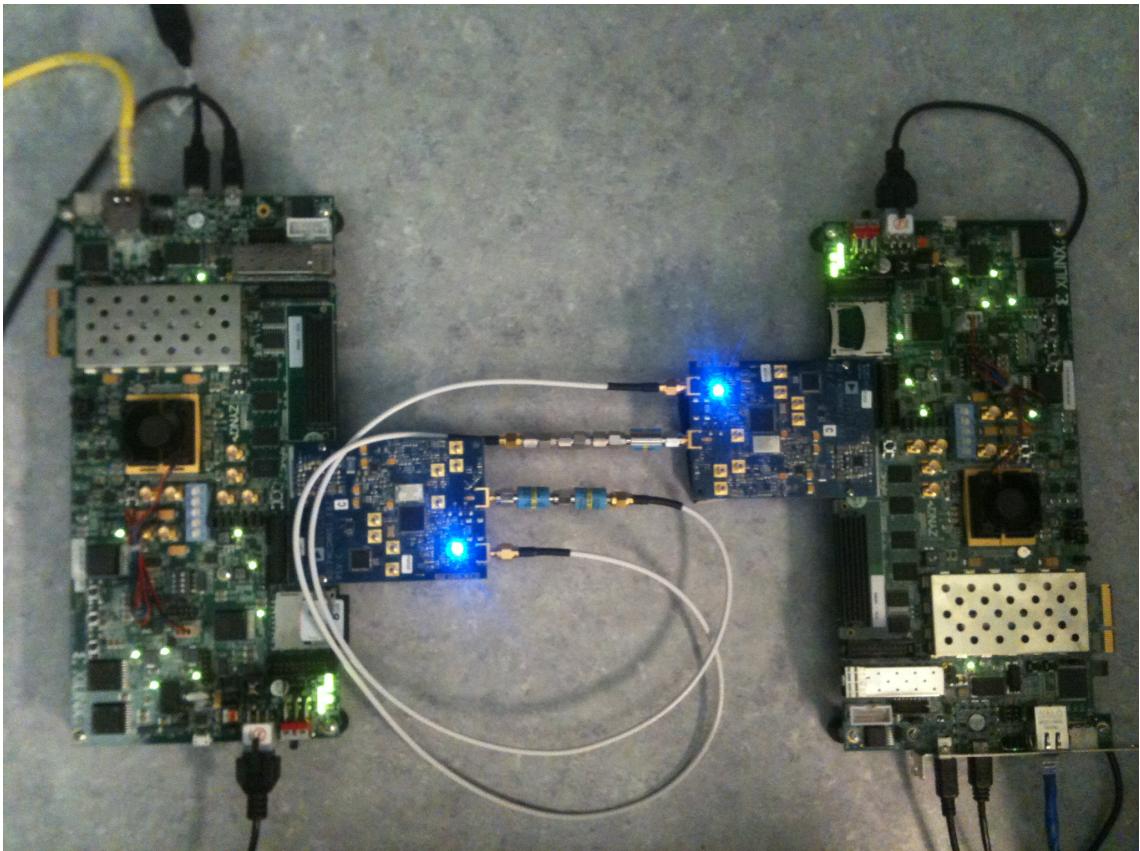


Figure 3.17. Hardware set-up

This configuration should be tested partially and have realistic estimation of the maximum possible bit-rate and then calculate SNR of channel. Having engineering steps to examine each hardware block, we check the loops illustrated in Figure 3.16. The maximum bit rate we could reach to exchange via Ethernet peripheral of the Zynq board which is called EMAC is $600Mbps$. This was a time consuming task to reach to this bit rate considering the complicated Direct Memory Access mechanism implemented in near contact of EMAC inside of ARM processor. Fortunately, there were many useful application examples dedicated by Xilinx but still it should study many document to understand the scheme in RX and TX of Ethernet.

Next, the correct configuration of the two TX and RX BRAMs are checked by directly data replacement between the two banks specified by the ARM processor. This was an important step because the maximum data rate is very depends on the

correct data reading and writing into these two blocks. There is an useful functionality of the RAM which is designed also in Zynq-7000 that called Error-Correction Code (ECC). This is a type of computer data storage that can detect and correct the most common kinds of internal data corruption. ECC memory is used in most computers where data corruption cannot be tolerated under any circumstances, such as for scientific or financial computing. The two BRAMs communicate with AXI Bus via an BRAM controllers. The configuration of the BRAMs and the controllers should be set accordingly.

The main part of the project is dedicated of the OFDM-PHY block with its sophisticated details.

Chapter 4

Sample Analysis and Conclusion

After explanation of the system architecture and hardware design, the test methodology and the analysis of the results from the hardware is defined. The results are captured thanks to ChipScope software which help for internal acquisitions in various points of the Programmable Logic of FPGA. There are other tests we did to have an estimation of the system in total. For instance, we send a random data stream from a PC and send it via Ethernet in different packet size to a board and compare it in another PC where we generate the same random set.

4.1 Hardware Samples and Analysis

A complete OFDM frame is illustrated in Figure 4.1. The position of preamble signal is completely distinguished in the figure which is comparable of the standard shown in previously except the *Training*. This section is special for our design for conveying some further information about the node. The detail of each preamble word is zoom in the next figures as defined in Section 2.2.3. Specifically, the STS section is shown in Figure 4.2.

The auto-correlation is perform to catch the begining of the preamble with is STS. The output of the block is shown in Figure 4.3 to detect the signal arrival.

The cross-correlation on the LTS with a pre-defined expected LTS is shown in Figure 4.4. As already described we have 2.5 LTS symbol in each frame. So, two peak are detected in the system.

The LTS section is supposed to carry a flat shape in frequency that is used for the channel response estimation. In the Figure 4.5 shows the frequency response of

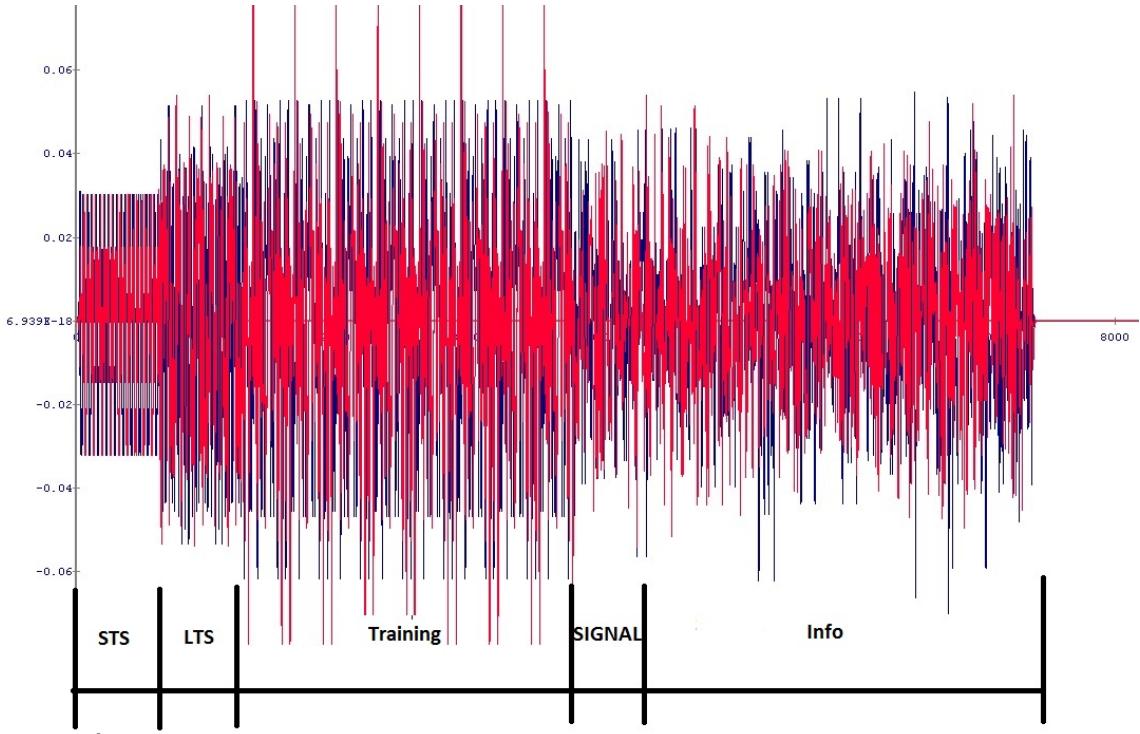


Figure 4.1. OFDM Frame (I/Q) detected in Chipscope.

the LTS in the in the input of the receiver side. Beginning of the system test, we make a loop after the DAC of the transmitter to the ADC of the receiver and there is no RF conversion. Practically, this shape represents the frequency response of the ADC/ DAC and some non-linear elements like induction and trances. Because, there are some traces in the path who rejects the DC frequencies we needed to use an IF filter to modulate the signal around $5MHz$.

Passing the signal through the RF side which modulate around $2.4GHz$ and demodulate it again we have a shape as shown in Figure 4.6. As you can see we still have the IF filter. The peak on 0 frequency is a result of the electronic elements which injects DC.

We have a shape of LTS frequency response like Figure 4.7 if the IF filter is not activated. As you can see we have the DC peak which can be rejected easily.

Figure 4.8 is the frequency response of the channel after an internal loop between the transmitter and the receiver on one FPGA board. As it is a perfect

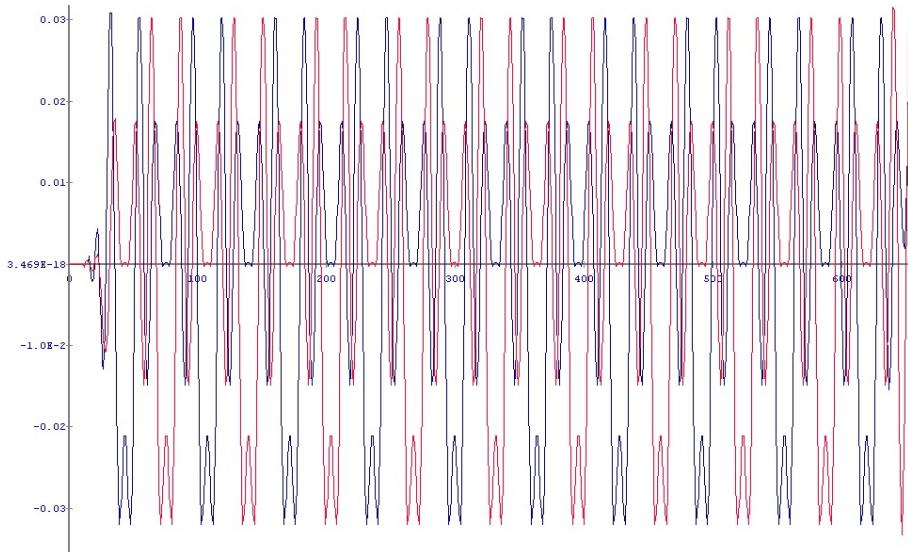


Figure 4.2. STS (I/Q) detected in Chipscope.

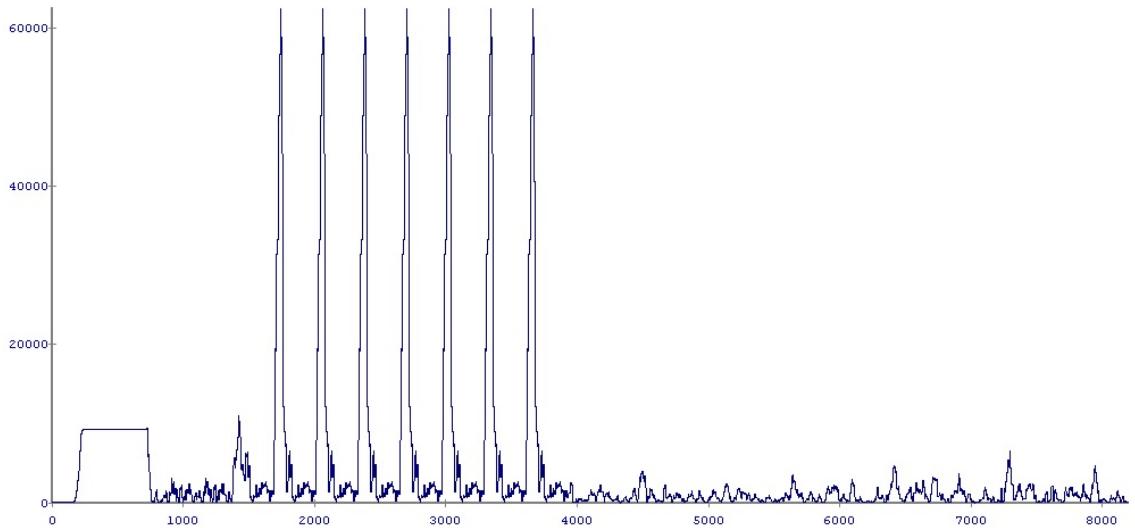


Figure 4.3. Auto-Correlation detected in Chipscope.

channel without any distortion we just have a perfect flat estimation of the channel. This response is used to compensate all the signal tones.

Figure 4.9 in a real channel after all the conversions of the transmitter and the receiver. The RF chain is activated in the scenario. Interestingly, we have a perfect detection of the symbols even after such the distortion.

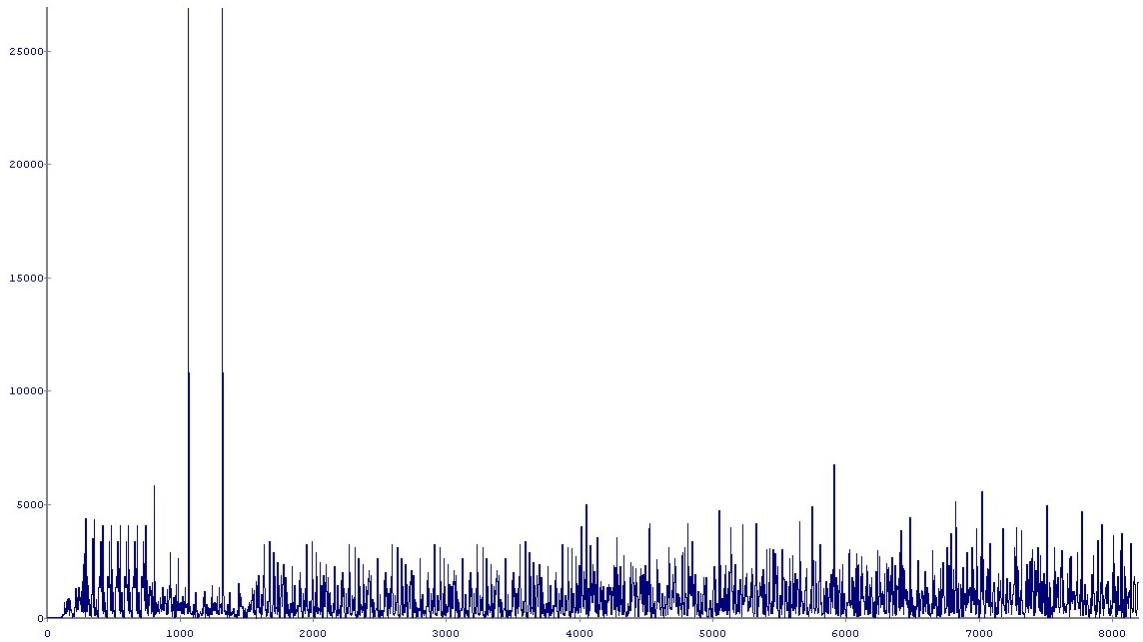


Figure 4.4. Cross-Correlation detected in Chipscope.

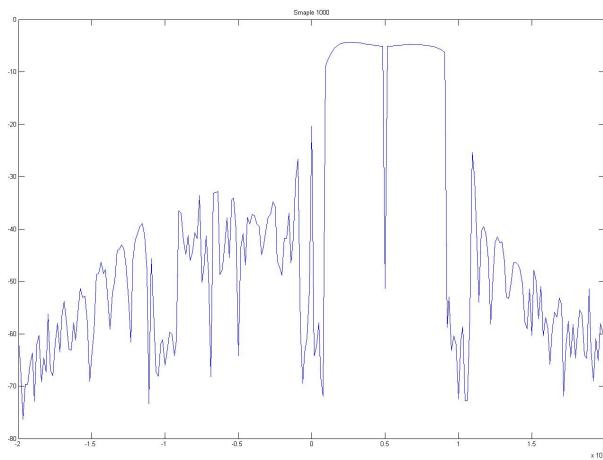


Figure 4.5. LTS Spectrum in Baseband chain (IF filter on 5MHz is enable)

Figures 4.10 and 4.11 compares two OFDM frames after the FFT block on the receiver side. The coding algorithm is a convolutional code which borrowed from another model and seen as a black-box for our system. Just do not consider the first

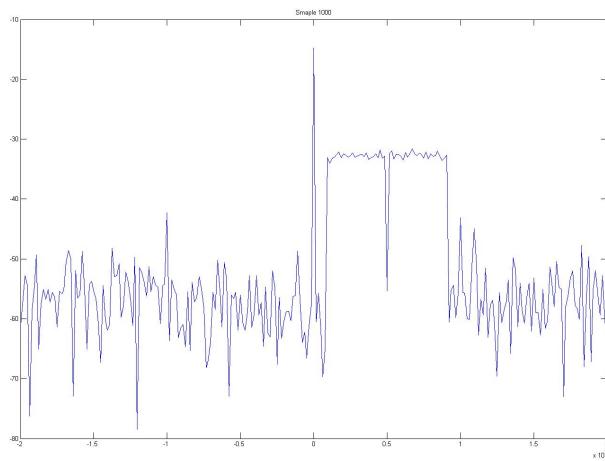


Figure 4.6. LTS Spectrum- passed RF chain (IF filter on 5MHz is enable)

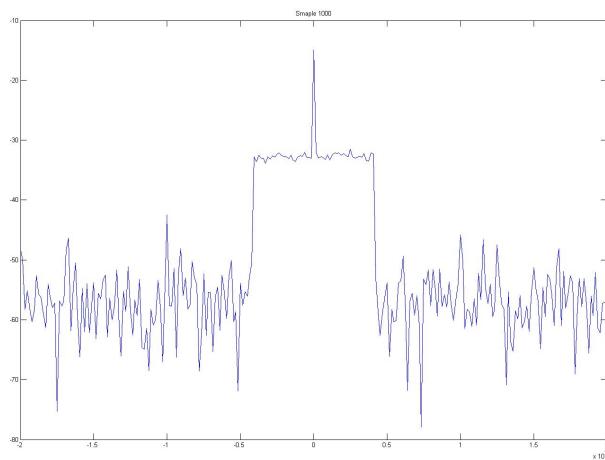


Figure 4.7. LTS Spectrum- passed RF chain (IF filter is disable)

8 symbols because they are the training block which are not defined in the standard and are the customization on our specific application. The next 8 frames in the coded and 4 in the non-coded are the base rate frames to report to the receiver about the modulation, size of message, sequence number and some other detection information. the rest are the message and the valuable information for the higher levels of the system.

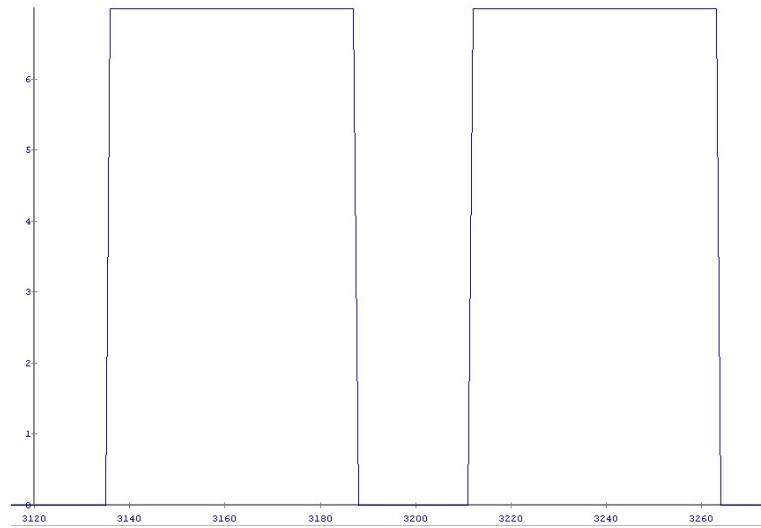


Figure 4.8. Frequency Response of a perfect channel detected in Chipscope

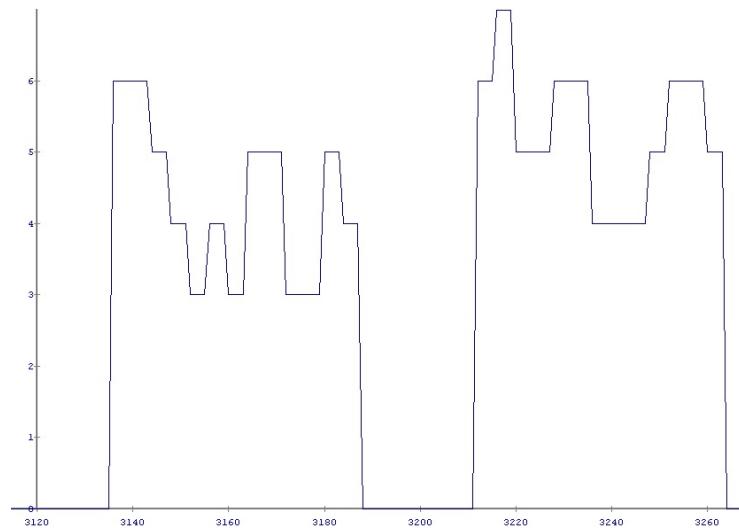


Figure 4.9. Frequency Response of an on-air channel detected in Chipscope

4.2 Bit-Error Rate Calculation

Will be add later!!!

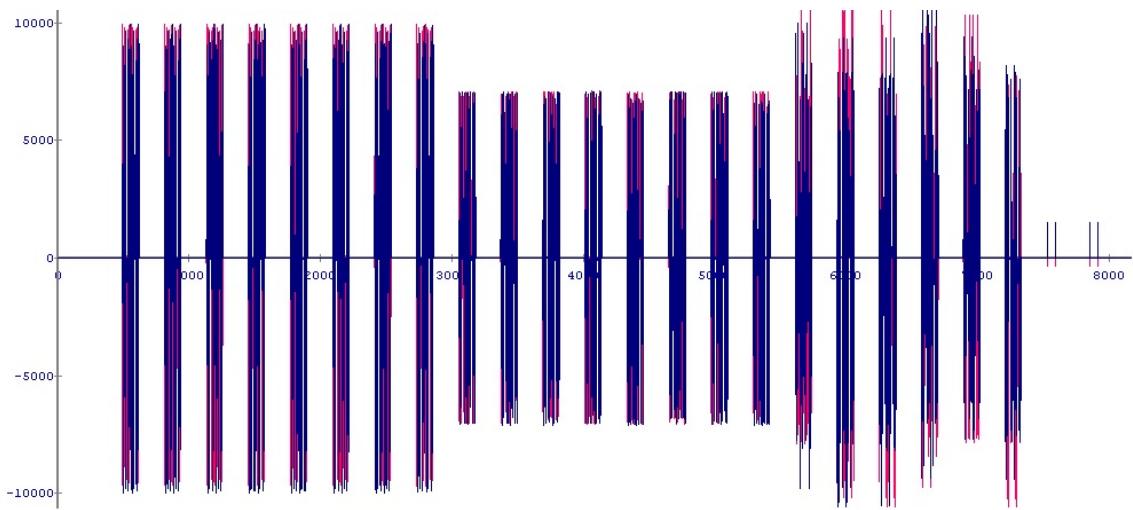


Figure 4.10. OFDM Symbols of a 16QAM 64-byte message Coded 1/2 rate

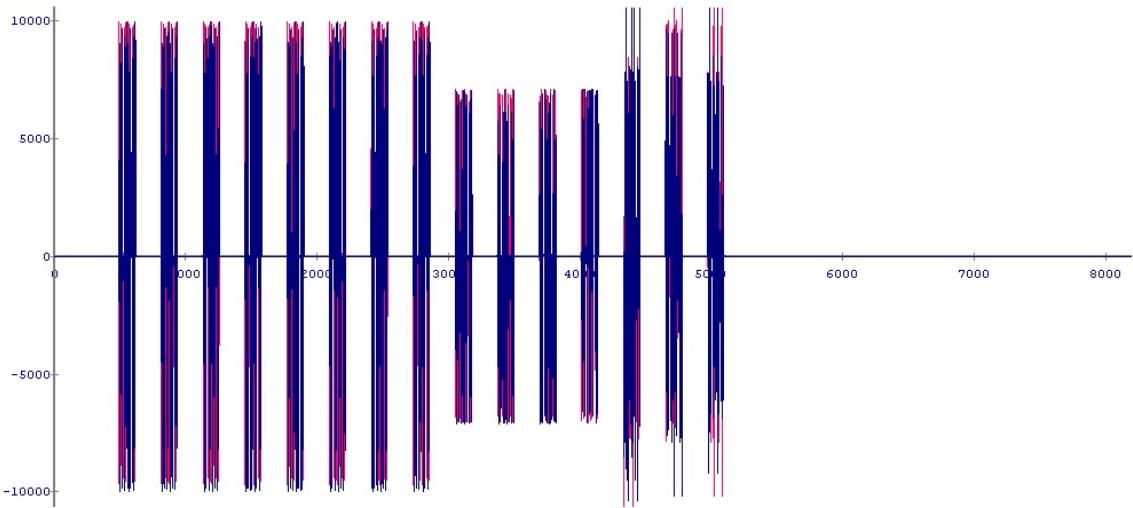


Figure 4.11. OFDM Symbols of a 16QAM 64-byte message no-Coded

4.3 FPGA Resource Consumption

The device used in our implementation is XC7Z045-22FFG900C and in Table 4.1 you can find the device utilization:

Keep in mind this table shows all the utilization of the ADC/DAC interface,

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	42,703	437,200	9%
Number of Slice LUTs	59,787	218,600	27%
Number used as Memory	5,384	70,400	7%
Number of occupied Slices	21,749	54,650	39%
Number of DSP48E1s	149	900	16%

Table 4.1. Device Utilization Summary (actual values)

BRAM, clock generator and OFDM PHY module. For OFDM PHY module we could reach Net Skew 0.51 ns and maximum Delay 1.91ns. About the ARM processor, we used one of the ARM cores although we have a dual core architecture. It works with 666.6MHz clock. It is almost 20% of the processor.

4.4 Conclusion

This thesis has presented the theoretical analysis and simulation and FPGA implementation details of a baseband OFDM system with channel estimation and timing synchronization. A radio board also is explained for the practical usage and proof of the feasibility. The OFDM system is prototyped based on IEEE 802.11a standard and transmits/receives signals on a 20 MHz bandwidth. The conceptual design is done in System Generator and ported on a FPGA. With QPSK modulation scheme, the system achieves a throughput of 24 Mbps.

Another critical part of the project is to reach an acceptable communication bit-rate in the processor side to the peripherals, more specifically Ethernet, for conveying data between two PC which is done perfectly.

4.4.1 Future Work

No doubt, System Generator is a very powerful tool for conceptual proof but is not still an industrial support. There are still difference in the Simulation in Matlab and what we have in the hardware. To overcome this issue we needed to make some changes to have a logical margin of the hardware difficulties. The resource consumption is not very optimized which expected to have better performance using VHDL programming. There are many low-level techniques to manage the power, speed and area which is not applicable in the system.

Some other problems like model-based system maintenance, extend for the future

and the bug detection difficulties make us to hesitate to industrialized a System Generator model at the moment. However, the accuracy of the theory is done perfectly in the current mechanism.

There are some projects in the department to extend the model for MIMO system design which is a good starting point. Besides, they try to upgrade the FFT point to higher levels for better bandwidth usage. A customized radio board is in the future program activities.

Bibliography

- [1] Patrick Murphy. Design, Implementation and Characterization of a Cooperative Communications System. *Thesis for Doctor of Philosophy at the Rice University* December 2010.
- [2] Hanzo L., Akhtman Y., Wang L. and Jiang M. MIMO-OFDM for LTE, Wi-Fi and WiMAX: Coherent versus Non-coherent and Cooperative Turbo-transceivers. *John Wiley and Sons Ltd* United Kingdom, 2011.
- [3] Xiong F. Digital Modulation Techniques. *2nd Edition* Artech House, Boston 2006.
- [4] Chris Dick. FPGA Implementation of an OFDM PHY. *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference IEEE* 2003.
- [5] Schmidl, T.M.; Cox, D.C. Low-Overhead, Low Complexity [Burst] Synchronization for OFDM. *Communications, 1996. ICC '96, Conference Record, Converging Technologies for Tomorrow's Applications.* IEEE 1996.
- [6] CVHD-950 Ultra-Low Phase Noise Oscillators. *Crystek Crystals.* <http://www.crystek.com/crystal/spec-sheets/vcxo/CVHD-950.pdf>.
- [7] AD9548 Quad/Octal Input Network Clock Generator/Synchronizer. *Analog Device.* <http://www.analog.com/en/products/clock-and-timing/clock-generation-distribution/ad9548.html>
- [8] AD9523 14-Output, Low Jitter Clock generator. *Analog Device.* <http://www.analog.com/en/products/clock-and-timing/clock-generation-distribution/ad9523.html>
- [9] AD-FMCOMMS1-EBZ User Guide. *Analog Device.* <http://wiki.analog.com/resources/eval/user-guides/ad-fmcomms1-ebz>
- [10] Zynq-7000 All Programmable SoC ZC706 Evaluation Kit. *Xilinx.* <http://www.xilinx.com/products/boards-and-kits/ek-z7-zc706-g.html>

BIBLIOGRAPHY

- [11] Masoud Salehi John Proakis. *Digital Communications*.
- [12] Andrea Goldsmith. *Wireless Communications*.
- [13] Berkeley David Tse University of California and Pramod Viswanath University of Illinois. *Fundamentals of Wireless Communication*. 2001.
- [14] IEEE 802.11: Wireless LANs *IEEE 802: LOCAL AND METROPOLITAN AREA NETWORK STANDARDS* 2012.