



TAMPERE UNIVERSITY OF TECHNOLOGY

**Farid Shamani**

**DESIGN OF A FLEXIBLE TIMING SYNCHRONIZATION SCHEME  
FOR COGNITIVE RADIO APPLICATIONS**

Master of Science Thesis

Examiners: Prof. Jari Nurmi

Dr. Roberto Airoidi

Examiners and topic approved by the  
Faculty Council of the Faculty of  
Computing and Electrical Engineering  
on 9 October 2013.

# ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

**SHAMANI, FARID: DESIGN OF A FLEXIBLE TIMING SYNCHRONIZATION SCHEME FOR COGNITIVE RADIO APPLICATIONS**

Master of Science Thesis, 75 Pages

November 2013

Major: Digital and Computer Electronics

Examiners: Prof. Jari Nurmi

Dr. Roberto Airoidi

Keywords: Software Defined Radio, Cognitive Radio, Synchronization, Flexible Timing Synchronization, FIR Filters, Partial Reconfiguration

Advancements in wireless technology have increased different applications to demand higher data rate wireless access. Spectrum scarcity has come more into picture day by day. In this case, Cognitive Radios (CR)s are new emerged promising technology which are an alternative solution to use spectrum more efficiently. In concept, CR is defined as an intelligent wireless device which is always alerted about its environment by continuously sensing the spectrum as well as having the ability to dynamically adopt its radio parameters. Although, CRs can mitigate spectrum scarcity to some extent, a variety of challenges have emerged of which synchronization is one the most prominent.

This thesis first presents some of common synchronization techniques used in conventional receivers and, based on them, presents a flexible timing synchronization scheme in which the CR receivers are able to adopt their radio parameters with new information regarding to the spectrum.

The core content of the synchronizer is based on Finite Impulse Response (FIR) filter which performs as a multicorrelator on demand. To do so, different synchronization architectures have been applied to the design, including Multiplier-Less based correlator as well as Transposed, Sequential and Pipelined Direct Form FIR filters. Consequently, all the architectures are compared to each other in terms of power consumption, chip area, maximum frequency, etc. Compiled results show that the best strategy is to employ Multiplier-Less based multicorrelator as the fundamental functional unit of the synchronizer.

The aforementioned synchronization block is implemented on an Altera family FPGA board series Stratix-V. All the components are written in VHDL language and simulated through ModelSim software. Quartus-II version 12.1 environment is used to compile simulated codes.

## PREFACE

This thesis is done as a completion of the Master of Science degree in department of Electronics and Communications Engineering at Tampere University of Technology.

I would like to give my sincere gratitude to Professor Jari Nurmi for the given opportunity; who made it possible for me to do my research work beyond his support. I would like to express my deep appreciations to Dr. Roberto Airoidi for all his precious and friendly support and advices which led to accomplish this thesis. Many many thanks to Jari Nurmi's team, specially Tapani Ahonen, Waqar Hussain and Leyla Ghazanfari for their supportive cooperations.

I would like to express all my deepest acknowledgments to my family Masoud, Ada, Saeed, Sepide, Saghar and specially my Mother. Who I am today and whatever I have achieved, it is just because of them. Without their non-stop support, nothing would have been possible.

I would like to extend my appreciation to all my friends at Tampere University of Technology in particular Orod Raeesi, Kamiar Radnosrati, Saeed Afrasiabi, Nader Daneshgar, Mona Aghababaeae for their warm friendship and nice moments we have shared together. The warmest gratitudes to Vida Fakour Sevom for being right beside me and never let me down in all ups and downs .

Finally, I am grateful again to my brother Saeed and my sister Ada for their unutterable support. Special thanks to my sister-in-law Shide for every little things she has already done for me.

Tampere, November, 2013

Farid Shamani

# CONTENTS

1. Introduction . . . . .	1
1.1 Motivation . . . . .	1
1.2 Thesis Outline . . . . .	2
2. Wireless Communication Systems . . . . .	3
2.1 Single-Carrier Modulation . . . . .	3
2.2 Multiple Access Methods . . . . .	4
2.2.1 Frequency Division Multiple Access (FDMA) . . . . .	4
2.2.2 Time Division Multiple Access (TDMA) . . . . .	5
2.2.3 Code Division Multiple Access (CDMA) . . . . .	7
2.3 Multi-Carrier Modulations . . . . .	8
2.3.1 Orthogonal Frequency Division Multiplexing (OFDM) . . . . .	8
2.3.2 Non-Contiguous Orthogonal Frequency Division Multiplexing (NC-OFDM) . . . . .	13
2.4 Introduction to Cognitive Radio (CR) . . . . .	16
2.4.1 What is Software Defined Radio (SDR) . . . . .	16
2.4.2 What is Cognitive Radio (CR) . . . . .	17
2.4.3 Evolution of Radio Technology . . . . .	17
2.4.4 Dynamic Spectrum Access (DSA) . . . . .	18
3. Synchronization . . . . .	21
3.1 Effects of Poor Synchronization . . . . .	21
3.2 Synchronization Errors . . . . .	22
3.3 OFDM Synchronization Issues . . . . .	23
3.3.1 Synchronization Methods . . . . .	25
3.3.2 Overview of 802.11a Packet Structure . . . . .	30
3.4 OFDM Synchronization Steps . . . . .	32
3.4.1 Coarse Symbol Timing Detection . . . . .	32
3.4.2 Fine Symbol Timing Detection . . . . .	35
3.5 NC-OFDM Systems . . . . .	37
3.5.1 NC-OFDM synchronization issues . . . . .	38
3.5.2 Primary User Filter . . . . .	42
3.5.3 NC-OFDM Synchronization Steps . . . . .	42
4. State of Art in Synchronizer Architecture . . . . .	45
5. FPGA Implementation of Multicorrelator . . . . .	49
5.1 Design Implementation Issues . . . . .	49
5.2 FIR Filter . . . . .	49
5.2.1 Sequential Direct Form FIR Filter Architecture . . . . .	51
5.2.2 Transposed FIR Filter Architecture . . . . .	53

5.2.3	Parallel FIR Filter Architecture . . . . .	54
5.3	Design Implementations . . . . .	56
5.3.1	Input Classifications . . . . .	57
5.3.2	Memory Block . . . . .	58
5.3.3	Threshold Detection Block . . . . .	59
5.3.4	Controller Block . . . . .	59
5.3.5	FIR Filter Core . . . . .	59
5.4	Compilation Results . . . . .	63
5.5	Partial Reconfiguration . . . . .	63
5.5.1	Partitioning for Partial Reconfiguration . . . . .	64
5.5.2	Wrapper Logic . . . . .	65
5.5.3	Freeze Logic . . . . .	66
5.5.4	Partial Reconfiguration Host . . . . .	66
6.	Conclusions . . . . .	69
	References . . . . .	71

## LIST OF FIGURES

2.1	Basic Single-Carrier Modulation Methods [9]. . . . .	4
2.2	Principles of Frequency Devision Multiple Access . . . . .	5
2.3	Principles of Time Devision Multiple Access . . . . .	6
2.4	Principles of Code Division Multiple Access . . . . .	7
2.5	(a)Conventional non-overlapping multi-carrier modulation. (b)Overlapping multi-carrier modulation. [6, p.26] . . . . .	9
2.6	Comparison between (a) single-carrier FSK modulation and (b) multi-carrier OFDM modulation. . . . .	9
2.7	OFDM transceiver architecture [4, p. 38] . . . . .	10
2.8	Inter-Carrier Interference due to frequency offset [7, p. 432] . . . . .	11
2.9	Structure of an OFDM block with cyclic prefix . . . . .	12
2.10	(a) Illustration of ISI due to multipath delay; (b) zero-padding guard interval to avoid ISI; (c) guard interval with cyclic prefix to eliminate ISI and ICI [6, p. 28]. . . . .	12
2.11	(a) OFDM and (b) NC-OFDM schemes. . . . .	14
2.12	NC-OFDM transceiver architecture . . . . .	15
2.13	CR and its relation to SDR . . . . .	18
2.14	Radio technology evolutions [23] . . . . .	19
2.15	Spectrum utilization snapshot at Berkeley [22, p. 163] . . . . .	19
2.16	Spectrum utilization by employing DSA technology [1, p. 151] . . . . .	20
3.1	Effect of bad synchronization (the effect of external impairments, such as noise, have not been considered) . . . . .	23
3.2	802.11a packet structure . . . . .	24
3.3	Synchronization based on received signal energy . . . . .	26
3.4	Incoming signal lost within the noise due to the low SNR . . . . .	27
3.5	Principle of double slide window packet detection . . . . .	27
3.6	802.11a preamble structure [8, p. 51] . . . . .	28
3.7	Synchronization Based on preamble structured packet . . . . .	29
3.8	Autocorrelation of the preamble [25, p.67] . . . . .	32
3.9	(a) Sampling without CFO, (b) Effect of CFO . . . . .	33
3.10	(a) Sampling without CPE and (b) Effect of CPE. . . . .	33
3.11	Coarse symbol timing detection algorithms . . . . .	36
3.12	Two switch channel model . . . . .	38
3.13	Out-Of-Band control systems. . . . .	39
3.14	Received waveform containing transmitted signal (a) before matched filter and (b) after matched filter . . . . .	41

3.15	NC-OFDM synchronization steps . . . . .	42
3.16	NC-OFDM waveform occupancy by primary and secondary user [37] .	43
5.1	Synchronization block architecture . . . . .	50
5.2	Sequential direct form FIR filter architecture . . . . .	51
5.3	Critical path flow in the design . . . . .	53
5.4	Transposed direct form FIR filter architecture . . . . .	54
5.5	Parallel direct form FIR filter architecture . . . . .	55
5.6	Parallel direct form FIR filter with $N = 8$ . . . . .	55
5.7	Pipelined direct form FIR filter with $N = 8$ . . . . .	56
5.8	Synchronizer inputs . . . . .	57
5.9	Splitting 32-bit signal into two 16-bits signals . . . . .	57
5.10	Creating several copies of a process using GENERATE command. . .	60
5.11	Part of the code where an N-tap Transposed Direct Form (TDF) Finite Impulse Response (FIR) filter are created using GENERATE command. . . . .	61
5.12	Truncation of 32-bit temporary result to 16-bit final result . . . . .	61
5.13	Part of the code where an N-tap Parallel Direct Form (PDF) FIR filter are created using GENERATE command. . . . .	62
5.14	Configuring data flow (a) SCRUB and (b) AND/OR modes . . . . .	65
5.15	Wrapper logic scheme for (a) Persona-1 uses all three ports and (b) persona-2 uses two ports . . . . .	66
5.16	Schematic of a freeze logic . . . . .	66
5.17	Representation of (a) internal host and (b) external host . . . . .	67
5.18	How the Partial Reconfiguration (PR) host can be connected to the hard PR control block . . . . .	68

## LIST OF ABBREVIATIONS

- ALMs** Adaptive Logic Modules
- APP** A Posterior Probability
- ASIC** Application Specific Integrated Circuit
- ASK** Amplitude Shift Keying
- AWGN** Additive White Gaussian Noise
- CDMA** Code Division Multiple Access
- CFO** Carrier Frequency Offset
- CIR** Channel Impulse Response
- CLB** Configurable Logic Block
- CP** Cyclic Prefix
- CPE** Carrier phase Error
- CPLD** Complex Programmable Logic Device
- CR** Cognitive Radio
- CRC** Cyclic Redundancy Check
- DC** Delay and Correlate
- DDF** D-Flip-Flop
- DFT** Discrete Fourier Transform
- DSA** Dynamic Spectrum Access
- DSP** Digital Signal Processor
- DVB-T** Digital Video Broadcasting-Terrestrial
- EDA** Electronic Design Automation
- EDGE** Enhance Data GSM Environment
- FCC** Federal Communication Commission
- FDMA** Frequency Division Multiple Access



**FFT** Fast Fourier Transform

**FIR** Finite Impulse Response

**FPGA** Field Programmable Gate Arrays

**FSK** Frequency Shift Keying

**FZC** Frank-Zadoff-Chu

**Gbps** Giga Bit Per Second

**GI** Guard Interval

**GPP** General Purpose Processor

**GSM** Global System for Mobile

**HDD** Hard Decision-based Detection

**HDL** Hardware Description Language

**I** In-phase

**IC** Integrated Circuit

**ICI** Inter-Carrier Interference

**IEEE** Institute of Electrical and Electronic Engineers

**IDFT** Inverse Discrete Fourier Transform

**IFFT** Inverse Fast Fourier Transform

**I/O** Input/Output

**IP** Intellectual Property

**I/Q** In-phase and Quadrature-phase

**ISI** Inter-Symbol Interference

**JTAG** Joint Test Action Group

**LABs** Logic Array Blocks

**LDPC** Low-Density Parity-Check

**LEDs** Light-Emitting Diodes

**LEs** Logical Elements

**LO** Local Oscillator

**LTI** Linear Time Invariant

**LTS** Long Training Symbols

**LUT** Look Up Table

**MAC** Multiplier-Accumulate

**ML** Maximum Likelihood

**M-L** Multiplier-Less

**MMSE** Minimum Mean Square Method

**NC-OFDM** Non-Contiguous Orthogonal Frequency Division Multiplexing

**OFDM** Orthogonal Frequency Division Multiplexing

**OOB** Out-Of-Band

**PAL** Programmable Array Logic

**PAPR** Peak-to-Average Power Ratio

**PCI** Peripheral Component Interconnect

**PDF** Parallel Direct Form

**PLA** Programmable Logic Array

**PLL** Phase-Locked Loop

**PN** Pseudo Noise

**PR** Partial Reconfiguration

**PSK** Phase Shift Keying

**Q** Quadrature-phase

**QAM** Quadrature Modulation

**RAM** Random Access Memory

**RPDF** Retimed-Pipelined Direct Form

**SCO** Sampling Clock Offset

**SDD** Soft Decision-based Detection

**SDF** Sequential Direct Form

**SDMA** Space Division Multiple Access

**SDR** Software Defined Radio

**SINR** Signal to Interference and Noise Ratio

**SNR** Signal-to-Noise Ratio

**SoC** System-on-Chip

**SRAM** Static Random Access Memory

**STO** Symbol Timing Offset

**STS** Short Training Symbols

**TDF** Transposed Direct Form

**TDMA** Time Division Multiple Access

**VC** Virtual Carrier

**VCD** Value Change Dump

**VHDL** Very-high-speed integrated circuit Hardware Description Language

**WiMAX** Worldwide Interoperability for Microwave Access

**WCDMA** Wide Code Division Multiple Access

**WiFi** Wireless Fidelity

**WLAN** Wireless Local Area Network

**ZP** Zero Padding

**ZP-OFDM** Zero Padding Orthogonal Frequency Division Multiplexing

# 1. INTRODUCTION

Wireless data communication networks are one of the major concerns of developed countries with respect to the finite resource of radio spectrum. Therefore, several challenges where the physical layer is more involved have come into picture in order to use spectrum as efficient as possible. Variety of applications have occupied the entire spectrum including financial transaction, social interactions, security, etc. Although, both wired and wireless devices are capable of performing various services, such as audio and video broadcasting, web browsing, etc, with rapid evolution of microelectronics, wireless transceivers surrogated the traditional wired systems due to being more versatile as well as being portable. However, spectrum scarcities are increased more and more in that case [1]. At the moment, most of the prime spectrum has been assigned to so called primary user or licensed user and it is bothersome to find some free spectrum for new wireless applications.

## 1.1 Motivation

Since 1991, the development of Software Defined Radio (SDR) has been enabled in which the transceiver carries out the entire baseband processing in software. SDR defines a radio platform in which, at least, a portion of the implementation is held in software. In other words, any waveform can be applied to any frequency band. In general, standard patterns such as IEEE-802.11a/b/g based in software, can be easily replace in an SDR platform, while in traditional systems a complete replacement of the radio frequency hardware is needed. Therefore, in order to swap from a standard to another one, an expensive hardware upgrade should be performed. [1]

The basic idea of the Cognitive Radio (CR), which can be implied as an intelligent and advanced version of SDR, was proposed in 1998 and published in in 1999 by Joseph Mitola and Gerald Q. Maguire [2]. In principle, CR is a platform which can rapidly change its operating parameters by considering new environment characteristics. CR changes corresponding parameters in such a way that the user is not even notified. It is a new promising technology capable of detecting particular unused segments of the radio spectrum and employs them for secondary usage without interfering with the licensed users.

Although the above mentioned approaches look simple, several obstacles pose challenges from both sender and receiver point of view. One of the major concerns

in a CR receiver is to establish an accurate and robust synchronization scheme. In other words, how the receiver should be notified that in which frequency bands the transmitter is transmitting.

This thesis developed a new technique where the receiver distinguishes the characteristics of the secondary transmitter without any prior knowledge about the frequency band and the standard employed by the secondary user. Once the presence of a secondary transmitter is discovered, the receiver adopts its primary parameters to that of the transmitter.

## 1.2 Thesis Outline

This thesis is organized as follows:

Chapter 2 focuses on different types of modulation including single-carrier, multi-carrier and multiple access methods. Although single-carrier modulations are primary modulation techniques, CR technology is based on multi-carrier modulations. Therefore, in order to know how an accurate synchronization can be established, two major multi-carrier techniques are explained in detail. Thereafter, the concepts of SDR, CR and Dynamic Spectrum Access (DSA) methods are studied.

Chapter 3 concentrates more on synchronization issues. Followed by a brief introduction to the context of synchronization, different effects of bad synchronization are studied. Explanations regarding how, where and when the synchronization should be performed are narrowed to multi-carrier techniques.

Chapter 4 discusses previous implementations of the synchronizer with respect to their limitations and the state of art in synchronization architecture.

Chapter 5 explains different implementations of the synchronizer. The development kit is based on Altera Stratix-V family Field Programmable Gate Arrays (FPGA). The fundamental core of the synchronization block is studied step by step. This chapter completely investigates new emerged PR feature. Eventually, the compilation results related to each implementation are discussed in several tables.

Chapter 6 summarizes the entire work with respect to different implementations of the synchronization block. This chapter also compares results with each other and discusses about the trade-off between them.

## 2. WIRELESS COMMUNICATION SYSTEMS

The pervasive applications of wireless communication in modern society have led to several fields of research in which the physical layer is more involved. The harsh nature of channel with respect to different type of impairments threatening the signal, including scattering, reflection and diffraction, robust modulation, synchronization, channel estimation, etc, can only be few examples of above-mentioned concerns. [4]

Since this thesis is more dealing with synchronization scheme for CR systems based on Non-Contiguous Orthogonal Frequency Division Multiplexing (NC-OFDM), in order to understand the entire concept, single-carrier modulation techniques are briefly discussed. Then few basic Multiple Access methods are presented. Next, the principle of multi-carrier modulations in context of Orthogonal Frequency Division Multiplexing (OFDM) will be covered in detail, and, finally, before proceeding to the concept of CR, NC-OFDM will be studied.

### 2.1 Single-Carrier Modulation

The basic idea of the digital communication is to transmit information from sender to receiver via a propagation channel. Modulation, in brief, is the process in which the information signal, which is considered as the message signal, propagates through the channel after being multiplied by the carrier signal [5]. Figure 2.1 illustrates three main digital single-carrier modulation methods. According to [9], fundamental digital single-carrier modulation methods are listed as follows:

- *Amplitude Shift Keying (ASK)*: When the amplitude is the element to be varied.
- *Frequency Shift Keying (FSK)*: When the frequency is considered to be varied for signal carrier.
- *Phase Shift Keying (PSK)*: When the phase is the candidate to be varied.
- *Quadrature Modulation (QAM)*: When both amplitude and the phase are candidates to be varied, thereby two In-phase (I) and Quadrature-phase (Q) components are produced.

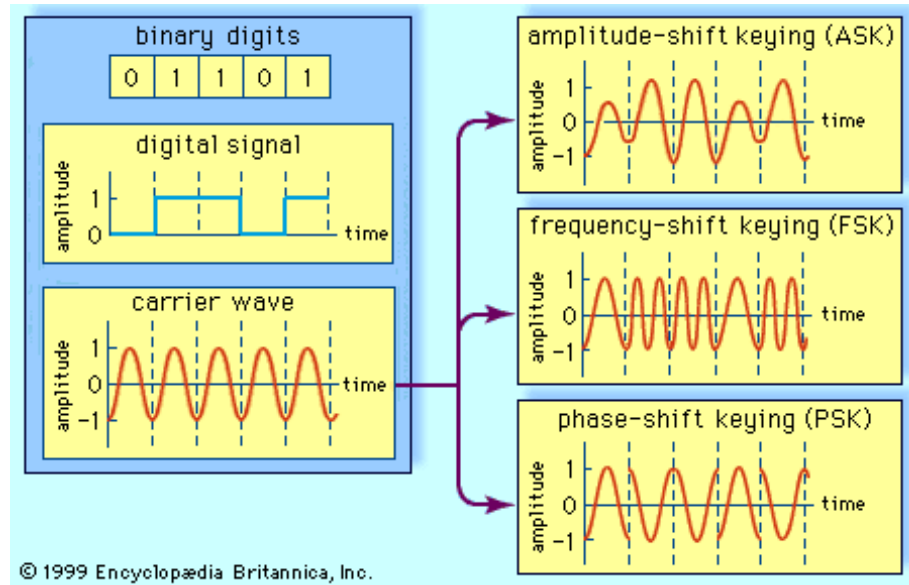


Figure 2.1. Basic Single-Carrier Modulation Methods [9].

## 2.2 Multiple Access Methods

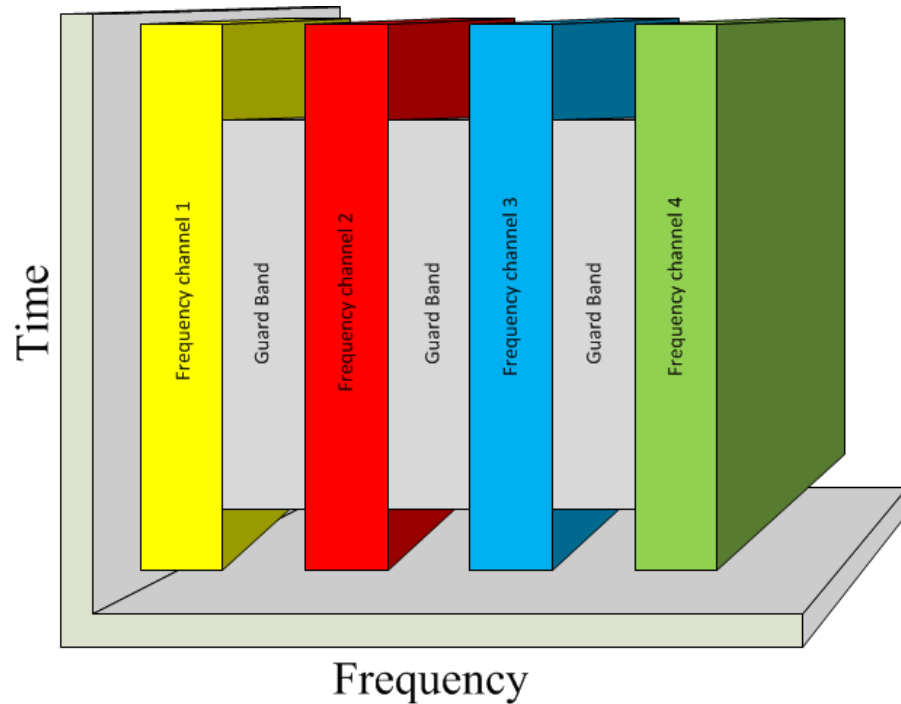
In contrast to wired systems, spectrum is a scarce resource in wireless system. In single-carrier modulation, data bits are modulated and the new produced modulated signals are transmitted sequentially through the spectrum. According to [6], single-carrier modulation not only wastes the frequency band due to its low data rate, but also requires a complex channel equalization. Furthermore, in most wireless systems there is a high demand of having multiple devices communicating in the same area. However, mostly, a certain frequency band is assigned to specific applications, due to scarcity of the spectrum, which can not be extended easily. Therefore multiple access techniques should be provided to use frequency band as efficiently as possible while it permits simultaneous communications of many users. The following subsections cover different multiple access methods [7].

### 2.2.1 Frequency Division Multiple Access (FDMA)

Frequency Division Multiple Access (FDMA) is the first and the simplest multiple access method. As it is depicted in Figure 2.2, a dedicated frequency band exists for each user in which the entire spectrum is exerted and will be released by the user [7]. This multiple access technique has the following advantages and disadvantages:

#### Advantages:

- Low complexity due to simple synchronization algorithm.
- No fading occurs during the transmission, due to using narrower frequency



*Figure 2.2. Principles of Frequency Devision Multiple Access*

band compared to other multiple access methods. Thus, very simple equalization is required.

- Since the transmission is continuous, a simple tracking algorithm is required.

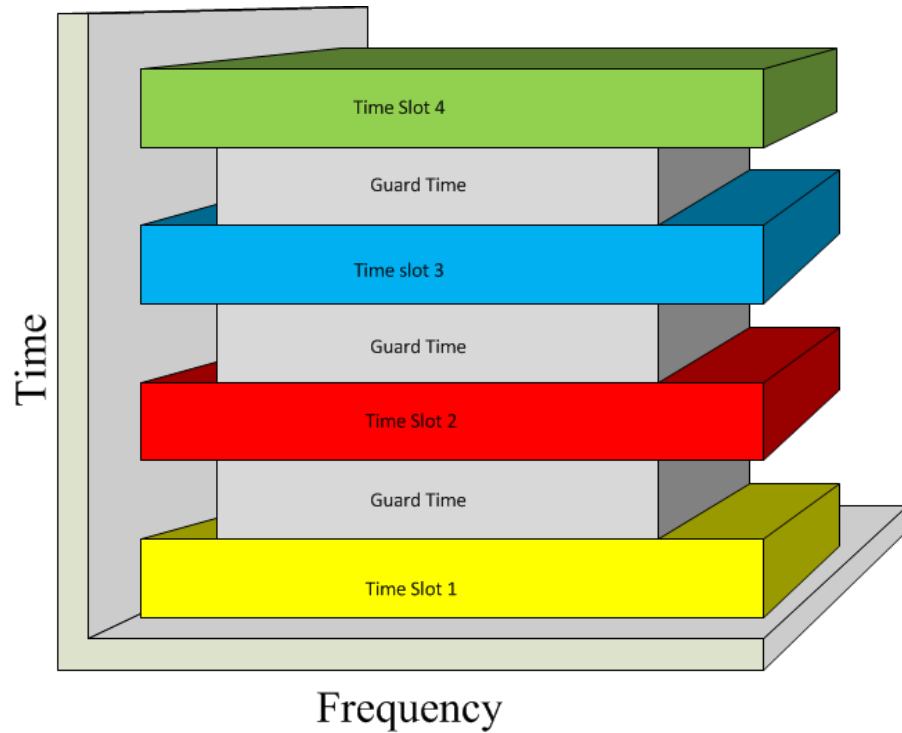
#### **Disadvantages:**

- Unused frequency bands, which is the major concern in spectrum efficiency, stay idle which result in wasting the spectrum.
- Guard bands are needed to cope with interferences caused by the adjacent frequency bands
- Sensitive to multipath effects.
- No frequency diversity due to narrow band.

### **2.2.2 Time Division Multiple Access (TDMA)**

However, Time Division Multiple Access (TDMA) is quite similar to FDMA, instead of allocating a narrow frequency band for a long time to a certain user, the whole frequency band is dedicated to the same user in a certain time known as time slot [7]. In other words, the time unit is divided into  $N$  time slots, each of which is assigned to a different user who is eligible to transmit over the entire frequency band. Figure





*Figure 2.3. Principles of Time Devision Multiple Access*

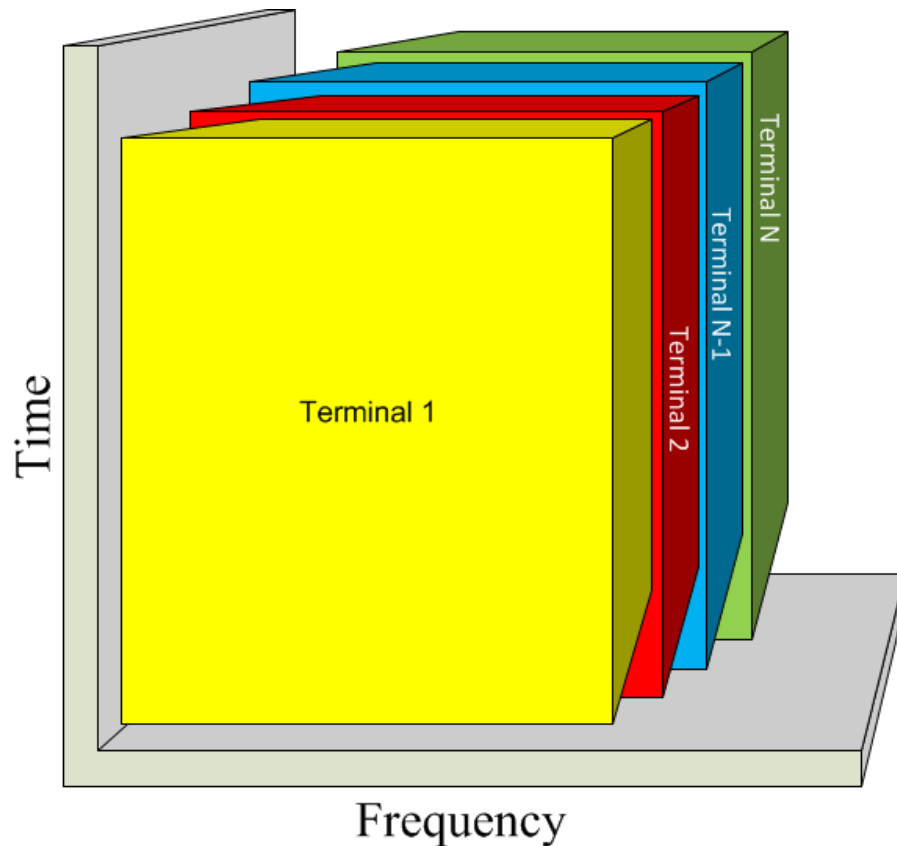
2.3 shows the concept of TDMA modulation. There are several advantages and disadvantages for TDMA similar to that of FDMA.

**Advantages:**

- Occupying a larger amount of frequency bandwidth results in exploitation of the frequency diversity.
- More flexibility compared to FDMA in terms of efficient usage of the bandwidth.
- Achieving a higher data rate by employing several time slots for a single user.

**Disadvantages:**

- Since the transmission is non-continuous, a precise synchronization is needed for each time slot.
- The duration of time slots needs to be optimized. In case of a short time slot, a large percentage of the time is wasted for synchronization. On the other hand, a long time slot produces longer latency.
- Time guards are required similar to that of FDMA.
- Adaptive channel equalization is always needed.



*Figure 2.4. Principles of Code Division Multiple Access*

### 2.2.3 Code Division Multiple Access (CDMA)

In addition to drawbacks of FDMA and TDMA access methods, another constraint induced researchers to develop Code Division Multiple Access (CDMA) access technique. Those methods could only serve finite number of users due to the limited available frequency bands or time slots. In other words, if all the frequency channels or time slots have been completely assigned to the users, a new incoming request should remain on hold until one of the channels or the slots is released.

As Figure 2.4 shows, in CDMA technique all the terminals are transmitting on the same frequency at the same time (while preserving the entire bandwidth), multiplied with a unique *signature*, also known as chip code, for each user. Thus, various users are able to communicate simultaneously at the same time. At the receiver, the desired signal can be demodulated by correlating the received signal with the chip code known by the receiver. At this point, other received signals are considered as interference from the receiver's point of view. This process is called *code acquisition*. There are several advantages and disadvantage by employing CDMA multiple access method, as well.

**Advantages:**

- Since all the terminals are using the same frequency band, no synchronization is needed.
- Huge code space makes the maximum number of users to be, theoretically, infinite.
- Interference caused by other terminals behave like noise.

**Disadvantages:**

- Precise power control is needed to compensate far-away low power users and prevent them from being blocked by near users (Near-Far problem).
- Chip timing is difficult to be acquired and maintained.
- Chip sequences of different users must be orthogonal to each other in order to have successful demodulation.

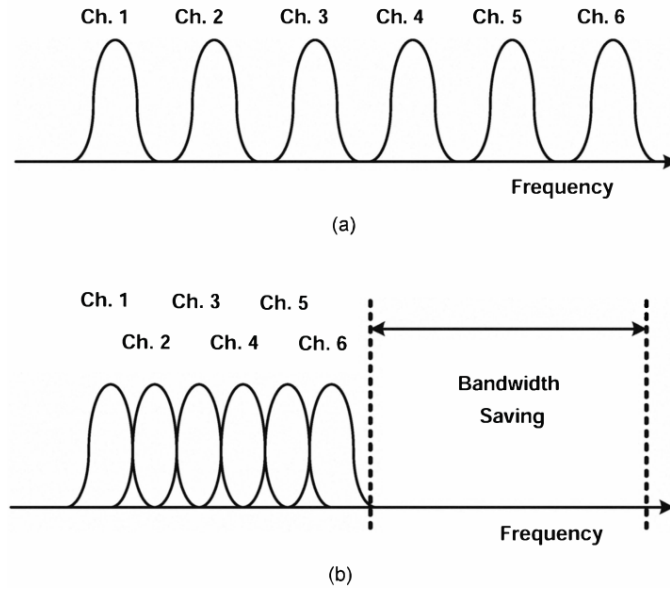
## 2.3 Multi-Carrier Modulations

Multi-carrier modulation is a method of transmitting a high speed data stream by splitting it into several sub streams and sending each of them over a separate carrier signals and, therefore, allowing system to support multiple users at the same time. The individual carriers have narrow bandwidth while the composite signal has a wide bandwidth. In this section, one of the most famous multi-carrier modulations named OFDM is investigated in detail.

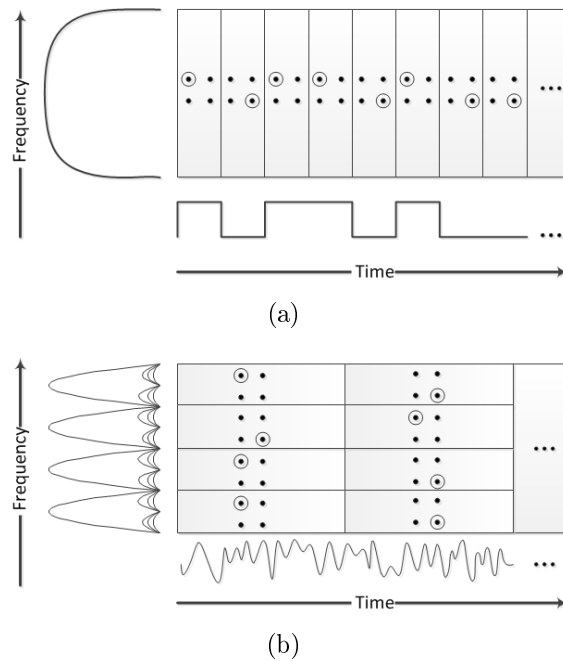
### 2.3.1 Orthogonal Frequency Division Multiplexing (OFDM)

OFDM is a robust technique used in many recent standardized wireless systems in order to achieve higher data rate as well as combating frequency selective fading while the synchronization is preserved at a satisfactory level. Only few subcarriers are distorted over deep fading or narrow band channels which can be compensated using error control mechanisms such as forward error correction. [12]

In principle, a high speed data stream is divided into  $N_u$  parallel substreams modulated onto  $N_u$  orthogonal subcarriers. It can be said that the OFDM is a hybrid of multi-carrier modulation and single-carrier FSK modulation [4]. Figure 2.5 shows how OFDM modulation saves the bandwidth by overlapping the adjacent subchannels while the orthogonality is preserved. In addition, Figure 2.6 illustrates how OFDM modulation is able to serve multiple users in the same frequency band as dedicated to single carrier FSK modulation.

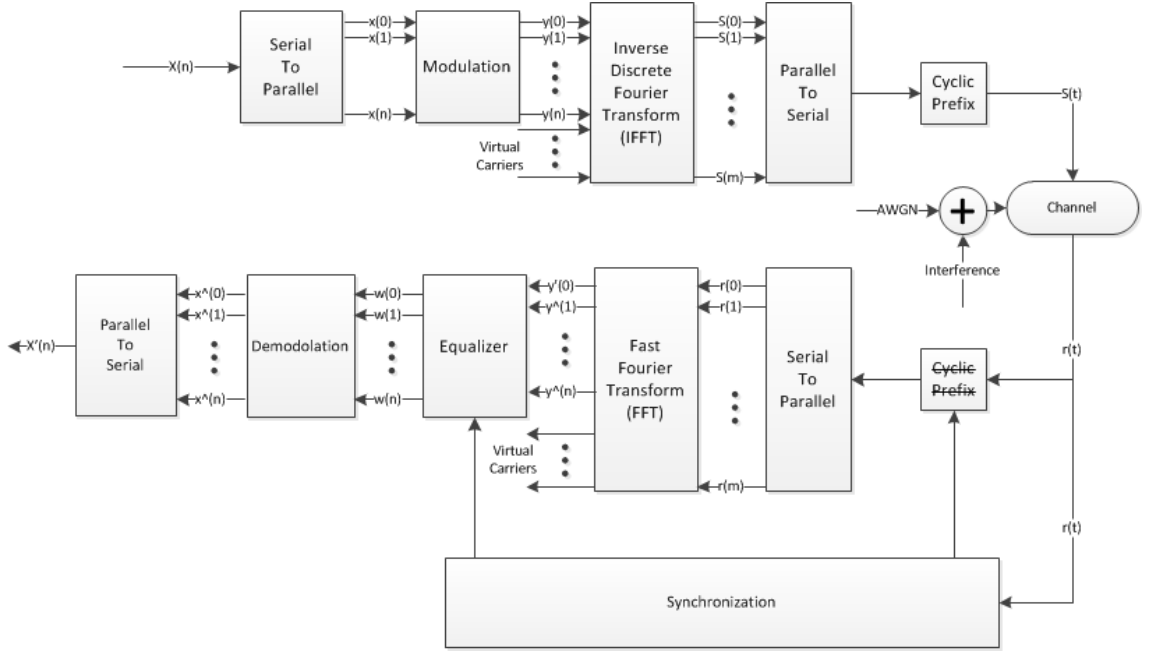


**Figure 2.5.** (a) Conventional non-overlapping multi-carrier modulation. (b) Overlapping multi-carrier modulation. [6, p.26]



**Figure 2.6.** Comparison between (a) single-carrier FSK modulation and (b) multi-carrier OFDM modulation.

An architecture of an OFDM transceiver is shown in Figure 2.7. A high-speed data stream  $X(n)$  is demultiplexed into  $N_u$  parallel ones  $x^{(k)}(n), k = 0, \dots, N_u$  by employing a serial to parallel converter to form a set of data subcarriers. Then each of them is individually modulated using either QAM or PSK modulation and produces  $y^{(k)}(n), k = 0, \dots, N_u$  [1]; typically, as long as the receiver knows the modulation pattern, each subcarrier is modulated with the same constellation [11].



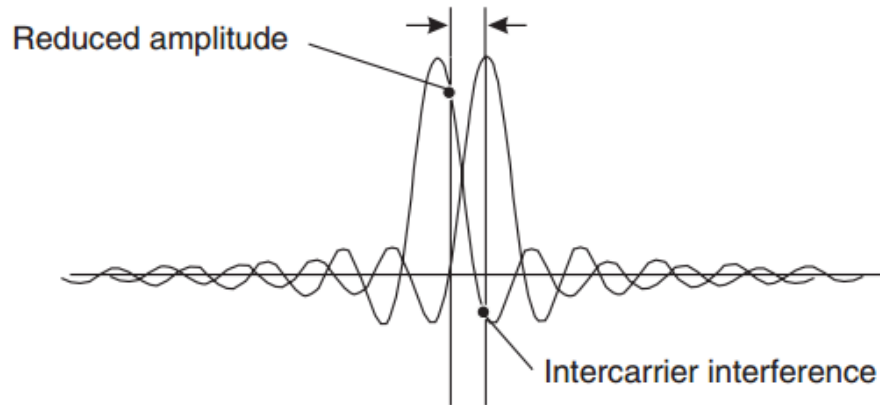
**Figure 2.7.** OFDM transceiver architecture [4, p. 38]

After the modulation is done, baseband OFDM waveform  $s^{(\ell)}(n)$ ,  $\ell = 0, \dots, N$  can be constructed as a  $N$ -input Inverse Discrete Fourier Transform (IDFT) unit with  $N \geq N_u$  defined as Equation (2.1).

The  $N - N_u$  unused inputs of the IDFT are set to zero and they are called Virtual Carrier (VC) which, typically, are dedicated to be used as guard bands in order to avoid interferences caused by transmission power of the adjacent subcarriers. This phenomenon is also known as Inter-Symbol Interference (ISI) and will be addressed later on. In general, IDFT can be implemented using an Inverse Fast Fourier Transform (IFFT) function. Finally, a Cyclic Prefix (CP) is added before converting the subcarriers to the composite signal  $s(n)$ . [4]

$$s^{(\ell)}(n) = \frac{1}{N} \sum_{k=0}^{N-1} y^{(k)}(n) e^{j2\pi k\ell/N} \quad (2.1)$$

At the receiver, first the CP is removed from the received signal  $r(n)$ . Next, a conversion between serial stream to parallel streams is applied by employing a serial to parallel demultiplexer. Then, with respect to Equation (2.2), the information can be extracted by performing Discrete Fourier Transform (DFT) function on the received parallel waveforms where  $r^{(\ell)}(n)$  are parallel input streams. DFT can be performed using a Fast Fourier Transform (FFT) function which produces  $\hat{y}^{(k)}(n)$ . Subcarriers are then equalized in order to compensate distortions caused by the channel. The equalized sub carriers  $\omega(n)$  are then demodulated and, finally, serial steam  $x'(n)$  is obtained using a parallel to serial converter on the parallel streams



**Figure 2.8.** Inter-Carrier Interference due to frequency offset [7, p. 432]

$x^{(k)}(n)$ . [4]

$$\hat{y}^{(\ell)}(n) = \sum_{\ell=0}^{N-1} r^{(\ell)}(n) e^{-j2\pi k\ell/N} \quad (2.2)$$

**Guard Interval (GI):** In wireless systems, the receiver might receive several copies of transmitted signal due to the multipath effects in which the original signal is arrived on time and the rest will be received by a small amount of delay. This phenomenon is called ISI where the tail of the first symbol is collided with the beginning of the second one which leads to destroy the entire symbol.

In order to cope with ISI, a Guard Interval (GI) with the length  $N_g$  is inserted to the first segment of each OFDM symbol. The length of the GI should be more than the length of the delay spread of the channel. For this reason, a degree of delay spread should always be considered while an OFDM symbol is constructed. [7]

During the guard interval, the transmitter sends a null waveform called Zero Padding (ZP). Although, Zero Padding Orthogonal Frequency Division Multiplexing (ZP-OFDM) has a simple and low power structure, it introduces another phenomenon called Inter-Carrier Interference (ICI) in which the orthogonality between subcarriers is destroyed due to receiving several copies of the time shifted ZP-OFDM waveform. Figure 2.8 depicts ICI caused by one subcarrier which affected many adjacent subcarriers. In order to eliminate the effect of ICI, a CP illustrated in Figure 2.9 is combined with the zero padding part which is exactly a duplication of a certain part from the end of the OFDM waveform to its beginning [6]. Figure 2.10 shows the whole above-mentioned scenario.

Similar to other single-carrier and multiple access methods, OFDM has its own advantages and disadvantages listed as following: [13]

**Advantages:**

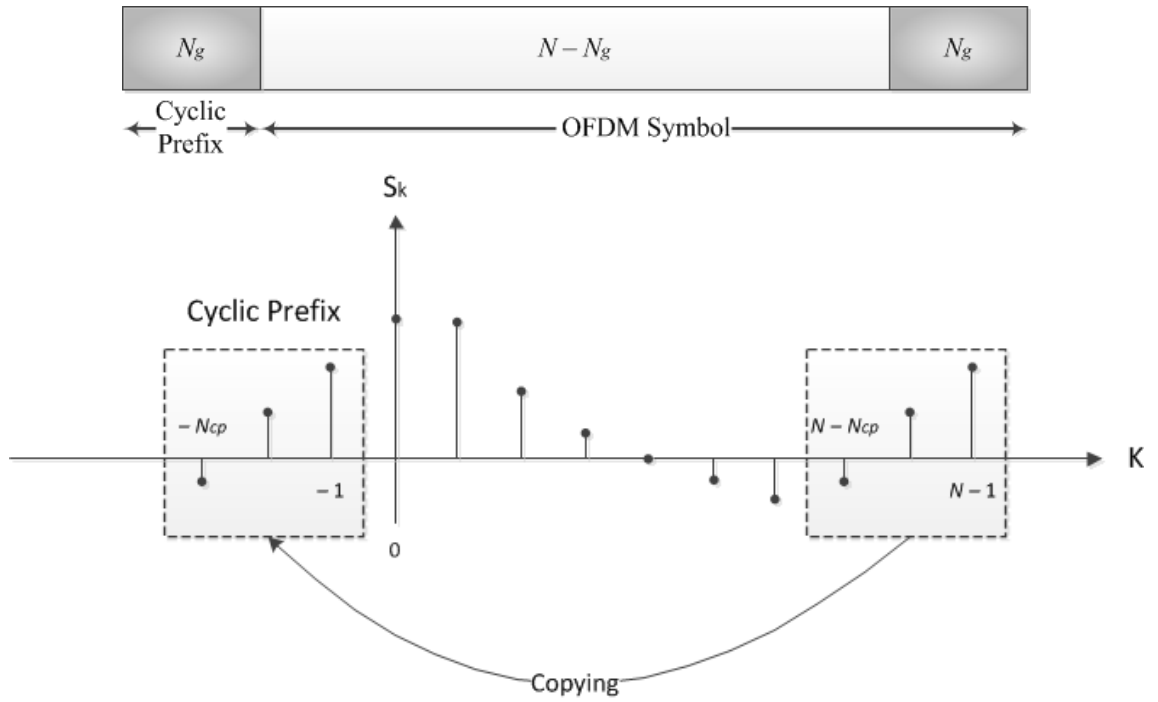


Figure 2.9. Structure of an OFDM block with cyclic prefix

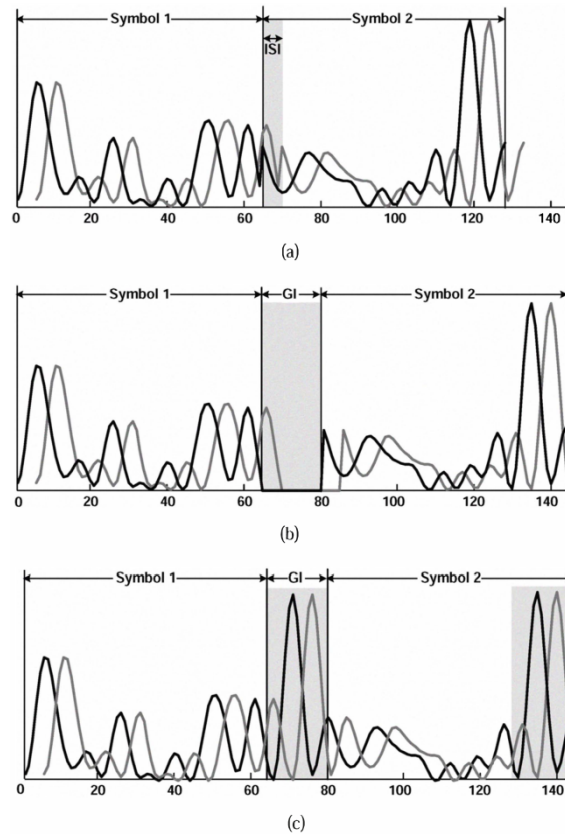


Figure 2.10. (a) Illustration of ISI due to multipath delay; (b) zero-padding guard interval to avoid ISI; (c) guard interval with cyclic prefix to eliminate ISI and ICI [6, p. 28].

- Efficient use of bandwidth.
- Suitable for high data rate transmission.
- More resistance to frequency selective fading.
- Simple channel equalization technique compared to other techniques.
- Less sensitive to sample timing offset.
- Elimination of ISI and ICI problems due to the use of CP.

**Disadvantages:**

- Peak-to-Average Power Ratio (PAPR) problem, having an amplitude with a large dynamic range due to the superposition of  $N$  sinusoidal signals on different subcarriers.
- Sensitive to carrier frequency offset.
- Extra overhead introduced by the CP.

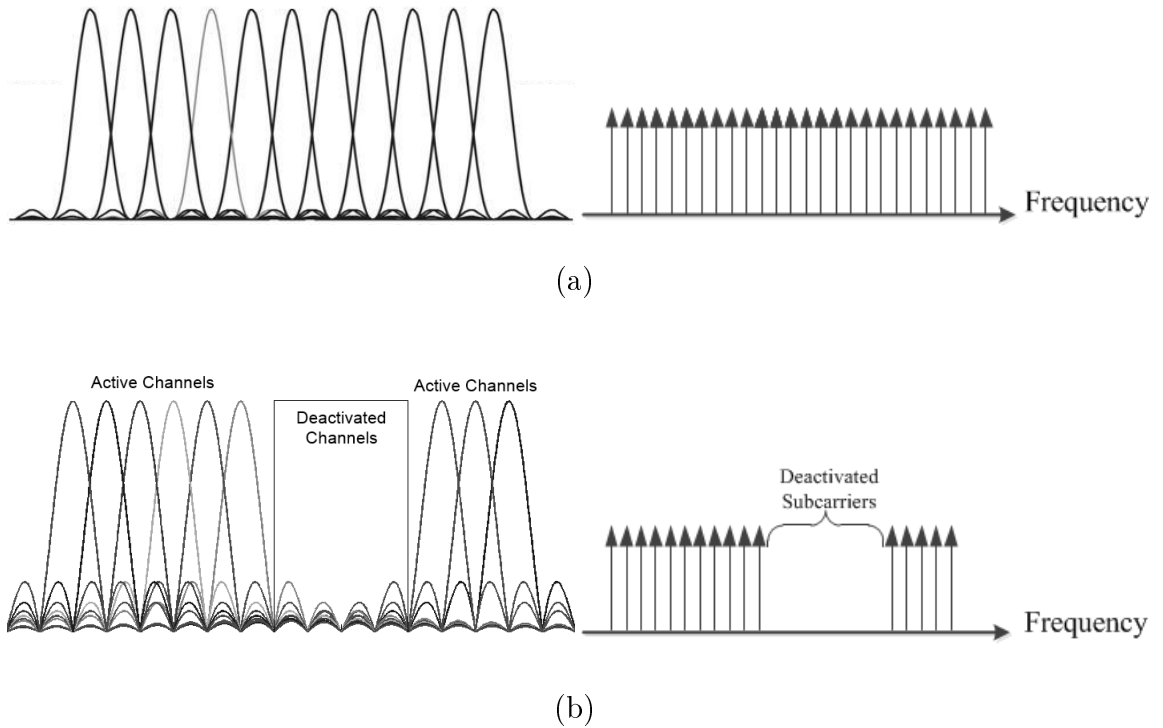
### 2.3.2 Non-Contiguous Orthogonal Frequency Division Multiplexing (NC-OFDM)

All of the techniques which are discussed so far, operate on contiguous spectrum frequencies. For example, a transceiver with 5 MHz bandwidth can operate, only if it detects an idle contiguous 5 MHz bandwidth among the whole spectrum. On the other hand, a narrow-band transceiver which has employed a 5 MHz bandwidth of the frequency band for transmitting 500 kHz, is wasting up to 90% of the scarce wireless bandwidth resource. Therefore, another multi-carrier modulation technique is needed to use these white spaces over the spectrum in such an optimum way not to interfere with adjacent users. [15]

In particular, the new technique must be agile enough to enable unlicensed users operate within unused spectrum dedicated to the licensed users while not interfering with the incumbent users. Moreover, it should be capable of handling high data rates transmission. One technique which meets all these criteria is a variant of OFDM modulation called NC-OFDM. In comparison with other techniques, NC-OFDM is capable of deactivating subcarriers which interfere with transmission of other users [1]. Figure 2.11 shows the difference between OFDM and NC-OFDM techniques.

Fundamental principles of NC-OFDM are quite similar to that of OFDM system. As depicted in 2.12, a high-speed data input  $X(n)$  is demultiplexed into  $N_u$  parallel data streams  $x^{(k)}(n)$ ,  $k = 0, \dots, N_u$  using a serial to parallel converter to form a set



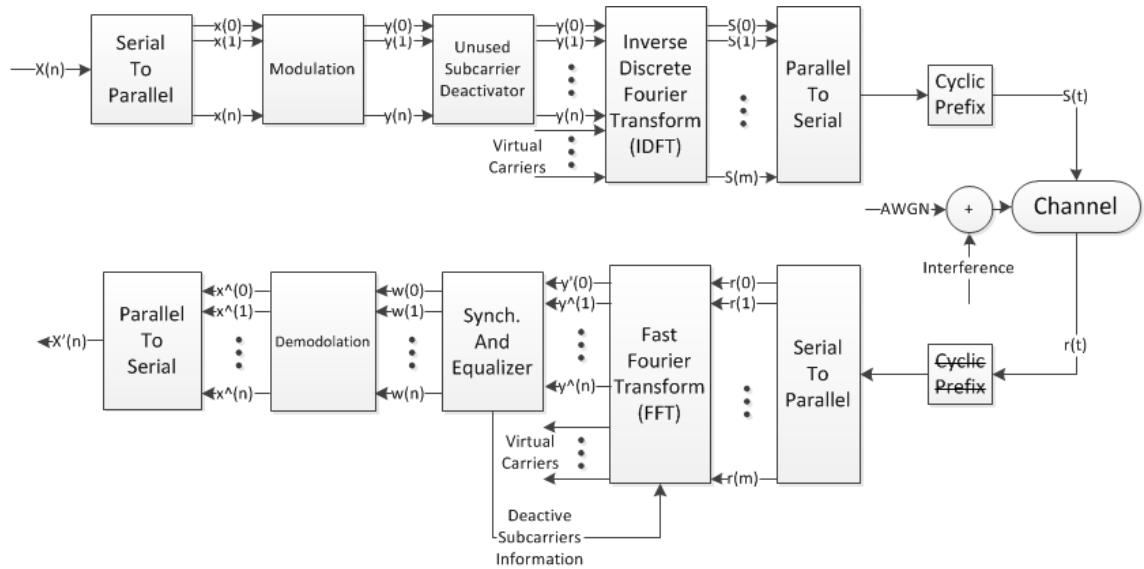


**Figure 2.11.** (a) OFDM and (b) NC-OFDM schemes.

of data subcarriers. Then each one of these  $x^{(k)}(n)$  is individually modulated using either QAM or PSK modulation and produce  $y^{(k)}(n)$ ,  $k = 0, \dots, N_u$ . At this point, a total number of unused subcarriers are deactivated by employing, for example, a controller unit. Next, baseband OFDM waveform  $s^{(\ell)}(n)$ ,  $\ell = 0, \dots, N$  can be constructed as a  $N$ -input IFFT unit with  $N \geq N_u$ . The  $N - N_u$  unused inputs of the IFFT are set to zero and use as virtual carriers dedicated to be used as guard bands. Finally, a CP is added before subcarriers convert to the composite signal  $s(n)$ . [1]

At the receiver, The cyclic prefix is removed from the received signal  $r(n)$  first. Next, a conversion from a serial stream to parallel streams is applied by employing a serial to parallel demultiplexer. Then, the information can be extracted by performing a FFT on the received parallel waveform where  $r^{(\ell)}(n)$  are parallel input streams produce  $\hat{y}^{(k)}(n)$ . Subcarriers are then synchronized and equalized in order to compensate the distortion caused by the channel. The equalized sub carriers  $\omega(n)$  are then demodulated and finally a serial steam  $x'(n)$  is obtained using a parallel to serial converter on parallel streams  $x^{(k)}(n)$ . [1]

Basically, by taking into account both Figure 2.7 and Figure 2.12, a fundamental difference can be seen between OFDM and NC-OFDM receivers which is the synchronization part. In contrast to OFDM where the synchronization is done before performing FFT, in NC-OFDM receiver the synchronization performed while the



*Figure 2.12. NC-OFDM transceiver architecture*

FFT is done. This trivial change in NC-OFDM system emerges challenges for receiver designers [1]. One of these challenges is how to keep the receiver synchronized with transmitter which is studied in detail at synchronization section.

Similar to other modulations, NC-OFDM contains some advantages, in addition to that of OFDM, and some drawbacks listed as following [16]:

#### Advantages:

- It is capable of turning off subcarriers across its transmission which are potentially interfere with adjacent subcarriers.
- NC-OFDM supports a high aggregate data rate using the rest of activated subcarriers.

#### Disadvantages:

- FFT pruning algorithms should be applied to compensate the effect of deactivated subcarriers on computation time.
- Synchronization scheme poses a lot of challenges, since the situation of the carrier frequency might be altered at any time.
- PAPR problem still exists.
- Precise synchronization requires more power consumption.

## 2.4 Introduction to Cognitive Radio (CR)

New evolutions in wireless communication technology have increased the demand of a more flexible, adoptable and intelligent transceivers due to the scarcity of wireless spectrum. Although, data communication networks are one the major challenges of developed countries with respect to the finite resource of the radio spectrum, wireless transceivers are more versatile and portable than the traditional wired systems. Before stepping forward to the concept of CR, It is crucial to address principle of SDR first.

### 2.4.1 What is Software Defined Radio (SDR)

With rapid evolution of microelectronics, wireless transceivers surrogated the traditional wired system due to be more versatile as well as being portable. Since 1991, the development of SDR has been enabled in which the transceiver carries out the entire baseband processing in software. SDR defines a radio platform in which, at least, a portion of the entire implementation is held in software. In a technical view, any waveform can be applied for any frequency band which permits the transceiver to be operated as a multi-function, multi-band and multi-mode wireless device. [18]

Over the years, the radio community has realized that most of the radio functions can be handled in the software. In general, wireless standards, such as IEEE-802.11a/g/n, which are based in software can be easily swapped in and out in an SDR platform. In traditional systems, a complete replacement of the radio frequency hardware is required in order to switch from a particular standard to another one which undergoes an expensive upgrade [1]. In an SDR platform, most of the signal processing is done in programmable processing technologies including General Purpose Processor (GPP), programmable System-on-Chip (SoC), Digital Signal Processor (DSP) and FPGA. Generally, architectural complexity of a SDR platform is more limited to run-time requirements and high computational workload of the algorithms. However, the scaling of the silicon technology permits to employ more number of the transistors for implementing computationally intensive architecture [19].

According to [20], two major advantages of SDR are, first, flexibility where the transceiver simply switches between channels and the second one is adaptability where the radio parameters, including channel modulation, frequency, power and bandwidth, can be simply changed due to the radio environment. Similar to other type of new evolved technologies, SDR platforms emerged from military researches and were then employed for civil usages. In general, SDR is the core enabler for a new technology named CR.

### 2.4.2 What is Cognitive Radio (CR)

The basic idea of CR, which can be implied as an intelligent version of SDR, was proposed by Joseph Mitola in 1998 and published in by Mitola and Gerald Q. Maguire in 1999 [2]. CR is basically an SDR platform which can rapidly change its operating parameters by considering new circumstances and criteria. In contrast with SDR, in CR these parameters are changed in such a way that its user is not even noticed. Technically, CR is smart enough to decide how, where and when it uses the spectrum without any prior knowledge. As Simon says in [21], "Cognitive radio is an intelligent wireless communication system that is aware of its surrounding environment (i.e., outside world) and uses the methodology of understanding-by-building to learn from the environment and adapt its internal states to statistical variations in the incoming RF stimuli by making corresponding changes in certain operating parameters (e.g., transmit-power, carrier-frequency, and modulation strategy) in real-time, with two primary objectives in mind:

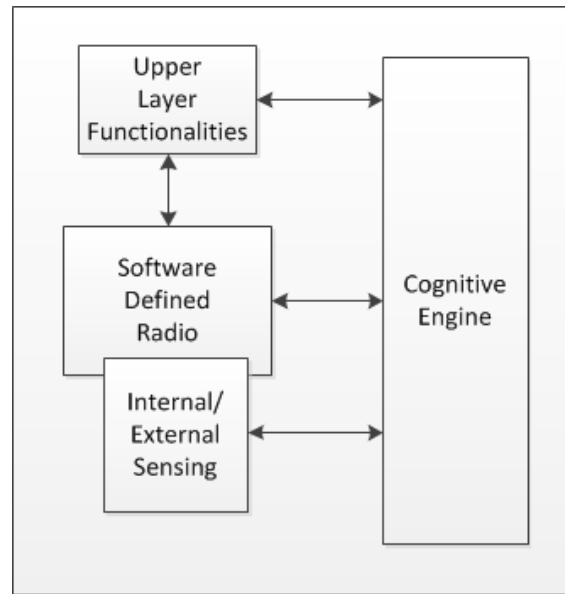
- highly reliable communications whenever and wherever needed;
- efficient utilization of the radio spectrum.

Six key words stand out in this definition: awareness, intelligence, learning, adaptivity, reliability, and efficiency." Figure 2.13 shows how CR is in relation to SDR. Based on that, cognitive engine is responsible to optimize and control SDR by simultaneously learning the environment using a sensing unit. Moreover, cognitive engine must be aware of hardware resources as well as other input parameters. Therefore, SDR becomes a flexible and common radio platform capable of supporting multiple standards, for example Global System for Mobile (GSM), Enhance Data GSM Environment (EDGE), Worldwide Interoperability for Microwave Access (WiMAX), Wireless Fidelity (WiFi) and Wide Code Division Multiple Access (WCDMA), as well as operating over a wide range of frequencies with different type of modulation techniques such as Space Division Multiple Access (SDMA), TDMA and OFDM [22].

### 2.4.3 Evolution of Radio Technology

Figure 2.14 shows the evolution of radio technology. An *Aware Radio* has sensors which enables the device to be aware of the environment. An *Adaptive Radio* is not only aware of its environment, but also is capable of changing its behavior in response. The final stage is CR. According to Polson's opinion in [23], CR is carrying the following characteristics:

- Sensors creating awareness in the environment.



*Figure 2.13. CR and its relation to SDR*

- Actuators enabling interaction with the environment.
- Memory and a model of the environment.
- Learns and models specific beneficial adaptations.
- Has specific performance goals.

#### 2.4.4 Dynamic Spectrum Access (DSA)

Due to propagation characteristics of the electromagnetic waves, a wide range of frequencies between 10 MHz to 6 GHz are suitable for wireless communication purposes. Although, this frequency range seems to be sufficient, a massive number of users are transmitting over the entire spectrum with, almost, the same transmission scheme. Therefore, since 1994 an American national organization called Federal Communication Commission (FCC) has conducted 33 spectrum auctions worth over 40 Billion dollars to some particular owners. A few of these spectrum owners examples are AM, FM, TV broadcast operators, telecommunication network operators, etc (also are known as licensed users or primary users).

An investigation in United States of America estimates that only 15% of the bandwidth is used in most of the cases. Figure 2.15 depicts how the primary user wastes the entire licensed bandwidth by not using the whole spectrum efficiently. Consequently, researchers focused more on a secondary usage of the bandwidth for unlicensed users over the licensed spectrum as their main objective, since almost none of licensed users are using the whole dedicated spectrum and only use particular

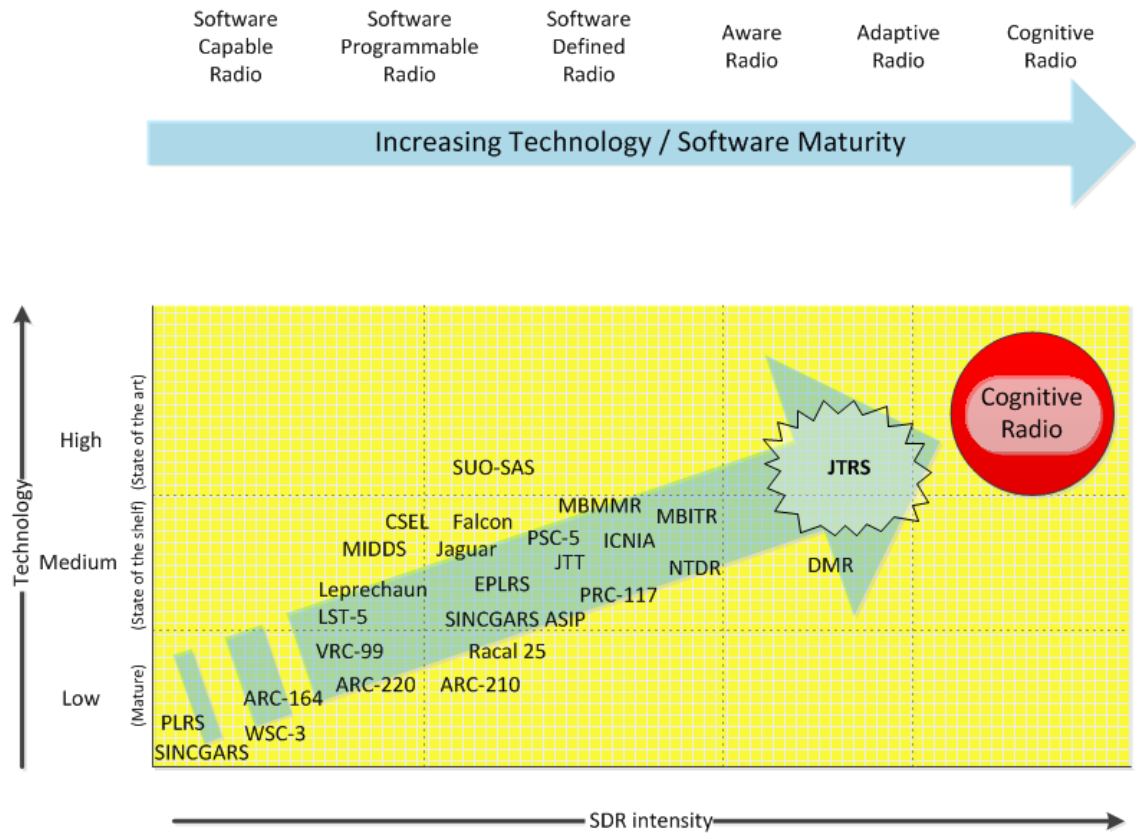


Figure 2.14. Radio technology evolutions [23]

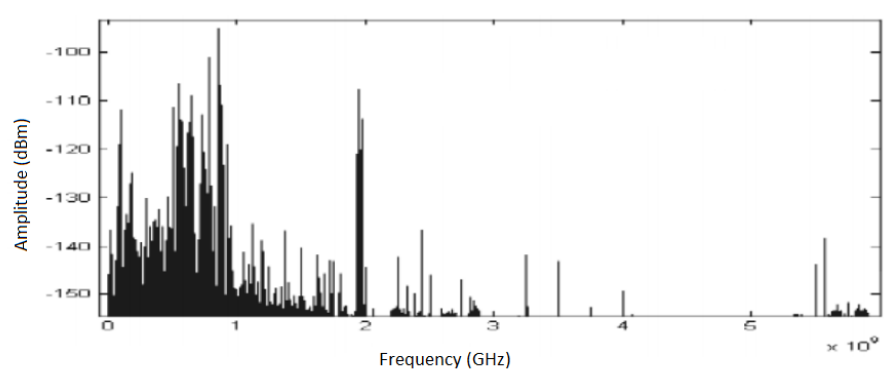


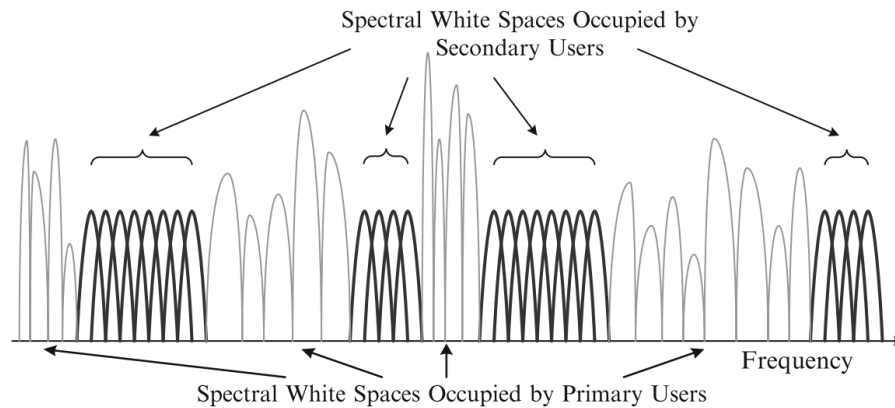
Figure 2.15. Spectrum utilization snapshot at Berkeley [22, p. 163]

part of the bandwidth. These efforts led to employ *white-spaces* as a secondary solution to be used for unlicensed users of what it is currently considered as DSA. The key motivation for DSA comes from the fact that the spectrum assigned to the licensed transmission band is not exploited to its full extent at all times. [7]

With recent developments in CR technology both licensed and unlicensed users can simultaneously communicate over the licensed spectrum as long as the unlicensed user respect to the right of incumbent licensed holder. In principle, full CR, also known as Mitola radio, is capable of adopting *ALL* transmission parameters, in-

cluding modulation format, accessing method, coding, center frequency, bandwidth, transmission times, etc, which is more likely to be as a science fiction view due to implementation complexities. Therefore, DSA is a spectrum sensing cognitive radio which only adapts the transmission frequency, bandwidth and time according to the environment circumstances [7]. Figure 2.16 illustrates how DSA enables the secondary usage of the licensed spectrum within white spaces without interrupting the primary user.

Traditionally, spectrum sharing between primary and secondary users was done manually. Secondary user monitored the primary user spectra and then intended to transmit over the whitespace or spectral holes. DSA extends this process by automating the processes of monitoring, selecting and using. Moreover, the frequency bands assigned to the secondary user must have the least probability of interfering with incumbent user. Therefore, a robust, accurate and reliable modulation technique performs a significant role at this stage. One of the most robust modulation candidates, which meets above-mentioned characteristics, is NC-OFDM due to its capability of turning off a portion of subcarriers interfering with the primary user and operates over a subset of non-contiguous subcarriers.



**Figure 2.16.** Spectrum utilization by employing DSA technology [1, p. 151]

### 3. SYNCHRONIZATION

In each digital communication system, synchronization is an essential mechanism in order to fetch useful data from the received signal. So far, designing a robust and accurate synchronization algorithm has been one of the major challenge for design engineers. Synchronization is the process in which the receiver firstly detects any incoming data from the received signal and secondly distinguishes both the beginning and the end of the received packet.

Although there are several methods to establish a reliable synchronization for different modulation schemes, since the goal of this thesis is to present a flexible timing synchronization scheme for cognitive radio applications, synchronization issues regarding to OFDM as well as NC-OFDM are studied.

NC-OFDM is an extension of OFDM technique in which unused subcarriers can be deactivated in order to eliminate any interferences with the primary user. Therefore, for understanding synchronization techniques related to NC-OFDM system, a deep understanding of what happens in OFDM synchronization is strongly required before stepping forward to NC-OFDM systems.

Most of the synchronization algorithms designed for single-carrier and other multi-carrier techniques are unusable for OFDM and, consequently, NC-OFDM systems due to the nature of its frequency domain. One of the most important constraints which is different in OFDM technique is the fact that the synchronization can be established either in time- or frequency-domain. This level of flexibility is not available in other modulation methods. Hence, a tradeoff between lower computational complexity and higher performance exist between different synchronization algorithms.

#### 3.1 Effects of Poor Synchronization

In digital transmission, the data bit streams are represented in discrete-time signal format while all the physical media are able to communicate over the continuous-time format due to their nature. Moreover, most of the transmission media are inefficient to transmit baseband signals. Therefore, the digital baseband signal should be converted into continuous-time waveform and then modulated to a higher frequency signal before propagating through the channel. Wireless receivers are equipped with a Local Oscillator (LO) whose carrier frequency and phase are the same as that of



the signal in received waveform. Thus, the original signal can be driven out from the received signal by executing an accurate sampling of the clock frequency and phase on incoming signal. This process seems to be simple enough, synchronization issues emerge from this point which the researchers are still involved with to some extent. Unfortunately, the receiver is unsynchronized with the transmitter and it should be synchronized for each transaction.

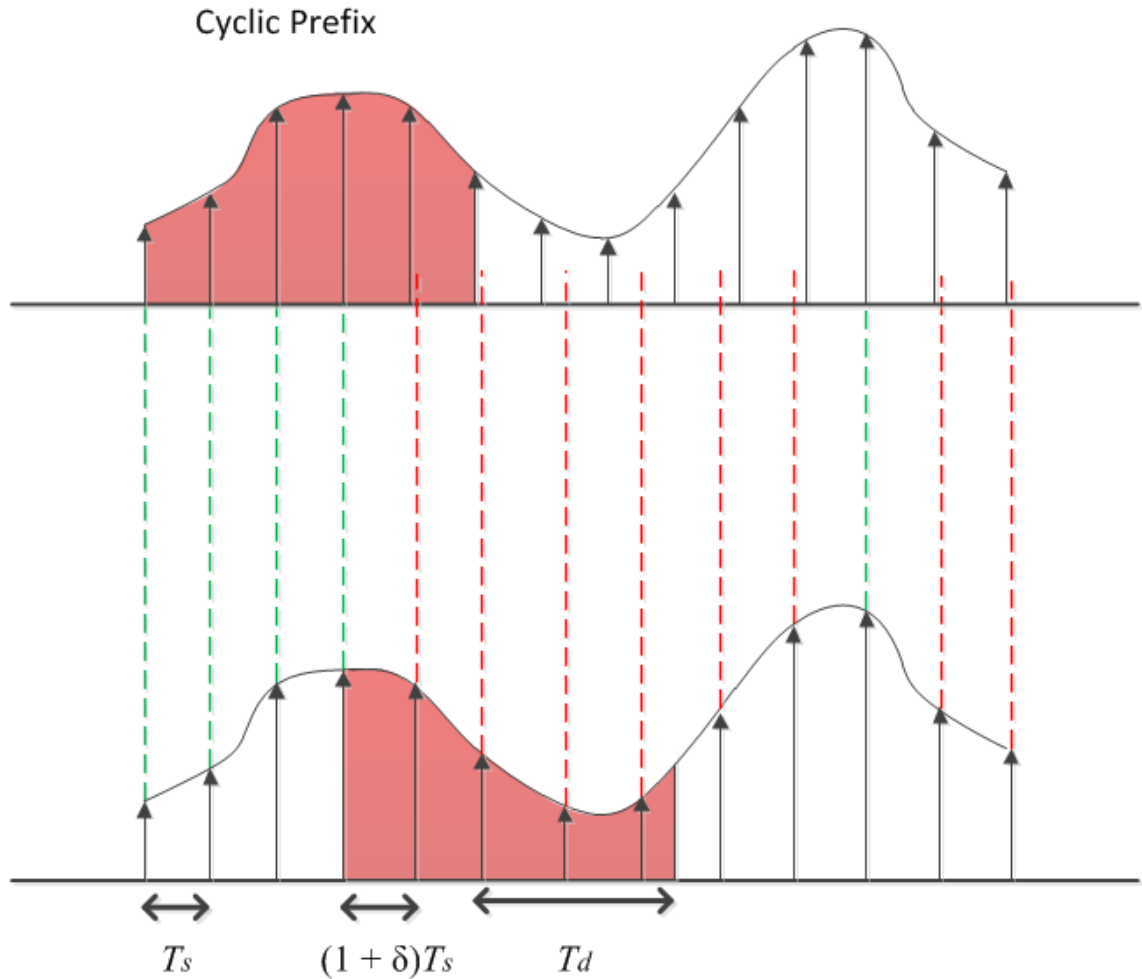
The following issues are most important impairments threatening a proper synchronization [6].

- Carrier frequency/phase errors caused LO clock.
- LO not only can not maintain the frequency and the phase, but also suffers from time variant phase noise.
- Additional phase rotation introduced in LO due to unknown propagation delay between sender and receiver.
- Frequency shift caused by Doppler effect.
- Cross-coupling of I/Q signals, also known as IQ imbalance, due to the front-end electronic mismatches.

### 3.2 Synchronization Errors

Synchronization errors can be yielded in either time, frequency or both. The basic concept in synchronization is that the receiver must know when to run the sampling process on the incoming wave stream. The sampling must be done exactly at the same time as it is supposed to be. Any alteration in sampling time causes the receiver to lose the data, miss the packet and terminate the transmission, meaning to waste the bandwidth. Therefore, perfect synchronization is one of the main concerns from the receiver's perspective. In the following, some major sources of synchronization impairments are explained briefly: [6]

- *Carrier Frequency Offset (CFO)*: Causes the received signal to be rotated with a magnitude of  $\Delta_f$ .
- *Carrier phase Error (CPE)*: Introduces additional phase rotation of magnitude  $\phi(t)$  to the received signal.
- *Sampling Clock Offset (SCO)*: Is caused by performing a sampling of a period  $(1 + \delta)T_s$  instead of  $T_s$  on the received continuous-time signal.
- *Symbol Timing Offset (STO)*: Occurs when the receiver loses the actual boundary of the received waveform.



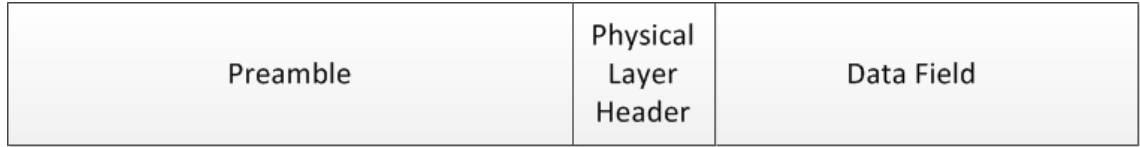
**Figure 3.1.** Effect of bad synchronization (the effect of external impairments, such as noise, have not been considered)

- *IQ imbalance:* Generates gain/phase mismatch in up/down conversions of I/Q paths.

The effect of a bad synchronization due to the SCO and STO errors are illustrated in Figure 3.1. As it can be seen, sampling at  $(1 + \delta)T_s$  interval caused the rest of the sampling process to be inaccurately done. Moreover, symbol boundary does not meet the criteria at all by the receiver either.

### 3.3 OFDM Synchronization Issues

Synchronization errors can be yielded in time, frequency or both. Although, single carrier modulations are more sensitive to time-domain errors (where OFDM has more resilience), OFDM suffers from frequency errors raised by performing FFT due to its frequency domain nature [6]. Equation (3.1) and (3.2) are used to approximately estimate the degradation caused by CFO and CPE for single carrier



**Figure 3.2.** 802.11a packet structure

and OFDM modulations, respectively where  $\Delta_f$  is the magnitude of CFO and  $\beta$  is a function of oscillator linewidth [25].

$$D \approx \begin{cases} \frac{10}{\ln 10} \frac{1}{3} (\pi \Delta_f T)^2 & \text{Single Carrier} \\ \frac{10}{\ln 10} \frac{1}{3} (\pi \Delta_f T)^2 \times SNR & \text{OFDM} \end{cases} \quad (3.1)$$

$$D \approx \begin{cases} \frac{10}{\ln 10} \frac{1}{60} (4\pi\beta T) \times SNR & \text{Single Carrier} \\ \frac{10}{\ln 10} \frac{11}{60} (4\pi\beta T) \times SNR & \text{OFDM} \end{cases} \quad (3.2)$$

Type of transmission is one of important issues in synchronization. Usually data is transmitted in either *packet based* or *frame based* format. In packet based systems, such as Institute of Electrical and Electronic Engineers (IEEE) 802.11a/g/n Wireless Local Area Network (WLAN), user data is divided into several so called packets with a limited size.

As shown in Figure 3.2, each packet starts with a known sequence named *preambles* which facilitate the synchronization process. Following by, a *header* which is consists of important information such as modulation order, code rate, etc. Finally, the user data is composed to form a complete packet. With this compressed structure the receiver does not have sufficient time to detect the signal. Therefore, the estimation and compensation processes for each error which might have occurred at the received signal must be done immediately. Since the FFT is completed after several cycles due to the heavy workload of computations in frequency domain, it is more likely to exploit time-domain synchronization in most of the receivers. Furthermore, periodic repetition of the preamble at the beginning of each packet, which has a good autocorrelation property, assists synchronization of OFDM packet-based system to be performed in time-domain.

In frame based OFDM systems, such as Digital Video Broadcasting-Terrestrial (DVB-T), data are transmitted continuously. Therefore, receiver has more time to

perform synchronization. Thus, receiver designers are more flexible to perform the synchronization in either time-domain or frequency-domain.

### 3.3.1 Synchronization Methods

As soon as the receiver is turned on, it should start searching for OFDM symbols in the received signal. According to the Equation (3.3), where  $\omega(n)$  is the amount of Additive White Gaussian Noise (AWGN), when no incoming signal is transmitted by the transmitter, the magnitude of the received signal is equal to noise level.

$$y(n) = \omega(n) \quad (3.3)$$

As soon as the sender starts transmission, as it can be seen from Equation (3.4), the magnitude of received signal is equal to the amount of AWGN plus the signal  $s(n)$  which should be detected by the receiver. [22]

$$y(n) = s(n) + \omega(n) \quad (3.4)$$

According to above-mentioned descriptions, there are several algorithms designed to perform synchronization in OFDM modulation as following:

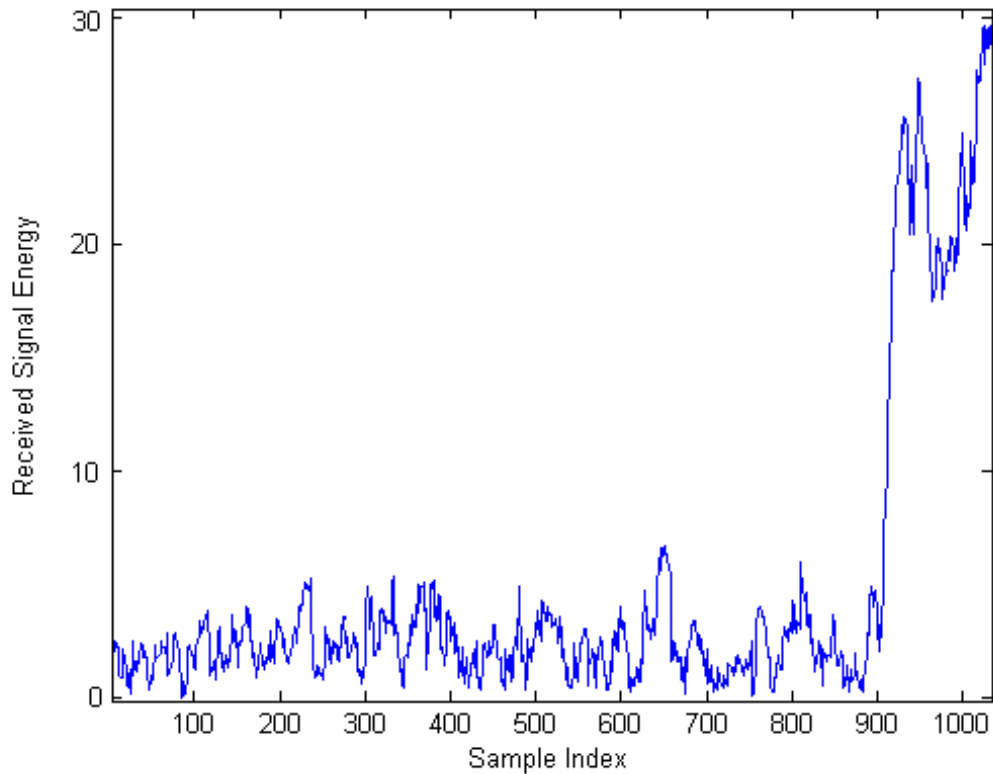
- Received Signal Energy Detection.
- Double Sliding Window Packet Detection.
- Preamble Structured Packet Detection.

#### Received Signal Energy Detection

This is the simplest packet detection algorithm to find the starting edge of the received signal by simultaneously measuring incoming signal energy. According to Equation (3.5), the received signal energy  $m_n$  is the summation of received signal energy over a window of length  $L$ . In other words, the calculation of  $m_n$  is a moving sum of received signal energy also known as *sliding window*. Figure 3.3 depicts how synchronization is performed based on received signal energy method. Whenever the received signal energy exceeds a particular threshold point, the receiver assumes an incoming transmission is ongoing. [8]

$$m_n = \sum_{k=0}^{L-1} r_{n-k} \times r_{n-k}^* = \sum_{k=0}^{L-1} |r_{n-k}|^2 \quad (3.5)$$

Although, the hardware implementation of this synchronization method is so simple due to running only one multiplication per sample, it requires a large memory to

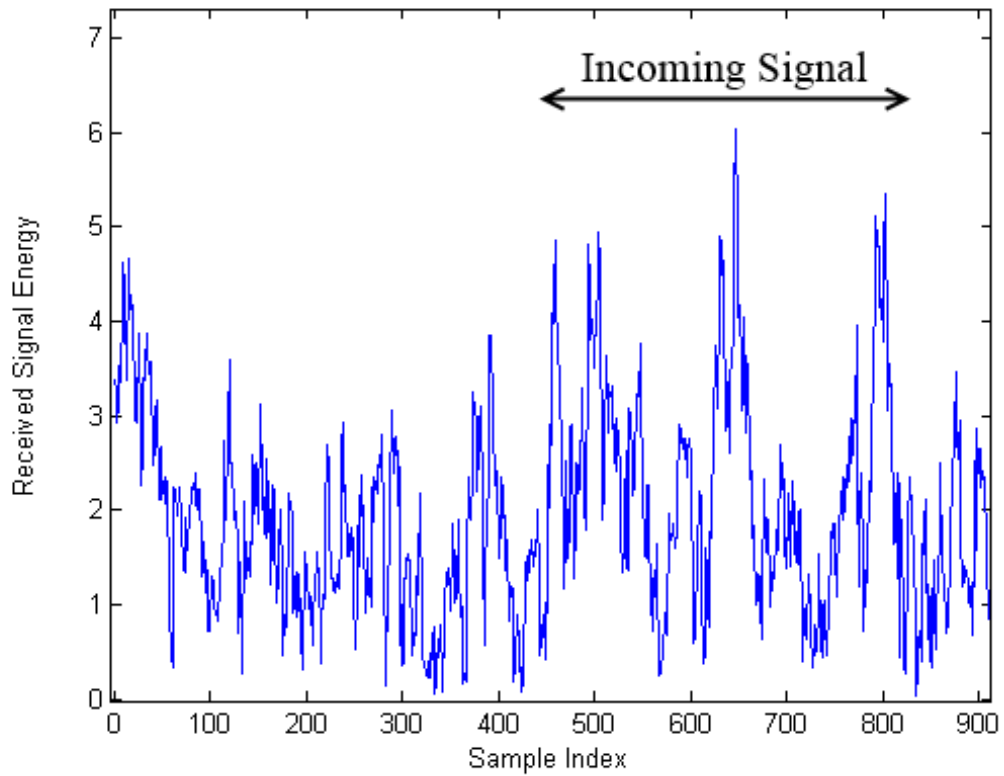


*Figure 3.3. Synchronization based on received signal energy*

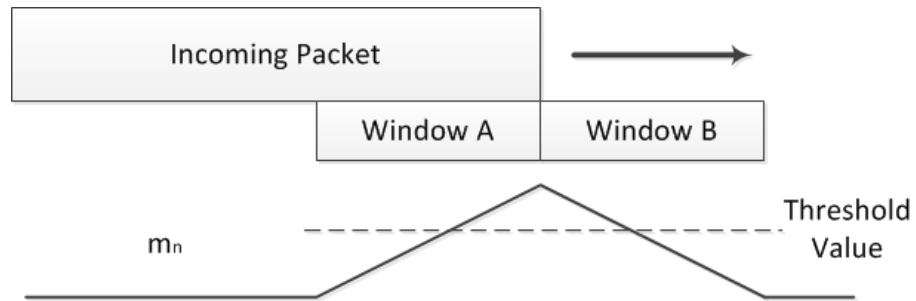
store all values of received signal. Another shortcoming is that the value of the threshold depends on the received signal energy. Therefore, it is hard for designers to set a fixed threshold value. If the value is set to high, the received signal might not be detected due to the low transmitter power; if the value is set lower than what it should be, the packet can not be detected in noisy channels due to the low Signal-to-Noise Ratio (SNR). In other words, in both scenarios the entire signal is either assumed as noise and will not be detected or lost within the noise. Figure 3.4 shows how signal is lost within the noise due to the low SNR.

### **Double Slide Window Packet Detection**

The basic concept of the double sliding window packet detection method is quite similar to received signal energy method. Figure 3.5 shows steps in which two sliding windows calculate the presence of any incoming packet. Based on Equations (3.6) and (3.7), In this method instead of estimating received signal energy just in one time, two sliding windows are employed to measure total energy of the incoming signal simultaneously. According to Equation (3.8), the threshold value is determined based on the maximum energy contained in both windows. [8]



*Figure 3.4. Incoming signal lost within the noise due to the low SNR*



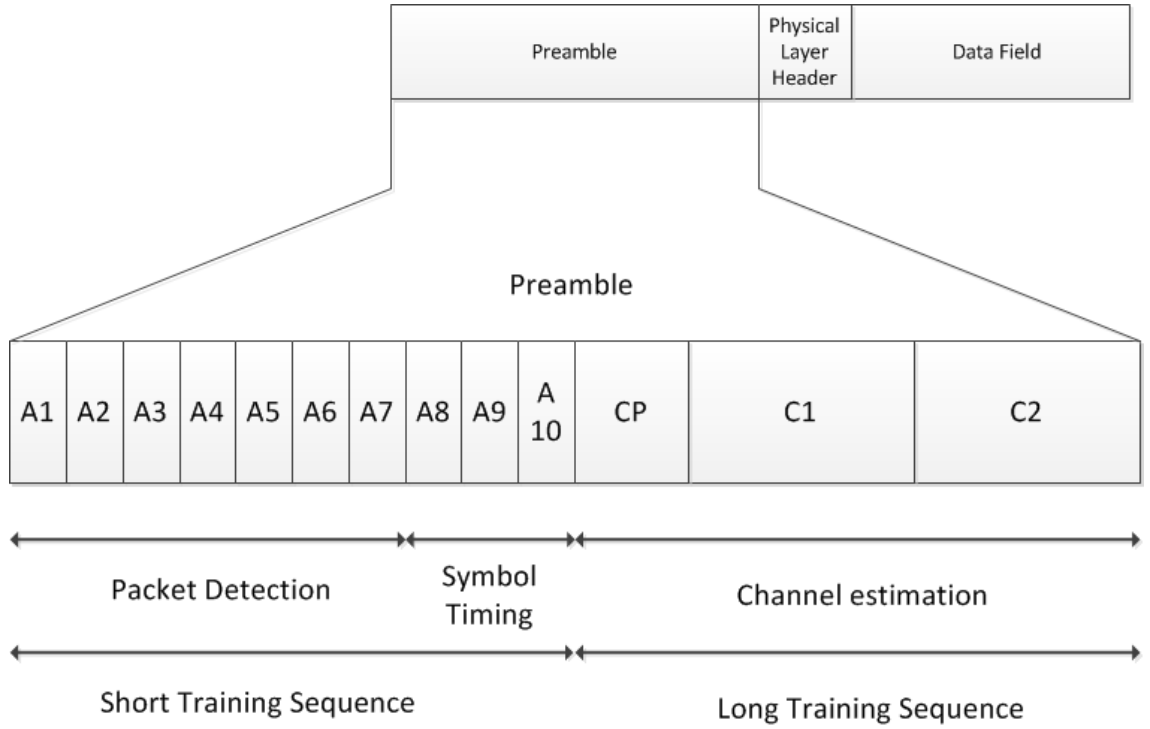
*Figure 3.5. Principle of double slide window packet detection*

$$a_n = \sum_{m=0}^{M-1} r_{n-m} \times r_{n-m}^* = \sum_{m=0}^{M-1} |r_{n-m}|^2 \quad (3.6)$$

$$b_n = \sum_{l=1}^L r_{n+l} \times r_{n+l}^* = \sum_{l=0}^L |r_{n+l}|^2 \quad (3.7)$$

$$m_n = \frac{a_n}{b_n} \quad (3.8)$$

As Figure 3.5 shows, when the value of  $m_n$  reaches to its maximum, it means that window A contains both signal and noise whereas window B contains only noise. Equation (3.9) shows the amount of SNR estimated at the peak level where  $S$  is the



*Figure 3.6. 802.11a preamble structure [8, p. 51]*

energy of signal and  $N$  is the magnitude of noise. This approach is suitable where the receiver has no prior information about when data is transmitted by the sender.

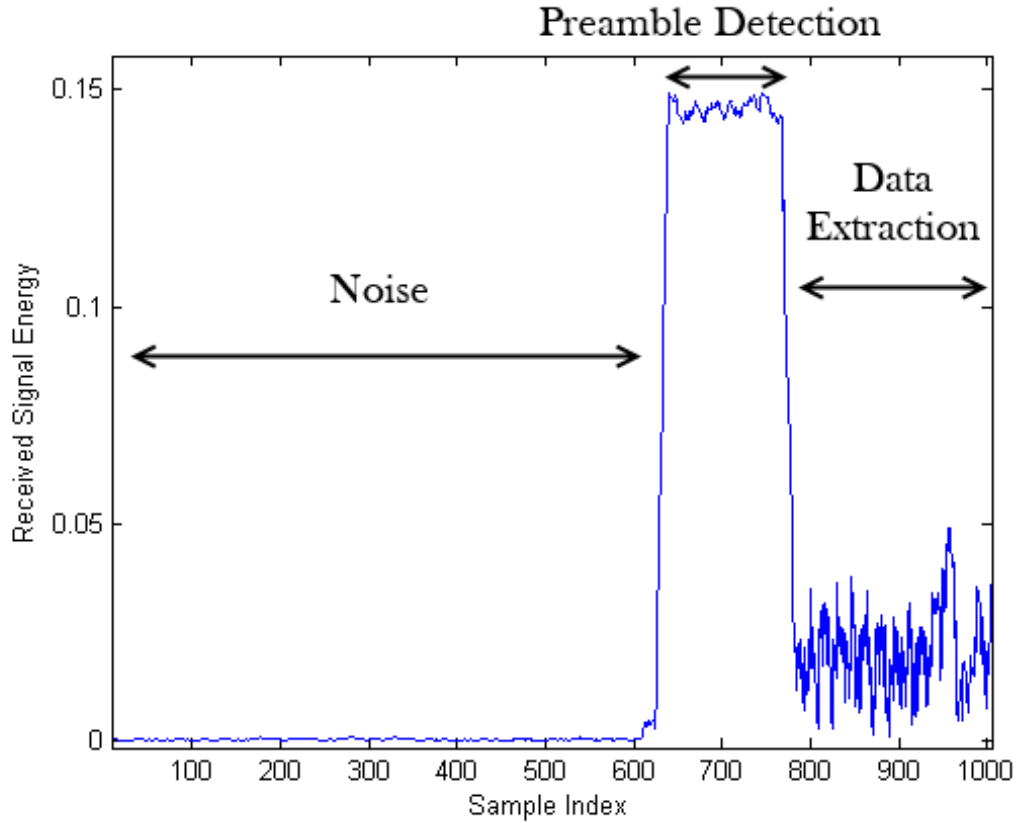
$$m_{peak} = \frac{a_{peak}}{b_{min}} = \frac{S + N}{N} = \frac{S}{N} + 1 \quad (3.9)$$

### Preamble Structured Packet Detection

**What is Preamble:** To aim synchronization process, a train of predefined bits called preamble are added to the beginning of each packet which is known by both sender and receiver. Since preamble is considered as an additional overhead to the header, its contents as well as its length must be carefully designed in order to provide significant information required for the synchronization process.

As it is shown in Figure 3.6, a preamble is composed of two major parts separated by a CP in between. First part,  $A_1$  to  $A_{10}$ , is called Short Training Symbols (STS), each of them composed of 16-bit sample. Following by, a 32-bit sample CP is inserted to compensate ISI introduced by STS. Conceptually, this CP is considered as a guard band preserving Long Training Symbols (LTS) from be distorted by channel impairments. Second part,  $C_1$  and  $C_2$ , consists of two identical 64-bit long samples known as LTS. Moreover, having this structure for the preamble, receiver is able to simply detect any incoming signal even in low SNR environments.

Overall, this method is an extension of double slide window packet technique by



*Figure 3.7. Synchronization Based on preamble structured packet*

taking advantage of 10 periodical STS at the beginning of each packet. Figure 3.7 illustrates how much preamble structured packet is accurate more than previous methods [8]. Equations (3.10), (3.11) and (3.12) were proposed by Schmidl and Cox in [24] and show how preamble assists synchronization, where  $c_n$  is the autocorrelation,  $p_n$  is the energy and  $m_n$  is the threshold value of the received data stream, respectively.

$$c_n = \sum_{k=0}^{L-1} r_{n+k} \times r_{n+k+D}^* \quad (3.10)$$

$$p_n = \sum_{k=0}^{L-1} r_{n+k+d} \times r_{n+k+D}^* = \sum_{k=0}^{L-1} |r_{n+k+D}|^2 \quad (3.11)$$

$$m_n = \frac{|c_n|^2}{(p_n)^2} \quad (3.12)$$



### 3.3.2 Overview of 802.11a Packet Structure

IEEE 802.11a is the basic standard mostly used in packet systems. Therefore, an overview of the packet structure might be generalized for other standards (such as 802.16, 802.22, etc). This standard is categorized as packet-based systems. In those systems, as it has been mentioned earlier, sender splits its data into several packets and then transmits them as quick as possible. Moreover, each packet composed of preamble, data, service field, padding, etc, should be transmitted in a certain time.

This standard specifies a 2.4 GHz OFDM based operating frequency which splits signal over 64 separate subcarriers where 12 subcarriers are usually used as guard band and the 52 remaining subcarriers are employed to transmit preamble, data, etc. This operating frequency enables data transmission at a rate of 6, 9, 12, 18, 24, 36, 48, or 54 Mbps where 6, 12, and 24 Mbps data rates are mandatory. Four subcarriers out of 52 subcarriers are employed as pilot subcarriers which are references to disregard frequency-phase shifts-rotations of the signal during transmission. Remaining 48 subcarriers are used to transmit information in parallel streams in which each subcarrier is spaced by a 0.3125 MHz Subcarrier frequency spacing  $\Delta_f$  (A total of 20 MHz for 64 subcarriers). Table 3.1 shows IEEE 802.11a timing analysis [26]. According to that, receiver has only  $4\mu s$  to detect and decode the entire packet. This is one of the reasons why synchronization in OFDM is performed in time-domain.

*Table 3.1. IEEE 802.11a Timing Analysis*

Index	Parameter	Abbreviation	Value
0	Total Subcarriers	N	64
1	Usable Subcarriers	$N_{\text{tot}}$	52
2	Data Subcarriers	$N_D$	48
3	Pilot subcarriers	$N_p$	4
4	Subcarrier Frequency Spacing	$\Delta_f$	0.3125 MHz
5	IFFT/FFT Duration	$T_{\text{FFT}}$	$3.2\mu s$
6	Preamble Duration	$T_p$	$16 \mu s(\frac{1}{\Delta_f})$
7	OFDM Symbol Duration	$T_{\text{sig}}$	$4 \mu s(T_{\text{GI}} + T_{\text{FFT}})$
8	Guard Interval Duration	$T_{\text{GI}}$	$0.8 \mu s(\frac{T_{\text{FFT}}}{4})$
9	Training Symbol GI Duration	$T_{\text{GT}}$	$1.6 \mu s(\frac{T_{\text{FFT}}}{2})$
10	Symbol Interval	$T_{\text{sym}}$	$4 \mu s(T_{\text{GI}} + T_{\text{FFT}})$
11	STS Duration	$T_{\text{STS}}$	$8 \mu s(10 \times (\frac{T_{\text{FFT}}}{4}))$
12	LTS Duration	$T_{\text{LTS}}$	$8 \mu s(T_{\text{GT}} + 2 \times T_{\text{FFT}})$

**Short Training Sequence** : STS is composed of ten repetitions of a  $0.8 \mu\text{s}$  symbol based on sequence given at Equation (3.13). The sequence is started from -26th subcarrier and ended to 26th subcarriers. Subcarriers -32 to -27 as well as 27 to 32 are used as the guard bands. The sequence is chosen to have good autocorrelation properties together with low PAPR. While the correlation peaks are used as an initial estimation for packet detection along with coarse frequency estimation and timing synchronization, the delayed responses degrade its throughput. Packet detection, which is achieved by correlating signal with a delayed version of itself, commonly said as autocorrelation due to the repetitive nature of STS. Frequency estimation is done by measuring in phase differences between two samples which are separated by  $0.8 \mu\text{s}$ . [25]

$$S_{-26,26} = \sqrt{\frac{13}{6}} \{0, 0, 1 + j, 0, 0, 0, -1 - j, 0, 0, 0, 1 + j, 0, 0, 0, -1 - j, \\ 0, 0, 0, -1 - j, 0, 0, 0, 1 + j, 0, 0, 0, 0, 0, 0, 0, -1 - j, 0, \\ 0, 0, -1 - j, 0, 0, 0, 1 + j, 0, 0, 0, 1 + j, 0, 0, 0, 1 + j, 0, \\ 0, 0, 1 + j, 0, 0\} \quad (3.13)$$

**Long Training Sequence** : LTS is composed of two  $3.2 \mu\text{s}$  symbols, based on sequence given at Equation (3.14), appended by a  $1.6 \mu\text{s}$  which is a replication of the last half of the long training symbols. Hence the total length of the LTS will take  $8 \mu\text{s}$  to be transmitted. Similar to STS, LTS begins from -26th subcarrier to 26th subcarrier. LTS may be used for more precise time acquisition due to the transition between STS and LTS. LTS along with with STS are used for a more accurate fine frequency estimation. Figure 3.8 depicts how synchronization is performed using STS and LTS. [25]

$$L_{-26,26} = \{1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, \\ 1, 1, 1, 0, 1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, \\ -1, 1, -1, 1, -1, 1, 1, 1, 1\} \quad (3.14)$$

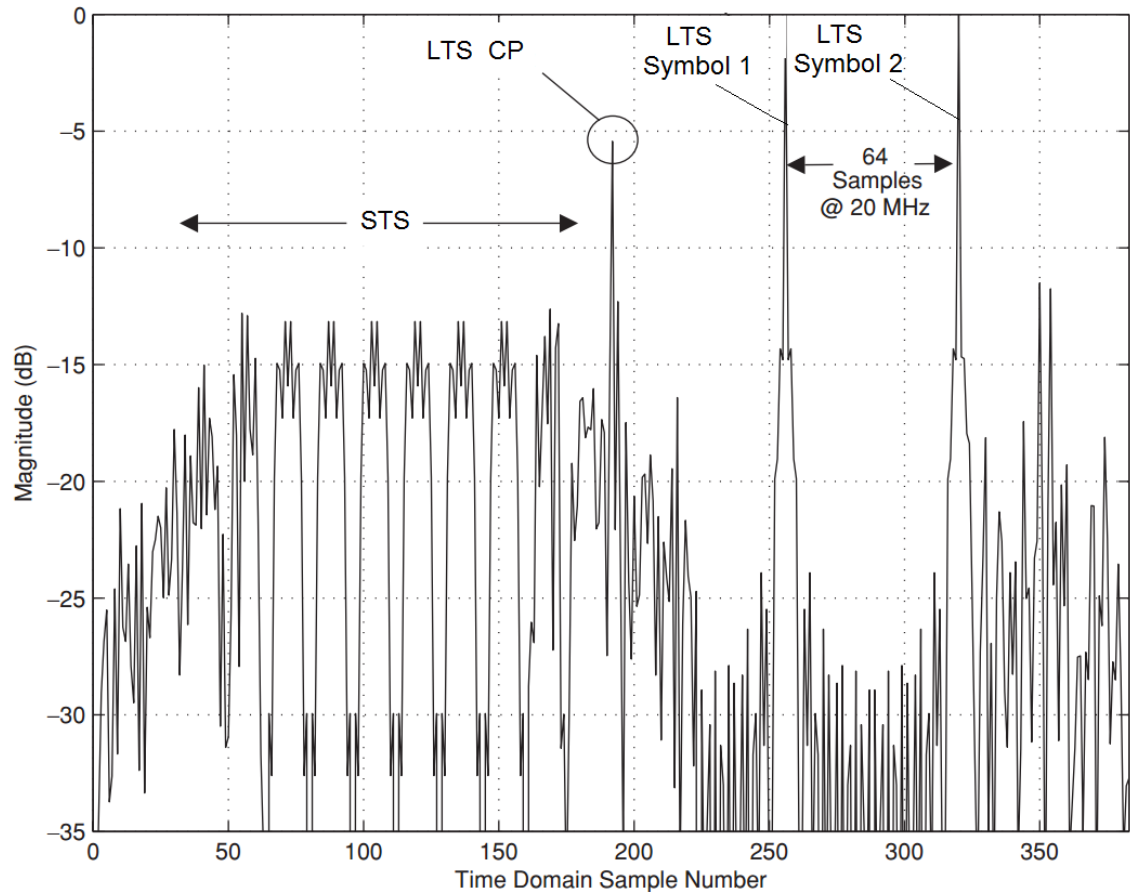


Figure 3.8. Autocorrelation of the preamble [25, p.67]

### 3.4 OFDM Synchronization Steps

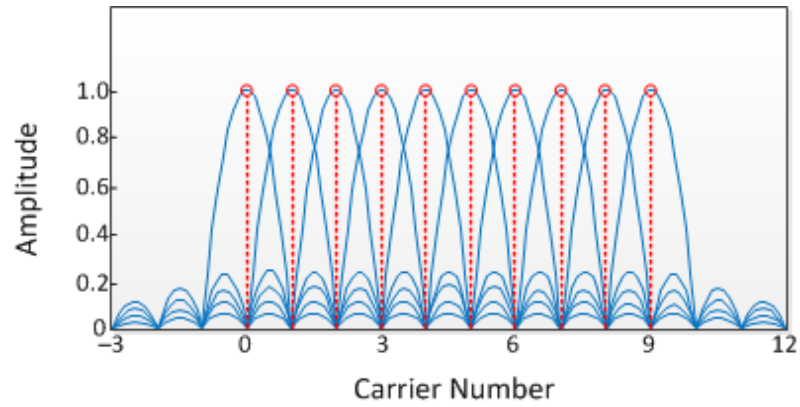
As earlier discussed, OFDM suffers from frequency-domain errors rather than time-domain errors. In that case, OFDM is vulnerable to CFO and CPE which cause  $\Delta_f$  and, subsequently,  $\phi(t)$  to the received signal. Figure 3.9 shows how existence of CFO introduces a shift of magnitude  $\Delta_f$  to the receiver in which the subcarriers lose their orthogonality with the receiver filter.

As previously mentioned, another issue which is a threat in OFDM systems is CPE. The presence of any offset in symbol timing causes phase noises. In simplest form, failure to detect the proper symbol boundaries introduces a phase noise where the constellation points are dispersed as it is illustrated in Figure 3.10.

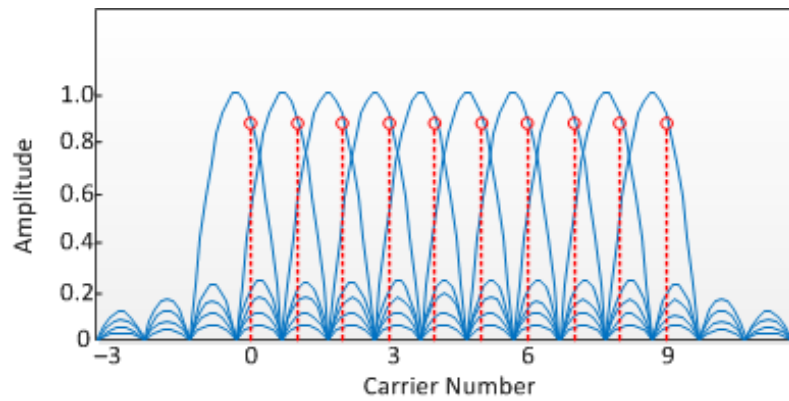
Basically, synchronization in time-domain is performed in two phases called *coarse symbol timing detection* and *fine symbol timing detection*.

#### 3.4.1 Coarse Symbol Timing Detection

The first phase to detect and extract the incoming signal waveform from the received signal is called coarse symbol training detection. In this regard, several methods

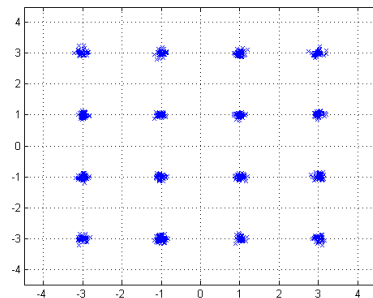


(a)

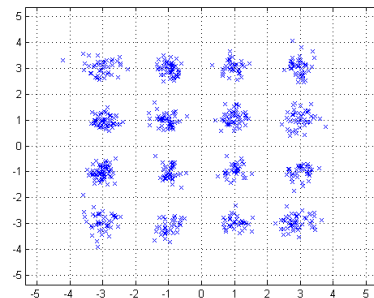


(b)

**Figure 3.9.** (a) Sampling without CFO, (b) Effect of CFO



(a)



(b)

**Figure 3.10.** (a) Sampling without CPE and (b) Effect of CPE.

have been proposed. Coarse symbol timing detection is achieved by using one of the following methods: [6]

### Delay and Correlate

The very straightforward algorithm for symbol timing detection is Delay and Correlate (DC). The principle of the DC algorithm is to detect the maximum autocorrelation of the signal based on Equation (3.15), where  $z_m$  is the received time-domain signal,  $R$  is the repetition interval and  $L$  is the separation between two symbol intervals.

$$\Phi_{DC}(m) = \left| \sum_{r=0}^{R-1} z_{m-r} z_{m-r-L}^* \right|,$$

$$\hat{m}_{DC} = \arg \max_m \Phi_{DC}(m). \quad (3.15)$$

Although DC algorithm is simple in term of complexity, it has two major drawbacks. Firstly, the peak magnitude of  $\Phi_{DC}$  varies due to different signal powers. Secondly, when the autocorrelation of the repetitive periods is done, the edge of the correlator output, specially in noisy environments, is not dropped sharply. In other words, it will take some time for a signal to reach to its lower level from the peak.

### Maximum Likelihood Metric

According to the Equation (3.16), principle of Maximum Likelihood (ML) algorithm is based on the assumption that the received signal is uncorrelated except for some replicas. This method is less reliable than other proposed algorithms, due to the high complexity of the hardware, which calculates magnitude of SNR ( $\rho$ ), as well as number of errors caused by bypassing SNR estimation. Equation (3.17) is an special case of ML, also known as Minimum Mean Square Method (MMSE), while magnitude of SNR is infinite.

$$\Phi_{ML}(m) = 2 \left| \sum_{r=0}^{R-1} z_{m-r} z_{m-r-L}^* \right| - \frac{\rho}{1+\rho} \sum_{r=0}^{R-1} (|z_{m-r}|^2 + |z_{m-r-L}|^2)$$

$$\hat{m}_{ML} = \arg \max_m \Phi_{ML}(m). \quad (3.16)$$

$$\Phi_{MMSE}(m) = \sum_{r=0}^{R-1} |z_{m-r}|^2 + \sum_{r=0}^{R-1} z_{m-r-L}^2 - 2 \left| \sum_{r=0}^{R-1} z_{m-r} z_{m-r-L}^* \right|$$

$$\hat{m}_{MMSE} = \arg \max_m \Phi_{MMSE}(m). \quad (3.17)$$

### Normalized Metrics

Schmidl and Cox in [24] proposed a group of timing detection algorithms known as preamble (which was discussed earlier). As Equation (3.18) shows, maximum  $\Phi_s(m)$  is achieved at the end of the preamble where two sequences of identical  $N/2$  samples are used. According to [6], Minns et al. proposed a more general preamble structure consisting of  $U$  identical segments which are varied in polarization. In case of  $U = 4$  the preamble structure is given as Equation (3.19) where  $A$  is the segment consisting of  $N/4$  samples.

$$\Phi_s(m) = \frac{\left| \sum_{r=0}^{N/2-1} z_{m-r} z_{m-r-N/2}^* \right|^2}{\left( \sum_{r=0}^{N/2-1} |z_{m-r}|^2 \right)^2} \quad (3.18)$$

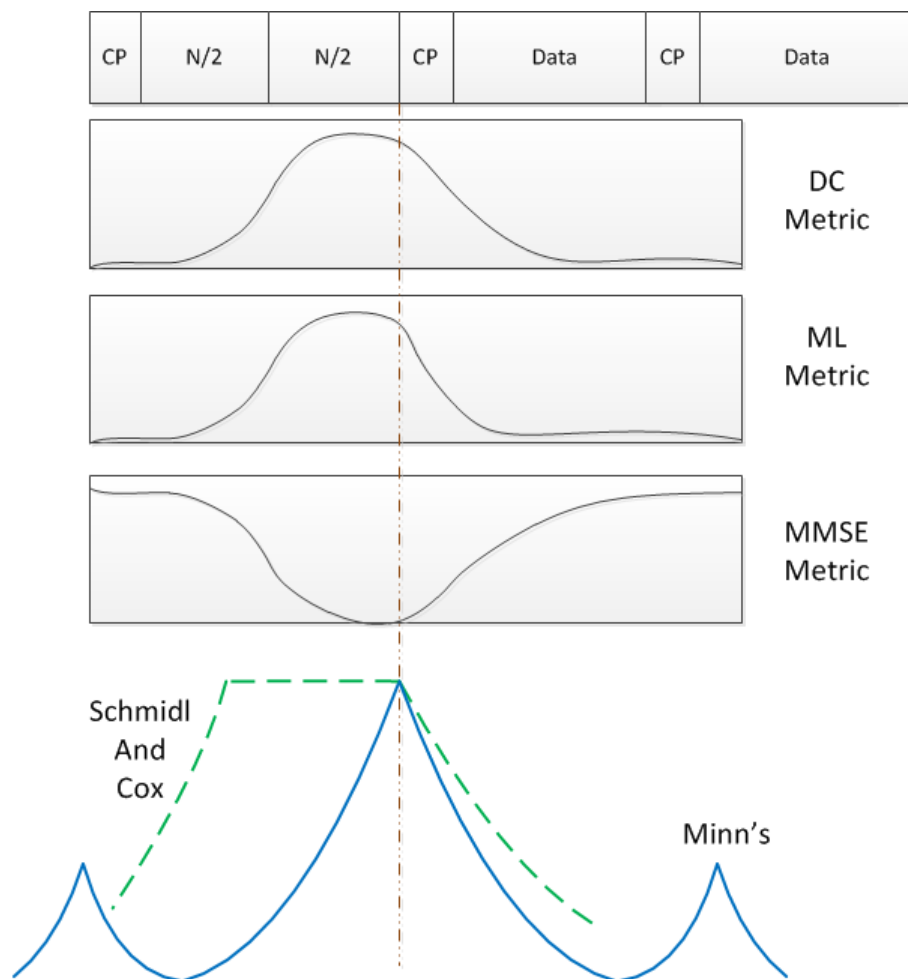
$$[-A \quad A \quad -A \quad -A] \quad (3.19)$$

As it is illustrated in Figure 3.11, a common phenomenon, which is the existence of a wide plateau, appears in all DC, ML, MMSE and in Schmidl's algorithms to some extent. When the number of identical preambles exceeds more than two segments, for example STS repetition in 802.11a, a wide plateau is created at the correlator output. Theoretically, this plateau indicates an ISI-free region to the FFT. In reality, additive noise widens the area of plateau. In that case, it is worth mentioning that the longer period in preamble, the larger FFT window, the more accurate detection. Eventually, Minn's algorithm defers any plateau and has a sharp peak in its metric. [6]

#### 3.4.2 Fine Symbol Timing Detection

As it is discussed in previous section, coarse symbol timing detection algorithms are only able to obtain timing information out of the received signal approximately. Thus, large timing errors might still exist. Therefore, in order to obtain accurate timing, another precise approximation is required which is called fine symbol timing detection. [6]

As previously said, *TIME* is a crucial issue in packet-based systems. Therefore, ISI-free DFT window should be prepared as soon as possible to perform channel estimation and packet header extraction in next steps. If fine symbol timing detection is not acquired in time, additional delay lines should be considered to buffer the input signal. Similar to coarse symbol timing detection algorithms, there are few



*Figure 3.11. Coarse symbol timing detection algorithms*

algorithms proposed to obtain fine symbol timing detection as following

### Time-domain Cross Correlation

Usually, fine symbol timing detection, in order to obtain Channel Impulse Response (CIR), is performed by matching time-domain received signal with a preamble waveform. In other words, instead of autocorrelating a noisy received waveform with a delayed version of itself, which is apparently noisy as in coarse symbol timing detection algorithms, a clean preamble waveform is correlated with the noisy received signal. The optimal timing estimation can be obtained according to Equations (3.20) and (3.21) where  $Q$  is the total length of preamble and  $p_q$  denotes the preamble samples. As a new approach, a threshold point can be set considering correlator which responses to any magnitude greater than the threshold point. [6]

$$\Phi_{zp}(m) = \sum_{q=0}^{Q-1} z_{m+q} P_q^* \quad (3.20)$$

$$\hat{m}_{MAX} = \arg \max_m \Phi_{zp}(m) \quad (3.21)$$

### Frequency Response Estimate

Another approach is to perform an IFFT in order to extract CIR information. According to Equation (3.22), CIR  $\hat{\mathbf{h}} = [\hat{h}_{-N/2} \hat{h}_{N/2+1} \dots \hat{h}_{N/2-1}]$  can be obtained, where  $\mathbf{F}^{-1}$  denotes an  $N \times N$  IFFT matrix,  $\mathbf{z}$  is a set of received frequency domain subcarriers and  $\mathbf{X}$  is a diagonal matrix whose  $i$ th diagonal element is the transmitted signal at  $i$ th subcarrier. Once  $\hat{h}_m$  exceeds a predefined threshold,  $m$  is considered as the symbol timing. Since IFFT requires extra operations, a long latency is introduced in this case.

$$\hat{\mathbf{h}} = \mathbf{F}^{-1} \times \mathbf{X}^{-1} \times \mathbf{z} \quad (3.22)$$

### Frequency-Domain Phase Shift

Conceptually, adjacent subcarriers suffer from similar channel fading. Useful information for fine symbol timing detection are only provided if the effect of channel fading is compensated. According to Equation (3.23), symbol timing offset  $\hat{m}_{PS}$  is obtained where  $\angle(\cdot)$  is the phase of complex number,  $Z_k$  is the received frequency-domain signal and  $X_k$  is the transmitted frequency-domain signal of the  $k$ th subcarrier. [6]

$$\hat{m}_{PS} = \frac{N}{2\pi} \angle \left( \sum_k (Z_{k+1} Z_k^*) (X_{k+1} X_k^*)^* \right) \quad (3.23)$$

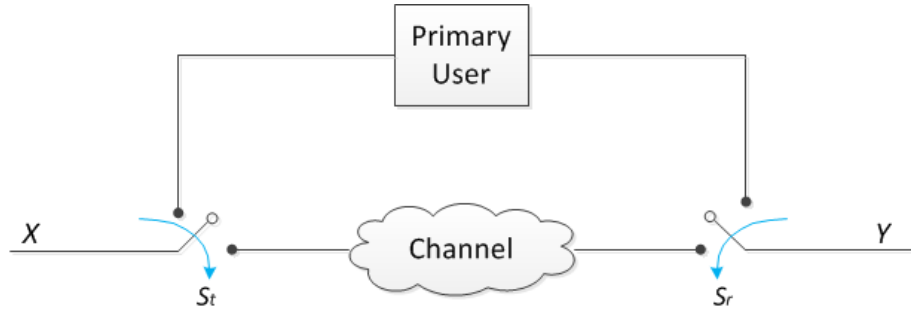
Nevertheless, all fine symbol timing detection algorithms can not handle received signals with large CFO. Therefore, before using these algorithms, CFO must be compensated in advance.

## 3.5 NC-OFDM Systems

NC-OFDM is a promising technique in CR systems to avoid any interference with the primary user, in which only unused subcarriers are employed to be transmitted. In NC-OFDM based CR systems, secondary transmitter instead of using a set of contiguous subbands might employ a variety of non-contiguous subbands due to the licensed user activities.

According to [27] in NC-OFDM systems, each subchannel is modeled as a two-switch channel in which the switch will open as soon as a primary user is detected. Figure 3.12 depicts how a subchannel is employed as a two-switch channel model.





*Figure 3.12. Two switch channel model*

This approach can be explained considering Equation (3.24) where signals  $X$  and  $Y$  are transmitted and received signal, respectively;  $N$  denoted as AWGN,  $S_t$  and, subsequently,  $S_r$  are transmitter and receiver switches  $\in \{0, 1\}$ . Therefore, it might happen that the sender and the receiver experience different wireless environment and, consequently, there is no guarantee that a subband occupied for secondary transmission still stays synchronized [28].

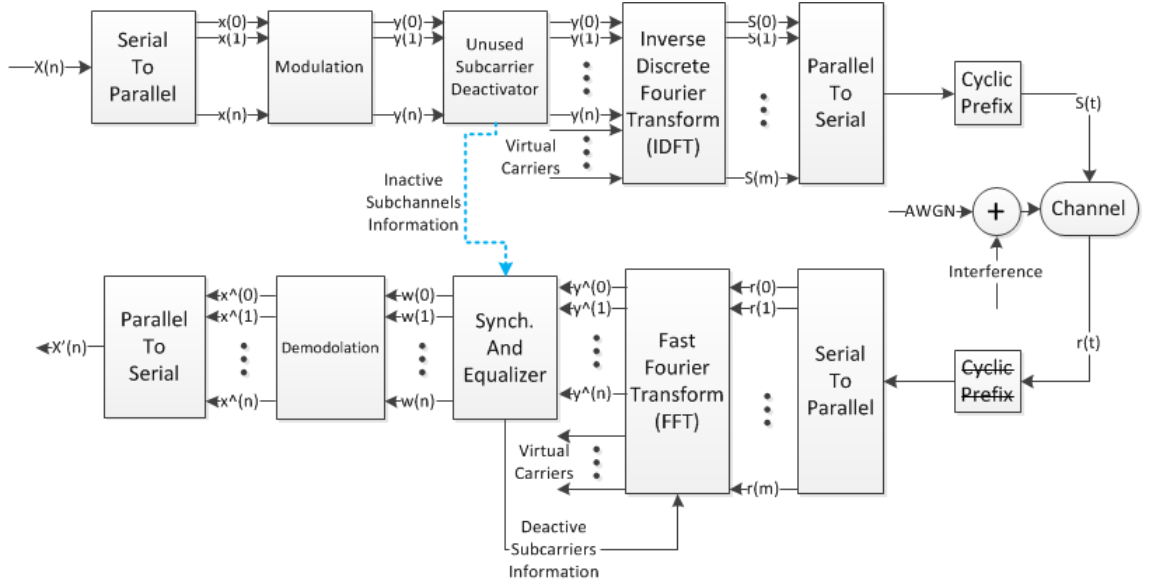
$$Y = (XS_t + N)S_r \quad (3.24)$$

### 3.5.1 NC-OFDM synchronization issues

Theoretically, synchronization in OFDM systems can be done either in time-, frequency-domain or both. In reality, the reason why most of the OFDM systems prefer to perform synchronization in time-domain is lack of time due to the packet-based system nature. As soon as the receiver is powered up, it should start autocorrelating and crosscorrelating on received waveform to extract useful information using preamble structure.

Symbol timing is extracted using a correlator whose coefficients are exactly the same samples of the preamble in time-domain representation. What makes the NC-OFDM receiver fail at this stage is the change in time-domain representation of the predefined preamble due to the *non-continuity* of the encoded signal. In other words, any activity by the primary user enforces the transmitter to alter its transmitting frequency which results in changing waveform of the signal in time-domain.

In addition to previously discussed synchronization problems in OFDM system, there are two new challenges which NC-OFDM receivers encounter. Firstly, only a part of subcarriers are available and the rest are occupied by licensed user which makes many traditional synchronization techniques used for OFDM systems disabled. Secondly, secondary users must reduce their transmission power to mitigate interference with the licensed user due to the sidelobe leakages. Therefore, a ro-



*Figure 3.13. Out-Of-Band control systems.*

best technique should be employed to reestablish the synchronization process in lower SNR region as well as subcarriers deficiency [29]. This can be achieved by employing either in-band or out-of-band control scheme.

### Out-Of-Band systems

In OFDM systems, location of the preamble, pilot and data carriers are fixed whereas in NC-OFDM a fixed location for these carriers might cause interferences with the primary user. Thus, a precise location of the carriers are not guaranteed. Moreover, the location of primary user is supposed to be changed across the spectrum over the time. Thus, useful carriers are changed over the spectrum as well. Henceforth, the receiver has no idea about the new locations of those useful carriers until it is notified.

One of the key challenges in NC-OFDM synchronization is how secondary receiver should distinguish which subchannels have been employed by the secondary transmitter. Recently, in some works, for example in [30], [31], [32], [33] and [34], another control channel is used to inform the receiver about the new information of the spectrum. This method is also known as Out-Of-Band (OOB) controlling mechanism. However OOB channel might not be suitable for NC-OFDM synchronization, since the dedicated channel may not be available in some practical situations [35]. Figure 3.13 represents a schematic of OOB control system. As it is illustrated, an extra dedicated channel has been established to transmit information regarding to active and inactive subchannels. Therefore, major challenges with respect to NC-OFDM synchronization are based on in-band signaling methods.

### **In-Band systems**

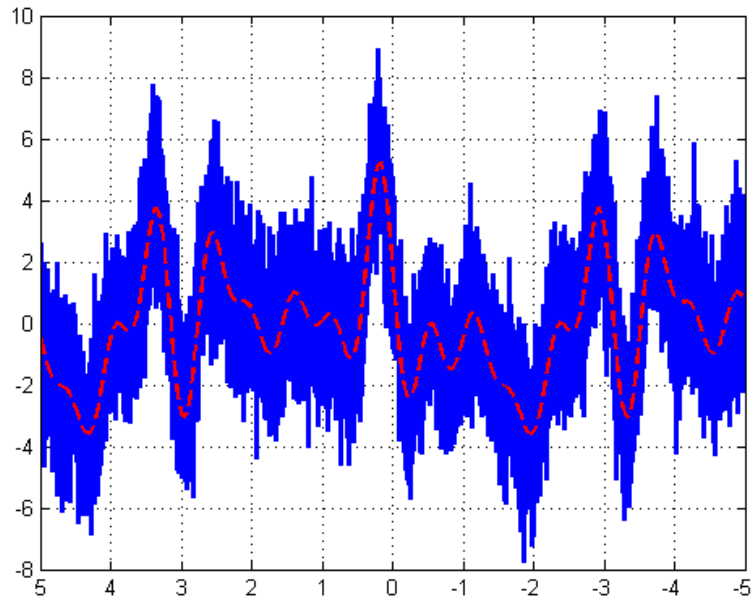
In contrast to OOB transmission, in in-band transmissions, information about spectrum aspects are embedded to the data packets themselves. Conceptually, embedded information are located at the preamble of each packet. Therefore, receiver must be intelligent enough to detect which subchannels are occupied by second transmitter. Hopefully, according to [39], secondary users willing to transmit over the licensed spectrum might have useful knowledge about the signal structure, power and location of the primary user. Hence, in all synchronization algorithms proposed for NC-OFDM receiver, it is assumed that the secondary receiver has prior information about the primary user's activities. Otherwise, besides secondary user synchronization concerns, secondary receiver would be responsible to detect and, subsequently, ignore the existence of primary user as well.

One of the main aspects of the secondary receiver is to ignore subbands, where primary user is transmitting in, based on its prior knowledge. As previously discussed, secondary user must reduce its transmission power to avoid any interference with the primary user due to the sidelobe leakages. Therefore the primary user signal power is massively higher than that of secondary. If the receiver fails to discard primary user's activities, a massive interference caused by the primary subband degrades the overall throughput of the NC-OFDM systems.

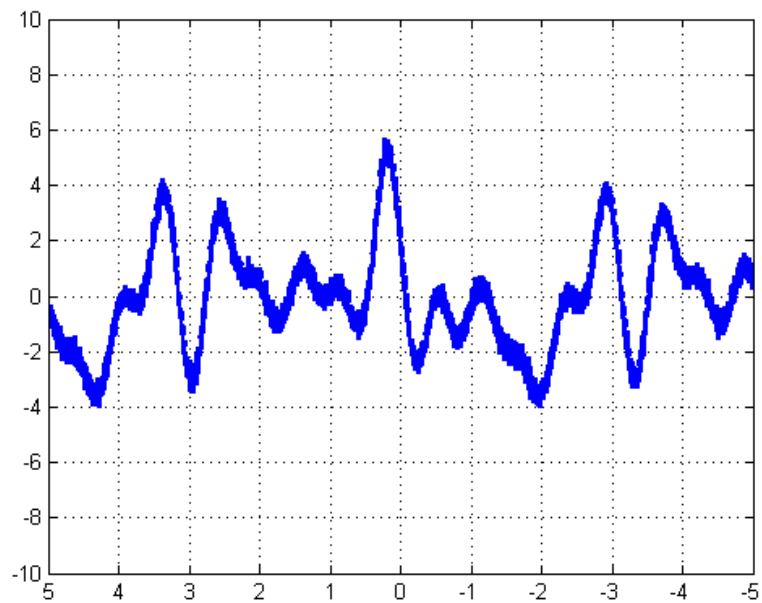
However, there are proposed techniques which are able to perform synchronization, most of them are suffering from interferences caused by primary user existence. For example in [15] an in-band solution is proposed using Frank-Zadoff-Chu (FZC) sequence to identify spectrum usage pattern. Although, this is the first in-band technique solution (according to the author's opinion), the algorithm heavily depends on available spectrum as well as Signal to Interference and Noise Ratio (SINR) in which one of the major sources of interferences is the existence of primary user.

In [36] a fractional bandwidth model has been proposed. However, the proposed algorithm performs synchronization with a special designed Pseudo Noise (PN) sequence, interferences caused by the primary user has not been considered. Therefore, this algorithm is only feasible for those environments where the primary signal power is lower than secondary transmitter.

In [35] a A Posterior Probability (APP) algorithm is employed to approximately calculates which channels are active, including primary user channels, and then by performing a Hard Decision-based Detection (HDD) scheme NC-OFDM symbols can be detected. Since HDD has a poor performance in sever interfering environments, a Soft Decision-based Detection (SDD) is performed to improve the performance. However, as author said "When subchannels are not close to the edge of the subband, the performance of the hard decision is perfect. However, the performance is poor when subchannels approach the edge of subband". According to [28], in this



(a)



(b)

*Figure 3.14. Received waveform containing transmitted signal (a) before matched filter and (b) after matched filter*

algorithm, the system code rate is  $1/4$  when only half of the subcarriers are active.

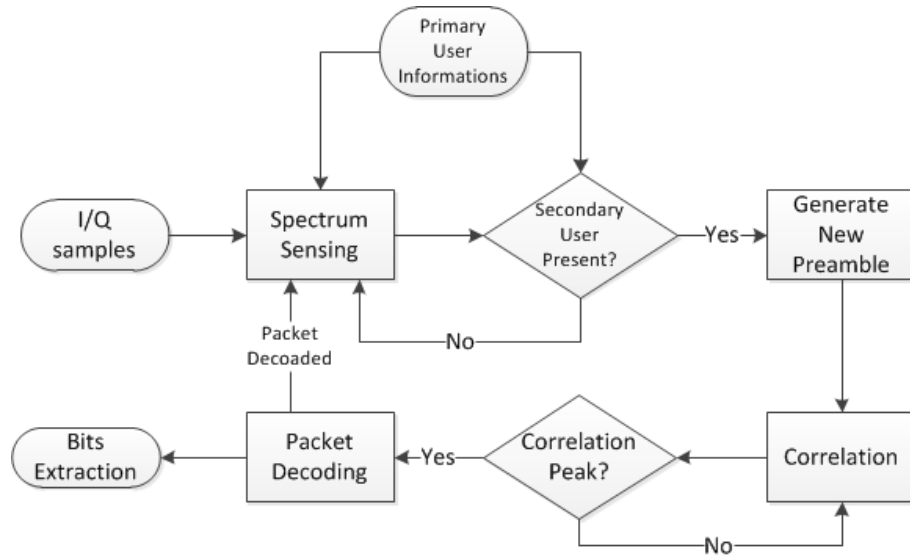


Figure 3.15. NC-OFDM synchronization steps

### 3.5.2 Primary User Filter

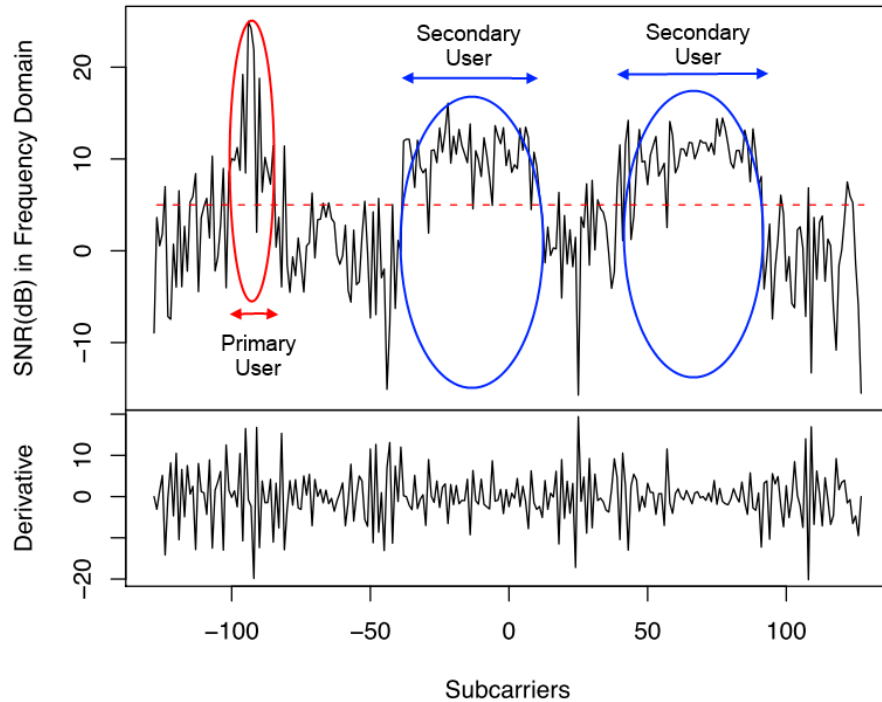
In order to filter out the primary user, often, the characteristics of the primary user such as pilot carriers and synchronization sequences are known for the secondary receiver. In context of CR, this approach can be done using *matched filters*.

In terms of signal processing, matched filter is obtained by correlating a known sequence with an unknown signal to discover the known sequence. In practice, this can be achieved by convolving an unknown incoming signal with the conjugated version of the known sequence. Indeed, matched filter is an optimal filter which maximizes the SNR to its maximum magnitude. Figure 3.14 depicts capability of matched filter to extract signal within the noisy received waveform. At the receiver, a matched filter can be implemented as a complex FIR filter capable of calculating one crosscorrelation per clock cycle. [1]

### 3.5.3 NC-OFDM Synchronization Steps

As previously addressed, synchronization in OFDM systems can be either in time-domain, frequency-domain or both whereas synchronization in frequency-domain is an obligation in NC-OFDM systems due to alteration in transmitted frequencies by the transmitter which changes the representation of the time-domain waveform. Hence, at the receiver, time-domain correlator does not yield satisfactory results anymore and might fail to perform the correlation until it is upgraded with new generated coefficients.

Usually, reestablishing the correlator coefficients are done in two stages: subcarrier detection and coefficients regeneration of the time-domain correlator. First step,



*Figure 3.16. NC-OFDM waveform occupancy by primary and secondary user [37]*

in brief, is to search through the entire spectrum, once a transmission is detected, new correlation properties should be regenerated using frequency-domain representation of the preamble coefficients. Figure 3.15 shows NC-OFDM synchronization, step by step. In three major steps a NC-OFDM packet can be decoded which are mentioned as following [37]

- *Spectrum Detection.*
- *Regenerate Time-Domain Correlator Coefficients.*
- *Update Correlator Coefficients.*

**Spectrum Detection:** Since the secondary receiver has no information about which subcarriers are employed by secondary transmitter, all subcarriers must be gathered at the receiving signal. Then spectrum is sensed using one of spectrum sensing methods discussed before, for example DC or ML. Then the results are compared to a threshold to decide which subcarriers contain information.

Note that the process works in a pipeline structure up to here. With any detection of an incoming packet, the pipeline process will be stalled. Figure 3.16 illustrates spectrum occupancy while both primary and secondary users are transmitting. By

taking into account both figure and prior information about the primary user activities, primary user shall be filtered out and the spectrum is considered to be occupied by the secondary user from the receiver aspect of view. [37]

**Regenerate Time-Domain Correlator Coefficients:** Following by the secondary transmission detection, time-domain coefficients of the correlator can be locally generated at the receiver using frequency-domain representation. Once the time-domain correlator is initialized by new generated coefficients, the incoming waveform can be implied similar to OFDM and is synchronized using discussed OFDM synchronization techniques.

Note that a copy of the received signal must be buffered in advance to extract time samples from the beginning of the packet. As long as the correlation shows satisfactory results, spectrum sensing unit shall be turned off to save energy. [38]

**Update Correlator Coefficients:** If the correlator fails to detect secondary user's incoming packet, spectrum sensing unit restarts sensing the entire spectrum to provide preamble re-generator with the new information regarding to the secondary user. [38]

## 4. STATE OF ART IN SYNCHRONIZER ARCHITECTURE

An NC-OFDM-based CR is capable of deactivating subcarriers which are not required for transmission. Since the primary user does not use the entire spectrum, the secondary users could occupy those white spaces as long as they are not interfering with primary subbands. In a similar manner, when an offline primary user is become online, the secondary users **must** terminate their transmissions in frequency bands which might cause any interference with primary user.

Although this procedure might seem to be simple from the transmitter point of view, the receiver is encountering several troubles where the major one is synchronization. The receiver has no idea about frequency bands occupied by the secondary user due to the change in primary user location. Furthermore, according to primary user activities, the location of preamble, pilot and data carriers are varied over time. Above mentioned issues enforce the receiver to be always compromised about the existence of secondary users.

Since NC-OFDM is chosen as a possible candidate in CR systems, a flexible transceiver is required to enable DSA feature. In this thesis, the transmitter part has been ignored and the main focus is on synchronization part of the receiver. The state of art of the design is given followed by explaining previous efforts and their respective limitations.

However, in [30], [31], [32], [33] and similarly in [34], some solutions have been proposed, these methods are considered in an OOB channel condition. In OOB systems, a particular channel is dedicated to both secondary transmitter and receiver in which new information about the spectrum occupancies in terms of active and deactivated subchannels are transferred. Therefore, the receiver is always alert about secondary users activities with the payoff of having an especially purposed channel assigned to secondary transmitters which undergoes additional cost as well as wasting the bandwidth. Hence, OOB control channels are not good candidates in context of CRs.

Irrespective of OOB transmission, up to my knowledge, there are only couple of researches in concept of in-band synchronization. In [36] a fractional bandwidth model has been proposed. Apart from OFDM based synchronization scheme, a special designed PN sequence is generated in frequency domain. In time domain,



preamble is represented with two identical halves whose sign bits are varied, while the interferences caused by the primary user have not been considered. Theoretically, this algorithm is based on the assumption that the power strength of the primary user signal is lower than secondary transmitter. In reality this algorithm will not work at all, since the secondary transmitter always keeps its transmission power lower than the primary user to mitigate interfering with primary band due to the sidelobe leakages. Therefore, this algorithm is not feasible for NC-OFDM based CR in practice.

In [35] the interference caused by primary user has been considered. Based on that, an APP algorithm is used to distinguish which subchannels are active by employing received training symbols in all subchannels and calculate the probability of active subchannels. When an active subchannel is detected, a HDD scheme is performed to detect NC-OFDM symbols based on calculating a threshold point. The threshold decision is varied based on signal strength as well as primary user power. However, in subchannels far from primary user band the HDD works fine, for those subchannels close to primary user band the performance is degraded drastically. Since HDD performance is degraded in noisy environments, another algorithm named SDD is performed to increase accuracy on those subchannels closed to the primary user band. SDD calculates another threshold point based on primary user activities. However, for subchannels located close to the edge of subbands, the performance is decreased due to not neglecting the primary user existence. Therefore, the performance of the proposed method is strongly depended on primary user activities.

According to [28], in the purposed algorithm in [35], the system code rate is  $1/4$  while only half of the subcarriers are active. Therefore, the authors try to improve the performance by employing an APP algorithm to identify active subcarriers and then used a Low-Density Parity-Check (LDPC) to improve system code rate. However, the authors emphasized more on how to create LDPC code and never explained the synchronization process, they just implied "with perfect spectrum synchronization, the BPSK symbols could be modulated onto only active subchannels, the performance of which is supposed to be ideal in this paper. However, in this ideal case, a perfect spectrum synchronization is necessary for the reason that any error in the synchronization information would lead to a serious mismatch between the spectrum detected at the receiver and actual spectrum used at the transmitter." Moreover, the purposed method requires additional hardware to decode received LDPC with iteration bound of maximum 80 times meaning more power consumption as well as chip area.

Authors in [37] and [38] proposed a blind synchronization method in which the receiver is able to regenerate time-domain preamble locally at the receiver by em-

ploying frequency domain representation of the preamble. In these articles, instead of performing a full 16-bit Multiplier-Accumulate (MAC) operations in frequency-domain, a Multiplier-Less (M-L)-based correlator has been investigated. Authors considered primary user activities and, hence, a "binary mask" (which is implemented as a simple XOR gate) extracts secondary transmission from incoming signal. As a result, primary user existence is eliminated after FFT is done. Thereafter, by regenerating time-domain correlator coefficients from frequency-domain representation of the preamble, the rest of the synchronization process is similar to that of OFDM.

The objective of this thesis is to implement a flexible timing synchronization scheme for CRs. The synchronization block is able to perform both autocorrelation and crosscorrelation functions on demand. These two functions are performed in only one block. In other words, by feeding either predefined coefficients inside its memory or incoming signal itself to the coefficient of the correlator, it enables multicorrelation functions. However, the work is quite similar to [38], the primary user is filtered before the FFT is done. According to [39], by having prior knowledge about licensed spectrum information, FFT computations regarding to primary user subchannels are ignored. Therefore, the inputs coming from FFT core, are white spaces where the secondary users might transmit within. Hence, massive workload with respect to primary user subchannels are eliminated meaning save more power conservation. In this thesis work, another diversity with what Aveek proposed in [38] is that the entire work is done in frequency-domain. Hence, there is no need to employ time-domain correlator, since the frequency-domain multicorrelator performs fast enough to decode the entire packet.

In addition, the synchronizer core, which is a multicorrelator, is implemented using different architectures as its core. Therefore, in different situations, different implementations can be considered. Moreover, with PR feature, which is a Beta version available only on Altera Stratix-V FPGAs, synchronizer is able to reconfigure some portion of itself while the rest of the design is operating. Hence, a multi-standards synchronization scheme which is able to change analog signal processing with digital signal processing is presented on a single chip.

Since the entire synchronization procedure is performed in frequency-domain (which demands a massive number of computations), major concerns related to the presented synchronization block are power consumption as well as area size. Moreover, in order to not to lose the next incoming packet, the multicorrelator must be able to detect current packet based on multicorrelation results as fast as possible.

According to [48] and [49], a Multiplier-Less multicorrelator is employed to mitigate power consumption as well as area size. MATLAB results in [38] show that the performance of the FIR filter is still at a satisfactory level. However, ModelSim

and compiling results depict that Multiplier-Less architecture is the best choice for multicorrelator mostly in all aspects.

The following section explain the core content of the synchronization block based on different implementations related to multicorrelator architecture in detail. Furthermore, in contrast to all above mentioned proposed algorithms in synchronization, the compilation results are given to make comparison between different architectures in terms of total number of registers, logic utilization, core dynamic thermal power consumption along with cell power consumption in detail, etc. Hence, this thesis work provides significant information to be a perfect resource for further research regarding to NC-OFDM-based CR.

## 5. FPGA IMPLEMENTATION OF MULTICORRELATOR

In this section, implementation of multicorrelator on Altera Stratix-V FPGA kit are studied in detail. As it will be seen, different architectures regarding to multicorrelator are fully explained. Following subsections describe the entire design implementation step by step.

### 5.1 Design Implementation Issues

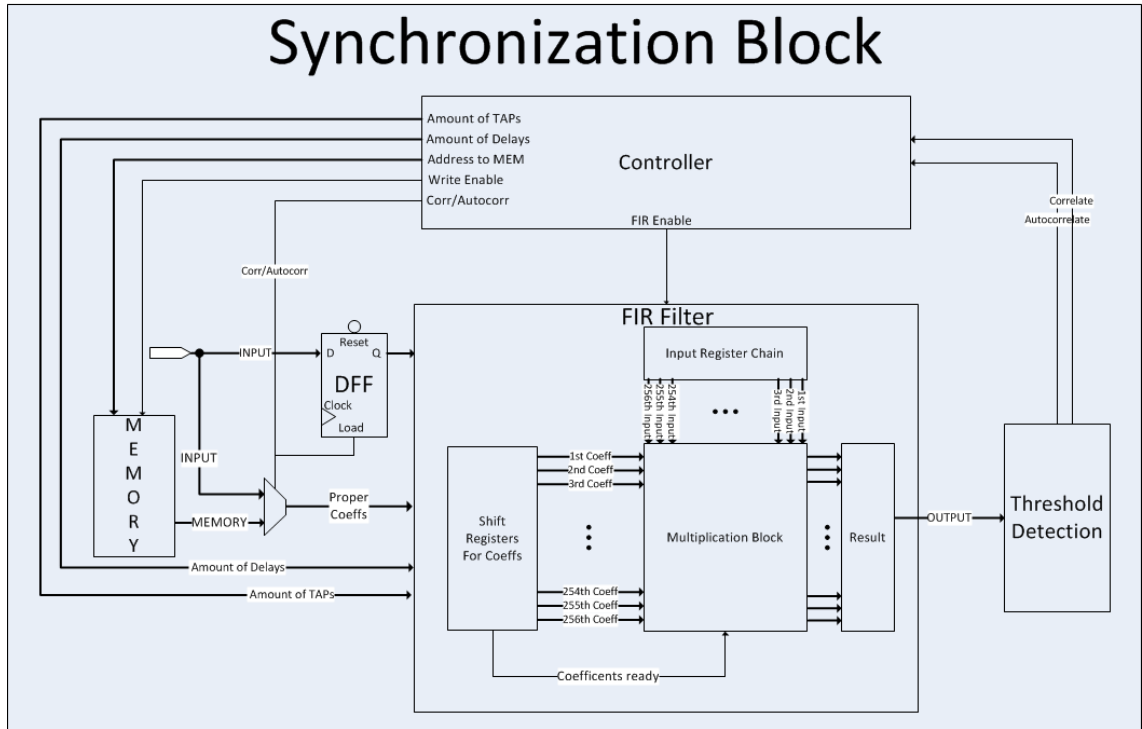
As previously discussed, Synchronization in NC-OFDM is done in frequency domain by performing a crosscorrelation between predefined coefficients and received signal, followed by an autocorrelation between the received sequence and a delayed version of itself. In this section, an overview of the entire implementation of synchronization block is given. Synchronization block is composed of several sub-blocks which are integrated together to form the entire design. As Figure 5.1 shows, the following blocks, which will be explained in detail later on, are presented in implementation of the synchronizer:

- *Complex FIR filter*
- *Memory Block*
- *Threshold Detector*
- *Controller*

Since FIR filter block has the most complex design among the above mentioned blocks, the basic concept of different architectures regarding to the FIR filters is explained, before stepping forward to designs explanations.

### 5.2 FIR Filter

FIR filters are widely used in DSP applications in which reconfigurability and low complexity are two major concerns. In concept of CR technology, reconfigurability is referred to FIR filters due to the basic idea of the CR. The basic idea is to enable a single hardware platform supports multi-standards communication on a single chip by swapping analog signal processing with digital signal processing.



*Figure 5.1. Synchronization block architecture*

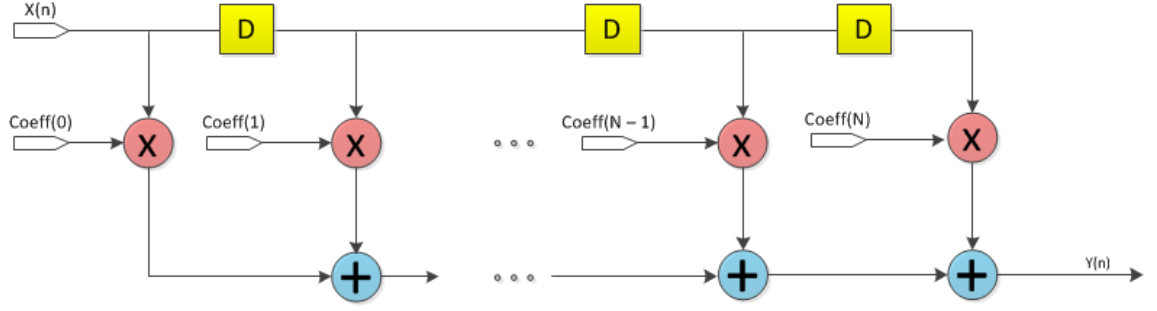
From power consumption point of view, FIR filters are intensive parts of a CR due to the massive workload of complex computations as well as operating at high speed to achieve highest sampling rate. Due to the nature of NC-OFDM-based CR, highest order of the filter, for example 4096-tap FIR filter, is required for performing synchronization. Therefore, power consumption of the receiver is increased along with chip area size [43].

According to Equation (5.1), a FIR filter periodically calculates  $y(n)$  as the products of coefficients  $h(k)$  with incoming signal  $x(n - k)$  on a window whose length is  $N$ , which is also known as filter tap or filter order. This operation is called MAC operation [44]. Therefore, a FIR filter with constant coefficients is a Linear Time Invariant (LTI) filter whose main characteristic is stability.

$$y(n) = h(n) * x(n) = \sum_{k=0}^{N-1} h(k)x(n - k) \quad (5.1)$$

Basically, FIR filters are composed of three fundamental functional units. These three major parts are adders, multipliers and delay elements integrated and arranged in different ways to form different FIR architectures. They are implemented in three different techniques known as Sequential Direct Form (SDF), TDF and PDF, each of which is able to obtain  $y(n)$  according to Equation (5.1).

Note that in this project, the multiplication block is not just a simple multiplier.



**Figure 5.2.** Sequential direct form FIR filter architecture

Since the incoming sequences from FFT outputs have In-phase and Quadrature-phase (I/Q) components, a multiplier must be able to compute complex multiplication. Indeed, complex multiplication can be performed using Equation (5.2) as following:

$$\begin{aligned}
 (a + bi) \times (c + di) &= a(c + di) + bi(c + di) \\
 &= ac + adi + cbi + bdi^2 \\
 &= ac + adi + bci + bd(-1) \\
 &= (ac - bd) + (ad + bc)i
 \end{aligned} \tag{5.2}$$

As previously discussed, in order to perform crosscorrelation and autocorrelation, coefficients should be conjugated first. This can be simply achieved by inverting the imaginary parts of the coefficients. Therefore, Equation (5.3) can be derived from Equation (5.2)

$$\begin{aligned}
 (a + bi) \times (c - di) &= a(c - di) + bi(c - di) \\
 &= ac - adi + bci - bdi^2 \\
 &= ac - adi + bci - bd(-1) \\
 &= (ac + bd) + (bc - ad)i
 \end{aligned} \tag{5.3}$$

Eventually, in this project, in order to perform the conjugation, instead of inverting the sign bit of incoming coefficients and, then, feeding them to the multiplier, simply the multiplication sub-block is modified to do exactly what Equation (5.3) depicts.

### 5.2.1 Sequential Direct Form FIR Filter Architecture

According to the Equation (5.1), SDF FIR filter is the first idea that come to the mind. Figure (5.2) shows the SDF architecture of FIR filter. N-tap filter consists of N delay elements, N multiplier and N-1 adders. For instant, assume that the order

of the filter is equal to 3. Hence,  $y(n)$  has a different value for each iteration which is equal to:

$$\begin{aligned} y(0) &= x(0)c(0) \\ y(1) &= x(1)c(0) + x(0)c(1) \\ y(2) &= x(2)c(0) + x(1)c(1) + x(0)c(2) \\ y(3) &= x(3)c(0) + x(2)c(1) + x(1)c(2) \\ y(4) &= x(4)c(0) + x(3)c(1) + x(2)c(2) \end{aligned}$$

As it can be seen, from third iteration onward, the number of coefficients is constant whereas the input  $x$  is shifted by one per iteration.  $y(n)$  is calculated by accumulating products of constant coefficients and shifted inputs. Considering the above example, assume that the order of filter is  $N$ , the result for the  $n$ th iteration is

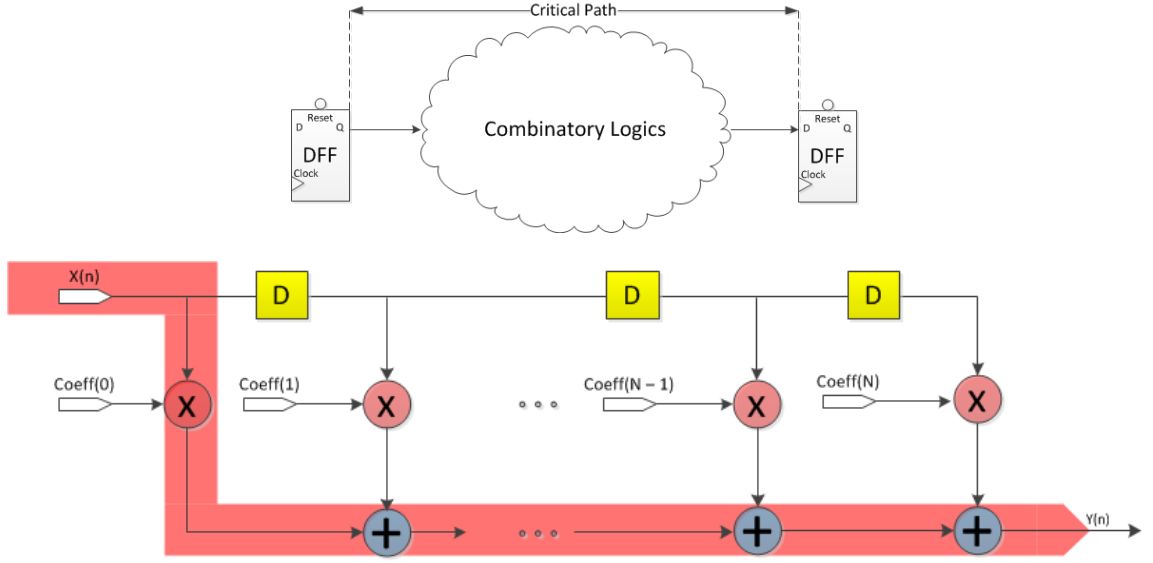
$$y(n) = x(n)c(0) + x(n-1)c(1) + \dots + x(0)c(N-1),$$

which is identical to Equation (5.1).

However, the output  $y(n)$  is valid for each sample of  $x(n)$ , the larger the number of taps in the filter, the longer critical path in this architecture. Critical path is the longest path in the design, which usually starts from an output of a register and ends at the input of another register. Figure 5.3 shows how critical path is flown in the circuit. Here, main drawbacks of a long critical path in the design are listed: [45]

1. Maximum possible clock rate is computed considering slowest logic path produced by worst-case critical path in the design.
2. Dynamic power is increased.
3. Core temperature is increased.

In order to understand the effect of critical path, consider that each addition takes  $T_a$  and each multiplication takes  $T_m$  units of time in the above mentioned example. The total critical path, for a 3-tap filter is  $T_{crit} = T_m + 2 \times T_a$ . Thus, the sampling period or sampling frequency, according to the Nyquist sampling criterion, is given by



*Figure 5.3. Critical path flow in the design*

$$\begin{aligned}
 T_{sampling} &\geq T_{crit} = T_m + 2 \times T_a \\
 F_{sampling} &\leq \frac{1}{T_{sampling}} = \frac{1}{T_m + 2 \times T_a}
 \end{aligned} \tag{5.4}$$

If Equation (5.4) is expanded for N-tap FIR filter, Equation (5.5) will be given as:

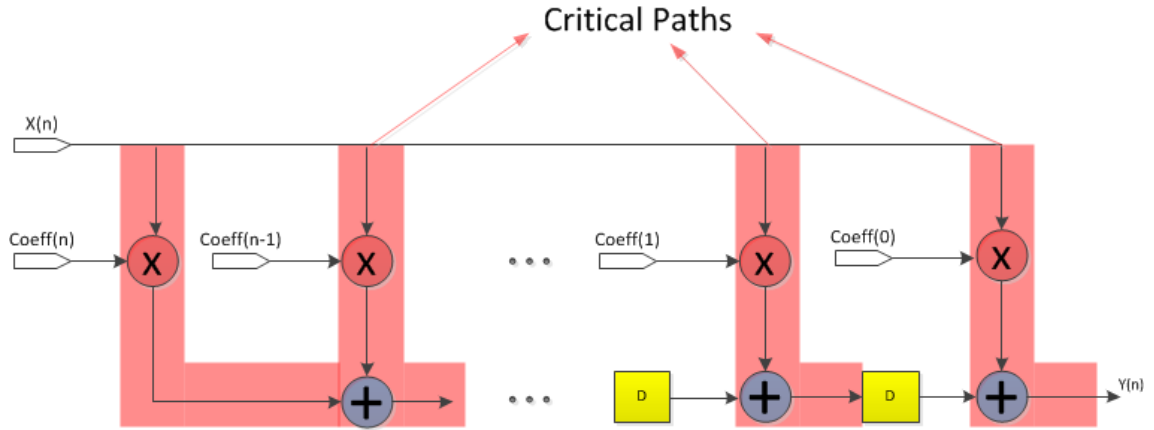
$$\begin{aligned}
 T_{sampling} &\geq T_{crit} = T_m + (N - 1) \times T_a \\
 F_{sampling} &\leq \frac{1}{T_{sampling}} = \frac{1}{T_m + (N - 1) \times T_a}
 \end{aligned} \tag{5.5}$$

Therefore, by altering the number of taps,  $T_{sampling}$  is increased and, consequently,  $F_{sampling}$  will be decreased which is not ideal in terms of digital circuit designs. However, there are few solutions such as pipelining and paralleling methods to mitigate critical path, all of them demanding additional cost as well as more complexity. Hence, in order to mitigate critical path problems, other architectures for FIR filter are proposed. However, the SDF architecture FIR filter has been designed in this project to compare its results with other FIR architectures.

### 5.2.2 Transposed FIR Filter Architecture

In order to mitigate critical path, in which SDF architecture FIR performs poorly, another variation of direct form FIR filter model named TDF architecture FIR filter





*Figure 5.4. Transposed direct form FIR filter architecture*

has been developed. In principle, TDF architecture FIR filters are structured based on the following modifications to the direct form FIR filter architecture [46].

- 1- Exchange input and output.
- 2- Put adders between delay elements.
- 3- Inverting direction of the signal flow.

Figure (5.4), illustrates the TDF architecture FIR filter and its corresponding critical paths. According to the figure, the longest critical path  $T_{crit}$ , irregardless of the number of taps, does not exceed  $T_m + T_a$  at any place due to the registers used between adder units. Therefore, TDF architecture can achieve higher throughput as well as lower power consumption, without any additional pipeline register. Thus, both  $T_{sampling}$  and  $F_{sampling}$  are constant given by Equation (5.6)

$$\begin{aligned} T_{sampling} &\geq T_{crit} = T_m + T_a \\ F_{sampling} &\leq \frac{1}{T_{sampling}} = \frac{1}{T_m + T_a} \end{aligned} \quad (5.6)$$

### 5.2.3 Parallel FIR Filter Architecture

PDF is another approach to mitigate critical path introduced in SDF. As Figure 5.5 shows, adder blocks are arranged in different levels of hierarchy. Depending on number of the taps, number of adder blocks increase in both vertical and horizontal directions. Thus, the total number of rows calculated by Equation (5.7) and, subsequently, the total number of adder units per row is given by Equation (5.8).

For best understanding of the concept of PDF, assume  $N = 8$ . Therefore, the total number of rows is equal to  $\lceil \log_2^8 \rceil = 3$  where the 1<sup>st</sup> row consists of  $(\frac{8}{2 \times 1} = 4)$ ,

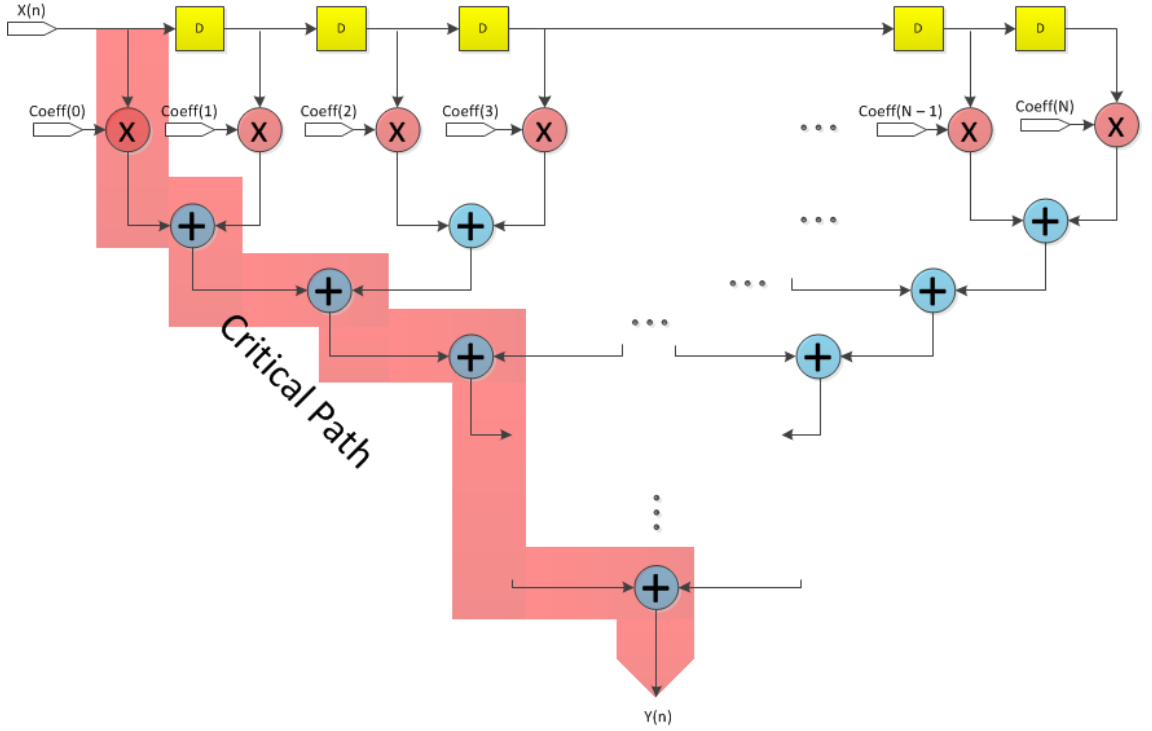


Figure 5.5. Parallel direct form FIR filter architecture

subsequently  $2^{nd}$  row composes of  $(\frac{8}{2 \times 2} = 2)$  and, eventually  $3^{rd}$  row includes  $(\frac{8}{2 \times 3} = 1)$  number of adder units. Figure 5.6 shows all these stages.

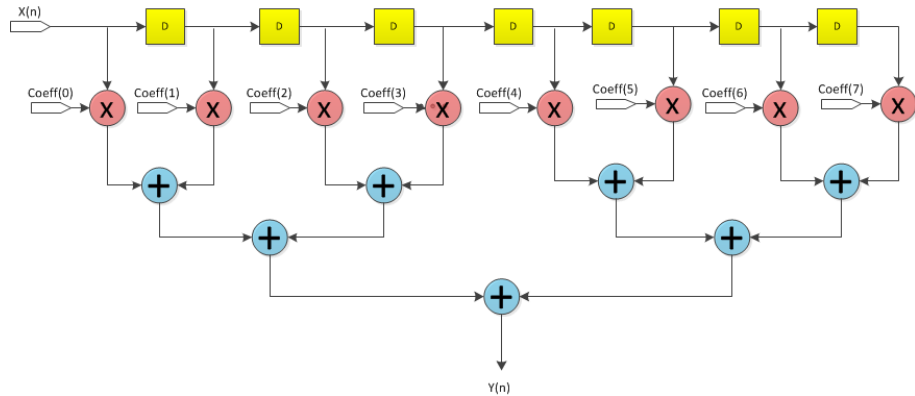


Figure 5.6. Parallel direct form FIR filter with  $N = 8$

$$Total\ Number\ of\ rows = \lceil \log_2^N \rceil \tag{5.7}$$

$$Total\ Number\ of\ adders\ per\ row = \left( \frac{N}{2 \times Row\ number} \right) \tag{5.8}$$

Therefore,  $T_{crit}$  is calculated as  $T_m + (\lceil \log_2^N \rceil \times T_a)$  and, consequently,  $T_{sampling}$  and  $F_{sampling}$  are given as

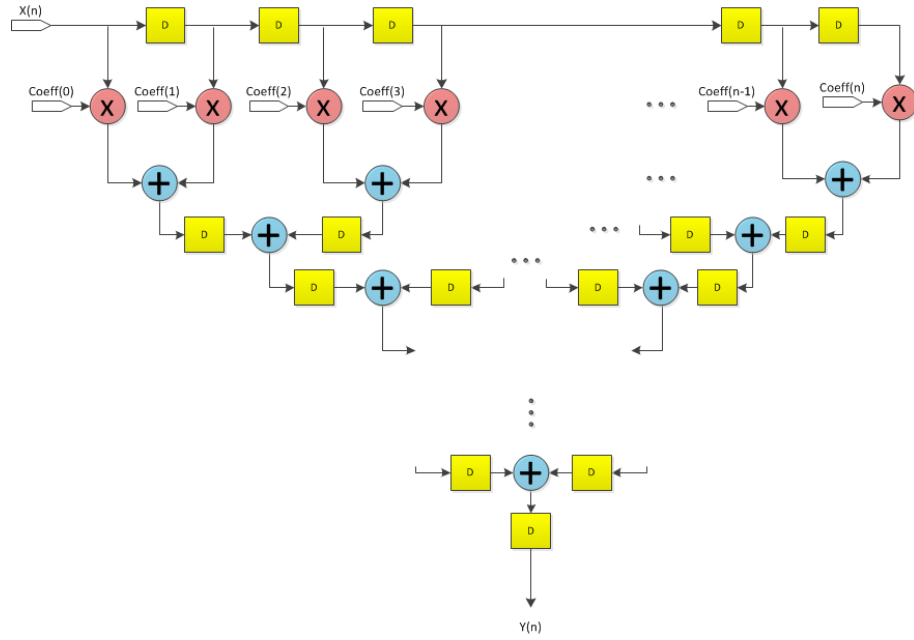


Figure 5.7. Pipelined direct form FIR filter with  $N = 8$

$$\begin{aligned}
 T_{\text{sampling}} &\geq T_{\text{crit}} = T_m + (\lceil \log_2^N \rceil \times T_a) \\
 F_{\text{sampling}} &\leq \frac{1}{T_{\text{sampling}}} = \frac{1}{T_m + (\lceil \log_2^N \rceil \times T_a)}
 \end{aligned} \tag{5.9}$$

Since in PDF architecture FIR filter the critical path is not the minimum possible, another improvement performed by employing *retiming* technique on critical path related to PDF. As Figure 5.7 shows, delay elements are inserted between each two adder units. In this case,  $T_{\text{crit}}$  is equal to that of Equation (5.4). It appears as if this scheme looks like a re-timed version of PDF, In this work, it will be referred as Retimed-Pipelined Direct Form (RPDF).

However the critical path is maintained at its minimum level, variety of D-Flip-Flop (DFF)s have been added to the design which increase the power consumption as well as chip area. The following section presents how the explained FIR filters can be implemented in hardware using Very-high-speed integrated circuit Hardware Description Language (VHDL).

### 5.3 Design Implementations

This project is done using VHDL as the hardware description language, simulated using ModelSim software, Compiled by Quartus-II version 12.1 environment and implemented on Altera Stratix-V family series (5SGSMD5K2F40C2N) development kit. The following section explains design elements.

### 5.3.1 Input Classifications

The aim of this project is to design a flexible timing synchronization scheme for CR applications. The core components of such synchronizer is to perform both cross-correlation and autocorrelation. Therefore, the synchronizer must be intelligently able to switch between crosscorrelation and autocorrelation in time. In other words, the presented synchronizer is able to perform *multi-correlation* on demand.

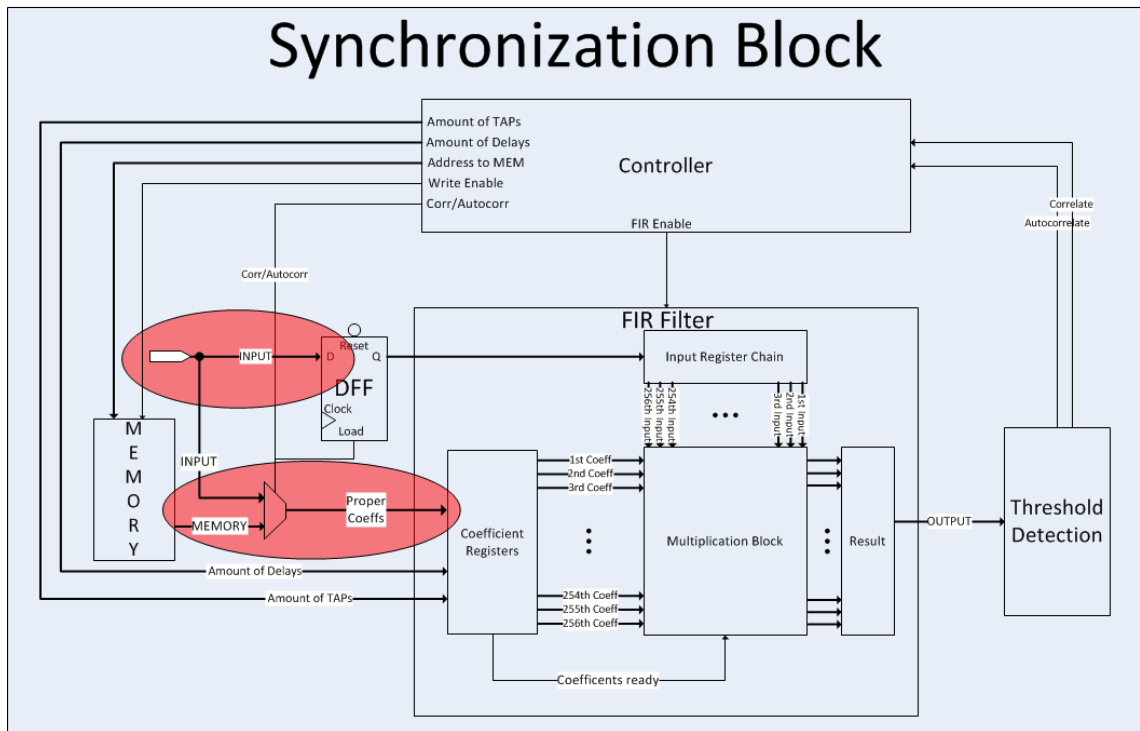


Figure 5.8. Synchronizer inputs

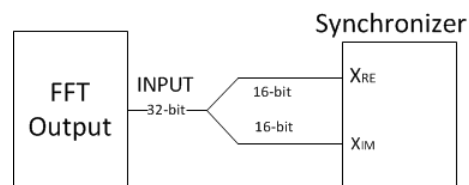


Figure 5.9. Splitting 32-bit signal into two 16-bit signals

According to Figure 5.8, there are two vectors of general input signal **INPUT** and **Proper Coeffs** fed to the FIR filter. In reality, as it is shown in Figure 5.9, a 32-bit signal **INPUT** is coming from FFT core. This 32-bit signal **INPUT** is split in two 16-bit sequences where the first 16-bit is the real part of the signal named  $X_{RE}$  and the second 16-bit is the imaginary part called  $X_{IM}$ . Henceforth,  $X_{RE}$  and  $X_{IM}$  are considered as complex input signals to the synchronizer block. However, for sake of simplicity and for keeping the figure understandable, only one of these two signals

is shown. Moreover, all the blocks and subblocks, except Multiplication subblock, are provided with CLOCK and ASYNCHRONOUS RESET which are respectively named "clk" and "rst\_n" in the design.

The second input labeled **Proper Coeffs** is fed to the FIR filter via a multiplexer. As the figure shows, the output of the multiplexer can be either **MEMORY** or **INPUT**. Indeed, based on performing crosscorrelation or autocorrelation, Proper Coeffs is connected to the MEMORY or INPUT, respectively. Note that in the code, the signal Proper Coeffs is split into two signals  $\text{Coeff}_{\text{RE}}$  and  $\text{Coeff}_{\text{IM}}$  representing the real and the imaginary parts, respectively.

In addition to these inputs, two "GENERIC" inputs **Amount of TAPs** and **Amount of DELAYs** are fed to the correlator, indicate the number of taps of the FIR filter (for performing correlation) and the number of delays of incoming signal (for performing autocorrelation).

### 5.3.2 Memory Block

The memory block considered for this design is an Static Random Access Memory (SRAM) initialized by a ".hex" file. Therefore, SRAM has a 32-bit output without any input port. Signal **Write Enable** is always set to '0', since there is no need to write to the memory cells. An 8-bit STD\_LOGIC\_VECTOR **Address to MEM** signal indicates which memory block should be loaded per clock cycle by taking its address into account. Since the maximum number of taps does not exceed 256, 8-bit is allocated to this signal. SRAM is employed to preserve frequency-domain representation of a predefined preamble. Thus, the values stored at the memory are used for performing crosscorrelation.

In order to initialize the SRAM by a ".hex" file, package TEXTIO corresponding to the library STD must be used as the header file. However, this library is known for ModelSim software to simulate the design, is not compatible during the compilation time with Quartus-II software. One solution which makes it possible to take the advantage of SRAM is to use the MEGA FUNCTION tools included in Quartus-II software.

MegaWizard Plug-In Manager assists to create and modify designs which contain custom variations, for example the aforementioned SRAM, using MEGA FUNCTION included in ALTERA library (Usually, is known as ALTERA\_MF). This wizard is able to create most of the common custom functional units such as different memory types, DSPs, arithmetic operations, Input/Output (I/O), etc. With respect to this, SRAM (created by the wizard) is capable of being initialized using ".hex" file as well.

MegaWizard Plug-In Manager is one of the most powerful tools included in Quartus-II software which is really helpful and handy for digital designers. How-

ever, the SRAM created by Quartus-II wizard is unknown for ModelSim due to using ALTERA\_MF library. Another solution is to switch between SRAMs created by VHDL code and the wizard while simulating and compiling, respectively. In other words, if the design needs to be simulated, SRAM written in VHDL must be used whereas during the compile time, SRAM created by the wizard must be applied.

### 5.3.3 Threshold Detection Block

There is one input and two output ports for Threshold Detection block. The input to this unit is a 16-bit signal labeled **OUTPUT** calculated by FIR filter. The decision is based on a comparison between  $|INPUT|^2$  and a predefined threshold level. Whenever the magnitude of the incoming signal is more than a threshold point, the Threshold Detection unit informs the controller to perform a proper action. There are two STD\_LOGIC output ports where the first one detects autocorrelation peak while the second one detects crosscorrelation peaks. Therefore two different threshold points can be set for the block.

### 5.3.4 Controller Block

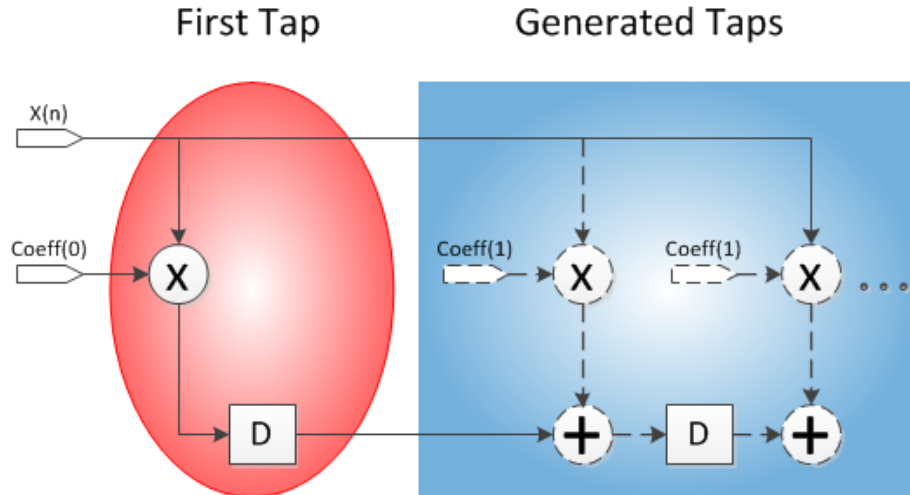
The Controller block is the most intelligent block of the design. One of the important responsibilities of this block is to make proper action based on received information by continuously monitoring the behavior of other subblocks. As it can be seen from Figure 5.8 as well, the Controller block is able to enable/disable FIR filter block as well as setting its input properly. If autocorrelation function is supposed to be performed, the Controller block routes the coefficient registers of the FIR filter to be fed by directing incoming signal, instead of flowing real coefficients held by the SRAM, via the multiplexer.

The Controller block is monitoring its input ports coming from the Threshold Detection block continuously. Any alteration through its inputs indicates that a peak has been found. Therefore, it changes the behavior of the FIR filter immediately. The Controller is able to prevent the input signal coming from FFT to be flown into the FIR filter for some purposes, for example power saving.

Moreover, the Controller block is capable of setting the number of taps as well as that of the delays at run time in the case that it is dynamically configured. This assists the FIR filter to reconfigure its taps based on the information provided by the Controller.

### 5.3.5 FIR Filter Core

As previously discussed about FIR filters in detail, different FIR schemes require different architectures. For example, SDF architecture FIR filter as well as PDF



**Figure 5.10.** Creating several copies of a process using *GENERATE* command.

architecture FIR filter require the input signal coming from FFT to be stored in some registers whereas in TDF architecture FIR filter scheme, there is no need to store incoming sequence inside the filter.

In principle, each FIR filter consists, at least, one tap, regardless of its architecture. There is a command in VHDL named *GENERATE* which creates several of copies of a process. FIR filters are usually created using *GENERATE* command. As Figure 5.11 depicts, once all subblocks, including multiplication block, coefficient register and delay element, created once, *GENERATE* command creates several copies of the same blocks. *GENERATE* command, usually, is used in a FOR loop whose iteration bound represents the number of generations.

Figure 5.11 represents a part of the code in which an  $N$ -tap TDF architecture FIR filter is created using *GENERATE* command, where  $N$  is denoted by generic `Num_of_TAP` signal. `temp_mul_RE` and `temp_mul_IM` are two *ARRAYs* of range  $(Num\_of\_TAP\ DOWNTO\ 0)\ OF\ STD\_LOGIC\_VECTOR\ (31\ DOWNTO\ 0)$  in which complex products of each tap are stored. However, both inputs to the multiplier, including the coefficient and the incoming signal, are 16-bit `STD\_LOGIC\_VECTOR`, the result must be saved in a 32-bit vector signal due to the multiplication nature from hardware aspect of view. Therefore, the product of  $N$ -bit signal with  $M$ -bit signal must be saved in a  $(N + M)$ -bit signal.

The *PROCESS* labeled `First_FIR` is creating first tap in which the temporary result is equal to the multiplication result. Moreover, *GENERATE* loop labeled `Other_FIRs`, creates other taps in which  $i$ th temporary result is the summation of  $i$ th multiplication result with  $(i - 1)$ th temporary result.

Note that the Coefficient Registers are a chain of DFFs whose inputs are fed by an *ARRAY* of range  $(Num\_of\_TAP\ DOWNTO\ 0)\ OF\ STD\_LOGIC\_VECTOR\ (15\ DOWNTO\ 0)$ . Each array block stores different values corresponding to either

```

First_FIR: PROCESS (clk, rst_n) -- CREATING A DFF
BEGIN
  IF rst_n = '0' THEN -- RESET ALL OUTPUTS
    temp_res_RE(0) <= (OTHERS => '0');
    temp_res_IM(0) <= (OTHERS => '0');

    ELSIF clk'EVENT AND clk = '1' THEN
      temp_res_RE(0) <= temp_mul_RE(0); -- Temporary result(0) = Multiplication result (0)
      temp_res_IM(0) <= temp_mul_IM(0); -- Temporary result(0) = Multiplication result (0)
    END IF;
  END PROCESS;
-----

Other_FIRs: FOR i IN 1 TO Num_of_TAP GENERATE

  PROCESS (clk, rst_n)
  BEGIN
    IF rst_n = '0' THEN
      temp_res_RE(i) <= (OTHERS => '0');
      temp_res_IM(i) <= (OTHERS => '0');
    ELSIF clk'EVENT AND clk = '1' THEN
      -- Representing Y(1) = X(1) + X(0)
      temp_res_RE(i) <= STD_LOGIC_VECTOR(SIGNED(temp_mul_RE(i)) + SIGNED(temp_res_RE(i-1)));
      temp_res_IM(i) <= STD_LOGIC_VECTOR(SIGNED(temp_mul_IM(i)) + SIGNED(temp_res_IM(i-1)));
    END IF;
  END PROCESS;
END GENERATE;

```

**Figure 5.11.** Part of the code where an  $N$ -tap TDF FIR filter are created using GENERATE command.

```

result_RE <= temp_res_RE(temp_res_RE'RIGHT)(31 DOWNT0 16);
result_IM <= temp_res_IM(temp_res_IM'RIGHT)(31 DOWNT0 16);

```

**Figure 5.12.** Truncation of 32-bit temporary result to 16-bit final result

the MEMORY or INPUT.

Eventually, after accumulating all the results, the final temporary result is truncated and copied to the OUTPUT. Figure 5.12 shows how truncation is done in this work. Signals  $temp\_res\_RE(temp\_res\_RE'RIGHT)(31\ DOWNT0\ 16)$  and  $temp\_res\_IM(temp\_res\_IM'RIGHT)(31\ DOWNT0\ 16)$  copy the last 16-bit of the corresponding MSB to the 16-bit  $res\_RE$  and  $res\_IM$  signals, respectively. The attribute "RIGHT" is the rightmost subscript of the array.

In a similar way, both SDF and PDF architecture FIR filters are implemented. Although, implementation of SDF architecture is simply achieved, PDF architecture has a complex implementation due to its tree hierarchy. Figure 5.13 shows a part of the code where an  $N$ -tap PDF architecture FIR filter are created using GENERATE command.

$CEIL\_LOG2(TO\_INTEGER(TO\_UNSIGNED(Num\_of\_TAP, 8)SRL1))$  is a PROCEDURE in which the magnitude of  $\lceil \log_2^N \rceil$  is calculated. The procedure is called from a separate PACKET. Note that in PDF architecture, the coefficient arrays is a 2-D array defined as:  $ARRAY(\lceil \log_2^N \rceil\ DOWNT0\ 0, (\frac{N}{2^{\lceil \log_2^N \rceil}})\ DOWNT0\ 0)$  OF INTEGER [47]. Theoretically, multi-dimension arrays can only be compiled in Quartus-II version 11.0 or later.



```

Generating_first_row : FOR i IN 0 TO (TO_INTEGER(TO_UNSIGNED(Num_of_TAP,8) SRL 1)) GENERATE
BEGIN
  Rows: PROCESS (clk, rst_n)
  BEGIN
    IF rst_n = '0' THEN
      temp_res_RE(0,i) <= 0;
      temp_res_IM(0,i) <= 0;
    ELSIF clk'EVENT AND clk = '1' THEN
      temp_res_RE(0,i) <= TO_INTEGER(SIGNED(temp_mul_RE(2*i)) + SIGNED(temp_mul_RE(2*i + 1)));
      temp_res_IM(0,i) <= TO_INTEGER(SIGNED(temp_mul_IM(2*i)) + SIGNED(temp_mul_IM(2*i + 1)));
    END IF;
  END PROCESS;
END GENERATE;

-----

Generating_Rows : FOR j IN 1 TO CEIL_LOG2(TO_INTEGER(TO_UNSIGNED(Num_of_TAP,8)SRL 1)) GENERATE
BEGIN
  Generating_Cols : FOR i IN 0 TO (TO_INTEGER(TO_UNSIGNED(Num_of_TAP,8) SRL j+1)) GENERATE
  BEGIN
    Parallel_adders: PROCESS (clk, rst_n)
    BEGIN
      IF rst_n = '0' THEN
        temp_res_RE(j,i) <= 0;
        temp_res_IM(j,i) <= 0;
      ELSIF clk'EVENT AND clk = '1' THEN
        temp_res_RE(j,i) <= temp_res_RE(j-1,TO_INTEGER(TO_UNSIGNED(i,7)SLL 1)) +
          temp_res_RE(j-1,(TO_INTEGER(TO_UNSIGNED(i,7)SLL 1)+1));
        temp_res_IM(j,i) <= temp_res_IM(j-1,TO_INTEGER(TO_UNSIGNED(i,7)SLL 1)) +
          temp_res_IM(j-1,(TO_INTEGER(TO_UNSIGNED(i,7)SLL 1)+1));
      END IF;
    END PROCESS parallel_adders;
  END GENERATE Generating_Cols;
END GENERATE Generating_Rows;

```

*Figure 5.13. Part of the code where an  $N$ -tap PDF FIR filter are created using GENERATE command.*

## Multiplier-Less Correlator

In a M-L correlator, the size of the correlator as well as the number of registers is depended on the number of input samples. According to [48] and [49], instead of performing 16-bit multiplication, a sign bit correlation between coefficients and input signal is done which can be simplified by a XNOR gate. Therefore, a massive number of registers and multiplications, which are the power-hungry operations in FIR filter due to their dynamic power consumption, are mitigated. Although the correlator uses only 1-bit instead of 16-bit, simulation results show that the performance of the FIR filter is still at a satisfactory level.

Aveek acknowledged M-L algorithm in [38] that the sign based correlator has satisfactory result even in 3dB SNR environments, which is the minimum SNR required to decode the lowest rate modulation. Intuitively, it works better in higher SNRs regions. By implementing both multiplier based correlator and a sign based correlator in MATLAB and tested both with 802.11g packets, The results were satisfactory. Based on that, in this work, a M-L based correlator implemented in hardware and the results are given in terms of summary, maximum frequency and

power consumption in the following section.

## 5.4 Compilation Results

Table 5.1 compares different synchronization schemes based on different FIR filter architectures in term of signal flow summary. For  $N = 256$ , the results show that M-L architecture is the best in all aspects whereas SDF is the worst one.

*Table 5.1. Summary of different synchronization blocks*

	<b>Available</b>	<b>TDF</b>	<b>SDF</b>	<b>PDF</b>	<b>RPDF</b>	<b>M-L</b>
<b>Logic utilization</b>	172,600	12,741	12,481	12,597	14,611	1,876
<b>Total pins</b>	864	68	68	68	68	68
<b>Total DSP blocks</b>	1,590	512	512	512	512	0
<b>Total registers</b>	-	16,561	16,869	16,764	25,265	2,886
<b>Maximum Freq (MHz)</b>	-	221.09	67.93	237.64	240.27	257.33

Power consumption analysis has been done based on results given by PowerPlay Power Analyzer Tool estimations. The analyzer directly reads the waveforms generated by the ModelSim software. Static probability and toggle rate for each signal are calculated based on the given Value Change Dump (VCD) file. VCD file contains signal activities and static probability information [50]. Table 5.2 shows the results estimated by PowerPlay Analyzer Tool. Table 5.3 shows power consumption for each cell. Here, the results show that M-L architecture achieved the best results almost in all aspects among the other architectures, whereas SDF is still the worst one in most of the cases.

*Table 5.2. Power consumption analysis*

<b>Power Consumption (mW)</b>	<b>TDF</b>	<b>SDF</b>	<b>PDF</b>	<b>RPDF</b>	<b>M-L</b>
<b>Total Thermal Power</b>	1365.46	3371.43	1133.33	1138.75	937.13
<b>Core Dynamic Thermal power</b>	283.20	200.59	153.86	160.10	9.42
<b>Core Static Thermal Power</b>	1056.36	1062.71	955.62	955.80	951.50
<b>I/O Thermal Power</b>	23.72	2108.14	22.86	22.86	26.20

## 5.5 Partial Reconfiguration

PR is the ability in which a portion of FPGA is reconfigured while the rest of the design is operating. In other words, a particular region of the design might have multiple personas while everything outside of this region is normally operating [51].

*Table 5.3. Flow summary of different synchronization blocks*

<b>Cells Power Consumption (mW)</b>	<b>TDF</b>	<b>SDF</b>	<b>PDF</b>	<b>RPDF</b>	<b>M-L</b>
<b>DSP block</b>	216.48	130.40	95.20	95.20	0
<b>Combinational cell</b>	15.43	10.78	1.28	1.30	1.87
<b>Register cell</b>	2.07	30.31	29.24	32.63	2.45
<b>I/O</b>	28.77	2090.09	4.31	4.23	7.44

PR has several applications and, generally, used in designs required to operate continuously while some particular regions can be reconfigured without disrupting the operating parts.

However, PR is a Beta feature in Quartus-II version 12.1 which only supports by Stratix-V family, additional license is required to make this feature available. Unfortunately, required licensed could not be obtained up to the date that this thesis is being done. However, the design is entirely ready to be uploaded on FPGA using PR feature.

Generally, PR region is configured using *configuration bits*. These bits are organized into frames representing a vertical area. When the PR is being performed, configuration data is send for the entire frame meaning that the area above and below the PR region is affected as well. This is done in two different modes called *SCRUB* and *AND/OR* mode.

As Figure 5.14 shows, in SCRUB mode all the bits in an entire column are reconfigured. Therefore, it is not possible to determine either the above PR or the below PR regions are being reconfigured. Hence, SCRUB mode can not be employed when there is an overlap between two PR regions.

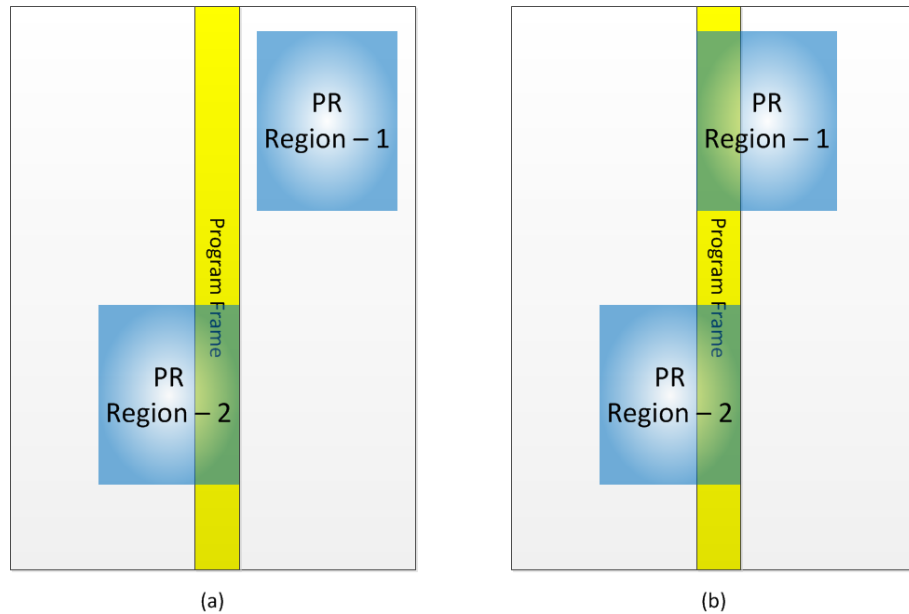
In AND/OR mode, the bits that don't need to be reconfigured are ANDed with a mask while rest of the bits are ORed. Therefore, this method is useful when multiple PR regions have overlap with each other.

In order to perform PR, following steps must be prepared beforehand:

- Partitioning for PR
- Wrapper Logic
- Freeze Logic

### 5.5.1 Partitioning for Partial Reconfiguration

In Quartus-II software, both PR regions and static regions must be differentiated from each other. This can be achieved by setting *LogicLock* Assignments as well as *Design Partitions* for different regions. Once they are done, *Revisions* manages



**Figure 5.14.** Configuring data flow (a) SCRUB and (b) AND/OR modes

multiple personas for all the PR regions. In other words, different source files corresponding to different personas can be managed using revisions.

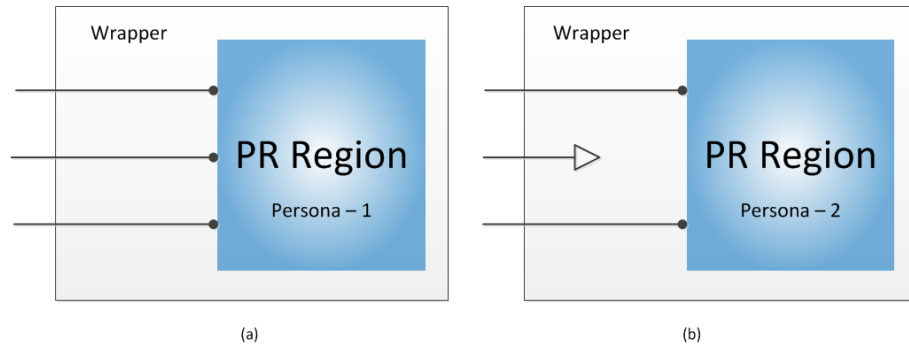
Note that only core logics, including Logic Array Blocks (LABs) consist of registers and Look Up Table (LUT)s, Random Access Memory (RAM) blocks and DSP blocks, can be reconfigured using PR. Peripheral blocks, including transceivers and Phase-Locked Loop (PLL), can not be reconfigured using PR feature. They might be reconfigured using different reconfiguration methods such as dynamic reconfiguration.

### 5.5.2 Wrapper Logic

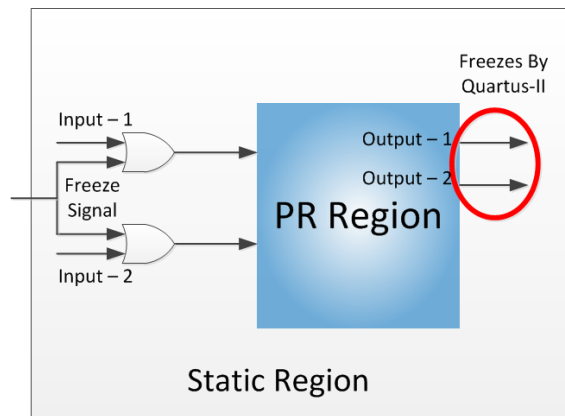
When a PR region has multiple personas, each of which has different functionalities. However, all PR regions must be interacted with the static region with a certain set of signals, some of the personas might need more ports to be able to interact with the static region.

A wrapper logic is an additional file whose extra ports, which are called Dummy Ports, cause all the personas look alike each other. Figure 5.15 shows how a wrapper logic can cause multiple personas with different required ports interact with static region.

Since all the personas require same set of ports in this project, wrapper logic file has been ignored while creating required steps.



**Figure 5.15.** Wrapper logic scheme for (a) Persona-1 uses all three ports and (b) persona-2 uses two ports



**Figure 5.16.** Schematic of a freeze logic

### 5.5.3 Freeze Logic

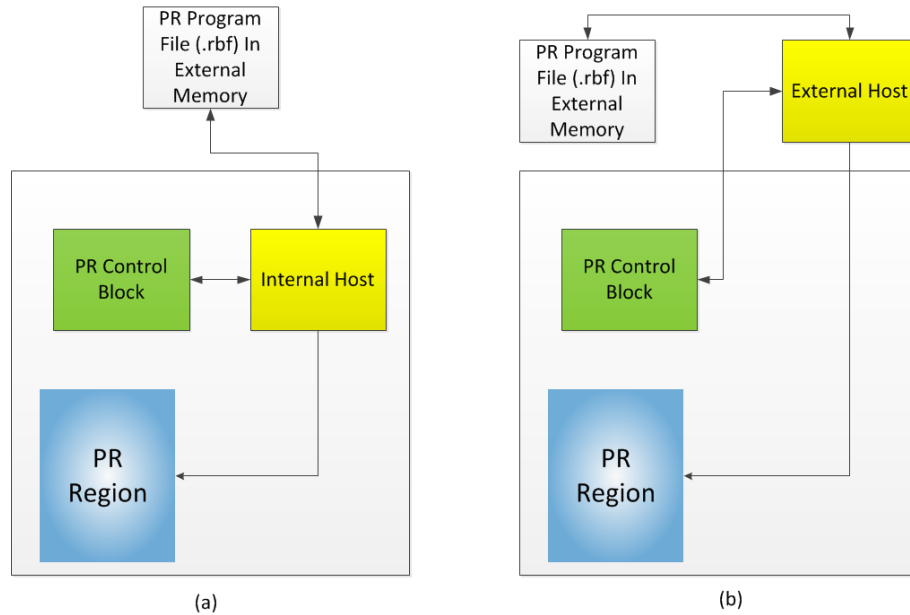
When a PR region is being reconfigured, any alteration in I/Os might cause contention. Therefore, it must be ensured that the I/Os between PR and static regions are constant during the reconfiguration process. This can be referred to what is called *Freezing*.

Usually, freezing is done by a region surrounding PR region in which all the input signals to the PR region are frozen before PR being started. This means setting all the inputs to the PR region to '1' while reconfiguration process is ongoing.

Moreover, during the PR process, static region should be independent to PR region. In order to do so, Quartus-II freezes all the outputs from PR region to static region automatically. Figure 5.16 shows how the freeze logic sets the inputs to the PR region to '1'.

### 5.5.4 Partial Reconfiguration Host

PR host is an extra logic created by the user themselves. This host is responsible to execute PR by communicating with hard PR control block on the FPGA. Usually,



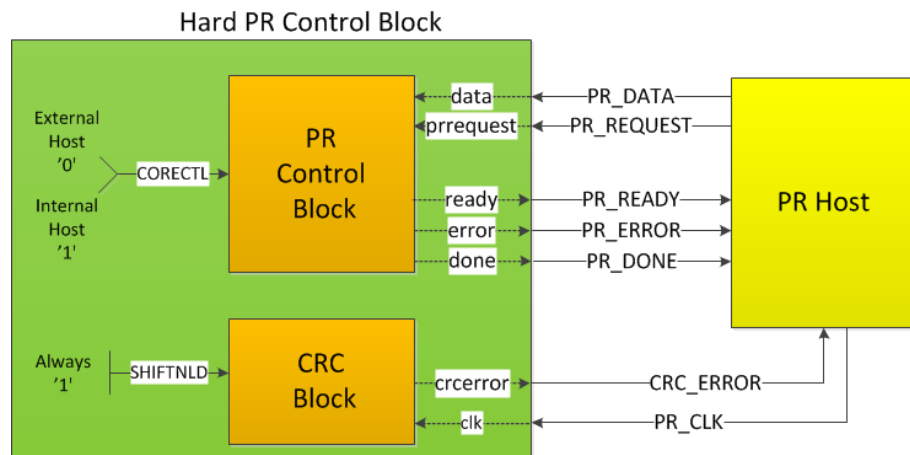
**Figure 5.17.** Representation of (a) internal host and (b) external host

PR host are implemented as state machines in either *internal* or *external* modes. As Figure 5.17 illustrates, internal host is implemented inside the FPGA whereas external hosts are implemented outside of the FPGA which communicate via the FPGA pins.

Before the PR process is being performed, the host sends the configuration bitstreams to the controller as well as issuing the freeze signal to the PR region. After the PR process is done, host is responsible to unfreeze and then resets the PR region. Note that configuration bitstreams are created by the Quartus-II software. Therefore, the user is not involved to even manipulate those bitstreams.

According to the Figure 5.18, there are seven predefined signals and two main hard blocks which need to be connected to the host. Predefined signals are listed as follows:

- *PR\_DATA* [15 to 0]: Contains bitstreams sent to the control block.
- *PR\_CLK*: Synchronize clock for *PR\_DATA*, supports up to 80 MHz frequency.
- *PR\_REQUEST*: Indicates PR host is requesting PR.
- *PR\_READY*: Indicates control block is ready for PR
- *PR\_ERROR*: Indicates any error during PR process.
- *PR\_DONE*: Indicates PR has done successfully.
- *PR\_DONE*: Indicates PR has done successfully.



*Figure 5.18. How the PR host can be connected to the hard PR control block*

Both PR control block and Cyclic Redundancy Check (CRC) block are needed to be instantiated in the design. PR control block controls the PR process inside the device and CRC block checks the CRC of the partial bitstream and flag errors. CORECTL signal should be connected to either  $V_{CC}$  or GND depending on whether the PR is going to be performed via the core or FPGA pins, respectively.

With respect to the CRC block, Signal SHIFTNLD must always be set to '1' and depends on performing PR either from core or external host, CRC\_ERROR signal should be connected to Hardware Description Language (HDL) code or FPGA pin, respectively. [51]

In this project it has been assumed that the PR host has been chosen to be inside the FPGA, hence, the CORECTL has been set to  $V_{CC}$  or '1'. PR\_REQUEST is managed using one of user defined push buttons, therefore it requires an additional debounce controller to manage the push button behavior. The debouncer is implemented as an state machine which checks the pressed button twice to ensure that the action is made intentionally. Eventually, CRC\_ERROR, PR\_ERROR and PR\_DONE notify user by turning on corresponding on-board Light-Emitting Diodes (LEDs) of the Stratix-V development kit.

## 6. CONCLUSIONS

In this thesis, a flexible timing synchronization scheme for Cognitive Radio application was developed. The synchronization block is able to be employed in both OFDM and NC-OFDM based Cognitive Radios. The major functionality of the synchronizer is based on performing crosscorrelation and autocorrelation functions in frequency domain between incoming signal with predefined preamble stored in an internal memory and incoming signal with a delayed version of itself, respectively.

Synchronizer is able to perform both crosscorrelation and autocorrelation functions in one functional block which can be implied as a multicorrelator. In other words, the proposed synchronizer is capable of performing multicorrelation on demand.

Moreover, synchronizer has been implemented using Partial Reconfiguration feature, Hence, it is capable of reconfiguring some portions of design during runtime without disrupting operating parts. In reality, this is the fundamental demand in context of Cognitive Radios.

In general, a wireless standard such as IEEE-802.11 can be simply replaced by IEEE-802.16 (dedicated for WiMAX) or IEEE-802.22 (assigned to TV-band), since synchronizer can adopt different patterns using Partial Reconfiguration which only needs to freeze the PR region, transmits new configuration bitstreams and then releases the PR region and resets it afterward.

The core content of the synchronizer, in terms of architecture, was based on FIR filters. Therefore, different structures considered to implement FIR filter including Sequential Direct Form, Transposed Direct Form, Pipelined Direct Form, Retimed-Pipelined Direct Form as well as Multiplier-Less architecture.

This project was done based on ModelSim software as the simulator and Quartus-II environment to compile the code. Simulation results showed that in terms of the same inputs as well as coefficients, multicorrelator with different architectures behaved the same and the final results were exactly matched to each other. Therefore, considering final results, there was no difference to select each architecture.

According to [38], MATLAB results had shown that the Multiplier-Less architecture had satisfactory results, even in environments with low SNR strength. In this project, compilation results inferred that the best architecture to be employed as FIR filter is, again, Multiplier-Less in terms of maximum frequency, chip area,



dynamic power consumption, etc compared with other architectures.

In contrast to Multiplier-Less architecture, the worst case architecture is Sequential Direct Form in, almost, all of the cases. In other words, Sequential Direct Form architecture was not recommended to be employed in any condition.

Remaining architectures had almost the same characteristics. However, Sequential Direct Form was a better selection in terms of chip area, it was slower in operating along with more power consumption in comparison with other architectures. Although differences were trivial, there was always a trade off between different architectures.

As experiences showed, the simplest architecture in terms of implementation, excluding Multiplier-Less architecture, belonged to Transposed Direct Form and, subsequently, Sequential Direct Form, whereas the most complex ones were Retimed-Pipelined Direct Form and Pipelined Direct Form, respectively.

## REFERENCES

- [1] A. M. Wyglinski, M. Nekovee and T. Hou, "Cognitive Radio Communications and Networks", 2010, Elsevier Inc., 714 p.
- [2] Joseph Mitola III, "Cognitive Radio Architecture. The Engineering Foundations of Radio XML", 2006, John Wiley and Sons Inc., 473 p.
- [3] Koski, E.; Furman, W.N., "Applying cognitive radio concepts to HF communications", The Institution of Engineering and Technology 11th International Conference on Ionospheric Radio Systems and Techniques (IRST), pp.1,6, 28-30 April 2009
- [4] Man-On Pun, M. Morelli, C-C Jay Kuo, "Communications and Signal Processing", Vol 3, Multi- Carrier Techniques for Broadband Wireless Communications. A Signal Processing Perspective, London, 2007, Imperial College Press, 257 p.
- [5] M. S. Alencar, V. C. DA ROCHA JR, "Communication Systems", United State of America, 2005, Springer Science+Business Media Inc., 415 p.
- [6] T. Chiueh, P. Tsai, I. Lai, "Baseband Receiver Design for Wireless MIMO-OFDM Communications", 2nd Edition, 2012, Wiley-IEEE Press, 346 p.
- [7] A. F. Molisch, "Wireless Communication", 2nd Edition, 2011, John Wiley and Sons Ltd., 816 p.
- [8] J. Heiskala, J. Terry, "OFDM Wireless LANs: A Theoretical and Practical Guide", 2001, Sams Publishing, 315 p.
- [9] "A Fiber Optics Primer" [WWW]. [Accessed on 06.10.2013]. Available at [http://home.roadrunner.com/~lifetime/mm-fiber\\_optics\\_primer.htm](http://home.roadrunner.com/~lifetime/mm-fiber_optics_primer.htm)
- [10] bin Senawi, A.S.; bin Sha'ameri, A.Z., "Performance evaluation of multi-carrier modulation techniques for HF frequency selective fading channel", Proceedings of 4th National Conference on Telecommunication Technology (NCTT), 2003, pp.151,154, 14-15 Jan. 2003
- [11] K. M. Bobrowski, "Practical Implementation Consideration for Spectrally Agile Waveforms in Cognitive Radio", M.Sc. thesis, Worcester, 2009, Worcester Polytechnic Institute, 119 p.
- [12] E. Perahia, R Stacey, "Next Generation Wireless LANs 802.11n and 802.11ac", 2nd Edition, United Kingdom, 2013, Cambridge University Press, 452 p.

- [13] "Advantages and Disadvantage of OFDM" [WWW]. [Accessed on 16.10.2013]. Available at <http://sna.csie.ndhu.edu.tw/~cnyang/MCCDMA/tsld021.htm>
- [14] Rajbanshi, R.; Wyglinski, Alexander M.; Minden, G.J., "Adaptive-Mode Peak-to-Average Power Ratio Reduction Algorithm for OFDM-Based Cognitive Radio", IEEE 64th Vehicular Technology Conference, pp.1,5, 25-28 Sept. 2006
- [15] Shulan Feng; Zheng, H.; Haiguang Wang; Jinnan Liu; Zhang, P., "Preamble Design for Non-Contiguous Spectrum Usage in Cognitive Radio Networks", IEEE Wireless Communications and Networking Conference (WCNC), pp.1, 6, 5-8 April 2009
- [16] E. Hossain, V. Bhargava , "Cognitive Wireless Communication Networks", 2007, Springer Science + Business Media, 440 p.
- [17] A. Jadhav, "OFDM as downlink transmission scheme for LTE", [WWW], WirelessCafe, 2009, [Accessed on 18.10.2013], Available at <http://wirelesscafe.wordpress.com/2009/04/20/ofdm-as-downlink-transmission-scheme-for-lte/>
- [18] M. Z. Parvez, Md. A. Al Baki, "Peak To Average Power Ratio (PAPR) Reduction in OFDM Based Radio Systems", M.Sc. thesis, Blekinge, 2010, Blekinge Institute of Technology, 67 p.
- [19] R. Airoidi, "Design and Implementation of Software Defined Radios on a Homogeneous Multi-processor Architecture", PhD. thesis, ,Tampere, Finland, 2013, Tampere University of Technology, 124 p.
- [20] Sadiku, M. N O; Akujuobi, C.M., "Software-defined radio: a brief overview", IEEE Potentials, vol.23, no.4, pp.14,15, Oct.-Nov. 2004
- [21] Haykin, Simon, "Cognitive radio: brain-empowered wireless communications", IEEE Journal on Selected Areas in Communications, vol.23, no.2, pp.201,220, Feb. 2005.
- [22] H. Arslan, "Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems", 2007, Springer, 469 p.
- [23] J. Polson, "Cognitive Radio Application in Software Defined Radio", Proceeding of the SDR 04 Technical Conference and Product Exposition, SDR forum, 2004.
- [24] Schmidl, T.M.; Cox, D.C., "Robust frequency and timing synchronization for OFDM", IEEE Transactions on Communications, vol.45, no.12, pp.1613,1621, Dec 1997.

- [25] E. Perahia, R. Stacy, "Next Generation Wireless LANs 802.11n and 802.11ac", 2nd edition, 2013, Cambridge University Press, 452 p.
- [26] "802.11a White Paper", [WWW], Vocal Technologies Ltd., 2012, [Accessed on 26.10.2013], Available at [http://www.vocal.com/wp-content/uploads/2012/05/80211a\\_wp1pdf.pdf](http://www.vocal.com/wp-content/uploads/2012/05/80211a_wp1pdf.pdf)
- [27] Jafar, S.A.; Srinivasa, S., "Capacity Limits of Cognitive Radio with Distributed and Dynamic Spectral Activity", IEEE International Conference on Communications (ICC), vol.12, pp.5742,5747, June 2006
- [28] Li Li; Daiming Qu; Tao Jiang; Jie Ding, "Design of LDPC codes for non-contiguous OFDM-based communication systems", IEEE International Conference on Communications (ICC), pp.4712,4716, 10-15 June 2012
- [29] Biao Huang; Jun Wang; Wanbin Tang; Shaoqian Li, "An effective synchronization scheme for NC-OFDM systems in cognitive radio context", IEEE International Conference on Wireless Information Technology and Systems (ICWITS), pp.1,4, Aug. 28 2010-Sept. 3 2010
- [30] Jinnan Liu; Shulan Feng; Haiguang Wang, "Comb-Type Pilot Aided Channel Estimation in Non-Contiguous OFDM Systems for Cognitive Radio", 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom), pp.1,4, 24-26 Sept. 2009
- [31] Xiao Zhou; Runhe Qiu, "An adaptive synchronization algorithm for Non-Contiguous OFDM cognitive radio systems", IET International Communication Conference on Wireless Mobile and Computing (CCWMC 2011), pp.102,106, 14-16 Nov. 2011
- [32] Wyglinski, Alexander M., "Effects of Bit Allocation on Non-Contiguous Multicarrier-Based Cognitive Radio Transceivers", IEEE 64th Vehicular Technology Conference, pp.1,5, 25-28 Sept. 2006
- [33] Rajbanshi, R.; Wyglinski, Alexander M.; Minden, G.J., "An Efficient Implementation of NC-OFDM Transceivers for Cognitive Radios", 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications, pp.1,5, 8-10 June 2006
- [34] Zhou Yuan; Pagadarai, S.; Wyglinski, Alexander M., "Feasibility of NC-OFDM transmission in dynamic spectrum access networks", IEEE Military Communications Conference (MILCOM), pp.1,5, 18-21 Oct. 2009

- [35] Daiming Qu; Jie Ding; Tao Jiang; XiaoJun Sun, "Detection of Non-Contiguous OFDM Symbols for Cognitive Radio Systems without Out-of-Band Spectrum Synchronization", IEEE Transactions on Wireless Communications, vol.10, no.2, pp.693,701, February 2011
- [36] Jae Yeon Won; Hyun Gu Kang; Yun Hee Kim; Ickho Song; Myung-Sun Song, "Fractional Bandwidth Mode Detection and Synchronization for OFDM-Based Cognitive Radio Systems", IEEE Vehicular Technology Conference (VTC), pp.1599,1603, 11-14 May 2008
- [37] A. Dutta, D. Saha, D. Grunwald, D. Sicker. "Practical Implementation of Blind Synchronization in NC-OFDM based Cognitive Radio Networks", Proceedings of the 2010 ACM workshop on Cognitive radio networks, pp. 1-6, 2010
- [38] Saha, D.; Dutta, A.; Grunwald, D.; Sicker, D., "Blind synchronization for NC-OFDM - When "channels" are conventions, not mandates", IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN), pp.552,563, 3-6 May 2011
- [39] "White Space Database Administrators Guide", [WWW], Federal Communications Commission, [Accessed on 29.10.2013], Available at <http://www.fcc.gov/encyclopedia/white-space-database-administrators-guide>
- [40] Huang, W-K; Lombardi, F., "An approach for testing programmable/configurable field programmable gate arrays", Proceedings of 14th VLSI Test Symposium, pp.450,455, 28 Apr-1 May 1996
- [41] "DSP Development Kit, Stratix V Edition", [WWW], Altera Corporation, [Accessed on 31.10.2013], Available at <http://www.altera.com/products/devkits/altera/kit-stratix-v-dsp.html#documentation>
- [42] "DSP Development Kit, Stratix V Edition Reference Manual", [WWW], Altera Corporation, July 2012, [Accessed on 31.10.2013], Available at [http://www.altera.com/literature/manual/rm\\_svgs\\_dsp\\_dev\\_board.pdf](http://www.altera.com/literature/manual/rm_svgs_dsp_dev_board.pdf)
- [43] Mahesh, R.; Vinod, A.P., "New Reconfigurable Architectures for Implementing FIR Filters With Low Complexity", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.29, no.2, pp.275,288, Feb. 2010
- [44] Ghosh, D.; Sharma, Deepak; Aziz, A., "A Novel Low Area and High Performance Programmable FIR Filter Design Using Dynamic Random Access Memory", 48th Midwest Symposium on Circuits and Systems, pp.1477,1480 Vol. 2, 7-10 Aug. 2005

- [45] Peng Li, "Critical Path Analysis Considering Temperature, Power Supply Variations and Temperature Induced Leakage", 7th International Symposium on Quality Electronic Design (ISQED), pp.6 pp., 259, 27-29 March 2006
- [46] T.Saramäki, Digital Linear Filtering I, Tampere, 2012, Tampere University of Technology, Lecture slide, 98 p.
- [47] Z. Navabi, " VHDL: Modular Design and Synthesis of Cores and Systems.", 3rd edition, 2007, McGraw-Hill Companies, 531 p.
- [48] Pham, T.H.; Fahmy, S.A.; McLoughlin, I.V., "Low-Power Correlation for IEEE 802.16 OFDM Synchronization on FPGA", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.21, no.8, pp.1549,1553, Aug. 2013
- [49] Diaz, I.; Wilhelmsson, L.; Rodrigues, J.; Lofgren, J.; Olsson, T.; Owall, V., "A Sign-Bit Auto-Correlation Architecture For Fractional Frequency Offset Estimation in OFDM", Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp.3765,3768, May 30 2010-June 2 2010
- [50] "PowerPlay Power Analysis Quartus-II Handbook Version 13.0", Vol. 3, November 2012, Altera Corporation, 28 p.
- [51] "Design Planning for Partial Reconfiguration Quartus-II Handbook Version 13.0", Vol. , May 2013, Altera Corporation, 46 p.