國 立 交 通 大 學

資訊工程系
碩 士 論 文

在雙模OFDM和DSSS無線網路下 低取樣率之封包同步器的設計與分析

Design and Analysis of Low Sampling Rate Packet Synchronizer in Dual OFDM/DSSS Wireless LAN

研 究 生：郭錦華

指導教授：許騰尹 教授

中 華 民 國 九 十 四 年 七 月

在雙模OFDM和DSSS無線網路下
低取樣率之封包同步器的設計與分析
Design and Analysis of Low Sampling Rate Packet Synchronizer in Dual OFDM/DSSS Wireless LAN

研 究 生：郭錦華　　Student : Jin-Hwa Guo

指導教授：許騰尹　　Advisor : Terng-Yin Hsu

國 立 交 通 大 學

資 訊 工 程 系

碩 士 論 文

A Thesis

Submitted to Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science and Information Engineering

July 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年七月

在雙模OFDM和DSSS無線網路下 低取樣率之封包同步器的設計與分析

學生：郭錦華　　指導教授：許騰尹 博士

國立交通大學資訊工程學系　研究所碩士班

摘要

在現代無線通訊系統中，直接序列展頻(DSSS)和正交頻率多重分割(OFDM)被廣泛的使用。有越來越多現實中的系統和規格，例如 IEEE 802.11g，將這兩種調變整合到同一個作業平台上。然而，不同的調變方式所佔據頻寬不同，導致需要兩套不同取樣頻率的類比/數位轉換器(ADC)，來達成偵測和同步所收到的訊號。

在這份論文中，將架構以 IEEE 802.11g 爲基準的參考平台。首先提出的接收器架構，只包括一套類比/數位轉換器和動態取樣來降低硬體成本。全數位鎖相迴路(ADPLL)所動態提供的取樣頻率和相位可由基頻處理器(base band processor)來控制，以達成最佳的符號時序(symbol timing)。接下來，在 DSSS/OFDM 共存系統中的封包偵測(packet detection) 演算法，以及 OFDM 符號時序估計演算法將被展現。並且分析封包漏失 (packet loss)和錯誤宣告(false alarm)的機率。最後，以．13 微米的製程實作一顆晶片，以驗証其可行性。

Design and Analysis of Low Sampling Rate Packet Synchronizer in Dual OFDM/DSSS Wireless LAN

Student: Jin-Hwa Guo Advisor: Dr. Terng-Yin Hsu

Department of Computer Science and Information Engineering,
National Chiao Tung University

Abstract

Direct Sequence Spreading Spectrum (DSSS) and Orthogonal Frequency Division Multiplexing (OFDM) are widely used in modern wireless communications systems. There are more and more practical systems and standards, such as IEEE 802.11g, integrating these two modulation schemes into one operating platform. Different modulation systems, however, occupy different bandwidth, whereas two sets of ADCs with two sampling rate have to be used to detect and synchronize the received signals.

In this thesis, a reference design based on IEEE 802.11g is constructed. First, the proposed receiver architecture consists of only one set of ADCs with dynamic sampling in order to reduce hardware cost. The sampling frequency and phase from ADPLL can be controlled by the base band processor and thus approaches the optimum symbol timing. Furthermore, a packet detection algorithm for the coexistence of DSSS/OFDM system, and an algorithm for OFDM symbol timing estimation are presented. The probabilities of packet loss and false alarm are analyzed. Finally, a test chip using 0.13 um technology is implemented to verify the feasibility.

Table of Contents

page

List of Figures

page

Chapter 1

Introduction

Orthogonal Frequency Division Multiplexing (OFDM) and Direct Sequence Spreading Spectrum (DSSS) are widely used in modern wireless communications systems [1-4]. There are more and more practical systems and standards integrating these two modulation schemes into one operating platform. Different modulation systems, however, occupy different bandwidth conventionally, and use two sampling rate to detect and synchronize the received signals. Conventionally, there are more than two sets of ADCs for I channel and Q channel existing in some receiver designs [5]. Some of the others, use much higher clock rate for ADC sampling, and obtain the other sampling rate by interpolator, as depicted in figure 1.1.

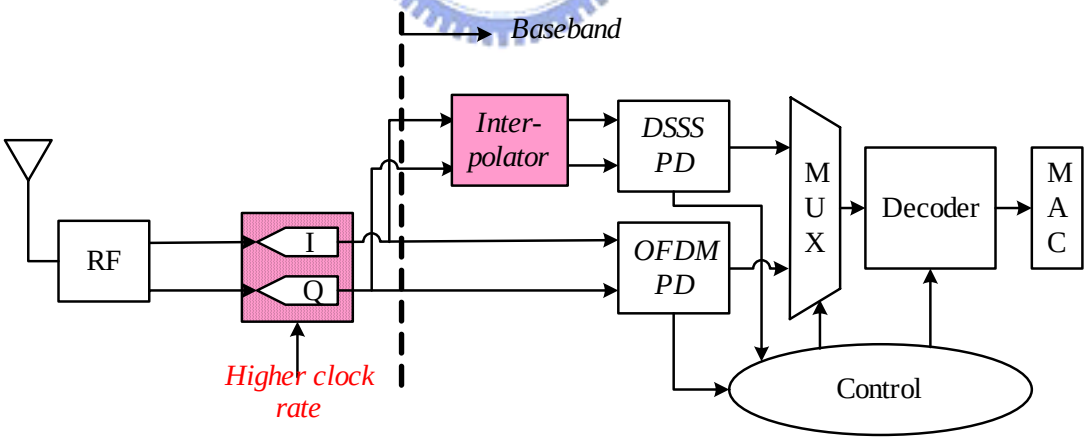

Figure 1.1 Packet synchronizer with interpolator

The higher clock sampling and interpolator consumes power and hardware area. To reduce sampling clock, we proposed a packet synchronizer in figure 1.2. Both modulations use only one set of ADCs and ADPLL with dynamic sampling

techniques. Furthermore, for low power issues, the initial ADC sampling rate is only half of OFDM sampling rate and is also smaller than the DSSS sampling rate

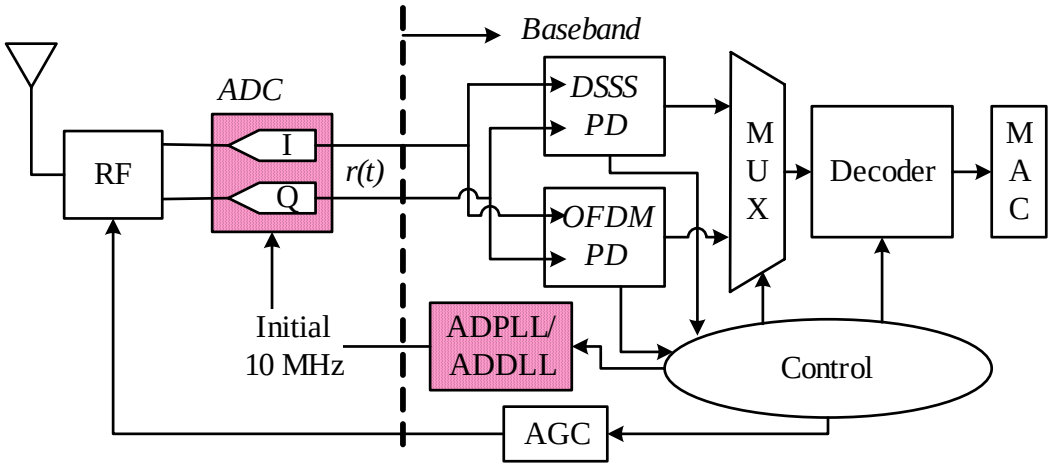

Figure 1.2 Proposed packet synchronizer with dynamic sampling

Even if the different sampling rate issues is solved, the separate OFDM and DSSS packet detection algorithms will not work jointly; the false-alarm rate is still high. The penalties for false-alarm in dual-mode system are much higher than in single-mode system, since we will miss the desired packet. To prevent from high false-alarm rate, a dual-mode OFDM/DSSS packet synchronizer is proposed.

This thesis is organized as follows. Chapter 2 introduces the basics of OFDM/DSSS systems and wireless channel model. Chapter 3 presents the detection and acquisition algorithms and the threshold decision. Chapter 4 discusses the Matlab simulation results under different conditions. Chapter 5 emphasizes the hardware architecture and shows the implementation results. Chapter 6 makes the conclusion and future work.

Chapter 2

OFDM/DSSS System and Channel Model

In this chapter, we are going to describe tree block of wireless communications, transmitter, channel model, and receiver. At first, we introduce the basic of OFDM and DSSS modulations. And then we present the IEEE 802.11g PHY specification, which combines both 802.11a (OFDM) and 802.11b (DSSS) at one system, and the IEEE 802.11g PHY transmitter block diagram. After that, we characterize some wireless channel models and parameters, such as Additive White Gaussian Noise (AWGN), carrier frequency offset (CFO), multi-path, and so on. Finally, in order to achieve robust synchronizer, we propose a universal receiver system model.

2.1 The basics of OFDM

Orthogonal Frequency Division Multiplexing (OFDM) is a multi-carrier modulation that achieves high data rate and combat multi-path fading in wireless networks. The main concept of OFDM is to divide available channel into several orthogonal sub-channels. All of the sub-channels are transmitted simultaneously, thus achieve a high spectral efficiency. Furthermore, individual data is carried on each sub-carrier, and this is the reason the equalizer can be implemented with low complexity in frequency domain.

2.3 The basics of DSSS

Direct Sequence Spread Spectrum (DSSS) is a spectrum technique whereby the data signals are modulated with spreading code. The principle of Direct Sequence is to spread a signal on a larger frequency band by multiplexing it with a signature or code to minimize localized interference and background noise. To spread the signal, each bit is modulated by a code. In the receiver, the original signal is recovered by receiving the whole spread channel and demodulating with the same code used by the transmitter. A fundamental issue in spread-spectrum systems is how much protection spreading can provide against interfering signals with finite power. Spread-spectrum techniques distribute a relatively low-dimension signal in a large dimensional signal.

2.3 IEEE 802.11g PHY specification

The 802.11g PHY defined in standard is known as the Extended Rate PHY (ERP), operating in the 2.4 GHz ISM band. Four operational modes are listed as followed:

A. ERP-DSSS/CCK – This mode builds on the payload data (PSDU) rates of 1, 2, 5.5, and 11 Mbit/s that use DSSS (DBPSK and DQPSK), CCK and optional PBCC modulation, and the PLCP Header operates on data rate 1 Mbit/s DBPSK for Long SYNC, 2 Mbit/s DQPSK for Short SYNC. Figure 2.1 shows the format for the interoperable PPDU that is the same with 802.11b PPDU format, and the details of components such as spreading code, scrambler, CRC implementation, and modulation refer to 802.11b standard.

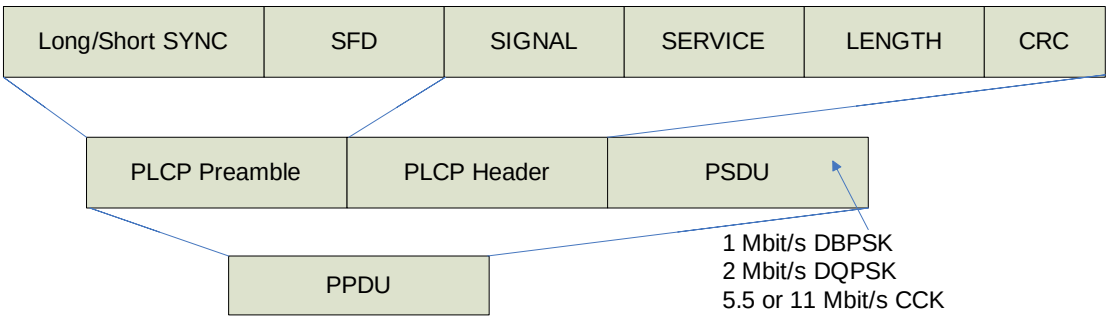

Figure 2.1 Frame format of ERP-DSSS/CCK

B. ERP-OFDM – This mode builds on the payload data rates of 6, 9, 12, 18 24, 36, 48, and 54 Mbit/s, based on different modulations (PSK and QAM) and coding rate, by means of OFDM technique. Except PLCP Preamble, SIGNAL field with data rate 6 Mbit/s and DATA are packaged (OFDM) symbol by symbol. The only difference from 802.11a is the operating ISM band (802.11a is in 5 GHz). Figure 2.2 is the PPDU format.

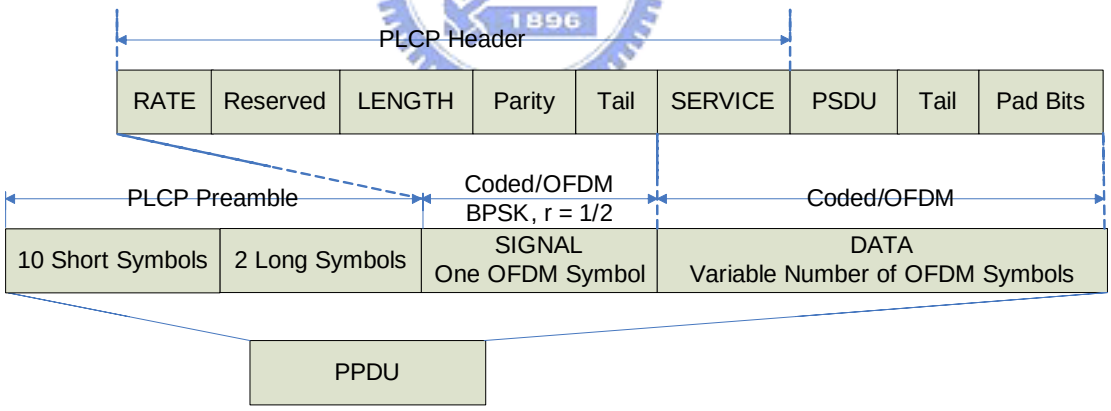

Figure 2.2 Frame format of ERP-OFDM

C. ERP-PBCC – This mode builds on the payload data (PSDU) rates of 22 and 33 Mbit/s and it is a single-carrier modulation scheme that encodes the payload using 256-state packet binary convolutional code. The PPDU format follows mode a).

D. DSSS-OFDM – This mode is a hybrid modulation combining a DSSS

preamble and header with an OFDM long preamble, signal field and payload transmission. In the boundary between DSSS and OFDM parts, it is single carrier to multi-carrier transition definition. The payload data rates are the same with those of b). The PPDU format is as followed in figure 2.3.

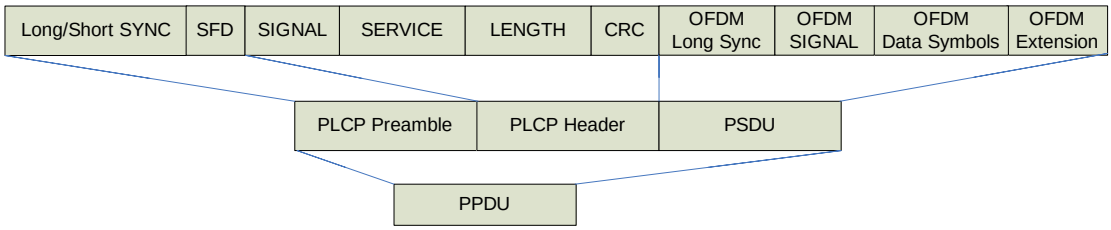

Figure 2.3 Frame format of DSSS-OFDM

An ERP BSS is capable of operating in any combination of available ERP modes and Non ERP modes. For example, if options are enabled, a BSS could operate in an ERP-OFDM-only mode, a mixed mode of ERP-OFDM and ERP-DSSS/CCK, or a mixed mode of ERP-DSSS/CCK and Non-ERP. Notice that since the first two modes are required to implement and considered as main operating modes and the others are optional, the discussion of platform will be located on the first two modes hereinafter.

Figure 2.4 is the transmitter block diagram. After the parameters of data rate and data length are decided, following the blocks one by one will generate the transmitted signals, and the MUX that depends on operation mode will select the signal that be sent to air by antenna after up-conversion the baseband signals to operating channel frequency.

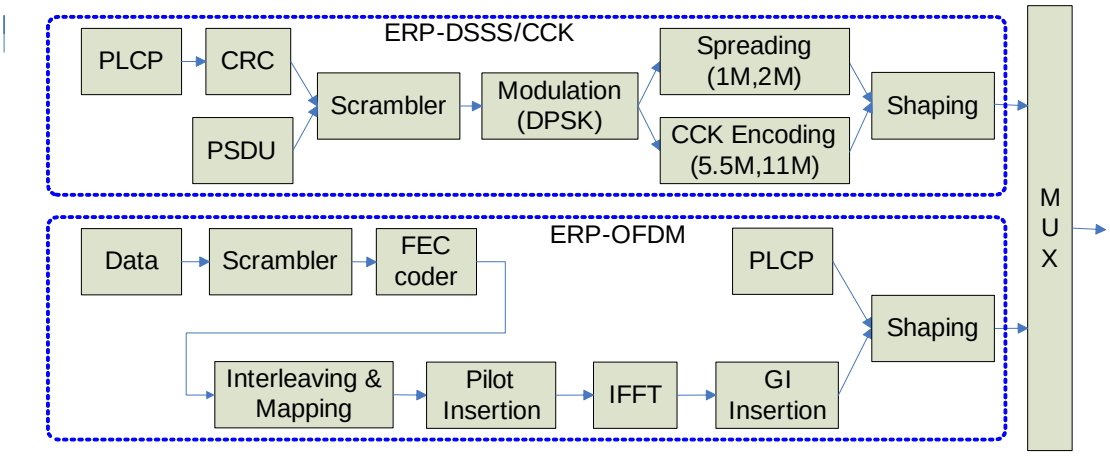

Figure 2.4 Block diagram of transmitter

2.4 Channel Model

There are many imperfect effects during transmitted signals through channels, such as Additive White Gaussian Noise (AWGN), carrier frequency offset (CFO), multi-path, and so on. These imperfections make the receiver design hard because it caused received signal distortion, rotation, delay, decay, and so on. We show the block diagram of channel model in figure 2.5, and describe these effects briefly in the following subsections.

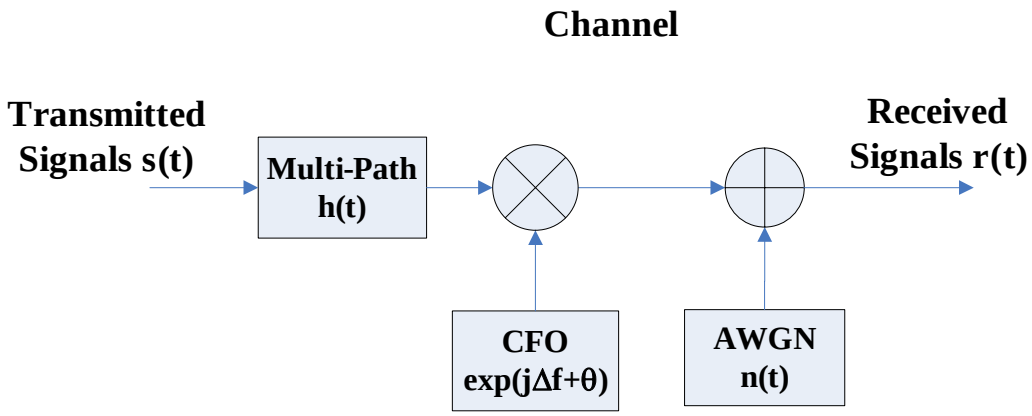

Figure 2.5 Block diagram of channel model

2.4.1 Additive White Gaussian Noise

The common wideband channel thermal noise impairment, on which SNR (Signal to Noise Ratio) is typically based. The primary spectral characteristic of thermal noise is that its power spectral density is the same for all frequencies of interest in most communication systems. The thermal noise is usually modeled as Additive White Gaussian Noise (AWGN).

2.4.2 Carrier Frequency Offset

Carrier Frequency Offset (CFO) is caused by the local oscillators' inconsistency between the transmitter and receiver. The received signals *r(t)* can be written as

$$r(t) = \sum_{t} s(t) \times e^{i2\pi\Delta ft+\theta} \tag{2-1}$$

where Δf and θ are the differences of carrier frequency and carrier phase between TX and RX, respectively. CFO will cause the constellation of the transmitted signal become a circle, that means phase of the transmitted signal will rotate with time.

2.4.3 Multi-path

Because there are obstacles and reflectors in the wireless propagation channel, the transmitted signal arrivals at the receiver from various directions over a multiplicity of paths. Such a phenomenon is called multi-path. It is an unpredictable set of reflections and/or direct waves each with its own degree of attenuation and

delay. Multi-path is usually described by two sorts:

A. Line-of-sight (LOS): the direct connection between the transmitter (TX) and the receiver (RX).

B. Non-line-of-sight (NLOS): the path arriving after reflection from reflectors.

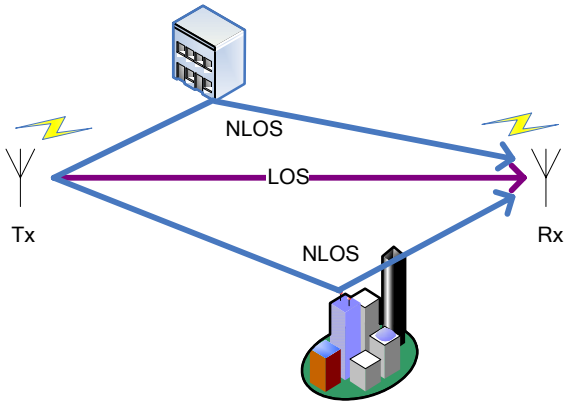

Figure 2.6 NLOS and LOS between transmitter and receiver

Multi-path will cause amplitude and phase fluctuations, and time delay in the received signals. When the waves of multi-path signals are out of phase, reduction of the signal strength at the receiver can occur. One such type of reduction is called the multi-path fading; the phenomenon is known as "Rayleigh fading" or "fast fading." A representation of Rayleigh fading and a measured received power-delay profile are shown in figure 2.7. Besides, multiple reflections of the transmitted signal may arrive at the receiver at different times; this can result in inter symbol interference (ISI) that the receiver cannot sort out. This time dispersion of the channel is called multi-path delay spread that is an important parameter to access the performance capabilities of wireless systems. A common measure of multi-path delay spread is the root mean square (rms) delay spread. For a reliable communication without using adaptive equalization or other anti-multi-path techniques, the transmitted data rate should be much smaller than the inverse of the rms delay spread (called coherence bandwidth).

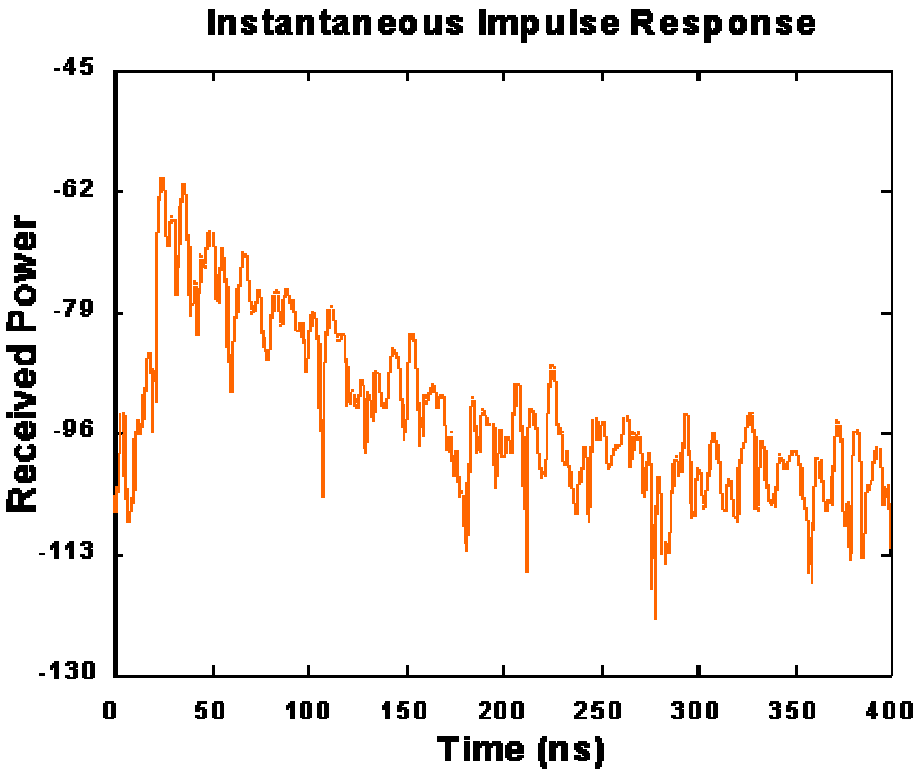

Figure 2.7 Instantaneous impulse responses

2.4.4 Sampling Clock Offset

The interfaces of RF and Baseband data are Digital to Analog Converter (DAC) in the transmitter and Analog to Digital Converter (ADC) in the receiver side. The ADC is the first stage of Base band, so it dominates the receiving signal to noise ratio (SNR). To get the highest input SNR, the ADC is hoped to sample at the eye open position where it has the maximum signal power. However, the initial sampling phase could be anywhere in the eye diagram, so timing synchronization is necessary. The ADC has two kinds of clock source: free running clock and phase lock loop (PLL) output clock. With free running clock, this method also called non-synchronous sampling or fix sampling, clock frequency and phase are fixed. Once timing error estimated, the compensation would be performed with interpolator. With PLL output clock, also called synchronous sampling or dynamic sampling, it receives the timing

error and adjusts its frequency and phase to compensate the error. There is a need to maintain synchronization while the accuracy and stability of the original clock reference in the receiver may not be ideal. These tasks are the responsibility of a specific module Delay Lock Loop (DLL).

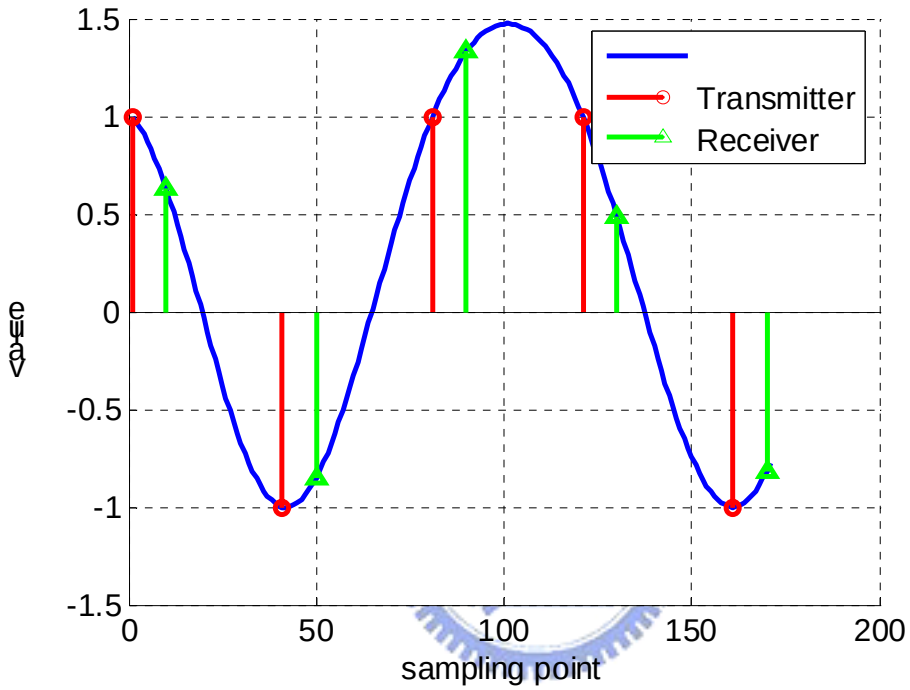

Figure 2.8 Sampling clock may no ideal in the receiver

2.5 Proposed Packet Synchronizer

In the proposed synchronizer, it needs to decode the data of OFDM and DSSS format when either of them is transmitted. We distinguish OFDM and DSSS format by their preambles, as depicted in figure 2.9. The OFDM preamble field consists of ten identical short preambles, and a guard interval followed by two identical long preambles. These OFDM preambles are transmitted at 20 MHz. In DSSS preamble, there is a series of scrambled positive and negative ones, whose symbol rate is 1

MHz. And then these scrambled ones are spread over the 11-chip Barker sequence *b(t)={ + − + + − + + + − − −}*. Thus, these DSSS preambles are transmitted at 11 MHz.

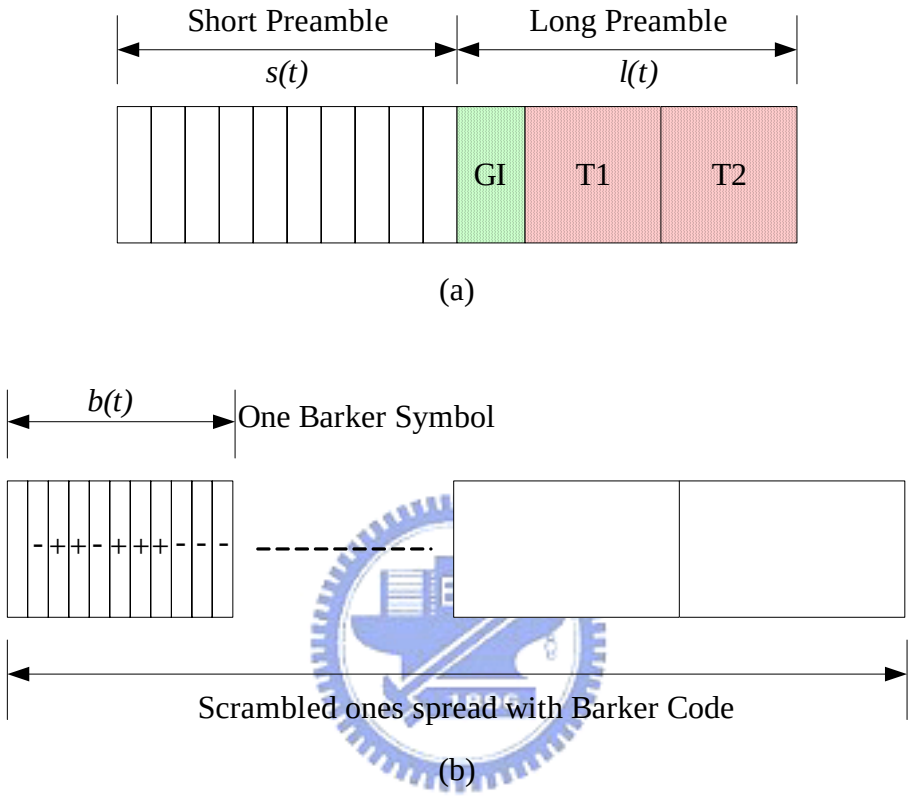

Figure 2.9 Structure of (a) OFDM preamble (b) DSSS preamble

To achieve the robust synchronization, universal receiver architecture is shown in figure 2.10. As the signal is down converted, the analog waveform is sampled by one pair of ADCs for I and Q channels whereas two pairs of ADCs in conventional design. The sampling clock is controlled by ADPLL and ADDLL. In the baseband, there are two match filters for the detection of OFDM and DSSS. The control unit switches the MUX according to the match filters' result, and change the clock rate of the ADPLL/DLL to a proper value. At the same time, AGC adjusts the power level in the front end to keep the system performance.

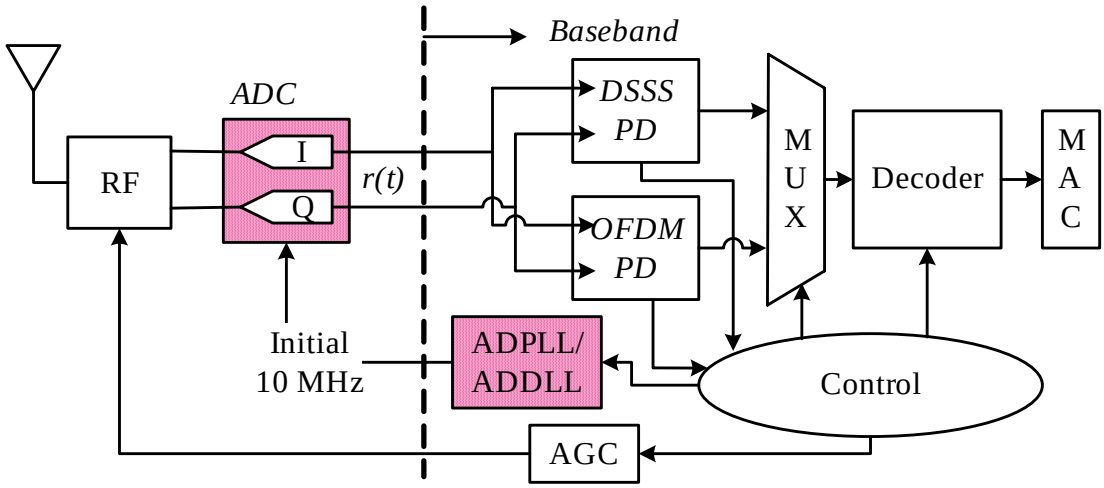

Figure 2.10 Proposed packet synchronizer with dynamic sampling

Chapter 3

Detection and Acquisition Algorithms

In dual DSSS/OFDM systems, the first problem we encounter is that how to determine the received signals are from either DSSS or OFDM transmitter. Although we can recognize each system by their distinct preamble structure, these two modulation systems also occupy different bandwidth and sampling rate. Thus, we may need two set of ADCs operating at different clock, or use high sampling rate ADC and obtains the other sampling rate via interpolation. Both methods require extra hardware and more power consumption. In contrast to those methods, we apply a dynamic ADC sampling methodology to simply hardware design and reduce power consumption..

With the method of dynamic sampling, the Control Unit, as mentioned in figure 2.10, can handle the generated clock. The initial clock rate is set at the higher one between OFDM and DSSS. When a specific type of preamble is declared, the frequency of ADPLL is then moved to two times of its symbol rate for timing acquisition. After that, the sampling rate is down to one time of the original symbol rate. Thus, we can achieve the packet detection by only one set of ADCs and dynamically control the sampling frequency for low power purposes.

The data flow is depicted in. figure 3.1. At first, RX is waiting for packets coming. The symbols are sampled at 20MHz. In the first case, assume an OFDM packet is received; we use its short preamble for packet detection. After that, we should determine the OFDM symbol timing, that is, to find out FFT window. This can be achieved by long preamble. In the second case, assume a DSSS packet is received; we use DSSS preamble for packet detection. Since DSSS preambles are scrambled ones spreading with Barker code, we can determine a DSSS packet is coming by

Barker code property.

In the beginning of synchronization, the received signal type is unknown, so it has three possible cases: OFDM preamble, DSSS preamble, and others. In subsection 3.2 and 3.3, we are going to define the condition by which the declaration of coming for OFDM or DSSS can be made.

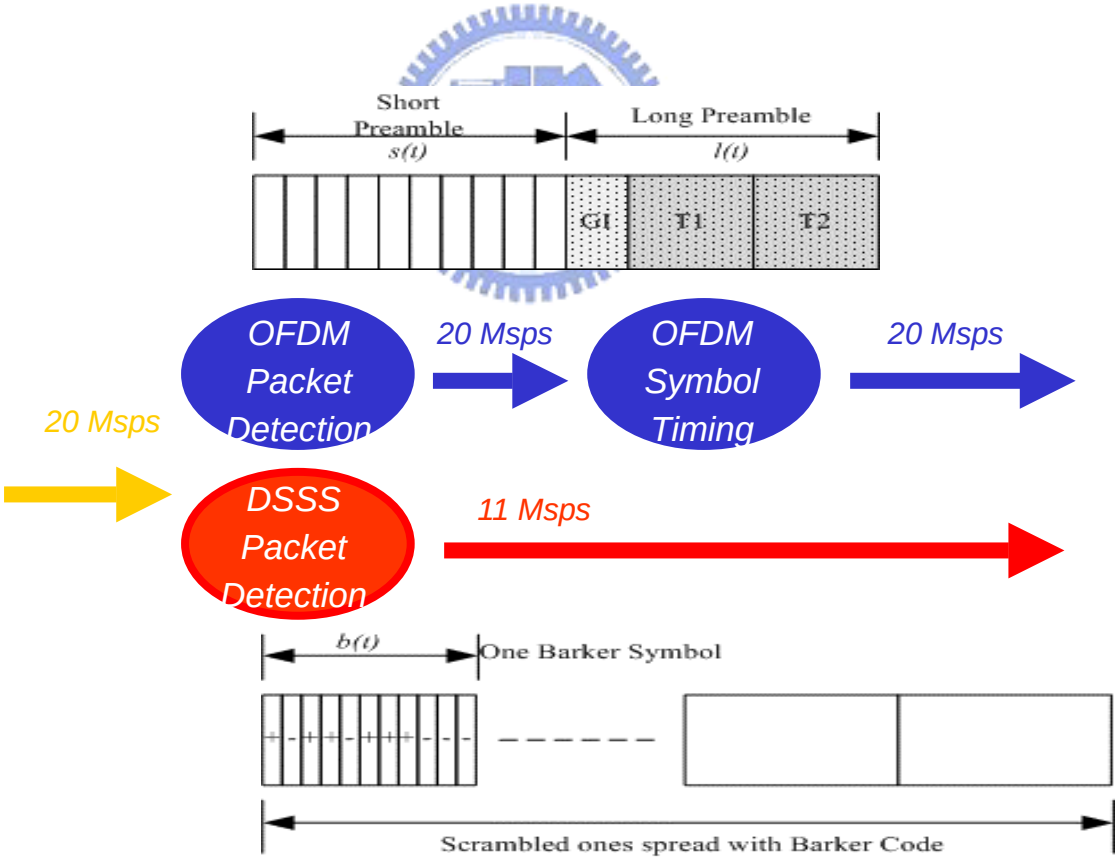

Figure 3.1 Detection data flow at 20 MHz

3.1 Detection of OFDM Packet

The OFDM Short preambles have been designed to help the detection of the start of the packet. These preambles are Pseudo Noise (PN) sequence in frequency domain, and they enable the receiver to utilize a very simple and efficient algorithm to detect the packet. The general approach was presented in Schimdl and Cox [6] using the special preamble composed of two identical symbols to synchronize the timing.

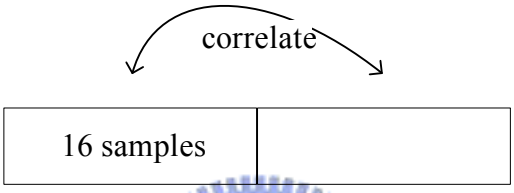

Figure 3.2 Using 2 short preambles for packet detection at 20 MHz

. The OFDM short training symbol *s(t)* is used for the packet detecting, as depicted in figure 3.2. The auto-correlation function *c(t)* is defined by

$$c(t) = \sum_{k=0}^{L_a - 1} r(t+k) r^*(t+k+L) \tag{3.1}$$

where * denotes complex conjugate and L_a is the length of each OFDM short preamble. It is the correlation of the present short preamble with the delayed one. The received power *p(t)* for the short preamble can be written as

$$p(t) = \sum_{k=0}^{L_a - 1} \left| r(t+k) \right|^2 \tag{3.2}$$

The resulting timing metric $\Lambda_{OFDM}(t)$ is defined by

$$\Lambda_{OFDM}(t) = \frac{|c(t)|^2}{p(t)^2} \tag{3.3}$$

If the timing metric $\Lambda_{OFDM}(t)$ exceeds a pre-defined threshold Γ, we declare the presence of OFDM packet. The criterion is usually transform into

$$|c(t)|^2 > \Gamma \times p(t)^2 \tag{3.4}$$

to omit the division operation. Figure 3.3 shows that noise has a great effect upon that timing metric $\Lambda_{OFDM}(t)$. Therefore we will discuss the decision of the threshold value Γ in the subsection 3.4.

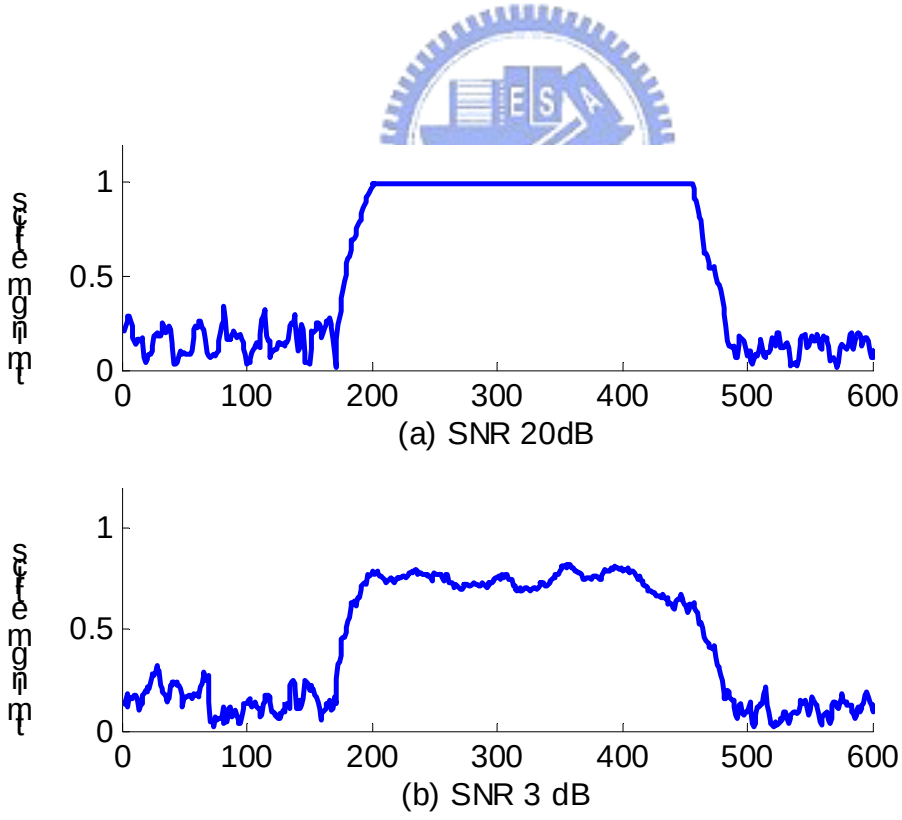

Figure 3.3 Timing metric $\Lambda_{OFDM}(t)$ versus different SNR

3.2 Detection of DSSS Packet

The DSSS preamble is scrambled ones spread with an 11-chip Barker sequence. The packet detection of DSSS is to correlate the received signal with a pre-known spreading sequence. If it is correctly matched, the correlation power will be greater than the others, and we may call it a 'Peak'. The cross correlation *d(t)* is defined by

$$d(t) = \left| \sum_{k=0}^{L_B - 1} r(t+k) \cdot b(k) \right|^2 \tag{3.5}$$

where *b(k)* is the 11-chip Barker code and L_b is the length of Barker code.

Unfortunately, there are usually mismatch between the initial OFDM sampling rate and chip-rate of DSSS preamble, and thus exists a timing shift. The shifted samples *r'(t)* can be derived as follow

$$r'(nT_I) = r(nT_I) \oplus \text{sinc}\left(\frac{nT_I - \Delta P_n}{T_I}\right) \tag{3.6}$$

where $\oplus$ denotes convolution operation, TI is sampling interval, and ΔPn is the clock offset. The peak varies from time to time even if the first chip is correctly aligned, as shown in figure 3.4.

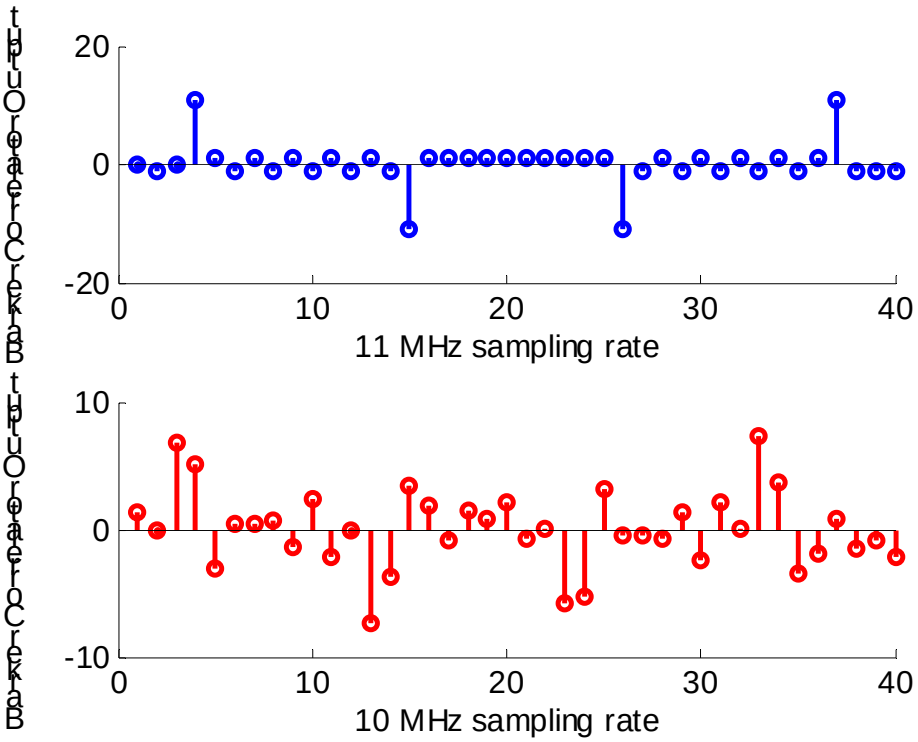

Figure 3.4 Barker correlator output at (a) 11 MHz and (b) 10 MHz

Due to the mismatch of sample rate, two characteristic are examined to ensure a DSSS packet is coming. The Baker correlator output is depicted in figure 3.5. First, the interval of two adjacent peaks remains a constant period. Second, the peak values should be greater than others in one period, and we can compare peak value with several "valley" values. Thus, the resulting timing metric $\Lambda_{DSSS}(t)$ is defined by

$$\Lambda_{DSSS}(nT_S) = d(nT_S) - \sum_{k=ts}^{te} d(nT_S + kT_C) \tag{3.7}$$

where T_S is the period of peak occurrence, T_C is the chip period of spreading code, and [*ts, te*] is the index window of summation. The criterion for the declaration of DSSS is to see if $\Lambda_{DSSS}(t)$ is positive.

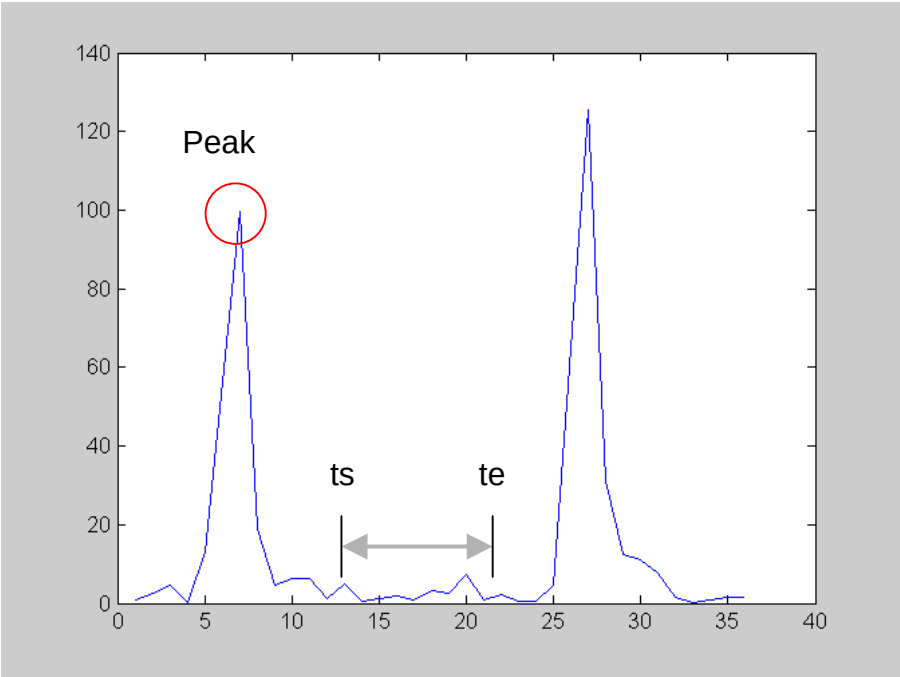

Figure 3.5 Barker correlator output of DSSS

We also compare the proposed DSSS packet detection with previous work done by Shih-Lin Lo [9], who is graduated isIP Lab member. As shown in figure 3.6, the method proposed by Lo is to compare the peak value with average value, in other word, Peak-to-Average Power Ratio (PAPR). The DSSS packet is declaration if the PAPR exceeds a pre-defined threshold. This method works fine when no multi-path is presence. If there is multi-path presence, due to the multi-path delay spread, peak will spread out and inference those correlations close to it. Thus the PAPR value will have great variation and decrease detect probability. The proposed work is to compare the peak value with valley value. Because it doesn't take those correlations close to peak into account, it has better detect probability than previous work.

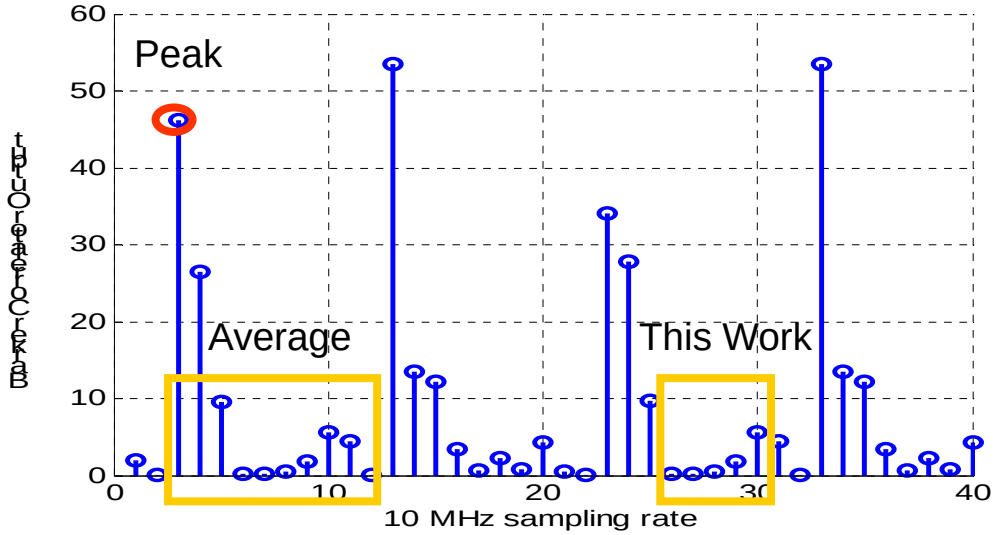

Figure 3.6 Comparisons between Lo's and this work

How to choose the length and location of index window [*ts* ,*te*] will be discussed later. Here we summarize the criterion for declaration of OFDM and DSSS as

$$\Lambda_{OFDM}(t) > \Gamma \quad \text{and} \quad \Lambda_{DSSS}(nT_S) > 0 \tag{3.8}$$

3.3 Threshold Decision

To evaluate the performance of packet detection in several cases, IEEE 802.11g are taken as the simulation platform. Thus the symbol rate and chip rate of OFDM and DSSS are 20 MHz and 11 MHz, respectively. The probability of packet loss and false alarm are to be analyzed.

In the first case, assume OFDM packets are transmitted, the distribution of timing metric $\Lambda_{OFDM}(t)$ is depicted in figure 3.7. It shows the timing metrics in different packet types transmitted, i.e. OFDM preamble, DSSS preamble, or noise. The OFDM preambles under different SNR are also shown in figure 3.7. Obviously, when the transmitted signal is either DSSS preamble or noise, the timing metrics $\Lambda_{OFDM}(t)$ are mostly low. On the other hand, when the transmitted signal is OFDM preamble, the timing metrics $\Lambda_{OFDM}(t)$ become more centralizing and higher as SNR increasing.

Now return to the threshold decision problem. The false alarm probability is the area of noise curve above threshold; the packet loss probability is the area of OFDM curve below threshold. If the threshold sets too high, the false alarm probability decreases but the packet loss probability increases. On the other hand, if the threshold sets too low, the packet loss probability decrease but the false alarm probability increases. The trade off can be made by setting the threshold Γ at the intersection of OFDM and Noise curves. Take the curve when SNR equals 6 dB as example, it can be seen that Γ is equal to 0.3 as illustrated in figure 3.7.

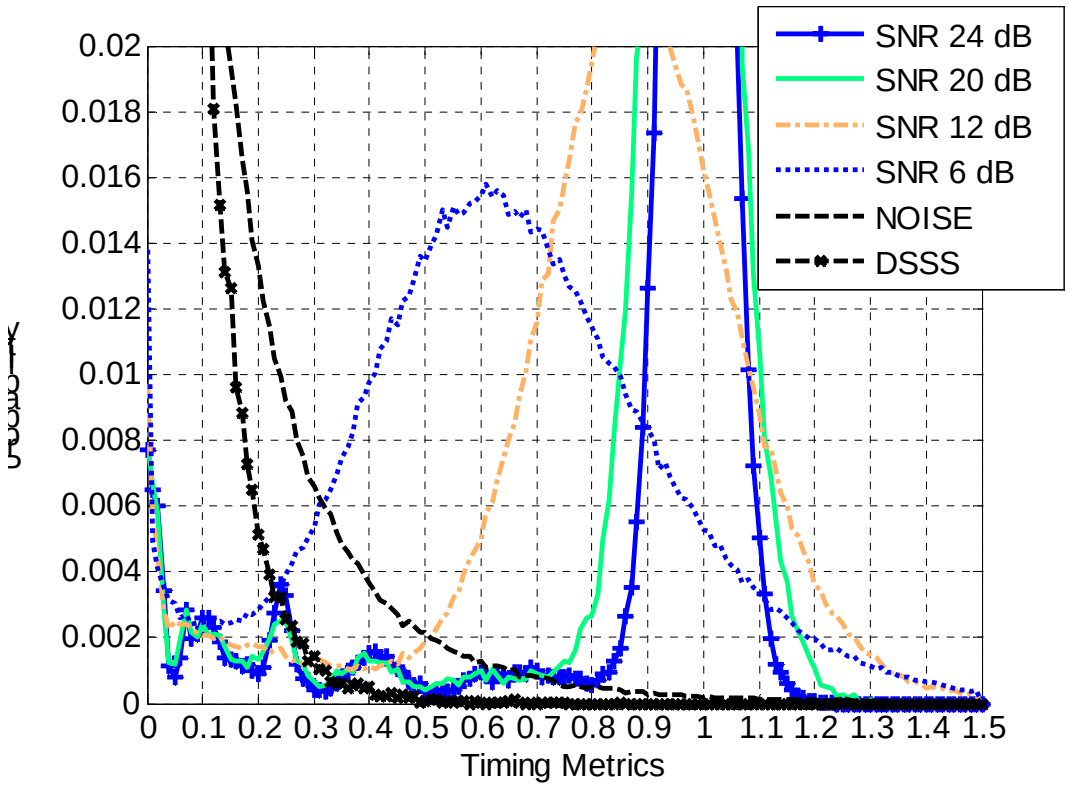

Figure 3.7 Distribution of timing metric $\Lambda_{OFDM}(t)$ at different SNR

Nevertheless it is possible to know that $\Lambda_{OFDM}(t)$ exceeds threshold whereas the received signals are only noise. In this situation, false alarm is occurred. In order to reduce false alarm, it is necessary to examine the times by which the timing metrics $\Lambda_{OFMD}(t)$ exceeds specific threshold Γ. Figure 3.8 shows the CDF of timing metrics $\Lambda_{OFDM}(t)$ transmitted noise. The threshold and the times exceeds threshold successively are also considered. In other word, the CDF shows the relationship between the false alarm probability and the threshold decision. To achieve 0.01 false alarm probability, for example, the threshold Γ equals 1.3 if the count is 1. But the threshold Γ can be 0.4 if the count is 10. The false alarm probability will decrease greatly if the count is high. It shows that even the timing metrics $\Lambda_{OFDM}(t)$ is high when the received signal is noise, it will not keep high consecutively.

But it is not the case when the received signal is OFDM preamble. Figure 3.9 shows CDF of timing metrics $\Lambda_{OFDM}(t)$ received OFDM preamble under SNR 6 dB. The CDF shows the ratio of OFDM preamble can be recognize by specific threshold and count. The detection probability is direct proportion to this CDF. We can see that to achieve 0.1 target line, for example, the threshold Γ equals 1 if the count is 1. But the threshold Γ can be 0.65 if the count is 10. It explains that low threshold will not reduce detection probability very much if we increase the count.

Hence we can use low threshold to decrease packet loss probability, whereas use high count to decrease false alarm probability. Thus, the OFDM preamble and noise can be distinguished by low threshold and high count.

3.3.1 Adaptive Threshold Decision

The fixed-threshold and high count scheme works fine, but it can be further improved by adaptive threshold approach. Although the adaptive threshold approach has not yet implemented in my thesis, the main idea is as follow. At first, assume the channel condition is bad, and we use low threshold and high count to detect the OFDM preamble. The CDF of timing metrics $\Lambda_{OFDM}(t)$ received noise determine the values of threshold and count. If the threshold and count can detect a particular OFDM preamble, we change the threshold and count to detect the same OFDM preamble. If the new threshold and count can also recognize the same OFDM preamble, we adopt the new threshold and count. Otherwise, we restore the original threshold and count.

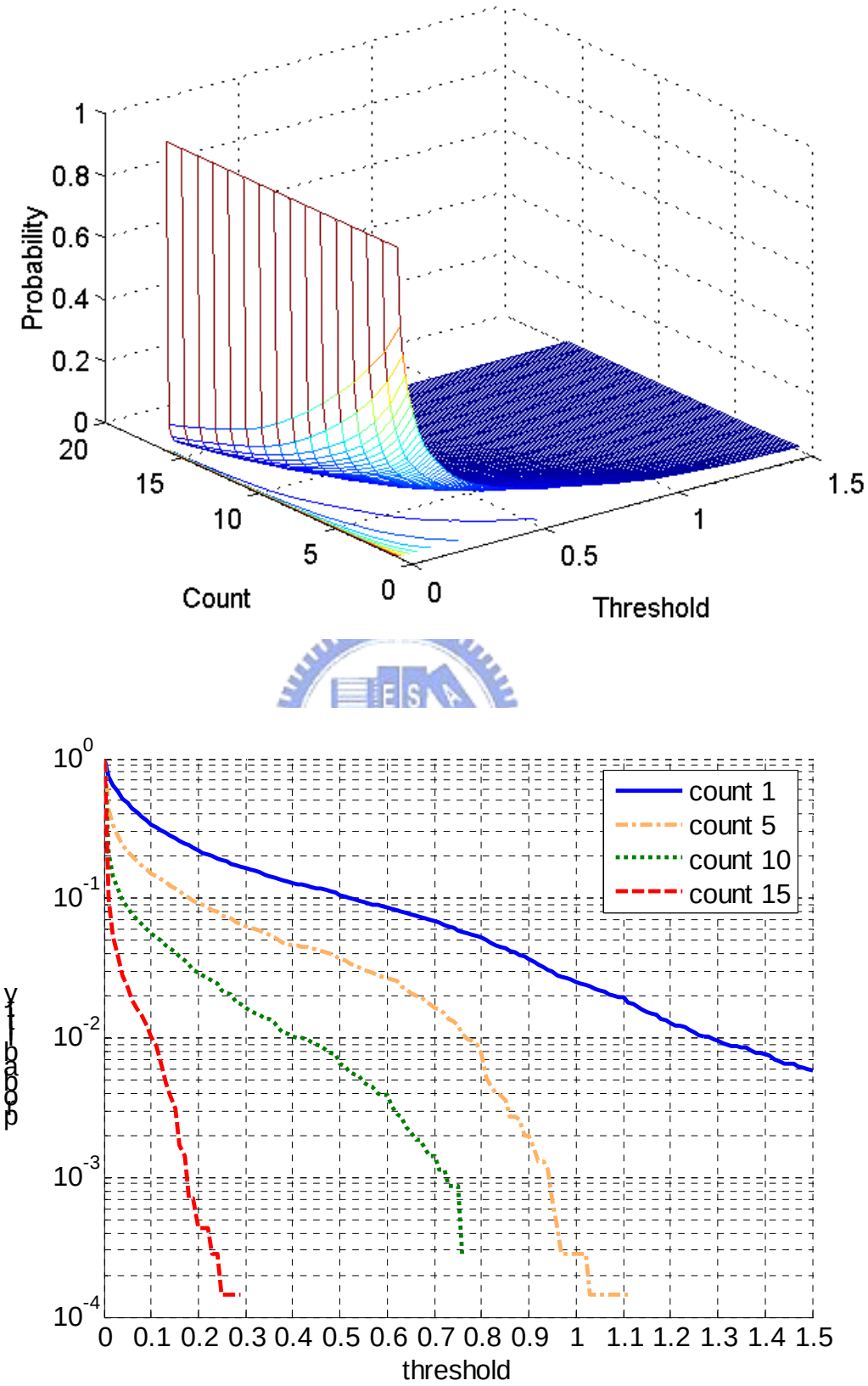

Figure 3.8 CDF of timing metrics $\Lambda_{OFDM}(t)$ received noise.

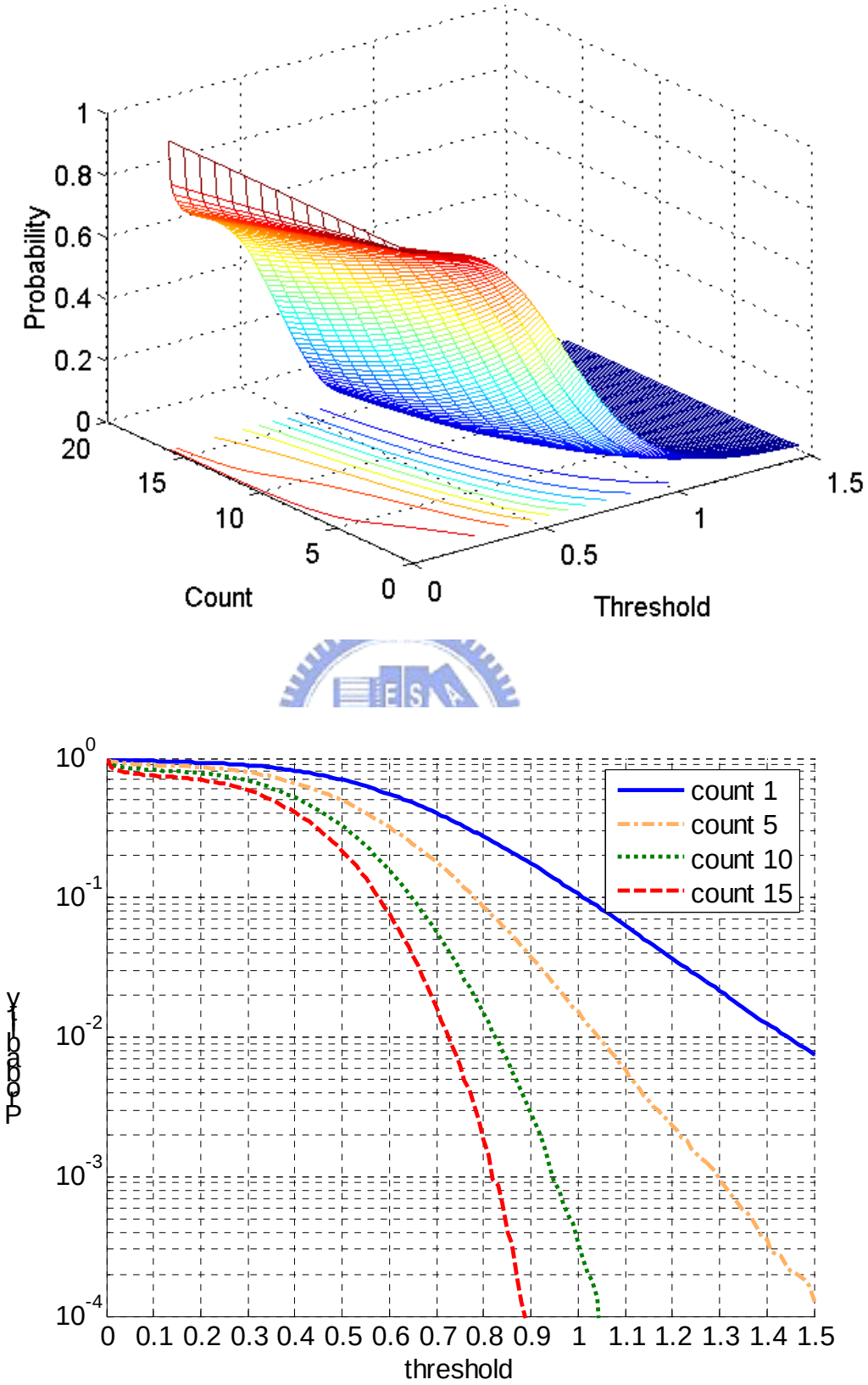

Figure 3.9 CDF of timing metrics $\Lambda_{OFDM}(t)$ received OFDM preamble under SNR 6 dB

In the second case, assume DSSS packets are transmitted. The probability under the condition $\Lambda_{DSSS}(t)>0$, which is the criterion to detect DSSS, is discussed by means of window length and location. We denote the window length by η, and the end index *te* can be defined by $te = ts + \eta$. As η increases, since the peak value is compared with more valley values, the probability of $\Lambda_{DSSS}(t)>0$ are decreasing. In the presence of multi-path, the peak will spread out and inference those correlations close to it. As a result, the probability of $\Lambda_{DSSS}(t)>0$ are also decreasing if we choose window location to be near the peak location.

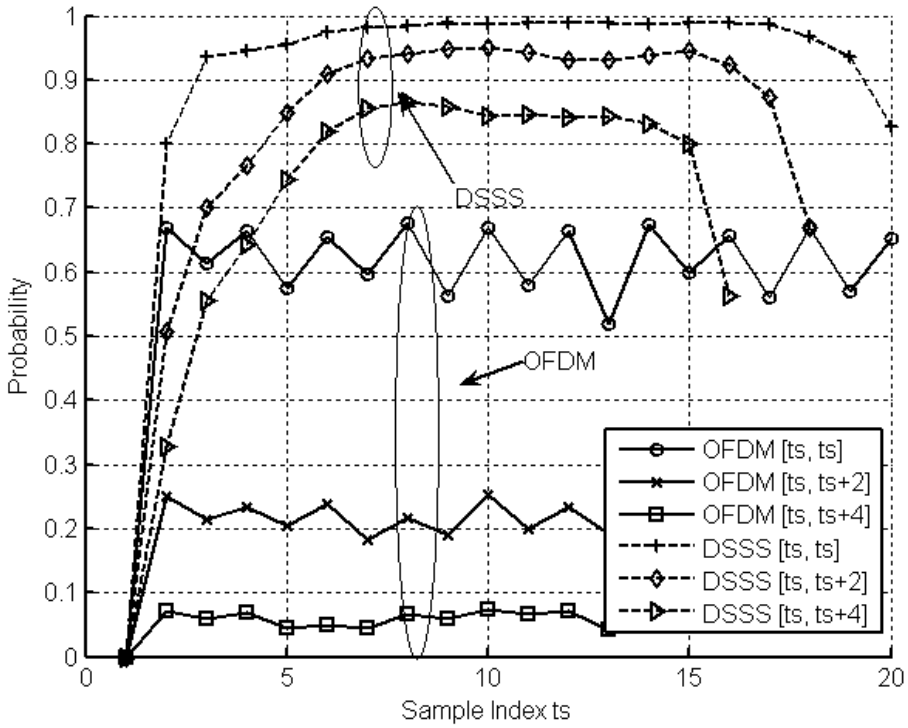

Figure 3.10 Probability of timing metrics $\Lambda_{DSSS}(t)$ @ SNR = 3 dB

The probability of detecting DSSS in transmitted different packet types is illustrated in figure 3.10. The start index *ts* are represented by x-axis and the window length η are illustrated in different curves. As η increasing, those curves in OFDM group are decreasing faster than those in DSSS group.

Hence the final criterions used for OFDM and DSSS detection are defined in the similar way by

$$\sum_{k=0}^{\alpha-1} sign(\Lambda_{\mathrm{OFDM}}(k)-\Gamma) == \alpha \tag{3.9}$$

$$\sum_{k=0}^{\beta-1} sign(\Lambda_{\mathrm{DSSS}}(kT)) == \beta \tag{3.10}$$

where *sign(.)* denote the operation of signum which return 1 if the element is positive, 0 if the element is negative. The criterion holds if timing metrics exceeds specific threshold consecutively a certain times. We summarize packet detection algorithm in FSM depicted in figure 3.11.

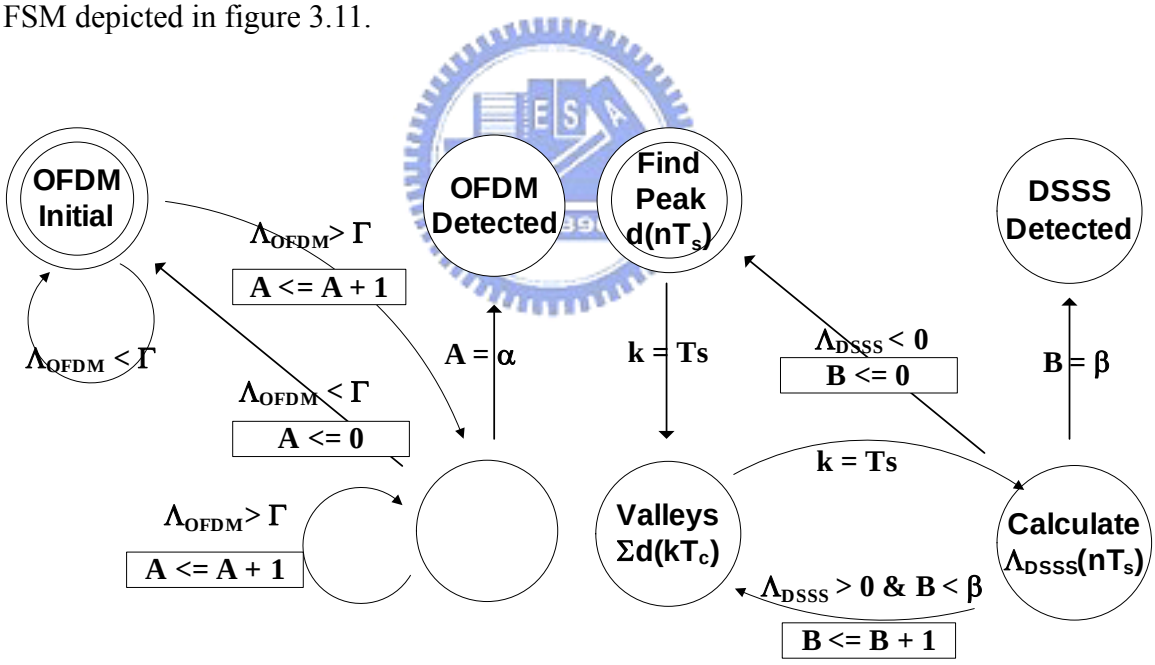

Figure 3.11 FSM of Packet Detection

3.4 Low Sampling Rate Packet Detection

Although the algorithms described in subsection 3.2 ~ 3.4 work fine at 20 MHz sampling rate, the power consumption issue may also be considered further. If we can decrease the sampling rate during packet detection period, we can reduce the power consumption very much.

Now consider the previous algorithms that work under 20 MHz sampling rate, but the DSSS packet detection only takes the odd samples to calculate, that is, the processing rate is only 10 MHz. On the other hand, the processing rate of OFDM system is 20 MHz, what happen if the processing rate is reduced from 20 MHz to 10 MHz? The only change is to let OFDM packet detection operates at 10 MHz. We still facility 2 short preamble to detect the packet, but since the sampling rate is 10 MHz, only half of 20 MHz. The operation needed is reduced to 8 samples per preamble, as depicted in figure 3.12. After running the simulation results, it shows that the 10 MHz sampling also satisfy the system requirement. The detection data flow is depicted in figure 3.13.

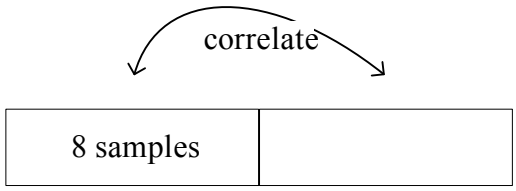

Figure 3.12 Using 2 short preambles for packet detection at 10 MHz

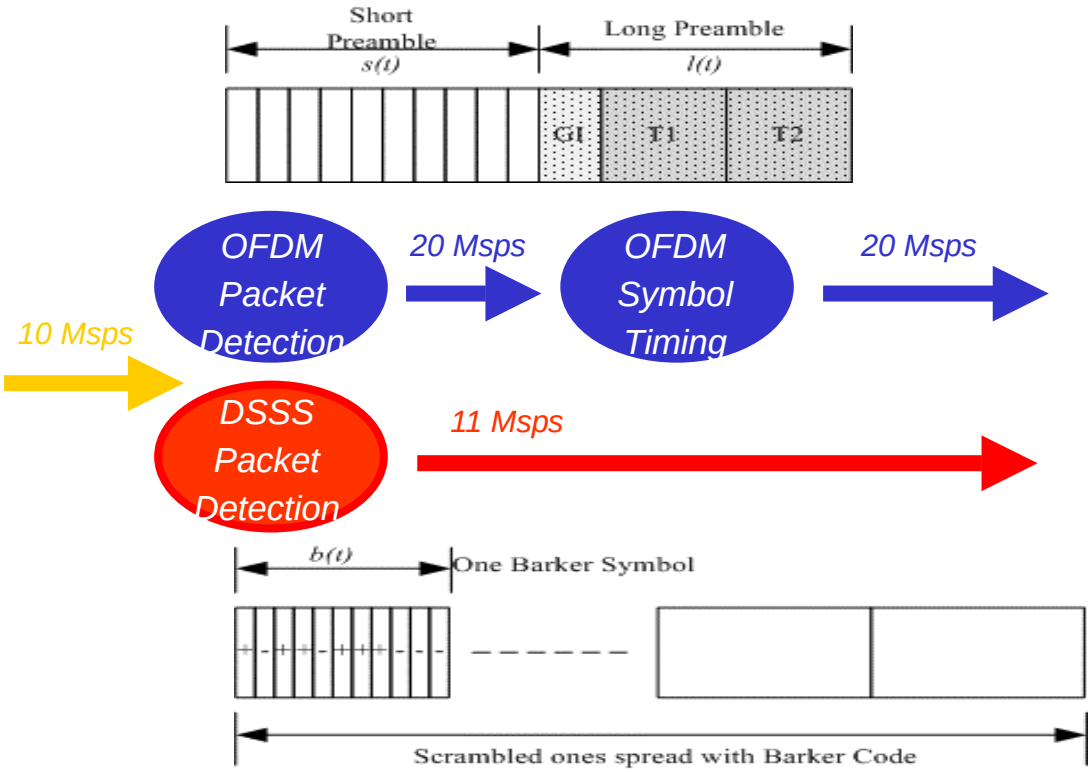

Figure 3.13 Detection data flow at 10 MHz

3.5 Estimation of OFDM Symbol Timing

After OFDM preamble has been detected, we need to decide the symbol timing. The task of symbol timing is to find the FFT window boundary. The miss-aligned FFT window will not only decay the original signal but also induce the inter carry interference (ICI) and inter symbol interference (ISI) [8]. Accordingly, we want the estimated symbol timing as close to the last sample in guard interval (GI) as possible.

The estimation of symbol timing τ can be obtained by observing the cross-correlation $\xi(t)$ and finding its maximum within a defined range

$$\tau = \arg \max_{t \in [\tau s, \tau e]} \{\xi(t)\} \tag{3.11}$$

$$\xi(t) = \left| \sum_{k=0}^{L_L - 1} r(t+k) LT(k)^* \right|^2 \tag{3.12}$$

where LT denotes the ideal long training sequence, L_L is the length of LT, and $[\tau s, \tau e]$ define the search window.

Figure 3.14 shows that the auto-correlation outputs keep high under the short preamble period, and attenuate enormous under the long preamble period. Hence the start of window search τs can be estimated by monitoring the timing metric $c(t)$ in (3.1). If it is below the threshold consecutively, we can assume it is at the end of short preamble and initiate the window search. Furthermore, since auto correlation is not robust to noise, τs is an approximately position that may be moved earlier or later than the real end of short preamble, as depicted in figure 3.15.

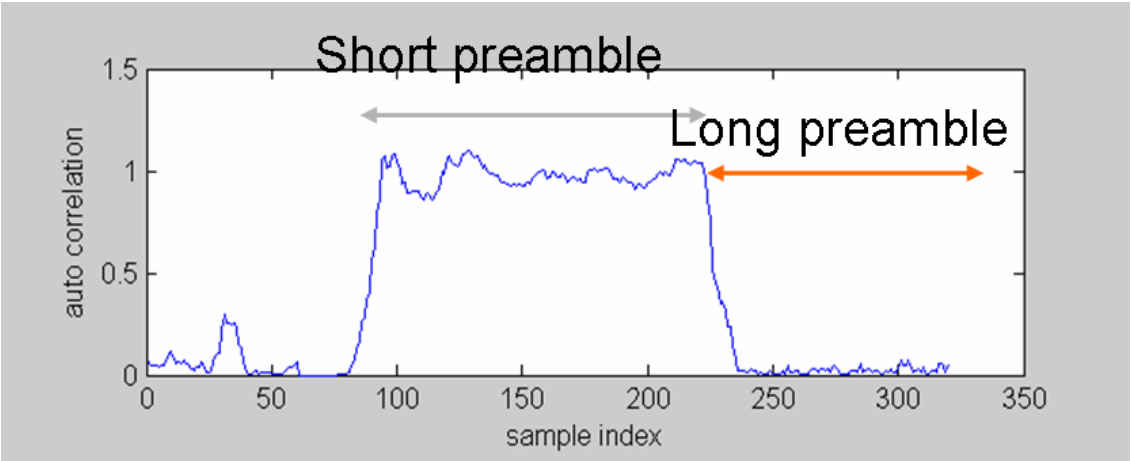

Figure 3.14 Auto-correlation outputs during preamble period

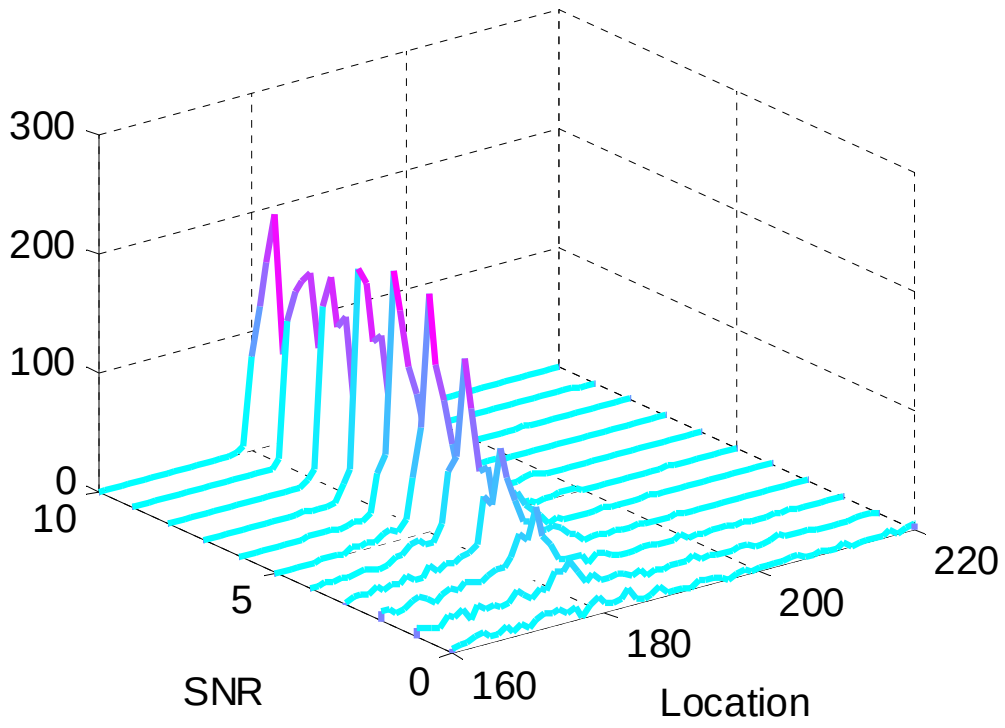

Figure 3.15 Distribution of SNR versus End of short preamble declaration

Conventionally, the end of search window τe is defined by a fixed length ε, say $\tau e = \tau s + \varepsilon$. If ε is too small, the range $[\tau s, \tau e]$ may not contain the true symbol timing,

whereas a large ε will result in more buffers to store the received long preamble, which is used to estimate channel impulse response. In contrast of fixed search window, we propose a dynamic search window algorithm by observing whether the maximum is greater than its successors consecutively. For this purpose, a variable ξ_{max} and a counter $\lambda(t)$ are defined by

$$\xi_{\max}(t) = \max\left\{\xi_{\max}(t-1), \xi(t)\right\} \tag{3.13}$$

And

$$\lambda(t) = \begin{cases} \lambda(t-1)+1 & \xi_{\max}(t-1) \geq \xi(t), \\ 0 & \text{otherwise}, \end{cases} \tag{3.14}$$

The symbol timing τ is determined immediately as the counter $\lambda(t)$ exceeds a pre-defined number p. In accordance with the proposed algorithm, the $\xi_{max}(\tau)$ is the local maximum over the window $[\tau s, \tau e]$ and the window length is at least p. In our reference system, the desired symbol timing is between sample index 177 and 192, and the estimated index in different p has the distribution as illustrated in figure 3.16 and 3.17. The distributions of estimated indexes are centralized by increasing p and SNR.

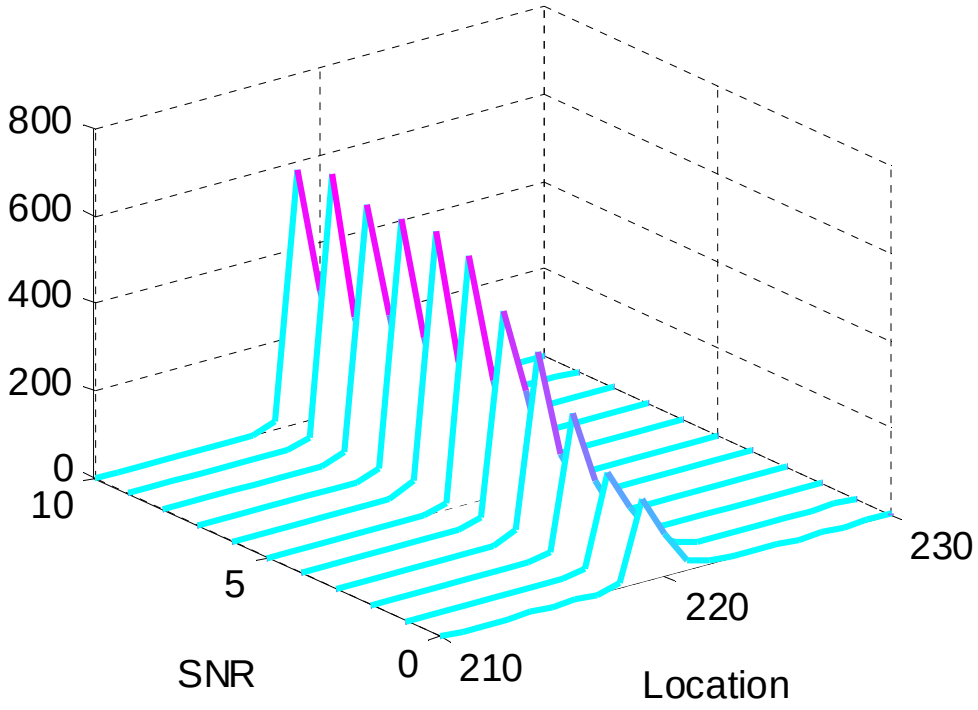

Figure 3.16 Distribution of symbol timing versus varied SNR

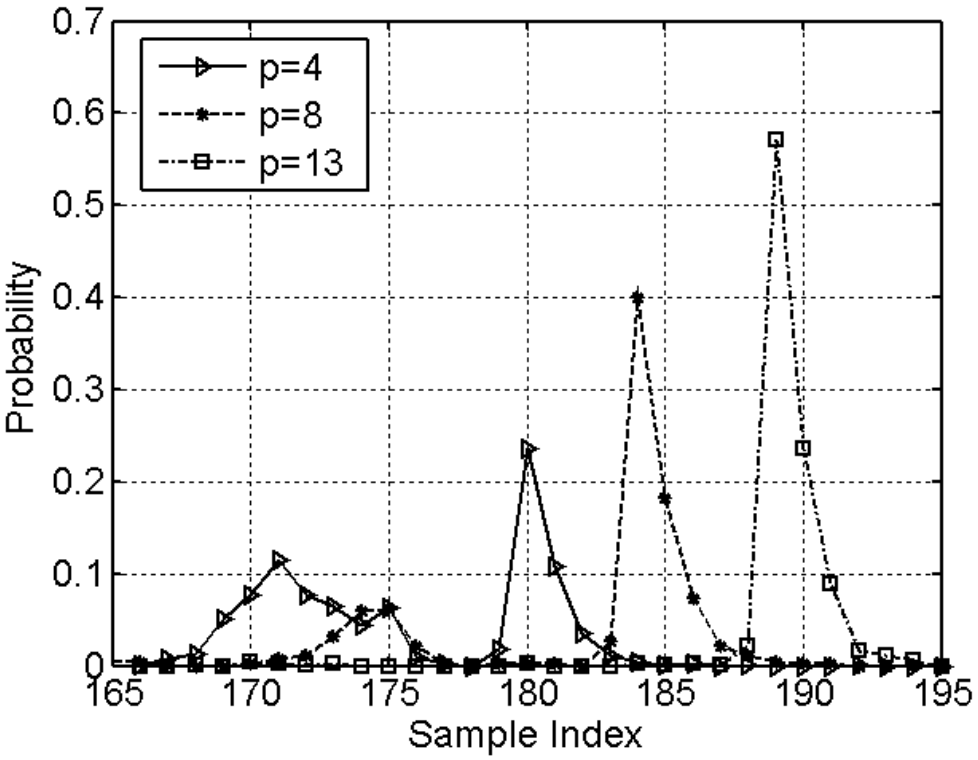

Figure 3.17 Distribution of symbol timing @ SNR = 6 dB

Chapter 4

Matlab Simulation

4.1 Simulation Platform

We chose Matlab as our simulation language due to its ability to mathematics, such as matrix operation, numerous math functions, and easily drawing figures.

The next problem is how to simulate the behavior between DSSS and OFDM system, the sampling rate difference between 20 MHz and 11 MHz, to be more accurate. To achieve the sampling rate difference, we first assume an ADC sampling at

$$2 \times \gcd(11, 20) = 440 \tag{4.1}$$

Mega samples per second. The OFDM transmitter, which are transmitted at 20 MHz, are 22x up-sampling using Raised Cosine Filter to achieve 440 M samples per second; in the same manner, the DSSS transmitted are 40x up-sampling. So that both satisfy 440 M samples per second. Whereas the transmitted *s(t)* signals suffer from various channel impairment, we can express the received signals *r(t)* as

$$r(t) = s(t) \otimes h(t) \times e^{i2\pi\Delta f t + \theta} + n(t) \tag{4.2}$$

Where the h(t) is the multi-path, the exponential term is CFO, and n(t) is the AWGN. At the receiver side, because the initial sampling rate is either 10 or 20 M samples per second, the received signals are 44/22x down sampling. Also, the sampling point is

not always the perfect one, there will exits a sampling offset. Finally, the DSSS/OFDM packet detector determines the packet type by the down-sampled signals Figure 4.1 illustrates the Simulation Flow Diagram.

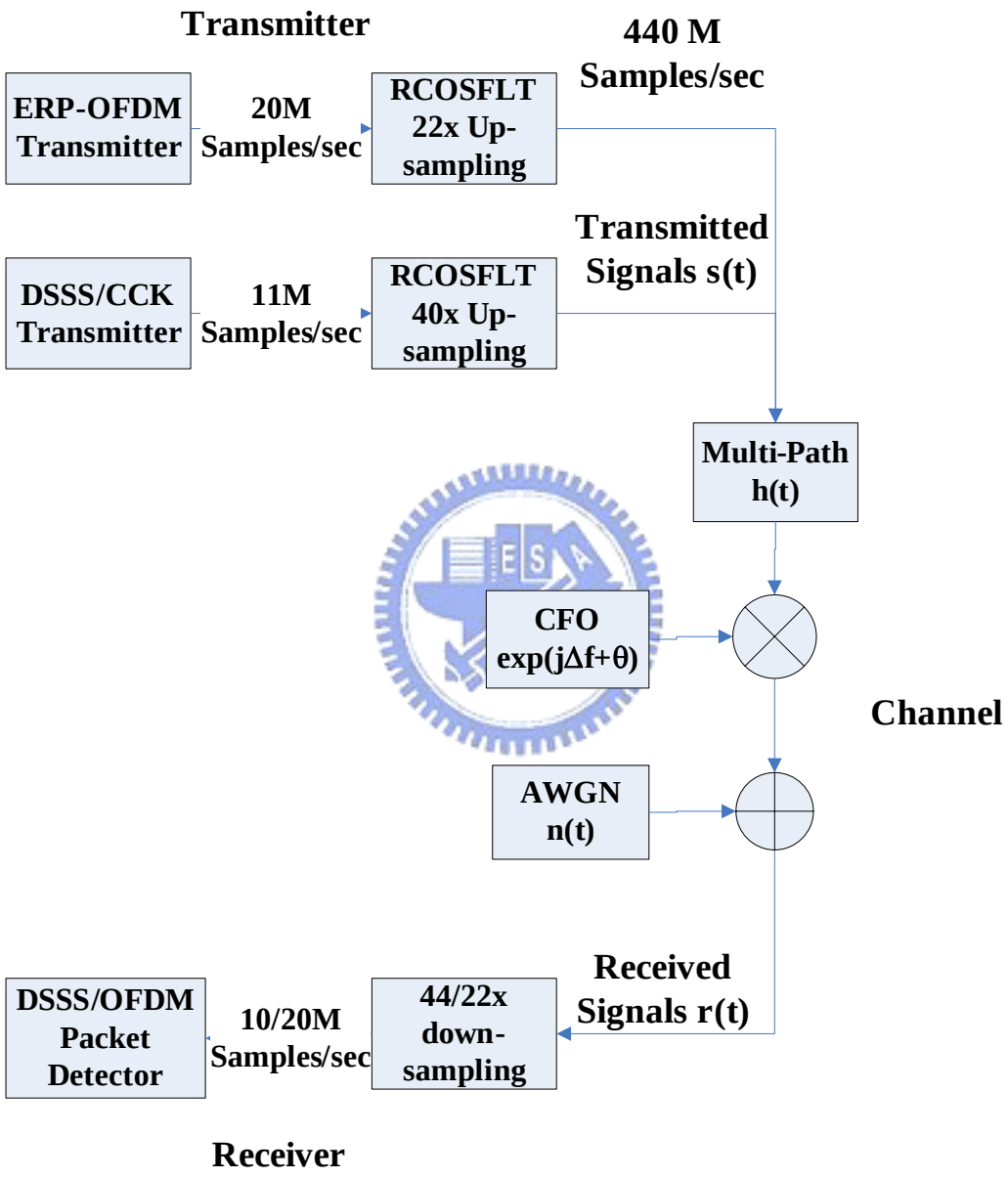

Figure 4.1 Simulation Flow Diagram

4.2 Simulation Result

To evaluate the performance of proposed packet detection and symbol timing estimation, IEEE 802.11g PHY based on dual modulation OFDM and DSSS [1][2] is considered. The simulation environment is under Rayleigh fading channel with RMS 50ns and CFO 50 p.p.m. The results are obtained by applying 2000 simulation runs with different SNR.

Figure 4.2 and 4.3 shows the packet detection performance at 20 MHz first. The required SNR for packet loss probability equal to 1% is about 1.5 dB under different channel conditions. The required SNR for packet false alarm probability equal to 1% is about 1~2 dB under different channel conditions.

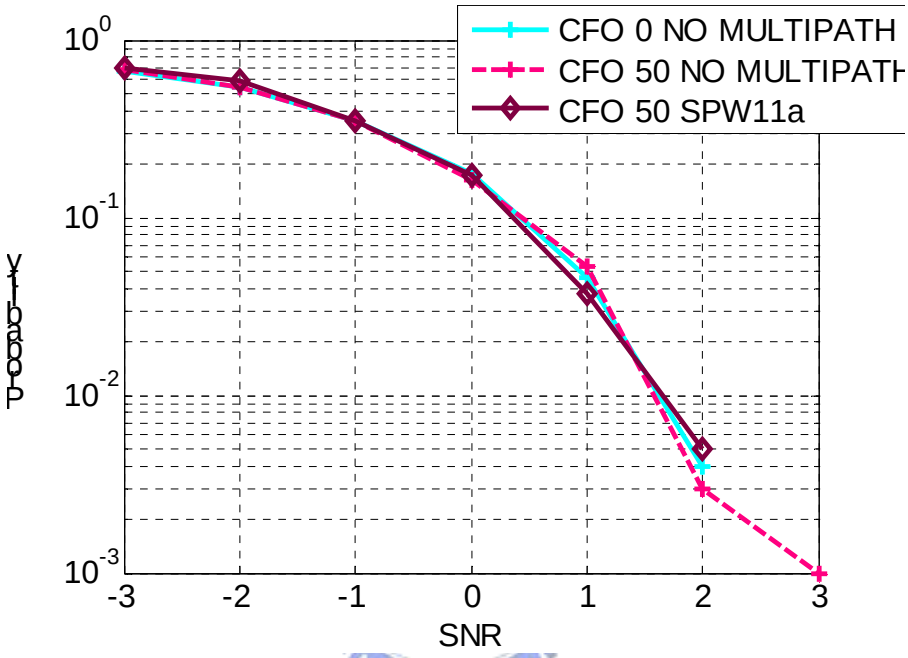

Figure 4.2 OFDM packet loss probabilities at 20 MHz

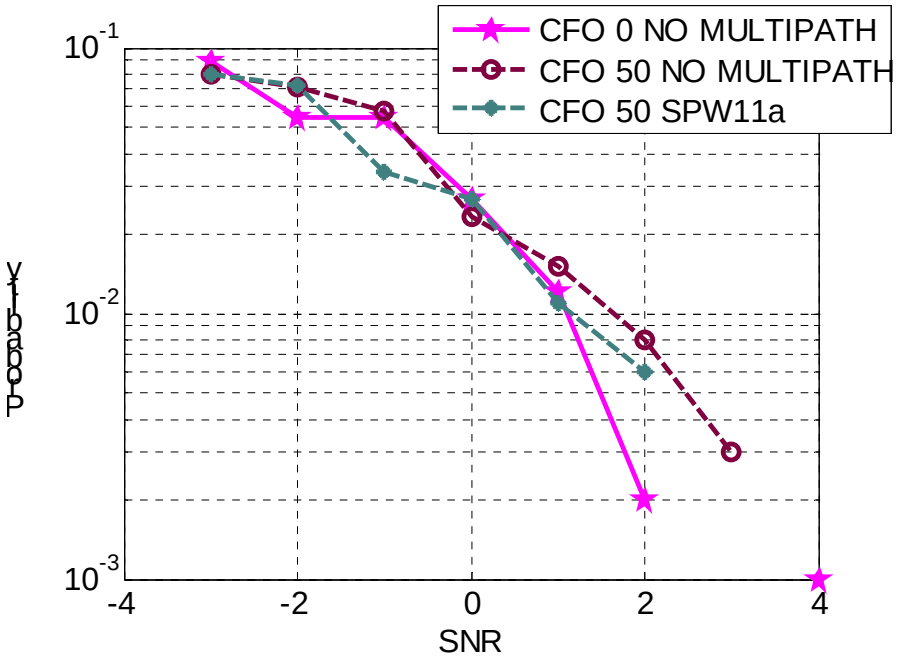

Figure 4.3 OFDM packet false alarm probabilities at 20 MHz

Figure 4.4 and 4.5 show the packet detector performance at 10 MHz. The required SNR for OFDM packet loss probabilities equal to 1% is 5.7 dB. The required SNR for DSSS packet loss and false alarm probabilities equal to 1% are -0.5 dB and -2.8 dB.

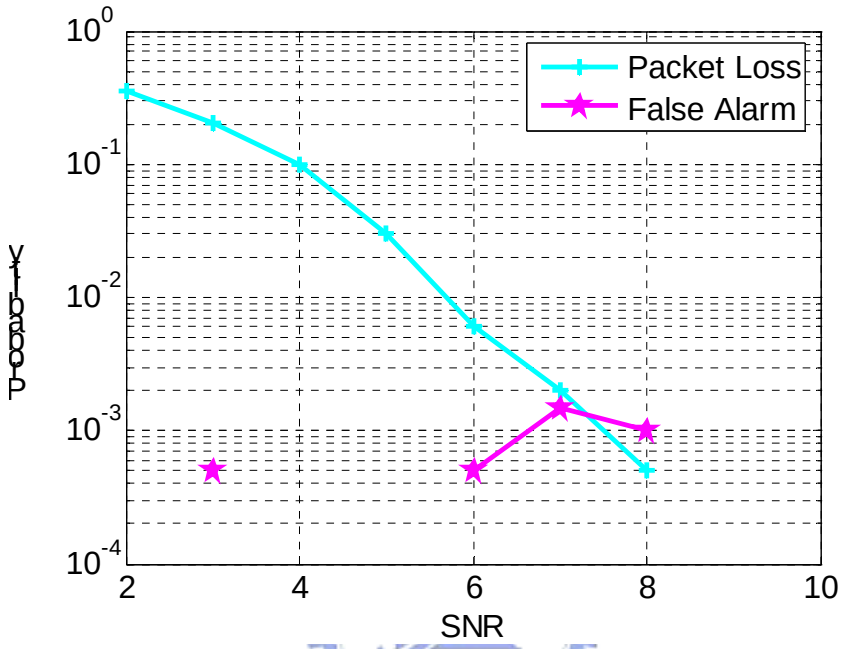

Figure 4.4 OFDM packet detection at 10 MHz

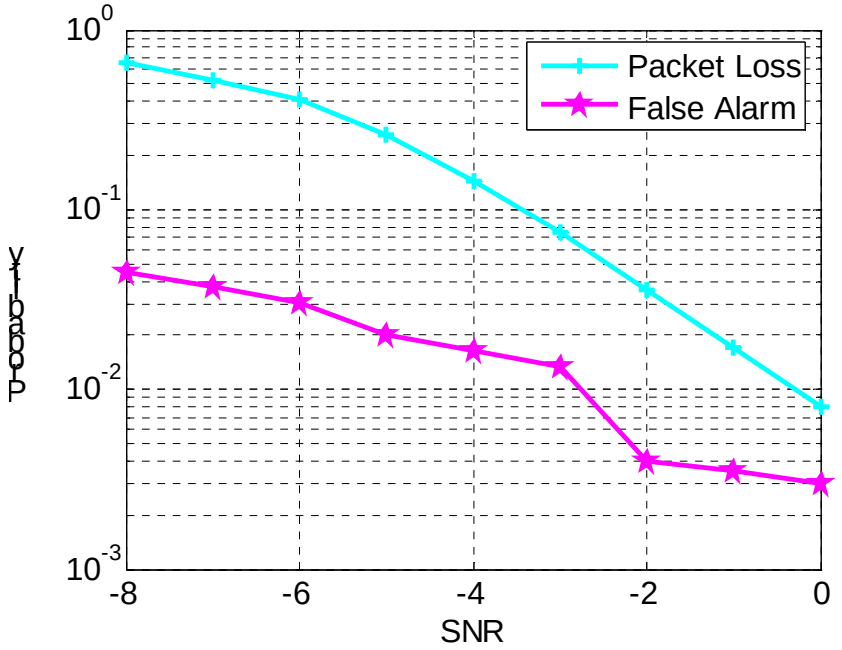

Figure 4.5 DSSS packet detection at 10 MHz

Figure 4.6 and 4.7 shows the symbol timing performance of two multi-path models SPW11a channel and IEEE fading channel. The packet loss and false alarm probabilities are very close in both channels. But the symbol timing error probability in IEEE fading channel is larger than in SPW11a channel. The performance loss is due to the pre-cursor channel impulse response.

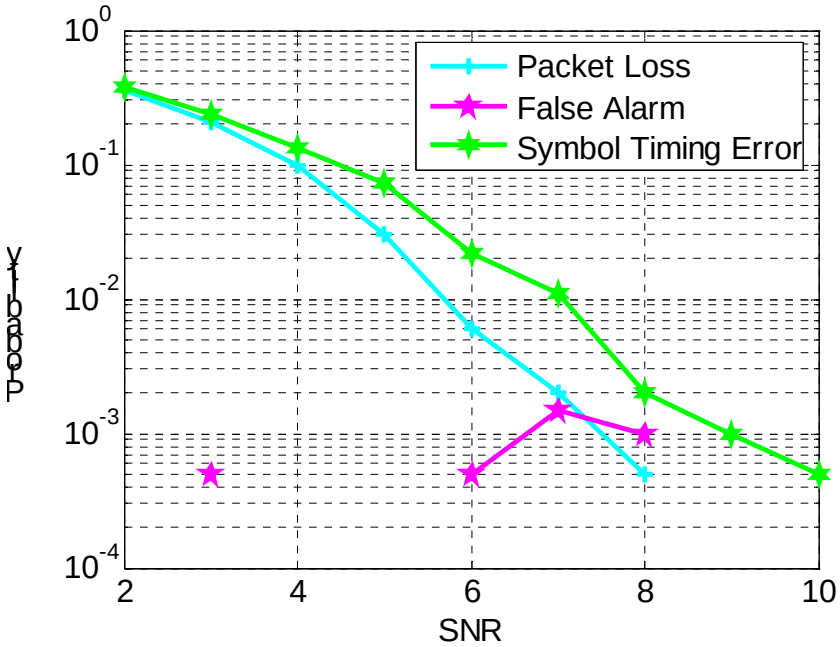

Figure 4.6 Symbol Timing Error in SPW11a channel

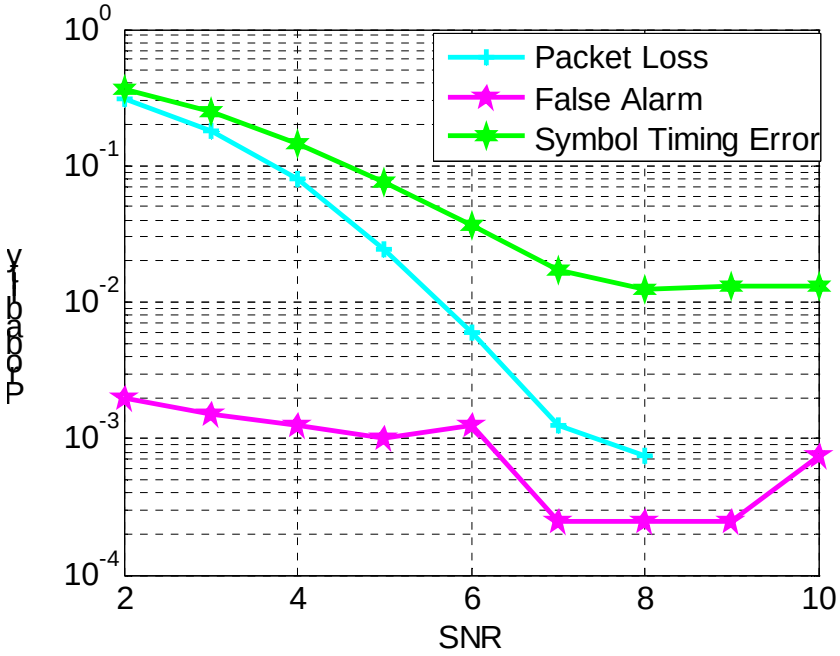

Figure 4.7 Symbol Timing Error in IEEE fading channel

Chapter 5

Hardware Implementation

5.1 Fixed Point Simulation Result

Figure 5.1 illustrate the fixed point simulation result. The simulation environment is under SPW11a fading channel with RMS 50ns and CFO 50 p.p.m. The results are obtained by applying 1000 simulation runs with different SNR. The data rate is 6 Mbps and PSDU length is 500 bytes. The required SNR for PER 8% is about 6.6 dB for both float-point and fixed-point, which satisfy the standard requirement.

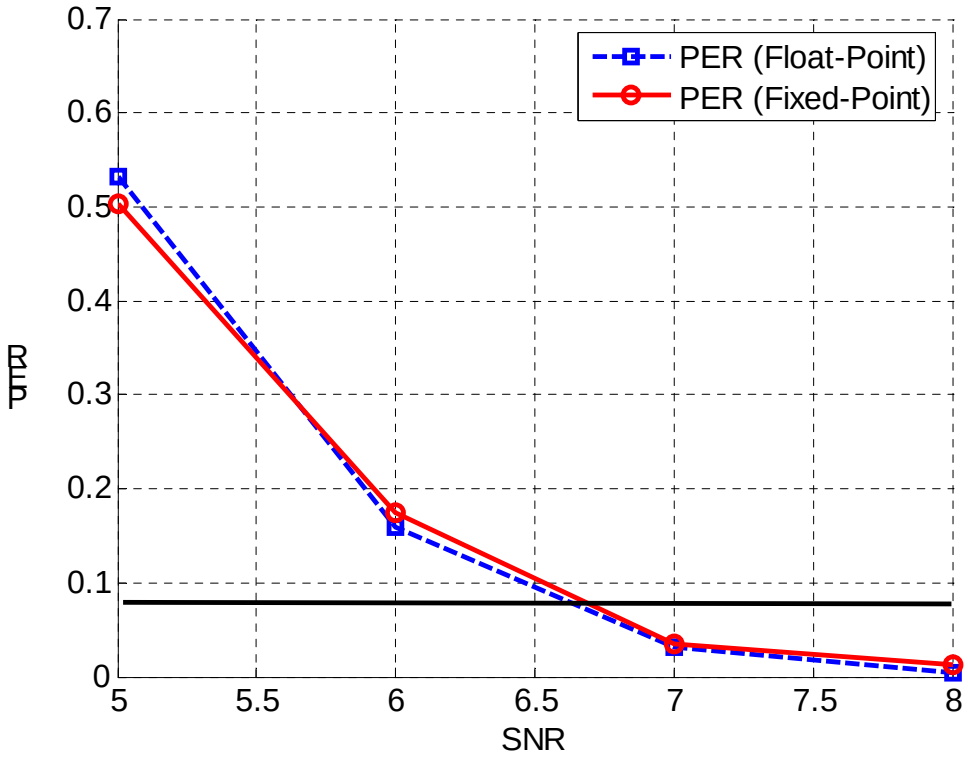

Figure 5.1 PER versus SNR with float-point and fixed-point

5.2 Matlab to Verilog Design Flow

To make sure that the Verilog behaviors 100% match Matlab simulation results, several things must be confirm. We show the flowchart of Matlab to Verilog design in figure 5.2. At First, we must find out the word length of variables in the Matlab platform, due to variables in Verilog only have finite precision. Fixed-point simulations are performed to ensure that these changes will not degrade system performance serious. It is trade-off between cost and performance. Second, once the fixed-point simulations are done, we use Matlab to generate input and output data of desired Matlab function, and feed the input data to the designed Verilog function. If the output data of Matlab are completely the same as the ones of Verilog, the Matlab to Verilog design is done.

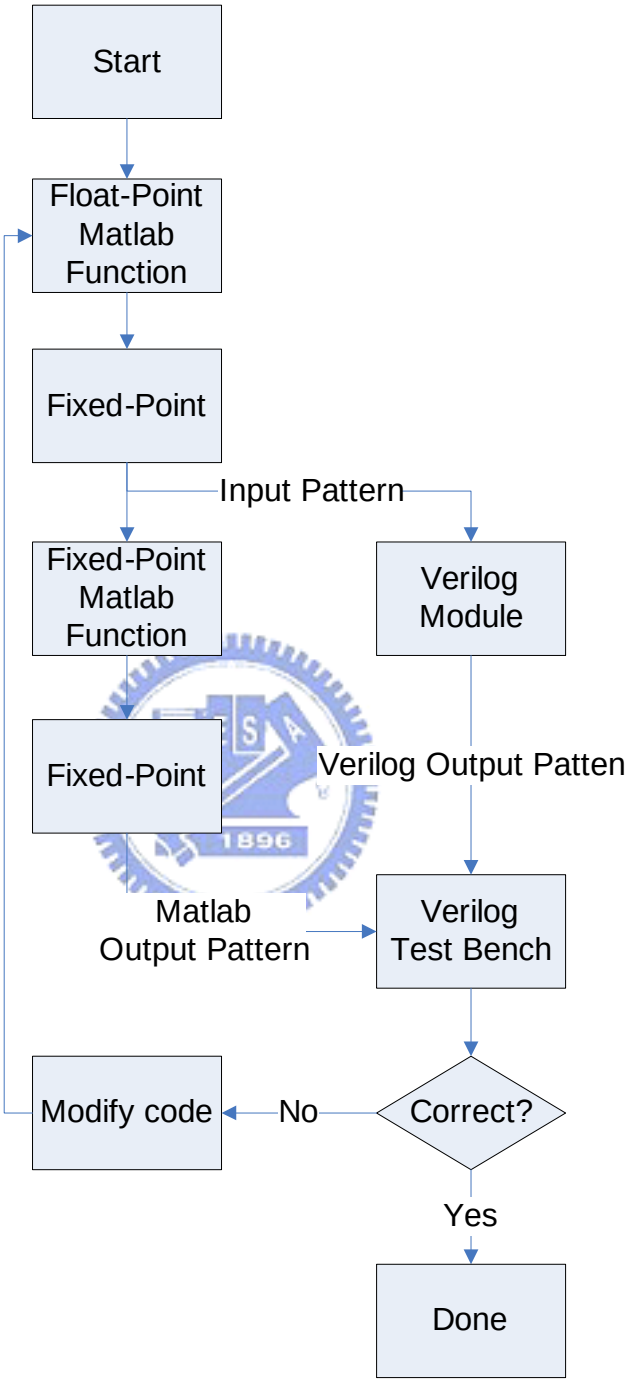

Figure 5.2 Flowchart of Matlab to Verilog design

5.3 Design Architecture

The whole architecture of the proposed algorithms can be divided into three parts, the OFDM packet detection, DSSS packet detection, and OFDM symbol timing estimation. The block diagram of OFDM/DSSS packet detection is depicted in figure 5.3. The detection control consist of two small self-run FSM, one for OFDM packet detection, and the other for DSSS packet detection. At first, the data are saved into 32-element shift registers. The OFDM packet detection, DSSS packet detection, and the OFDM symbol timing estimator share the shift registers.

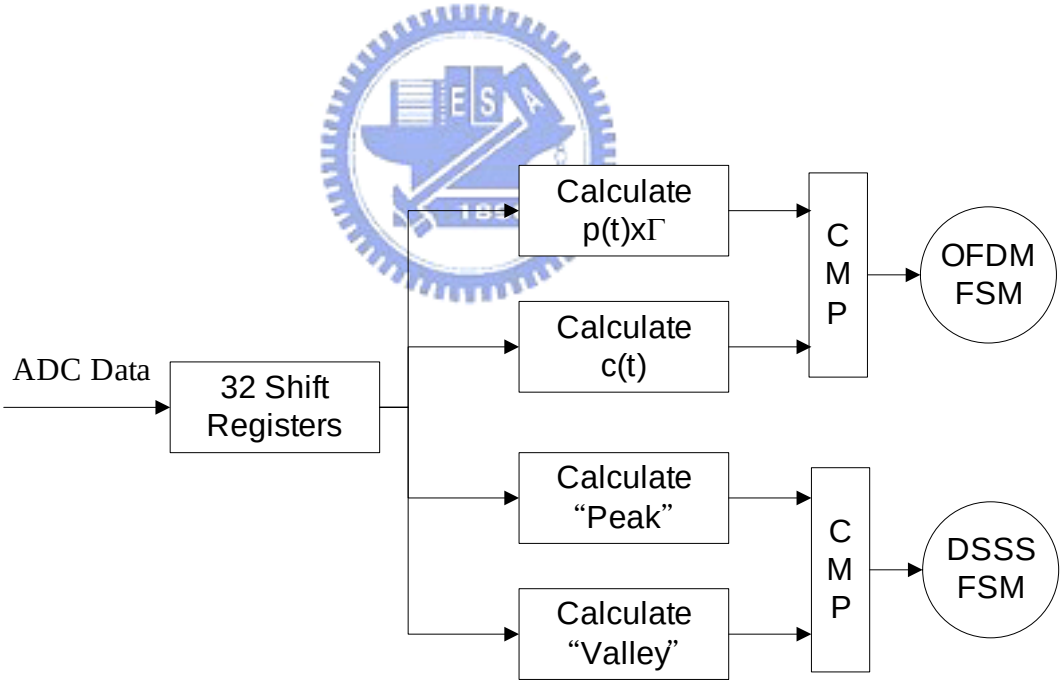

Figure 5.3: block diagram of OFDM/DSSS packet detection

Figure 5.4 shows the architecture of OFDM packet detection. The architecture of OFDM packet detection can be implemented with the iterative formula by transform the equation (3.1) and (3.2) into

$$\begin{aligned} c(t) &= c(t-1) \\ &\quad -r(t-1)r^*(t-1+L) \\ &\quad +r(t+L_a-1)r^*(t+L_a-1+L) \end{aligned} \tag{5.1}$$

And

$$p(t) = p(t-1) - \left|r(t-1)\right|^2 + \left|r(t+L_a-1)\right|^2 \tag{5.2}$$

Thus, we can save a lot of adder and complex multiplier into only one adder, one complex multiplier, and extra shift registers to save the subsequence calculations. And these calculations are rounded to nearest integer before further usage. The FSM control SEL_10M to indicate the current sampling rate, 10 MHz or 20 MHz. If the sampling rate is 10 MHz, one short preamble only consists of 8 samples. The output of complex multiplier is the autocorrelation of one sample and the square value is the corresponding power. The square part can be implementing with Look-up Table (LUT) rather than multiplier to reduce hardware cost. The auto-correlation and power of one sample are saved in the Corr_FIFO and Power_FIFO, respectively. According to equation (5.1) and (5.2), the summations in equation (3.1) and (3.2) are replaced with accumulation. Finally, the criterion for OFDM packet detection can be achieved by comparing the auto-correlation *c(t)* and received power *p(t)*, as mention in equation (3.4). The comparison results are sending to the OFDM FSM to generate the control signal.

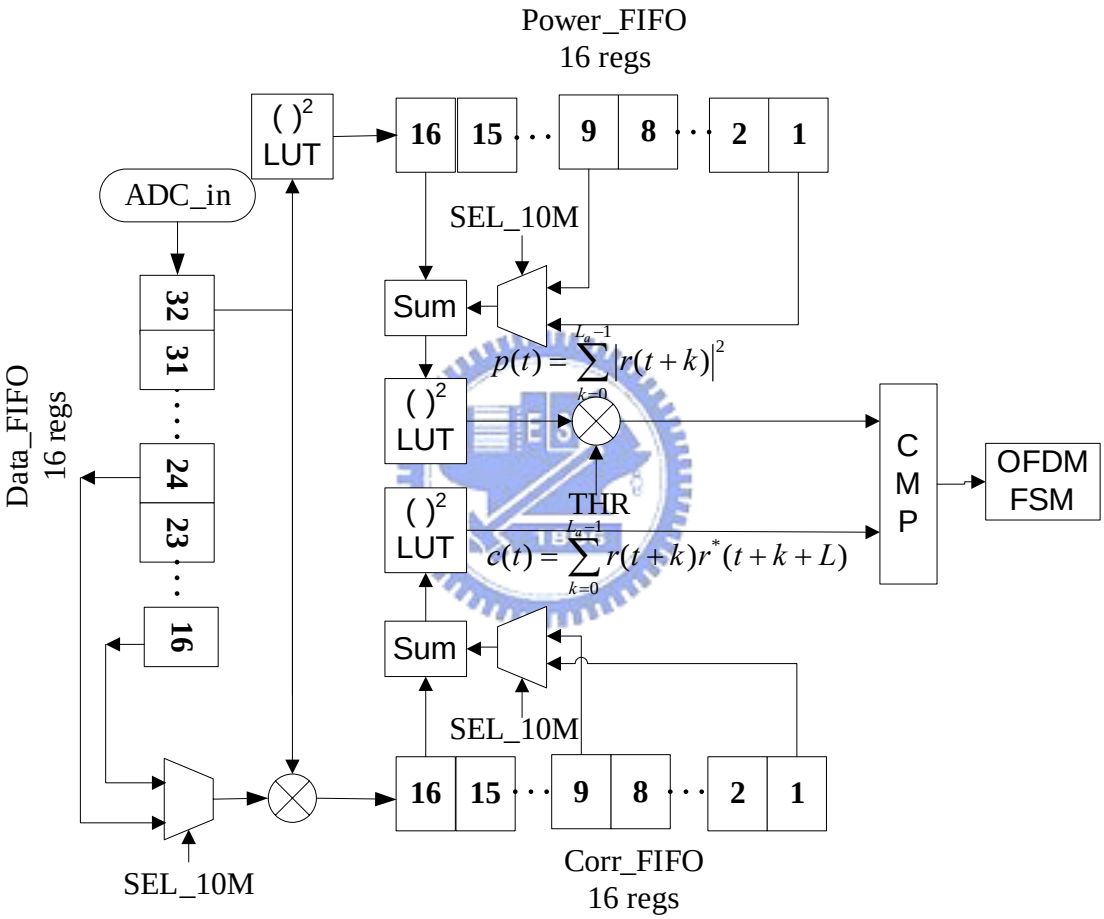

Figure 5.4 Architecture of OFDM Packet Detection

Figure 5.4 shows the architecture of DSSS packet detection. The Barker peak is obtained by de-spreading the received data with 11-chip Barker sequence. And then the data path can be divided into two branches. The upper branch is to calculate the "Peak" in equation (3.7). First, it finds every max peak in each window, and then checks the interval of every adjacent peak to see if the interval is the same as 100 ns. The lower branch is to calculate the "Valley" in equation (3.7). It utilizes a shifter register to calculate the summation of "Valleys". Finally, the "Peak" value and the "Valley" value are compared, and the results are used by the DSSS FSM to make the decision.

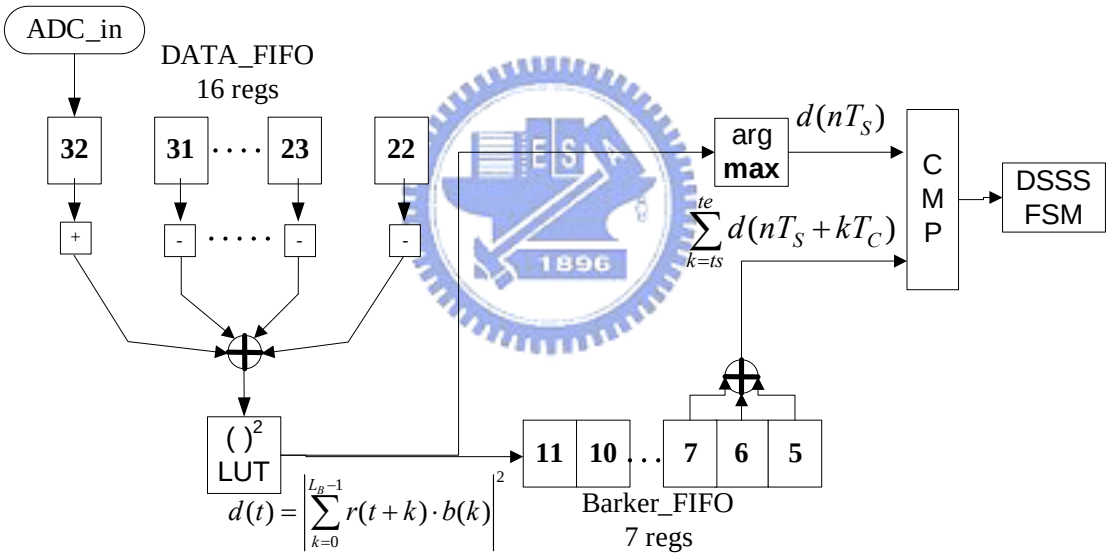

Figure 5.5 Architecture of DSSS Packet Detection

If both detector claims simultaneous, the OFDM packet is declared in consequence of the length of OFDM preamble is much shorter than DSSS preamble.

Figure 5.5 shows the architecture of OFDM symbol timing estimator. The data in shift registers are correlated with a pre-known long preamble, we only user the first half of long preamble, which are 32 samples. After that, the correlation value is used to calculate the ξ_{max} and a counter $\lambda(t)$ in equation (3.13) and (3.14).

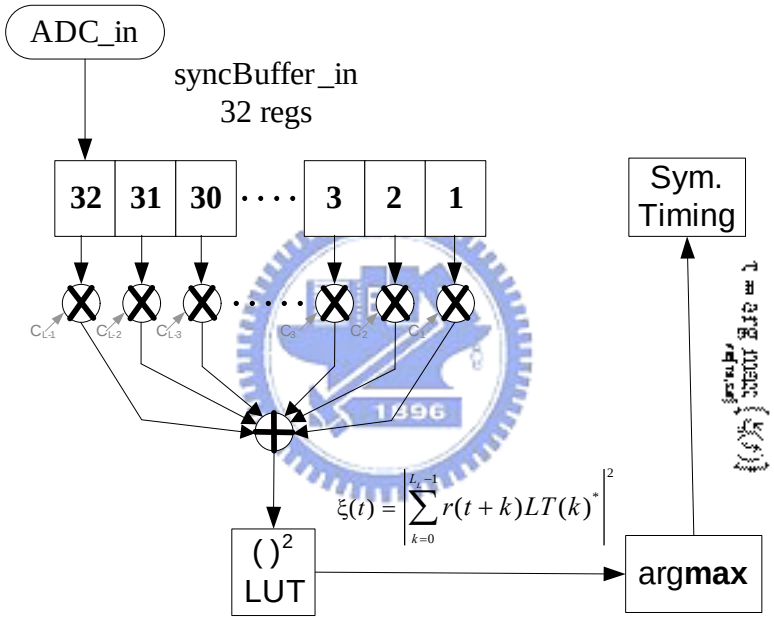

Figure 5.6 Architecture of OFDM Symbol Timing Estimator

5.4 Implementation Result

An IEEE 802.g PHY receiver is implemented to verify the feasibility. Table 5.1 lists the chip characteristics of the IEEE 802.11g PHY receiver. The process is UMC .13um standard cell library and Cadence BGX is used for the synthesis. The maximum available clock rate is 20 MHz and the area of synchronizer is 82105 um^2, about 20k gate counts. The implantation loss is slight and conforms to the requirements. In addition, the proposed synchronizer is applied in the chip of 11g baseband receiver by isIP Lab as demonstrated in the Figure 5.6. Due to we only have limit area, the shape of layout is rectangle.

Table 5.1: IEEE 802.11g PHY receiver chip characteristics

Process	UMC .13um standard cell	
PAD Number	43 pins (PAD limit)	
System Clock	20 MHz	
	SYNC	TOTAL
Gate Count	20 K	170 K

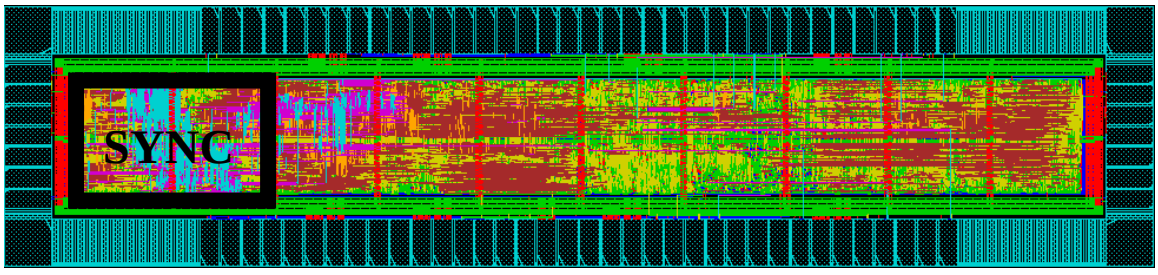

Figure 5.7 Layout of 802.11g PHY receiver

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we propose a packet synchronizer that can handle successful both OFDM and DSSS packets in dual-mode (OFDM/DSSS) systems with only one set of ADCs and dynamically controlling the sampling frequency. By this way, we can achieve the low-power approach in the wireless baseband processor, whereas keep the overall performance met standard requirements. Not only the algorithm itself, but also the parameters of algorithm, threshold decisions, are carefully discussed. The threshold decision is as important as the proposed algorithm. To hardware implementation, the fixed-point Matlab simulations decide the word length of every module. According to the fixed-point simulation result, a test chip using 0.13 um technology is implemented to verify the feasibility.

6.2 Future Work

There are some possible improvements in the future works. First, although the ADPLL can dynamically controlling the sampling frequency, the practical and detail behavior of ADPLL are not considered in this thesis. We just assume that the sampling frequency will change as soon as we want. But it is impossible in a real world, and there will be some transaction time and jitter will occur. So a detailed ADPLL model is necessary in the Matlab simulation. Once the ADPLL model is constructed, a real ADPLL can be hardware implemented in the proposed packet synchronizer. Second,

the adaptive threshold decision as mentioned in subsection 3.3.1 is a good approach to further improve the performance and robustness in packet detection. In the adaptive threshold approach, the threshold can adapt to the channel condition, and the false alarm and packet loss probability can be reduced.

References

[1] *Supplement to standard for LAN/MAN part 11: Wireless MAC and PHY specifications: High-speed physical layer in the 5GHz band*, IEEE 802.11a, 1999.

[2] *Supplement to standard for LAN/MAN part 11: Wireless MAC and PHY specifications: Higher-Speed Physical Layer Extension in the 2.4GHz Band*, IEEE 802.11b, 1999.

[3] *Multi-band OFDM Physical Layer Proposal for IEEE 802.15 Task Group 3a*, IEEE P802.15-03/268r3, 2003.

[4] *DS-UWB Physical Layer Submission to 802.15 Task Group 3a*, IEEE P802.15-04/0137r3, 2004.

[5] Nakajima, N.; Kohno, R.; Kubota, S., "*Research and developments of software-defined radio technologies in Japan*", Communications Magazine, IEEE, Vol. 39, Issue: 8, Aug. 2001

[6] Timothy M. Schmidl and Donald C. Cox, "*Robust Frequency and Timing Synchronization for OFDM*", IEEE Transactions on Communications, Vol. 45, NO. 12, Dec 1997

[7] Alan J.Coulson, "*Maximum Likelihood Synchronization for OFDM, Using a Pilot Symbol: Algorithms*", IEEE Journal on Selected Areas in Communications, Vol. 19, NO. 12, Dec 2001

[8] Speth, M., Gunnar Fock, D., Meyr, H., "*Optimum Receiver Design for Wireless Broad-Band Systems Using OFDM-Part I*", IEEE Transactions on Communications, Vol. 47, NO. 11, Nov 1999.

[9] Shih-Lin Lo, "*The study of Front-End Signal Process for Wireless Baseband Applications* ", NCTU, master thesis, June 2003

[10] Jan-Jaap van de Beek, Magnus Sandell, and Per Ola B¨orjesson, "*ML Estimation of Time and Frequency Offset in OFDM systems*", IEEE Transactions on Signal Processing, Vol. 45, NO 7, July 1997

[11] H. Minn, M. Zeng, and V.K. Bhargava, "*On timing offset estimation for OFDM systems*", IEEE Communcations Letters, Vol. 4, NO. 7, July 2000

[12] Kun-Wah Yip, Yik-Chung Wu, and Tung-Sang Ng, "*Timing-Synchronization Analysis for IEEE 802.11a Wireless LANs in Frequency-Nonselective Rician Fading Environments*", IEEE Transactions on Wireless Communications, Vol. 3, NO. 2, March 2004