Robust, Low Complexity and Energy Efficient Baseband Receiver Design for MB-OFDM UWB

FAN, Wen

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

in

Electronic Engineering

The Chinese University of Hong Kong

March 2011

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Abstracts

Ultra-wideband (UWB) technology, targeting at wireless personal area networks (WPANs), brings the convenience of high-speed and short-range wireless interconnects. As a novel communication technique, multi-band orthogonal frequency division multiplexing (MB-OFDM) UWB has the features of high spectrum efficiency, multiple access capability and robustness against narrow band interference. However, its inherent high complexity and the requirement of powerful processing for good performance are the obstacles for practical application.

Synchronization plays the key role for the performance of the whole baseband receiver system. In this dissertation, a dual-threshold (DT) packet detection scheme is proposed for timing synchronization. Compared with traditional cross-correlation algorithm, DT has much better detection performance, especially in high noise environment.

As an OFDM-based system, MB-OFDM UWB is vulnerable and sensitive to carrier frequency offset (CFO). We employ multipartite table method (MTM) to implement arctangent and sin/cos functions for frequency synchronization. Compared with traditional algorithms, MTM has the advantages of low cost, low power consumption and higher processing speed. Residual phase distortion is corrected by a highly simplified phase tracking method, which also has better performance compared with traditional phase tracking scheme.

Architectures of matched filter in packet detector, CFO corrector, FFT output reorder buffer and phase tracking block are optimized for low complexity. Implemented in 0.13-μm CMOS technology, the proposed baseband receiver system has the core area of 2.5 mm^2 and the estimated power consumption is 170 mW, which is equivalent to the energy efficiency of 88 pJ/b at 480 Mbps data rate. The implementation results verify the robustness, low complexity and power efficiency of the proposed MB-OFDM UWB baseband receiver.

摘 要

應用於無線個人局域網的超寬頻技術為高速和短距離的無線互連帶來了便利。多頻帶的超寬頻正交頻分複用作為一項新的通訊技術，有著頻譜利用率高，能夠多重訪問和有效對抗窄頻信號干擾的特點。但是，這種超寬頻技術自身的高複雜度以及爲了保證系統性能，對信號處理能力的較高要求，成為將這項技術應用于實際產品的障礙。

同步技術對於整個基帶接收器系統的性能起著決定性作用。在這篇論文中提出了用於時序同步的雙閾值數據包探測方案。與傳統的交叉相關算法比較，我們提出的雙閾值算法有更好的性能，尤其是在高噪聲的環境中。

作為基於正交頻分複用的系統，多頻帶的超寬頻正交頻分複用技術對於載波頻率偏移非常敏感。我們在頻率同步的反三角函數，正弦和餘弦函數的實現中採用一種壓縮型查找表方案。與其它傳統方案相比，這種壓縮型查找表方案在硬件實現上具有低成本，低功耗和高速的優點。剩餘的相位偏差由一種高度簡化的相位跟蹤方法來矯正，這種相位跟蹤方法與傳統方法相比有更好的相位矯正性能。

在架構方面，數據包探測器中的匹配濾波器，載波頻率偏移矯正器，快速傅利葉變換輸出重排緩衝器和相位跟蹤器的結構上都進行了優化，以實現低複雜度的設計。提出的基帶接收器設計方案用 0.13-μm CMOS 技術實現，核心面積為 2.5 mm^2，版圖后的功耗估計為 170 mW，該功耗在 480 Mbps 數據速率下，等效于 88 pJ/b 的能量效率。實現結果證實了提出的多頻帶的超寬頻正交頻分複用基帶接收器具有高性能，低複雜度和低功耗的特點。

Contents

List of Figure

List of Table

Acknowledgement

This large project could not have been completed without the support and assistance of many others. Firstly, I would like to express my thanks to my advisor, Professor Choy Chiu Sing, for his years of support and patient guidance. His rigorous scientific research approach is an example to follow. I would also like to thank Professor LEUNG Ka Nang and Professor PUN Kong Pang for their great efforts on project funding application. And I would like to express my gratitude to laboratory technician, Mr. YEUNG Wing Yee, who has helped me a lot for CAD tools and tape-out works.

AI Yan Qing, Steve and Brian also deserve thanks for helping implementation and optimization on the UWB baseband receiver design. Xu Ke, Xin Lin, Zhang Ming, Guo Jianping, Jia Jingbing, Shi Weiwei, Tang Xian, Zhao Yingsheng and Liu Zhouyi gave me much help on code writing and physical design. Without their company and encouragement, I cannot finish my research in three years.

Finally, I would like to appreciate my family, my parents and my husband Guo Xu. Although there is a long distance between us, their concern and love are the power of my study. I apologize if I fail to acknowledge everyone who deserves it.

Chapter 1 Introduction

According to the rules set forth by the FCC on February 14, 2002 [1], ultra-wideband (UWB) is defined as the system that occupies more than 20% of a center frequency or more than 500 MHz bandwidth. For communications systems, the available spectrum is 7.5 GHz, from 3.1 to 10.6 GHz.

UWB has several features that differentiate it from conventional narrowband systems [2]:

- Large instantaneous bandwidth enables fine time resolution for network time distribution, precision location capability.
- Short duration pulses are able to provide robust performance in dense multi-path environments by exploiting more paths that are resolvable.
- Low power spectral density allows coexistence with existing users and has a Low Probability of Intercept (LPI).
- Data rate may be traded for power spectral density and multipath performance.

The potential for UWB technology includes to be used for very high-throughput and short-range applications like high-speed cable replacement (wireless USB), video distribution within the room and fast image downloads from a camera to a wireless kiosk [3]. However, there are also a number of other uses of the technology that are currently being developed. These include low-rate, low-power sensors, inventory tracking and cataloging devices, building material analysis and radar and position location-based applications.

1.1 Motivation

The exciting new applications that are emerging now or will emerge over the next few years will demand extremely fast speed. Speed is required for some reasons. Either the application involves a large file transfer, or high-resolution video streaming.

In the case of large file transfers, speed translates into consumer wait time. Instead of taking hours to transfer the contents of a DVD on to an 802.11a wireless local area network (WLAN), a UWB link could be expected to reduce this time to minutes in the current generation, moving to two minutes in the second [2], [4].

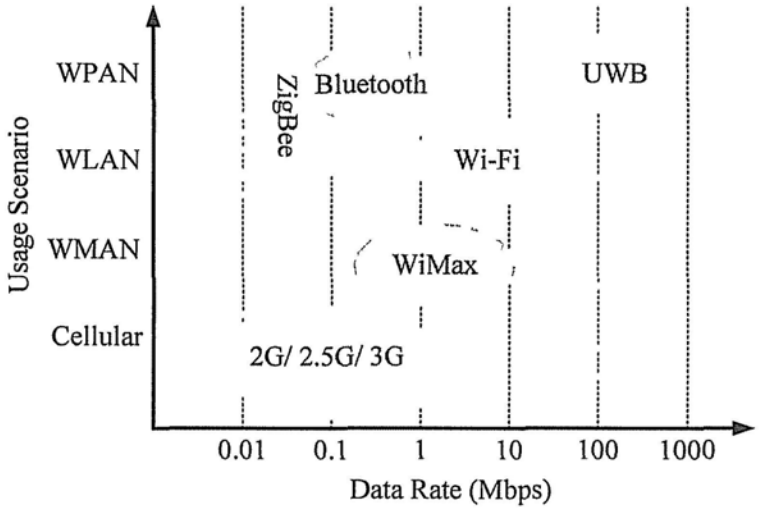


Figure 1.1 Data rate and usage range for various wireless standards

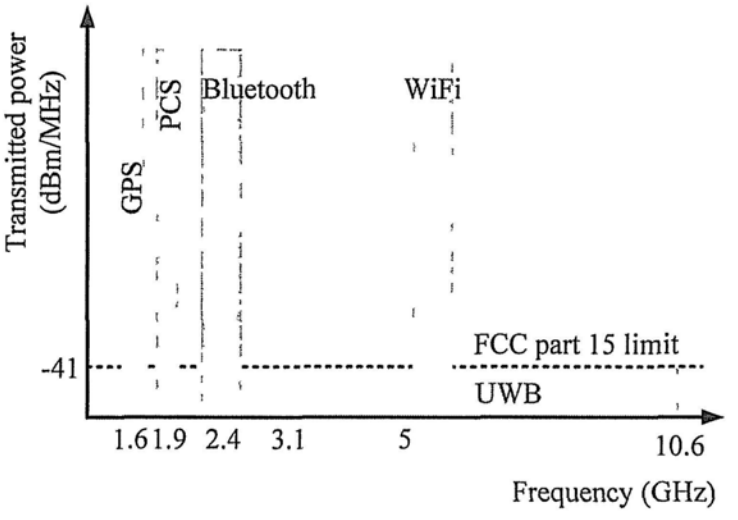


Figure 1.2 Spectrum allocation of different wireless standards

The data rate and usage range for different wireless standards are shown in Figure 1.1. The existing wireless communication techniques such as Wi-Fi and Bluetooth are not suitable for transferring large files due to the data rate and power consumption

limitations. Take a head-to-head technology comparison, UWB is five times faster than Wi-Fi. It can transmit data up to 200 times faster than Bluetooth.

The spectrum allocation of different wireless standards is shown in Fig. 1.2. FCC limits the emission power spectral density to -41.3 dBm/MHz, which is extremely small compared with other wireless systems. Compared with Wi-Fi, there is tenfold increase in power efficiency of UWB.

In addition, UWB offers better user density due to the feature of multiple access capability. In addition, as a WPAN technology, Wi-Fi has the limitations of interference and usage models, which are not associated with UWB.

The recent developments and designs for UWB wireless communication systems are based on two technologies, impulse radio (IR) and multi-band orthogonal frequency division multiplexing (MB-OFDM). Compared with IR-UWB, MB-OFDM approach has the best potential for very high rate and allows for good coexistence with narrowband systems (such as 802.11a), adaptation to different regulatory environments, future scalability and backward compatibility.

UWB has the potential to be applied in radar imaging systems, medical imaging systems, wireless high-definition multimedia interface and the wireless connectivity of consumer electronic devices. Some of the more notable applications to other communication systems include wireless universal series bus (WUSB), IEEE 1394, the next generation of high data rate Bluetooth and Universal Plug and Play (UPnP).

For MB-OFDM UWB receiver, synchronization mechanism plays a key role in the performance of the whole system. The mission of timing synchronization includes packet detection, symbol boundary detection and fast Fourier transform (FFT) window adjustment. It can avoid or alleviate the packet loss and inter-symbol interference (ISI) caused by timing error. As an OFDM-based system, MB-OFDM UWB is vulnerable and sensitive to carrier frequency offset (CFO), which will lead to inter-carrier interference (ICI), Doppler Effect on phase shift and degradation in signal noise ratio (SNR). Therefore, accurate frequency synchronization, aiming at CFO estimation and compensation, is the guarantee of the system reliability. Although CFO can be coarsely estimated and compensated using preambles in time domain, the residual

CFO (RCFO), sampling frequency offset (SFO) and the common phase error will lead to accumulated phase shift over a period and thus degrade the system performance if they are not well tracked and compensated. In MB-OFDM UWB systems, pilot subcarriers can help to solve the phase error problem in frequency domain.

In OFDM-based system, FFT module transforms signals between the time domain and the frequency domain and occupies a large portion of the circuit area as well as the power consumption. The buffers also contribute a significant portion of area and power in MB-OFDM UWB baseband receiver. The low power and low complexity strategies on FFT module and the buffers will make the system optimization more efficient.

1.2 Challenge

While UWB has many reasons to be an exciting and useful technology for future wireless communications, there are many challenges still to overcome. The challenging issues are summarized into three categories below and investigation into innovative methods is therefore a pressing issue.

- Large dynamic range of received power and SNR

  Since UWB occupies such a wide band, the interference from other communication systems could be very serious, which makes the packet much more difficult be detected.

- High complexity in hardware

  The order of hardware complexity is proportional to both the frequency diversity and the type of Time-Frequency Codes (TFC), which enables multiple access. High hardware complexity is therefore expected.

- RF-baseband iteration control

  The long synchronization processing delay of the feedback loop from baseband to RF limits the system operation speed and degrades the performance.

The inherent high complexity, the requirements of low cost, low power and good performance indicate the implementation of MB-OFDM UWB baseband receiver is quite a challenging work.

The synchronization technique has been extensively studied for years. Some works are based on OFDM systems [5]-[14]. Although MB-OFDM UWB systems can leverage on the successful experiences of OFDM, they cannot utilize the traditional synchronization technology directly due to the distinct features of UWB and the requirement of the quality of service (QoS). Some of the works just dedicate to the algorithm aspect of MB-OFDM UWB systems [15]-[21], but not provide the approaches for real applications. Most of the recent circuit solutions for UWB digital baseband receiver design aim to IR-UWB systems [22]-[28] but very few on MB-OFDM UWB systems.

The challenge for UWB radio implementation is fully exploit the wideband nature for lower power and a less costly solution than existing narrowband techniques. Therefore, we aim at developing a robust, low-complexity, high-speed and energy efficient MB-OFDM UWB baseband receiver to meet the practical application requirements.

1.3 Dissertation Outline

This dissertation is organized as follows.

Chapter 2 provides an introduction to UWB system, including the system architecture, band planning, frame format, channel model, constellation mapping, modulation scheme and link budget.

In Chapter 3, synchronization issues are discussed. They are divided into three parts: timing synchronization, coarse frequency synchronization and fine frequency synchronization. In each part, we firstly introduce some traditional algorithms. Then propose the novel methods to improve the performance and simplify the hardware

design. Finally, the architecture design of each part is presented.

In Chapter 4, FFT algorithms and architecture design are proposed. In addition, a new address encoding approach in FFT output reorder buffer is presented, which saves memory units significantly.

Channel estimation technique is discussed in Chapter 5.

In Chapter 6, after integrating the packet detector, CFO corrector, FFT, channel estimator and phase tracking block into a baseband receiver system, we analyze the bit-true implementation results and compare the proposed design with other similar baseband receiver designs.

Chapter 7 summarizes the thesis and lists the future work.

Chapter 2 Ultra-Wideband System

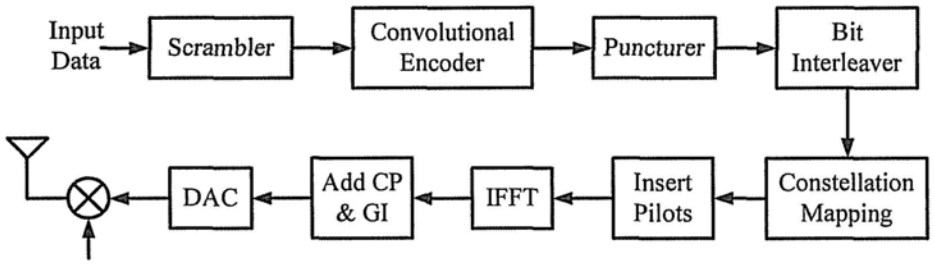
The basic idea of MB-OFDM is to divide spectrum into several 528 MHz bands. Information is transmitted using OFDM modulation on each band. The 128-point FFT/IFFT generates OFDM carriers efficiently. The constellation size of MB-OFDM is limited to quadrature phase shift keying (QPSK) for low complexity, but meanwhile which will reduce the internal precision [29]. Information bits are interleaved across all bands to exploit frequency diversity and provide robustness against multi-path and interference. Prefix provides robustness against multi-path even in the worst channel environments and the guard interval provides sufficient time for switching between bands.

In this chapter, we will introduce physical layer design of MB-OFDM UWB system as proposed in the IEEE 802.15.3a standard, including the architecture of transmitter and receiver, band planning for flexibility in spectrum management, the frame format, the channel model and the important system parameters, such as link budgets.

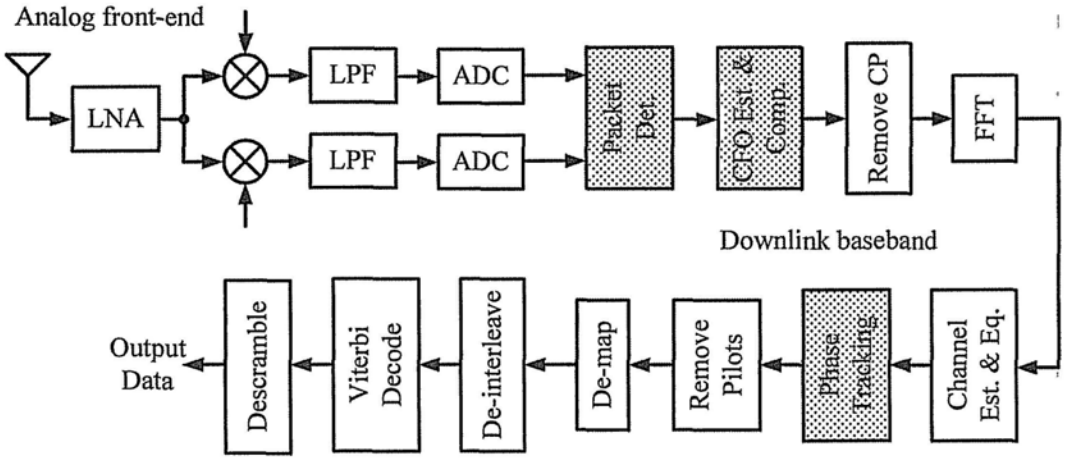
2.1 System Architecture

The architecture of MB-OFDM UWB transceiver is shown in Fig. 2.1. Fig. 2.1(a) is the structure of transmitter. The data bit stream is scrambled, encoded with the convolutional code, punctured, interleaved, mapped into a sequence of QPSK samples and OFDM-modulated. The resulting OFDM baseband signal is up-converted to a specified sub-band, passed through a power amplifier, and finally transmitted. A different pattern of band switching is assigned to different users in order to gain frequency diversity while minimizing the multiple access interference.

The corresponding receiver architecture is shown in Fig. 2.1(b). The RF signal received is passed through a preselect filter, amplified with a low-noise amplifier (LNA), down-converted to baseband, scaled in amplitude by a variable gain amplifier



(a) Block diagram of transmitter



(b) Block diagram of receiver

Figure 2.1 Transceiver architecture of MB-OFDM UWB system

(VGA) and then digitized. Before synchronization, the amplitude of the signal should be adjusted by auto gain control (AGC). After removing cyclic prefix (CP), the signal will be transformed from time domain to frequency domain. Channel effects will be removed by channel estimation and equalization blocks and the residual phase error will be corrected by phase tracking block. QPSK symbols are demodulated from the OFDM baseband signal. From the QPSK symbols, an estimated transmitted bit sequence is reconstructed and de-interleaved. Then the sequence is channel-decoded using a Viterbi decoder. Finally, the decoder output is descrambled, yielding an estimated sequence of the bit information transmitted.

2.2 Band Planning

In the MB-OFDM system, the whole available UWB spectrum is divided into several sub-bands with smaller bandwidth. This simplifies the design of the analog RF front end and analog-to-digital (ADC) and digital-to-analog converters (DAC). It also decreases overall power consumption. The bandwidth of each sub-band is larger than 500 MHz in compliance with the FCC rules for UWB transmission. Table 2.1 shows the band planning for the current MB-OFDM system. The bandwidth of each sub-band is 528 MHz. The sub-bands are assigned into 5 different groups. Group 1~4 have three sub-bands each and group 5 has two sub-bands.

Table 2.1 MB-OFDM band planning: divide UWB spectrum into 14 sub-bands [30]

Band group	Sub-band	Lower frequency (MHz)	Center frequency (MHz)	Upper frequency (MHz)
1	1	3168	3432	3696
	2	3696	3960	4224
	3	4224	4488	4752
2	4	4752	5016	5280
	5	5280	5544	5808
	6	5808	6072	6336
3	7	6336	6600	6864
	8	6864	7128	7392
	9	7392	7656	7920
4	10	7920	8184	8448
	11	8448	8712	8976
	12	8976	9240	9504
5	13	9504	9768	10032
	14	10032	10296	10560

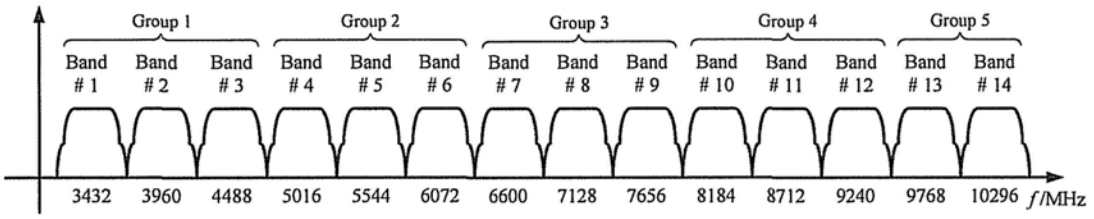


Figure 2.2 Band allocation for five distinct groups

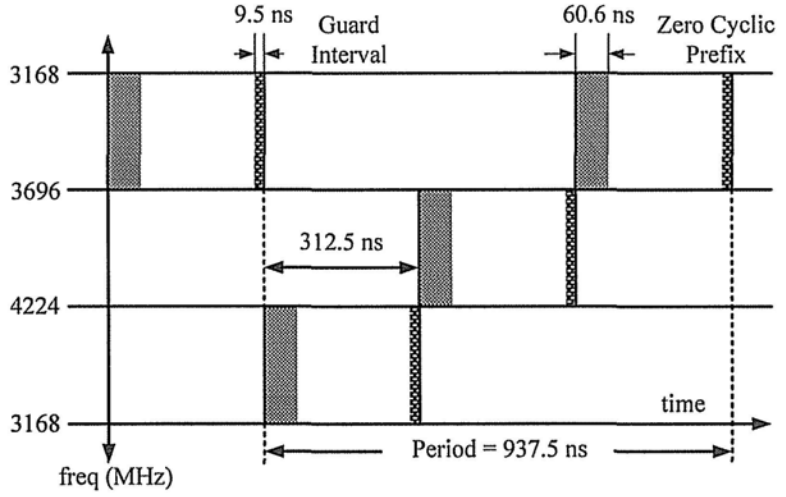


Figure 2.3 Time-frequency representation of MB-OFDM system with TFC 2

Table 2.2 Time-frequency codes for different piconets

Piconet number	Time-frequency codes
1	1 2 3 1 2 3
2	1 3 2 1 3 2
3	1 1 2 2 3 3
4	1 1 3 3 2 2

It was argued that due to path loss, attenuation, and losses in current RF technology, the use of frequency bands above 5 GHz yields only minor improvements in capacity at the cost of increased receiver complexity. For this reason, it was suggested that only the three sub-bands be located below 5 GHz (band group 1) as a mandatory mode. Support for the other band groups is optional at this time and will be added in the future. Fig. 2.2 shows the band allocation for five distinct groups.

Another technique for multi-user access is sub-band hopping. Every device uses only one sub-band group to transmit and receive data. The carrier frequency hops between bands according to time-frequency code (TFC) of length 6. Each code specifies the center frequency for the transmission of each symbol. Fig. 2.3 illustrates a time-frequency representation of MB-OFDM signal with TFC {1 3 2 1 3 2}. The guard interval for TX/RX switching is 9.5 ns and the zero cyclic prefix is 60.6 ns. The TFC is used to provide frequency diversity as well as to enable simultaneous piconet operation with little multiple access interference.

In MB-OFDM proposal for the IEEE 802.15.3a standard, a set of four TFCs are proposed, as shown in Table 2.2. These codes are designed such that the average number of collisions between any two codes is 1/3 [29].

2.3 Frame Format

The frame format including physical layer convergence procedure (PLCP) preamble, header, multimedia access control (MAC) frame body, tail bits and pad bits of MB-OFDM is shown in Fig. 2.4. Additionally, an optional sequence will be included after the physical (PHY) header when frame payload is transmitted using Mode 2. The PHY layer first pre-appends the PHY header plus the tail bits to the MAC header and then calculates the header check sequence (HCS) over the combined headers and tail bits. The tail bits are added after the PHY header in order to return the convolutional encoder to the "zero state". The resulting HCS is appended to the end of the MAC header along with an additional set of tail bits. Pad bits are also adder after the tail bits in order to align the data stream on an MB-OFDM symbol boundary. Tail bits are also added to the MAC frame body in order to return the convolutional encoder to "zero state". If the size of the MAC frame body plus tail bits are not an integer multiple of the MB-OFDM symbol, then pad bits are added to the end of the pad bits in order to align the data stream on the MB-OFDM symbol boundary.

As shown in Fig. 2.4, the PLCP header is always sent at an information data rate of 55 Mb/s and is always transmitted using Mode 1. The remainder of the PLCP frame

is sent at the desired information data rate of 55~480 Mb/s using either Mode 1 or Mode 2.

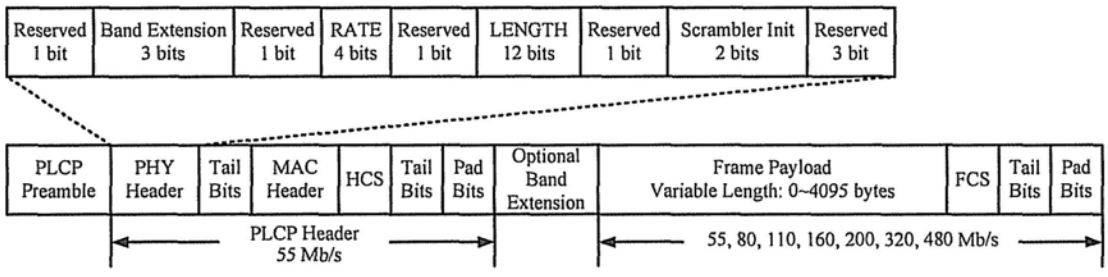


Figure 2.4 Frame format for a Mode 1 device

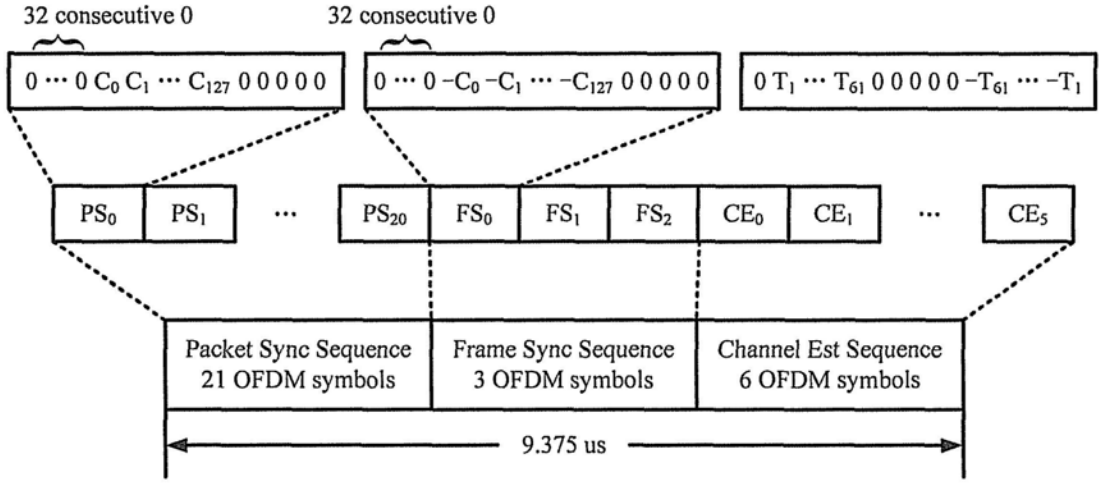


Figure 2.5 Standard preamble format for MB-OFDM UWB

The preamble of MB-OFDM UWB system aids receiver algorithms related to synchronization, carrier offset recovery and channel estimation. The standard preamble, which is shown in Fig. 2.5, consists of three distinct portions: packet synchronization sequence, frame synchronization sequence and channel estimation sequence. The packet synchronization sequence shall be constructed by successively appending 21 periods of a time domain sequence. Each piconet will use a distinct time domain sequence. Each period of the timing synchronization sequence shall be constructed by pre-appending 32 "zero samples" and by appending a guard interval of 5 "zero samples" to the 128 preamble coefficients. This portion of the preamble can be

used for packet detection and acquisition, coarse carrier frequency estimation and coarse symbol timing.

Similarly, the frame synchronization sequence shall be constructed by successively appending 3 periods of the 180 degree rotated version of the 128 preamble coefficients. This portion of the preamble can be used to synchronize the receiver algorithm within the preamble.

Finally, the channel estimation sequence shall be constructed by successively appending 6 periods of the MB-OFDM training symbol generated by passing the frequency domain sequence. This portion of the preamble can be used to estimate the channel frequency response, for fine carrier frequency estimation and fine symbol timing.

2.4 Channel Model

Analysis and design of UWB systems require an accurate channel. In this subsection, we will introduce the UWB channel model in IEEE 802.15.3a standard. The propagation models are divided into two categories: large- and small-scale models. The large-scale models characterize signal power over long distances and the small-scale models characterize signal behavior over a very short distance (up to 30m outdoors or a few meters in indoor scenarios). Large-scale models are described by pass loss models and shadowing. Specifically, path loss models provide an average value of the signal power as a function of the propagation distance. Shadowing characterizes the slow variation of the signal envelop over time around the deterministic path loss value. The path loss, shadowing and small-scale fading models of the standard UWB channel are provided below.

- *Path loss*: is defined as the ratio between the signal power at the transmitter and the signal power at the receiver. The path loss specified in the standard is based on free-space path loss, with the center frequency f_c given by $f_c = \sqrt{f_L f_H}$, where f_L and f_H are obtained at the −10 dB edges of the waveform spectrum.

- *Shadowing*: is defined as the slow vibration in signal attenuation around the mean path loss value. The shadowing is assumed lognormally distributed with standard deviation of 3 dB.
- *Small-scale fading*: adopted in the IEEE 802.15.3a standard is based on the S-V model. Although the path amplitude may follow the lognormal distribution, the Nakagami distribution, or the Rayleigh distribution, the lognormal distribution is adopted in the standard. Four sets of channel model parameters for different measurement environments were defined: CM1, CM2, CM3 and CM4. CM1 describes a line-of-sight (LOS) scenario with a distance between transmitter and receiver of less than 4m. CM2 describes the same range but for a non-LOS (NLOS) situation. CM3 describes a NLOS scenario for distances of 4~10m. CM4 describes an environment with strong delay dispersion, resulting in a delay spread of 25 ns.

The Saleh-Valenzuela (S-V) model was introduced for a wideband indoor channel. In the S-V model multipath arrivals are grouped into two categories: a cluster arrival and a ray arrival with a cluster. This model requires four main parameters: the cluster arrival rate Λ, the ray arrival rate within a cluster λ, the cluster decay factor Γ and the ray decay factor γ. The channel impulse response of the S-V model is expressed as

$$h(t) = \sum_{c=0}^{C} \sum_{l=0}^{L} \alpha(c,l)\delta(t - T_c - \tau_{c,l}) \tag{2.1}$$

where $\alpha(c, l)$ denotes the gain of the l-th multipath component in the c-th cluster. C is the total number of clusters and L is the total number of rays in each cluster. The time duration T_c represents the delay of the c-th cluster and $\tau_{c,l}$ is the delay of the l-th path in the c-th cluster relative to the cluster arrival time. The cluster and path arrivals within each cluster are modeled by Passion processes [29]:

$$p_{T_c}\left(T_c \middle| T_{c-1}\right) = \Lambda \exp\left[-\Lambda\left(T_c - T_{c-1}\right)\right], \quad c > 0 \tag{2.2}$$

$$p_{\tau_{c,l}}\left(\tau_{c,l} \middle| \tau_{c,l-1}\right) = \lambda \exp\left[-\lambda\left(\tau_{c,l} - \tau_{c,l-1}\right)\right], \quad l > 0 \tag{2.3}$$

The path amplitude $|\alpha(c, l)|$ follows the Rayleigh distribution. Specifically, the multipath gain coefficient $\alpha(c, l)$ is modeled as a zero-mean complex Gaussian random variable with variance [29]

$$\Omega_{c,l} = E[|\alpha(c,l)|^2] = \Omega_{0,0} \exp(-\frac{T_c}{\Gamma} - \frac{\tau_{c,l}}{\gamma}) \tag{2.4}$$

The four main parameters can be changed for different environments. They provide great flexibility to model very different environments. Fig. 2.6 illustrates the various parameters in the S-V model. Table 2.3 provides the model parameters of CM1 to CM4 and Fig. 2.7 shows the channel response of CM1 to CM4.

Table 2.3 Multipath channel model parameters

Parameters	CM1	CM2	CM3	CM4
Λ (ns^{-1})	0.0233	0.4	0.0667	0.0667
λ (ns^{-1})	2.5	0.5	2.1	2.1
Γ	7.1	5.5	14	24
γ	4.3	6.7	7.9	12

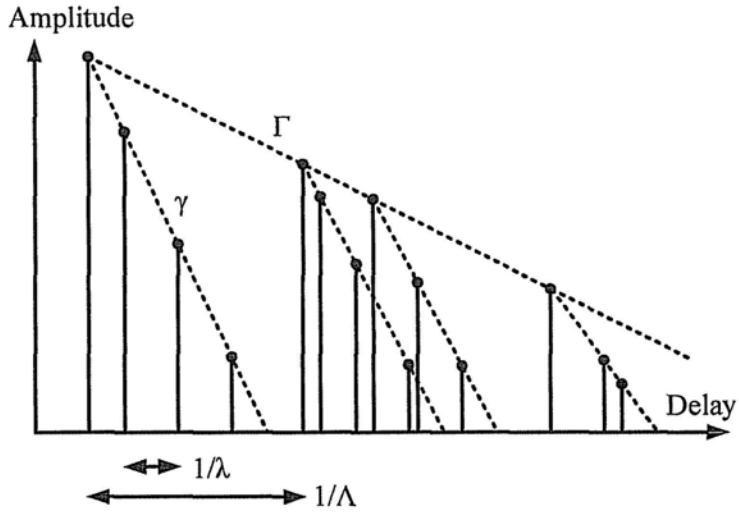


Figure 2.6 Principle of the S-V fading model

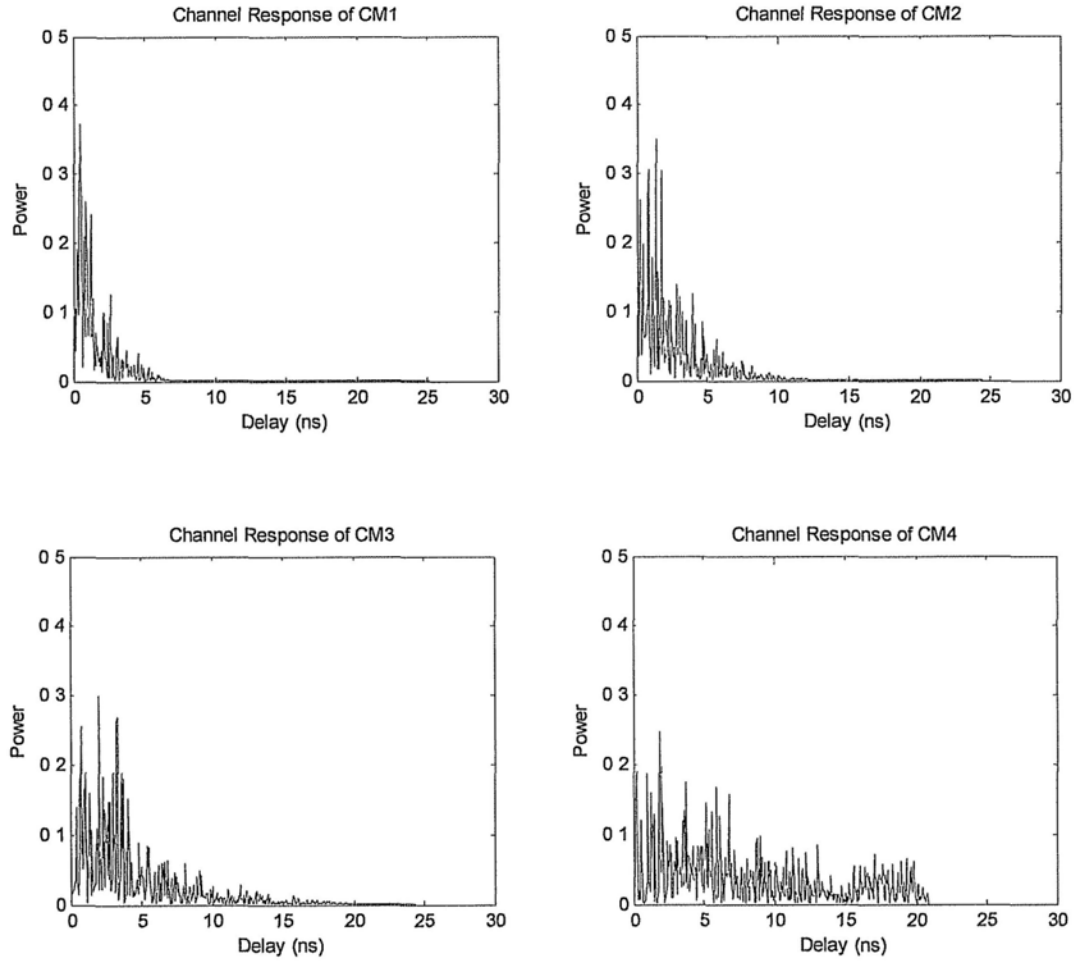


Figure 2.7 Channel response of CM1 to CM4

2.5 Constellation Mapping

The MB-OFDM subcarriers shall be modulated using QPSK modulation. The encoded and interleaved binary serial input data will be divided into groups of two bits and converted into complex number to represent QPSK constellation points. The conversion will be performed according to the Gray-coded constellation mappings, as shown in Fig. 2.8. The output value are formed by multiplying the resulting (I+*j*Q) value by a normalization factor as described below.

$$d = (I + jQ)K_{MOD} \tag{2.5}$$

The normalized factor K_{MOD} depends on the base modulation mode. For QPSK modulation scheme, $K_{MOD} = 1/\sqrt{2}$.

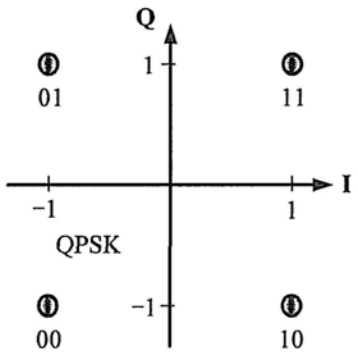


Figure 2.8 QPSK constellation bit encoding

2.6 OFDM Modulation

The complex-valued sequence generated from the constellation mapper is ready for OFDM modulation. The sequence in series is now converted to parallel, and the pilots, guards, and nulls are also inserted to the OFDM symbols before IFFT is taken. Each OFDM symbol contains 128 subcarriers. The duration for the OFDM symbol is 242.42 ns. Among the 128 subcarriers of the OFDM symbol, 100 data tones are used to transmit information. Twelve pilot tones are used to ensure the coherent detection robust against frequency offset and phase noise. Ten guard tones are used for a number of purposes, including relaxing the specifications on transmitting and receiving filters.

For modes with data rate of 50 and 80 Mb/s, the complex-valued sequence is divided into groups of 50 complex numbers. The data tones c_n (n is the subcarrier index) in Fig. 2.9 for the OFDM symbol k relate to the complex sequence d can be expressed as

$$\begin{aligned} c_{n,k} &= d_{n+50k} \\ c_{(n+50),k} &= d^*_{(49-n)+50k} \end{aligned} \tag{2.6}$$

where $n = 0, 1, \ldots, 49$, and $k = 0, 1, \ldots, N_{\mathrm{SYM}} - 1$, with N_{SYM} denoting the number of OFDM symbols. Since the same information is transmitted twice with two subcarriers, we obtain frequency diversity with a spreading gain factor of two in these two modes.

For modes with data rates larger than 80 Mb/s, such as 110, 160, 200, 320 and 480 Mb/s, the complex sequence is divided into groups of 100 complex numbers. The data

tones c_n for symbol k relate to the complex sequence d can be expressed as

$$c_{n,k} = d_{n+100k} \tag{2.7}$$

where n = 0, 1, …, 49, and k = 0, 1, …, N_{SYM} −1. In these modes, there is not frequency-diversity gain since each subcarrier conveys different information.

The subcarrier frequency allocation is shown in Fig. 2.9. To avoid difficulties in DAC and ADC offsets and carrier feed-through in the RF system, the subcarrier falling at DC is not used.

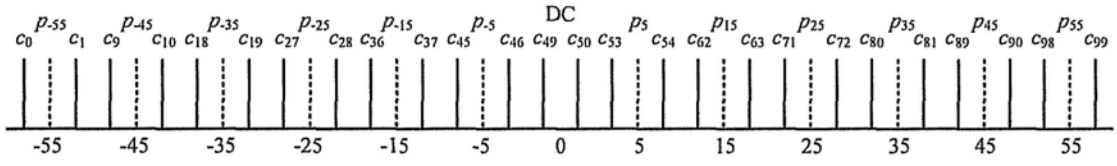


Figure 2.9 Subcarrier frequency allocation

In each MB-OFDM UWB symbol, the twelve pilots are located in subcarriers of -55, -45, -35, -25, -15, -5, 5, 15, 25, 35, 45 and 55. The pilots shall be binary phase shift keying (BPSK) modulated by a pseudo-random binary sequence p_l, which is generated by a linear feedback shift register (LFSR), to prevent the generation of spectral lines. The contribution due to the pilot subcarriers for the k-th symbol is given by the IFFT of the sequence P_n:

$$P_n = \begin{cases} (1+j)/\sqrt{2} & n = 15, 45 \\ (-1+j)/\sqrt{2} & n = -5, -25, -35, -55 \\ (-1-j)/\sqrt{2} & n = 5, 25, 35, 55 \\ (1-j)/\sqrt{2} & n = -15, -45 \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

The ten guard tones are located in subcarriers of -61, -60, …, -57 and 57, 58, …, 61. The same LFSR sequence p_l is used to generate the modulating data for the guard tones. The guard subcarrier symbol definition for the n-th subcarrier of the k-th symbol is given as follows:

$$\begin{aligned} P_{n,k} &= p_{\mathrm{mod}(k+l,127)}\left(\frac{1+j}{\sqrt{2}}\right), \quad l=0,1,2,3,4; \quad n=-61+l \\ P_{n,k} &= P^{*}_{-n,k}, \qquad\qquad\qquad\quad n=57,\ldots,61 \end{aligned} \tag{2.9}$$

In this numbering, it is assumed that $k = 0$ corresponds to the first channel estimation symbol CE0. The elements from the sequence p_l are selected independently for the pilots and the guard subcarriers.

2.7 System Parameters

In this subsection, we will introduce the data rate dependent modulation parameters, timing related parameters, link budget and receiver sensitivity of MB-OFDM UWB system.

The data rate dependant modulation parameters are listed in Table 2.4. The data rates of 55, 110 and 200 Mb/s are mandatory information data rate and the left data rates are optional information data rate. For all the data rates, the modulation scheme is QPSK. The data rates of 55 and 110 Mb/s have conjugate symmetric inputs to IFFT while others are not, which means just the data rates of 55 and 110 Mb/s have frequency spreading. And only 320 and 480 Mb/s data rate modes do not have timing spread.

Table 2.4 Data rate dependant parameters

Data rate (Mb/s)	*Modulation scheme*	*Coding rate*	*Conjugate symmetric input to IFFT*	*Timing spread*	*Spreading gain*	*Coded bites per symbol*
55	QPSK	11/32	yes	yes	4	100
80	QPSK	1/2	yes	yes	4	100
110	QPSK	11/32	no	yes	2	200
160	QPSK	1/2	no	yes	2	200
200	QPSK	5/8	no	yes	2	200
320	QPSK	1/2	no	no	1	200
480	QPSK	3/4	no	no	1	200

Table 2.5 Timing related parameters

Parameter	*Value*
Number of data subcarriers	100
Number of pilots subcarriers	12
Number of guard tones	10
Subcarrier frequency spacing	4.125 MHz (528 MHz /128)
IFFT/FFT period	242.42 ns
Cyclic prefix duration	60.61 ns (32/528 MHz)
Guard interval duration	9.47 ns (5/528 MHz)
Symbol interval	312.5 ns (242.42+60.61+9.47)

Table 2.6 Parameters of link budget and receiver sensitivity

Parameter	*Value*	*Value*	*Value*
Information data rate (Mb/s)	110	200	480
Average TX power (dBm)	−10.3	−10.3	−10.3
TX antenna gain (dBi)	0	0	0
Geometric center frequency of waveform (MHz)	3882	3882	3882
Total path loss (dB)	64.2 (@ 10m)	56.2 (@ 4m)	50.2 (@ 2m)
Average RX power (dBm)	−74.5	−66.5	−60.5
Noise power per bit (dBm)	−93.6	−91.0	−87.2
RX noise figure (dB)	6.6	6.6	6.6
Total noise power (dBm)	−87	−84.4	−80.6
Required E_b/N_0 (dB)	4.0	4.7	4.9
Implementation loss (dB)	2.5	2.5	3.0
RX antenna gain (dBi)	0	0	0
Link margin (dB)	6.0	10.7	12.2
Proposes min. RX sensitivity level (dBm)	−80.5	−77.2	−72.7

The timing related parameters of MB-OFDM are listed in Table 2.5. The interval of each symbol is 312.5 ns. The number of total subcarriers used is 122, including 100 data subcarriers, 12 pilot subcarriers and 10 guard tones. The subcarrier frequency spacing is 4.125 MHz. The durations of IFFT/FFT, cyclic prefix and guard interval are 242.42 ns, 60.61 ns and 9.47 ns respectively.

The link budget for high rate MB-OFDM of 110, 200 and 480 Mb/s are shown in Table 2.6.

Chapter 3 Synchronization

Synchronization issue is inevitable in all signal transmission systems. In wireless communication receivers, coherent demodulation needs to make use of local oscillator (LO) that has exactly the same carrier frequency and phase as the transmitted signal. In addition, accurate sampling clock frequency and phase allow the demodulator to recover the transmitted signal more efficiently. Unfortunately, the receiver is unsynchronized with the transmitter most of the time and thus does not have matching timing reference from which the carrier signal and the sampling clock signal can be regenerated. In general, LO mismatch causes phase error of carrier frequency offset (CFO) and sampling frequency offset (SFO). In reality, the controlled LO not only maintains unstable phase of the output signal, but also suffers from the time-varying phase noise [31]. Even with the perfect LO matching, the unknown propagation delay between transmitter and receiver introduces additional phase offset. And the Doppler effect due to relative motion will increase the impact of phase shift of received signal. All these unavoidable impairments undermine the demodulation performance if they are not properly tackled. As an OFDM-based system, MB-OFDM UWB is more vulnerable to these synchronization errors than other single-carrier communication systems.

Fig. 3.1 illustrate possible synchronization errors in OFDM-based signals. The carrier frequency offset (CFO) causes the received complex baseband signal to rotate as a frequency of Δf. The carrier phase error, $\varphi(t)$, introduces an additional phase rotation term in the received baseband signal. The sampling frequency offset (SFO) results in sampling the received continuous waveform at an interval of $(1+\Delta t)T_s$ instead of the idea T_s. And the symbol timing offset, T_d, refers to the error in the symbol boundary at the receiver from the actual boundary in the received waveform.

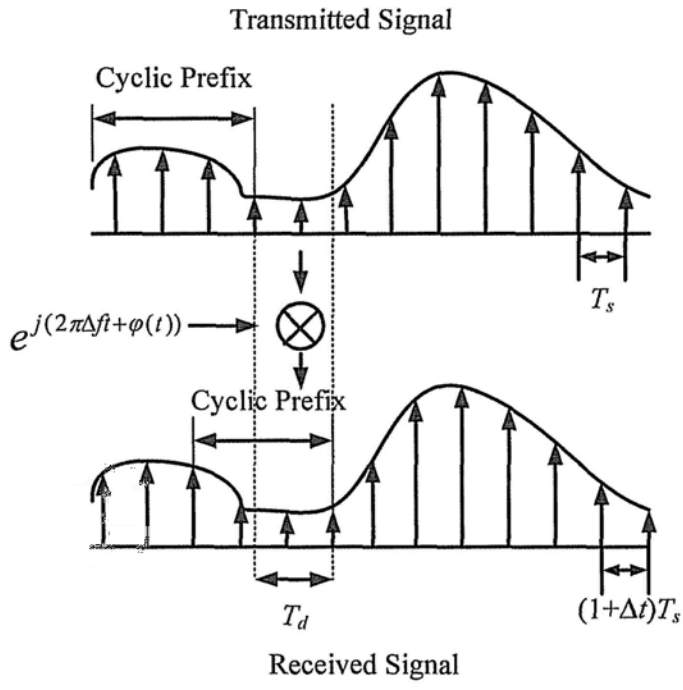


Figure 3.1 Synchronization errors in OFDM-based signals

This chapter will discuss synchronization errors and explain their effects. Timing and frequency synchronization algorithms will be introduced. And the optimal synchronization algorithms in the proposed system for low complexity and high performance will be present. Finally, we will show the architecture design of the synchronizer.

3.1 Timing Synchronization

As specified in [1], the transmitted MB-OFDM UWB signals with N = 128 subcarriers followed by 32 samplings of zero-padding and 5 samples for RF band switching can be expressed as

$$s(t) = \frac{1}{\sqrt{NT}} \sum_{l} \sum_{k} a_{k,l} e^{j2\pi k(t-lT_s)/(NT)} u(t - lT_s) \tag{3.1}$$

where the subscripts l denotes symbol index and k denotes the subcarrier index. $a_{k,l}$ is

the QPSK modulated data. T and T_s stand for the sampling interval and the duration time of each symbol respectively. $u(t)$ is a rectangular pulse function with duration of T_s. The power of transmitted signal $a_{k,l}$ is normalized to unity $E[|a_{k,l}|^2] = 1$ and $E[|s(t)|^2] = 1$.

3.1.1 Effects of timing offset

Suppose the channel maximum excess delay is shorter than the guard interval, the position of FFT window can have several situations, as shown in Fig. 3.2. If the start position is within region B, the signal in FFT window is not contaminated by the previous symbol and thus no ISI occurs. The only effect is introducing phase shift. After demodulation, the received signal with timing offset in region B is expressed in (3.2). Δn is defined as the delayed samples to the correct FFT window position.

$$R_{k,l} = S_{k,l} H_{k,l} e^{-j2\pi\Delta n/N} + W_{k,l} \tag{3.2}$$

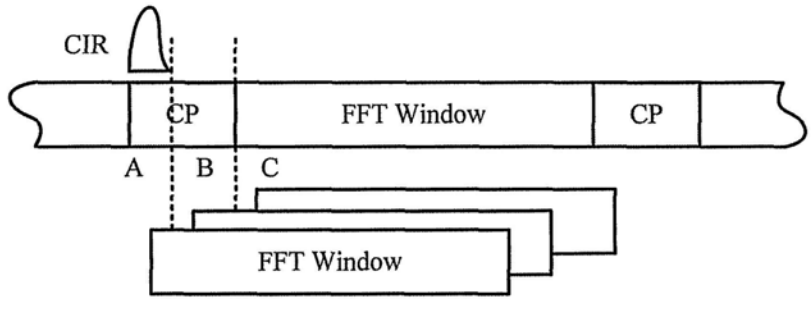


Figure 3.2 The scenario of timing offset

When the FFT window leads or lags by a large degree, such as in region A or C, ISI will be introduced and both the magnitude and phase of received signal will be distorted as shown in (3.3) [32]

$$R_{k,l} = S_{k,l} H_{k,l} e^{-j2\pi\Delta n/N} \frac{N-\Delta n}{N} + W_{k,l} + W_{ISI} \tag{3.3}$$

W_{ISI} is the introduced ISI noise. Due to the introduced ISI and the phase rotation, there is a slight attenuation in the transformed signal magnitude.

3.1.2 Traditional timing synchronization algorithms

Synchronization techniques for OFDM system have been extensively studied for years. Usually there are two phases of timing synchronization in OFDM receivers: coarse timing detection and fine timing detection.

Coarse timing synchronization is usually based on auto-correlation (AC), which is robust against multi-path. While fine timing synchronization is based on cross-correlation (CC), which is more robust in situation of low SNR. Both methods have interesting by-products: AC easily provides an estimate of the CFO, while CC provides channel impulse response.

In this subsection, we will introduce four algorithms for timing synchronization: AC, maximum likelihood (ML), minimum mean square error (MMSE) and CC.

- **AC**

In [5], a simple and direct AC power-normalized metric is shown. The AC algorithm for coarse timing synchronization is quite straightforward. It searches for the repetition in the received signal using a correlator and a maximum searcher. Let the repetition interval length be denoted as L. r_n is the received signal in time domain. The timing metric can be defined as

$$M(n) = \frac{|\sum_{k=0}^{L-1} r_{n+k}^{*} r_{n+k+L}|^2}{(\sum_{k=0}^{L-1} |r_{n+k+L}|^2)^2} \tag{3.4}$$

where * is the conjugated operation. And the estimated time index of the maximum AC can be expressed as

$$\hat{n} = \arg\max_{n} M(n) \tag{3.5}$$

If the maximum AC is over the threshold, the packet is presented and the estimated timing index is the symbol boundary. The drawback of this scheme is when the correlation window moves away from the repeated period, the power of timing metric

$M(n)$ may not fall off as expected, especially in low SNR. In this case, there may be a large error in the detected symbol boundary.

- **ML**

ML algorithm [33], [34] improves the performance of AC. Express ML function as

$$M(n) = 2\left|\sum_{k=0}^{L-1} r_{n+k}^{*} r_{n+k+L}\right| - \rho \sum_{k=0}^{L-1} (|r_{n+k}|^2 + |r_{n+k+L}|^2) \tag{3.6}$$

where

$$\rho \triangleq \left| \frac{E\{r_{n+k}^{*} r_{n+k+L}\}}{\sqrt{E\{|r_{n+k}|^2\} E\{|r_{n+k+L}|^2\}}} \right| = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_n^2} = \frac{SNR}{SNR+1} \tag{3.7}$$

σ_s^2 / σ_n^2 is the signal-to-noise ratio (SNR). The estimated symbol boundary is derived by the same way as AC, which is searching the maximum output of ML function. The complexity of ML is quite high because the estimation of SNR is difficult and errors in SNR estimation will make the system less reliable.

- **MMSE**

MMSE metric [6] is equivalent to a special case of the ML metric with $\rho = 1$. It shows almost the same timing estimation performance as ML. The criterion is to search the minimum output of the metric, as shown in (3.8).

$$M(n) = \sum_{k=0}^{L-1} |r_{n+k}|^2 + \sum_{k=0}^{L-1} |r_{n+k+L}|^2 - 2\left|\sum_{k=0}^{L-1} r_{n+k}^{*} r_{n+k+L}\right| \tag{3.8}$$

The main drawback of AC, ML and MMSE algorithm is that when the preamble has more than two identical segments (e.g. the preambles in MB-OFDM system), there will be a plateau or a wide basin in the correlator output waveforms. The width of the plateau or basin is approximately equal to the length of the extra cyclic prefix minus the length of the maximum channel excess delay. Theoretically, the plateau or basin indicates the ISI-free region for FFT window. However, noise contained in the received signal may cause the maximum/minimum to drift away from the optimal

point. So AC, ML and MMSE are the methods to detect packet coarsely and the definition of exact symbol boundary or FFT window needs fine timing synchronization scheme, such as CC.

- **CC**

CC is the mechanism for fine timing synchronization. Instead of correlating the noisy received waveform with a delay version of it, CC is defined as correlating the received signal with preamble waveform using a matched filter [8], [35]. It can fit into the low SNR situation. The function of CC can be expressed as

$$M(n) = \sum_{k=0}^{Q-1} r_{n+k} c_k^* \tag{3.9}$$

where c_k is the preamble coefficients and Q is the length of preamble.

3.1.3 Proposed timing synchronization algorithm

For the packet-based system, an ISI-free FFT window must be derived as soon as possible in order to proceed the following tasks, such as FFT and channel estimation. If the accurate symbol timing cannot be obtained in time, delay lines are necessary to buffer the received signal, which will increase the system cost. In this case, the timing detection algorithm in frequency domain is not feasible due to the long delay of FFT and high-complexity hardware cost.

In various conventional timing synchronization algorithms, CC is the most straightforward and low complexity solution for fine signal detection. It performs an exhaustive search for the peak energy of the output of the matched filter in each symbol to determine if the packet is detected and the optimal FFT window position. We propose a dual-threshold (DT) judgment scheme, which is based on the idea we presented in [36]. The packet detection scheme in [36] directly judges the correlation values of cascaded auto-correlator, which can improve detection performance compared with traditional CC method. But the extra memory units are involved to store six-symbol length correlation values. The proposed DT scheme just needs the extra memory to store peak

correlation values in each symbol and the corresponding sample index after first level judgment. Therefore, it has nearly the same detection performance as [36], but saves a large amount of memory.

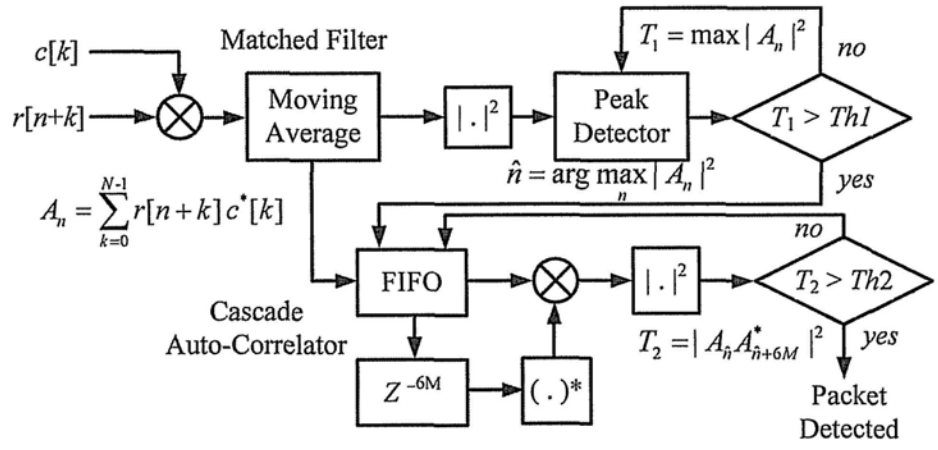


Figure 3.3 Block diagram of the propose DT packet detection scheme

Fig. 3.3 shows the block diagram of proposed DT packet detection scheme. The signal detection process is divided into two steps. The first step is based on CC algorithm. The peak CC energy of each symbol can be expressed as

$$T_1 = \max_n |A_n|^2 = \max_n |\sum_{k=0}^{N-1} r[n+k]c^*[k]|^2 \tag{3.10}$$

$$\sum_{k=0}^{N-1} c^2[k] = N \tag{3.11}$$

where A_n is the moving sum of CC value; $c[k]$ is the preamble coefficient; $r[k]$ is the received signal and N is the FFT window size. If the peak CC energy is over the first threshold $Th1$, the estimated sample index of symbol boundary and the corresponding moving sum $A_{\hat{n}}$ will be stored in FIFO for further use by the following auto-correlator. Otherwise, the peak CC energy of the next symbol will be calculated.

The second step is to read $A_{\hat{n}}$ from FIFO and auto-correlating with its six-symbol delayed version, the energy of the cascaded auto-correlator can be derived and

expressed as

$$T_2 = | A_{\hat{n}} A^*_{\hat{n}+6M} |^2 = | B_n |^2 \tag{3.12}$$

where M = 165 is the repeated preamble interval length. The delay interval of auto-correlator is decided by the period of the TFC. In order to ensure $A_{\hat{n}}$ and its delayed version are in the same band no matter which kind of TFC is adopted, the delay interval of auto-correlator is set to six-symbol length. If the output energy of cascaded auto-correlator is over the second threshold $Th2$, the packet is detected. Otherwise, fetch the next $A_{\hat{n}}$ value in FIFO and calculate the energy of the cascaded auto-correlator again.

The detection problem can be treated to distinguish between the hypotheses

$$\begin{aligned} \mathcal{H}_0 &: r[n] = w[n] \\ \mathcal{H}_1 &: r[n] = s[n] + w[n] \end{aligned} \tag{3.13}$$

To determine the detection performance, we will derive the probability of detection P_D for a given probability of false alarm P_{FA}. The channel impulse response is assumed to affect the transmitted signal $s[n]$ only by complex AWGN $w[n]$. Let $var(x; \mathcal{H}_i)$ denotes the variance of x under hypothesis $\mathcal{H}_i$ (i = 0, 1). The variances of noise $w[n]$ and signal $s[n]$ are σ_s^2 and σ_n^2 respectively. Then during the first threshold judgment, we can get the variance of A_n as

$$var(A_n; \mathcal{H}_0) = var(A_n; \mathcal{H}_1) = N^2 \sigma_n^2 / 2 \tag{3.14}$$

When under the hypothesis $\mathcal{H}_0$, only noises are presented and T_1 follows central Chi-Squared distribution [37]. The probability of false alarm can be expressed as

$$\begin{aligned} P_{FA1} &= Pr\{T_1 > Th1; \mathcal{H}_0\} \\ &\approx exp(-\frac{Th1}{2var(A_n; \mathcal{H}_0)}) = exp(-\frac{Th1}{N^2 \sigma_n^2}) \end{aligned} \tag{3.15}$$

When under the hypothesis $\mathcal{H}_1$, both signals and noises are presented and T_1 follows non-central Chi-Squared distribution. Thus the detection probability can be expresses as

$$\begin{aligned} P_{D1} &= Pr\{T_1 > Th1; \mathcal{H}_1\} \\ &\approx Q(\frac{\sqrt{2}\sigma_s}{\sigma_n}, \frac{\sqrt{2Th1}}{N\sigma_n}) \end{aligned} \tag{3.16}$$

where $Q(a, b)$ denote the generalized Marcum's Q function.

During the second level threshold judgment, the variance of B_n can be expressed as

$$\begin{aligned} var(B_n; \mathcal{H}_0) &= N^4\sigma_n^4 \\ var(B_n; \mathcal{H}_1) &= N^4(\sigma_n^2 + \sigma_s^2)^2 \end{aligned} \tag{3.17}$$

The false alarm probability of T_2 over the second level threshold can be expressed as

$$\begin{aligned} P_{FA2} &= Pr\{T_2 > Th2; \mathcal{H}_0\} \\ &\approx exp(-\frac{Th2}{2var(T_2; \mathcal{H}_0)}) = exp(-\frac{Th2}{2N^4\sigma_n^4}) \end{aligned} \tag{3.18}$$

And the detection probability of the second level threshold judgment is written as

$$\begin{aligned} P_{D2} &= Pr\{T_2 > Th2; \mathcal{H}_1\} \\ &\approx Q(\frac{\sigma_s^2}{\sigma_n^2 + \sigma_s^2}, \frac{\sqrt{Th2}}{N^2(\sigma_n^2 + \sigma_s^2)}) \end{aligned} \tag{3.19}$$

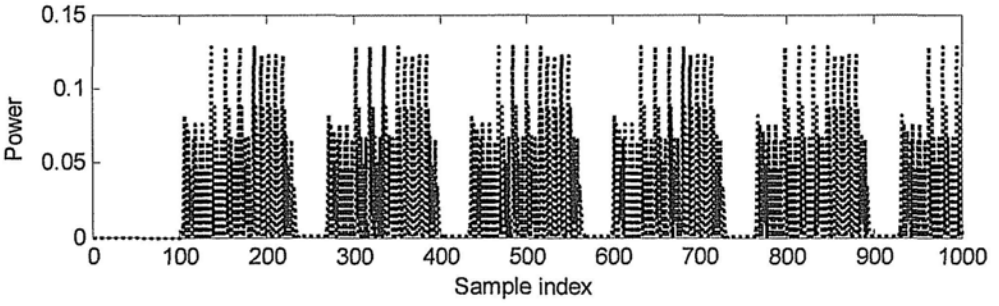
For the proposed DT detection scheme, we can derive the approximated detection probability and false alarm probability as

$$\begin{aligned} P_D &= Pr\{T_2 > Th2, T_1 > Th1; \mathcal{H}_1\} \\ &= \iint p(t_1; \mathcal{H}_1)p(t_2; \mathcal{H}_1)dt_1dt_2 \end{aligned} \tag{3.20}$$

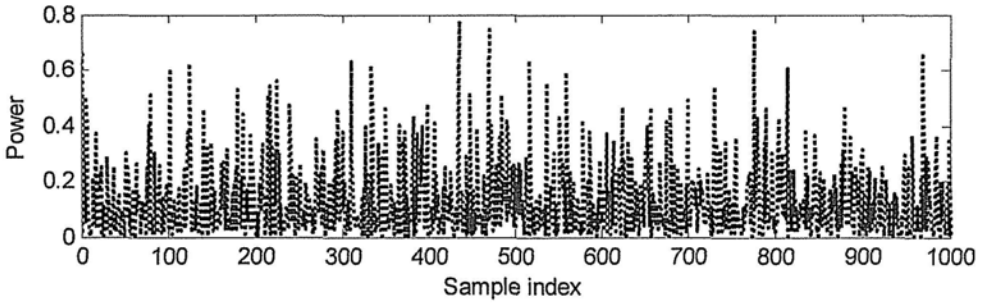
$$\begin{aligned} P_{FA} &= Pr\{T_2 > Th2, T_1 > Th1; \mathcal{H}_0\} \\ &= \iint p(t_1; \mathcal{H}_0)p(t_2; \mathcal{H}_0)dt_1dt_2 \end{aligned} \tag{3.21}$$

where $p(t_1; H_i)$ and $p(t_2; H_i)$ are the probability density functions (PDF) of T_1 and T_2 respectively.

3.1.4 Simulation results and comparisons



(a) Power of transmitted signal



(b) Power of received signal at 10 dB SNR

Figure 3.4 Power of transmitted and received signal

Fig. 3.4 (a) and (b) show the power of transmitted and received signal respectively. The delay of 100 samples is added to the transmitted signal. At 10 dB SNR environment, the received signal is severely distorted by the noise. Hence, the detector robust against noise is necessary.

Fig. 3.5 shows the output waveforms of different timing synchronization algorithms with SNR =10 dB. Obviously, the output waveforms of ML and MMSE schemes have plateaus and basins, which make the peak energy ambiguous. CC algorithm is for fine timing synchronization and is much easier to find accurate timing information. However, there are glitches in CC output waveform, which will corrupt the detection of symbol boundary and increase the false alarm probability. The waveform of the proposed DT algorithm has much lower noise floor than CC algorithm and there is not any glitch. Since ML and MMSE algorithms cannot perform as good timing synchronization as CC and DT algorithms, the following performance will just conduct between CC and DT schemes.

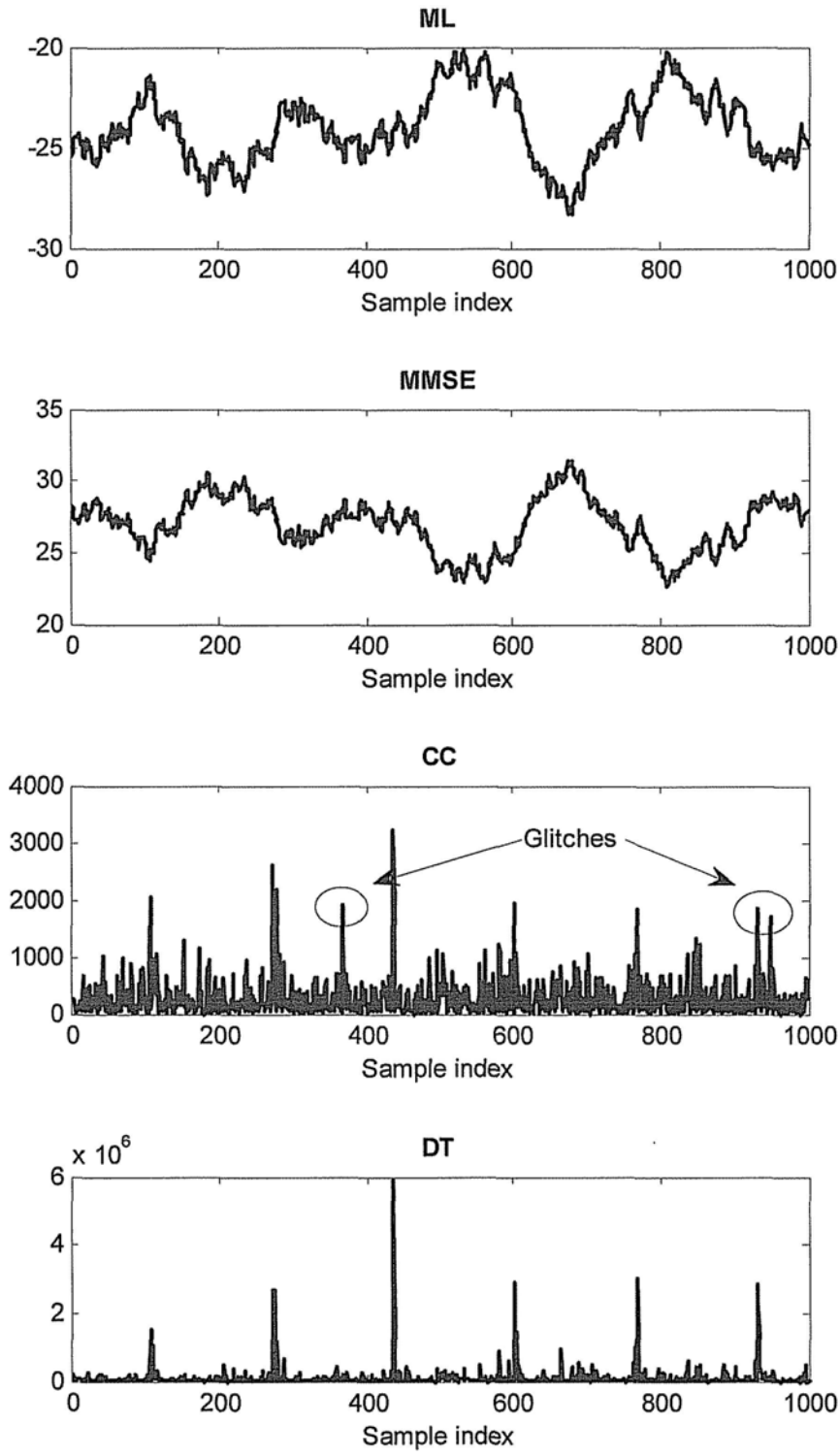
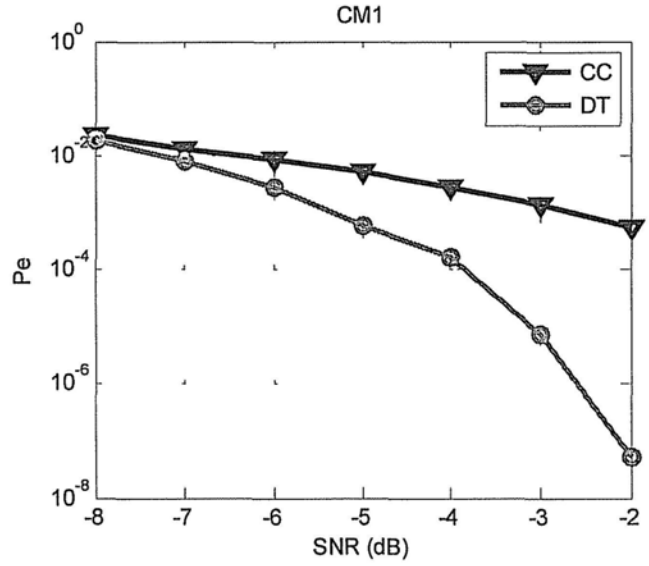
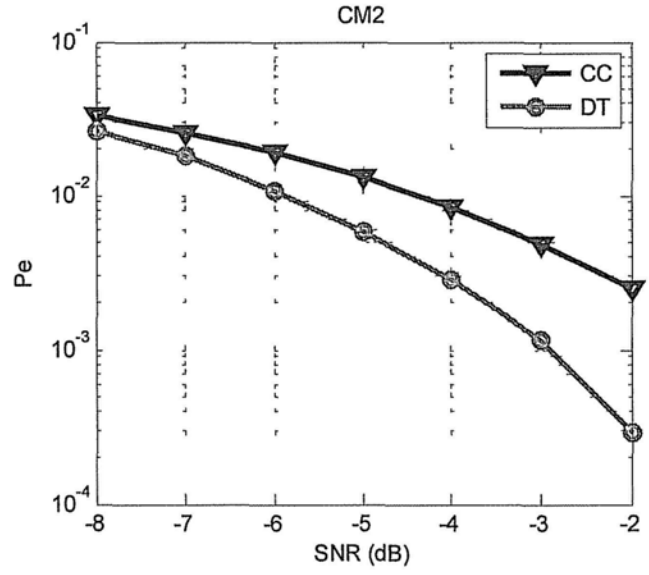
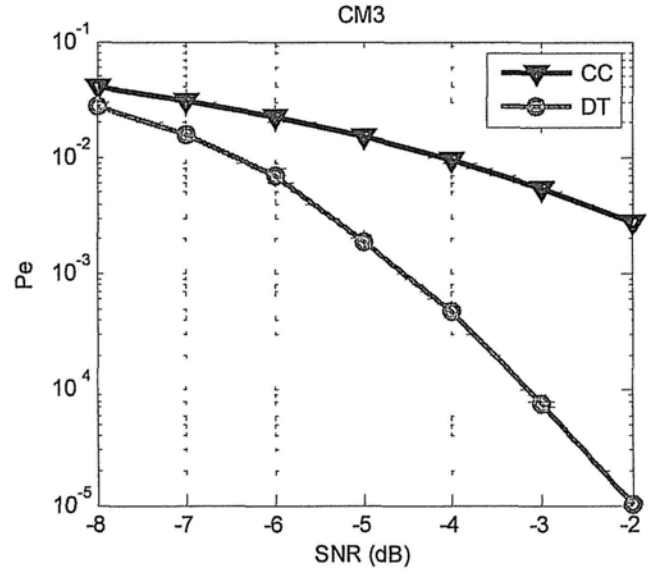


Figure 3.5 Output waveforms of four different timing synchronization algorithms at 10 dB SNR


CM1
CC
DT
Pe
SNR (dB)


CM2
CC
DT
Pe
SNR (dB)


CM3
CC
DT
Pe
SNR (dB)

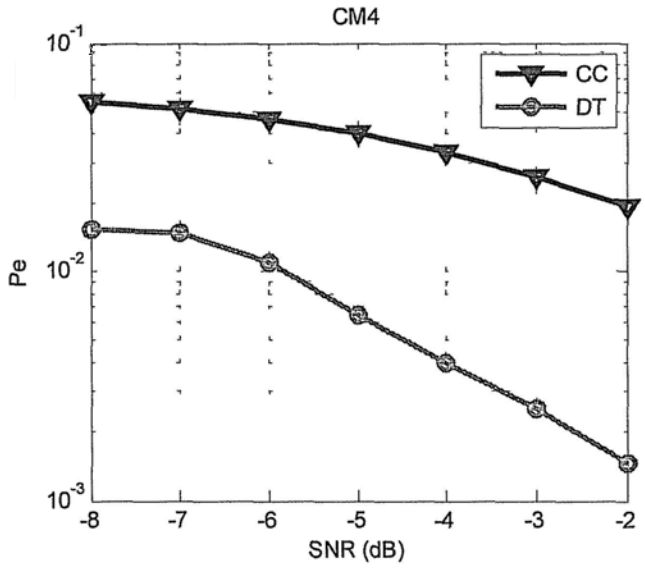


Figure 3.6 Error detection probability comparison between CC and DT in CM1 to CM4

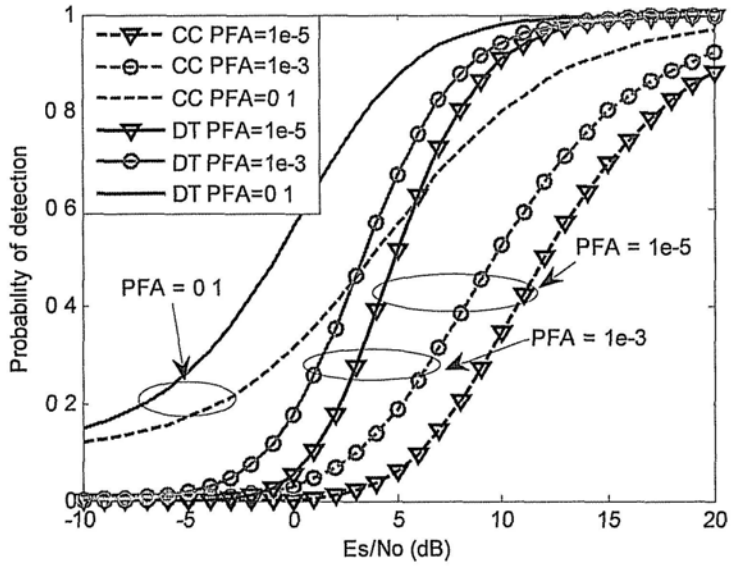


Figure 3.7 Detection probability comparison between CC and DT at different P_{FA}

Fig. 3.6 shows the comparison of packet error detection probability between CC and DT algorithms in the channel environment of CM1 to CM4. It is easy to find that the proposed DT scheme has much lower error detection probability P_e than the traditional CC algorithm in all the channel modes.

Fig 3.7 shows the detection probability comparison between CC and DT at

different false alarm probability. The false alarm probability P_{FA} is set to 0.1, 0.001 and 10^{-5} respectively. E_s/N_o is the ratio of symbol energy to noise power spectral density. For the same given false alarm probability, the proposed DT scheme has much higher detection probability than the traditional CC scheme. And from the detection probability curves, we can also find that the proposed DT scheme can achieve the same probability as the traditional CC algorithm in lower E_s/N_o, which means the proposed DT algorithm is much more robust against noise immunity. Fig. 3.5, Fig. 3.6 and Fig. 3.7 illustrate that the proposed DT algorithm has much better detection performance than the traditional CC algorithm due to the cascaded auto-correlator. Although it will increase the hardware cost slightly, its effect on the whole baseband system complexity can be ignored, which will be proved and addressed later.

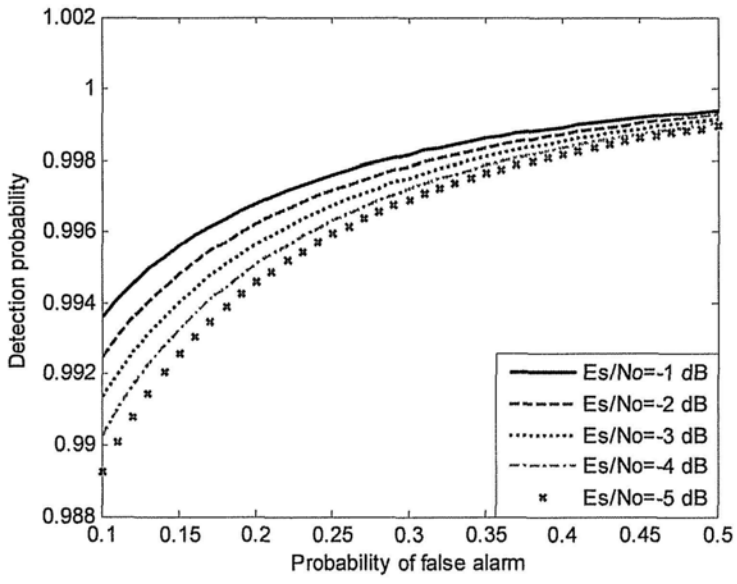
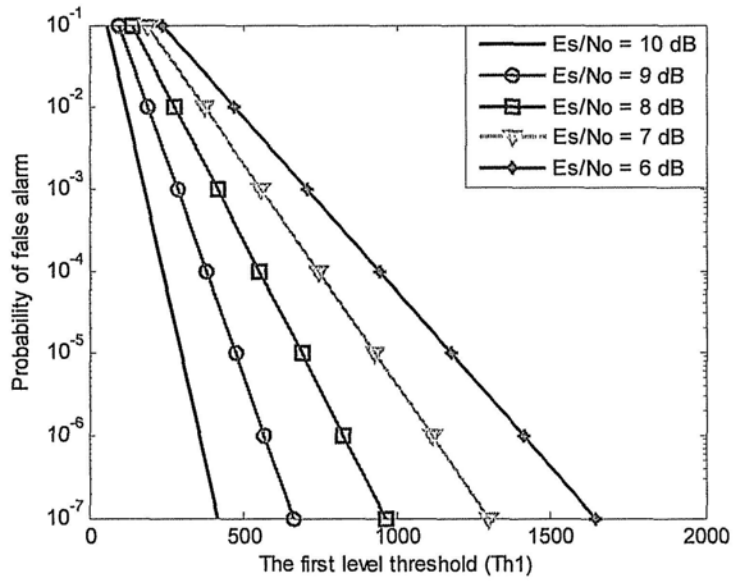
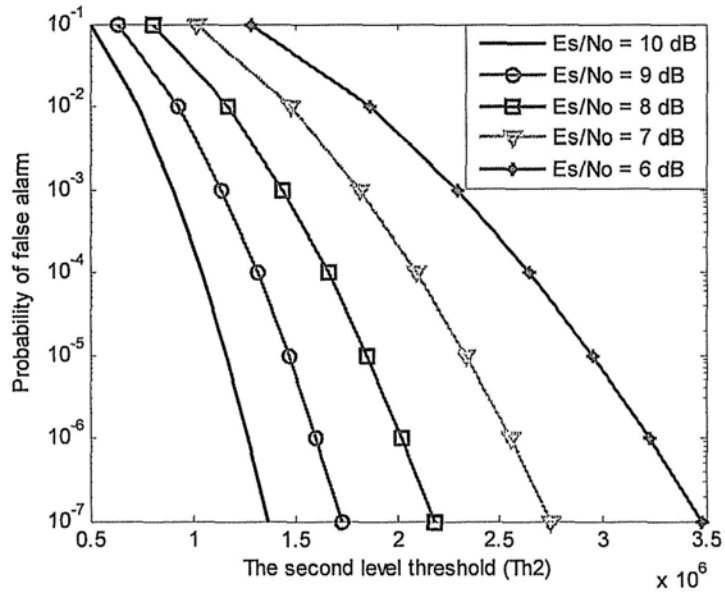


Figure 3.8 Relationship of the detection probability and the false alarm probability

Fig. 3.8 illustrates the relationship of detection probability and false alarm probability of the proposed DT algorithm. With increasing of false alarm probability, the detection probability will be increased accordingly.



(a) Probability of false alarm versus the first level threshold



(a) Probability of false alarm versus the second level threshold

Figure 3.9 Relationship of false alarm probability and two threshold levels in different E_s/N_o

Fig. 3.9 shows the probability of false alarm versus the first and second level threshold respectively. In the condition of 6 dB to 10 dB E_s/N_o, the first level threshold should be in the range of 500 ~ 1700 to guarantee the good detection performance. And

the optimum range of the second level threshold is $1.5 \times 10^6 \sim 3.5 \times 10^6$. The lower E_s/N_o means more ambiguous correlation peaks and thus lead to higher false alarm probability. In this case, the threshold should be higher to suppress the false alarm probability and ensure the detection performance.

Synchronization performance will be affected by auto gain control (AGC). Either the saturation case in high SNR or the high noise level caused by low SNR will decrease the probability of detection. In the proposed design, suppose AGC is settled in the analog front end within three symbols. As specified, there are 21 preambles for packet synchronization. Timing and frequency synchronization will use 13 preambles. Therefore, even three preambles are lost during AGC settling, there are still enough preambles left for synchronization.

3.1.5 Architecture design of packet detector

Quadrature 5-bit signals of ADC are passed to the packet detector of the digital baseband. To simplify the architecture of matched filter, both the received signal and the preamble coefficients are truncated to sign-bit. In this case, the $5b \times 5b$ multipliers can be replaced with NXOR gates and the workload of the cascaded auto-correlator can be alleviated. The 128 sign-bit of preamble coefficients are generated by spreading a 16-length sequence by a 8-length sequence as follows

$$\begin{aligned} &sgn(c_{16(j-1)+i}) = a_i \times b_j \\ &i = 1,2,...,16 \quad j = 1,2,...,8 \end{aligned} \tag{3.22}$$

where a_i and b_j are 1 or −1. The values of sequence a_i and b_j are listed in Table 3.1 and 3.2 respectively. According to (3.22), the 128 taps matched filter can be decomposed to 16 taps cascaded with 8 taps, as shown in Fig. 3.10. With the decomposition, the delay of the matched filter can be reduced to (16+8)/128=18.75% and the length of the circle shift register can be reduced to 20.

Table 3.1 The signs of sequence a_i

Preamble pattern	Sequence a_i															
1	1	1	1	1	-1	-1	1	1	-1	-1	1	-1	1	-1	1	1
2	1	-1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	-1	1
3	1	1	-1	-1	-1	1	-1	-1	-1	1	-1	-1	1	-1	1	1
4	1	-1	-1	1	-1	1	-1	-1	1	1	-1	-1	-1	-1	-1	1

Table 3.2 The signs of sequence b_i

Preamble pattern	Sequence b_j							
1	1	-1	-1	-1	1	1	-1	1
2	1	-1	1	1	-1	-1	-1	1
3	1	1	-1	1	1	-1	-1	-1
4	1	1	1	-1	-1	1	-1	-1

If the shift register is full, shift the data in the address of [5:20] to [1:16] and save the coming four sign-bits to the address of [17:20]. The proposed matched filter is designed with four parallel data paths and 12-level pipelines. The data in the addresses of [1:16], [2:17], [3:18] and [4:19] are distributed to four parallel data paths and cross-correlated with the corresponding coefficient a_i. In the stage of 8 taps, the FIFOs are used to store the accumulated sum from the stage of 16 taps. The sum of the 8 taps, *PosSum*, is just counting the number of "1", the output of matched filter should be obtained by subtracting the number of "−1" in 128 CC values, which can be implemented by left-shifting 1 bit of the *PosSum* and subtracting 128. The efficient combination of pipeline and parallel MF structure not only guarantees the high throughput, but also reduces the cost of hardware.

The cascaded auto-correlator at the second stage is implemented with four $8b \times 8b$ multipliers, two $10b \times 10b$ multipliers, two $14b$ adders and a $20b$ adder. Moreover, a FIFO designed to store the peak values of MF and the corresponding sample index has the size of $36b \times 6$. Compared with the former design proposed in [36], which includes an $18b \times 248$ SRAM, this architecture of DT packet detector saves 64% area.

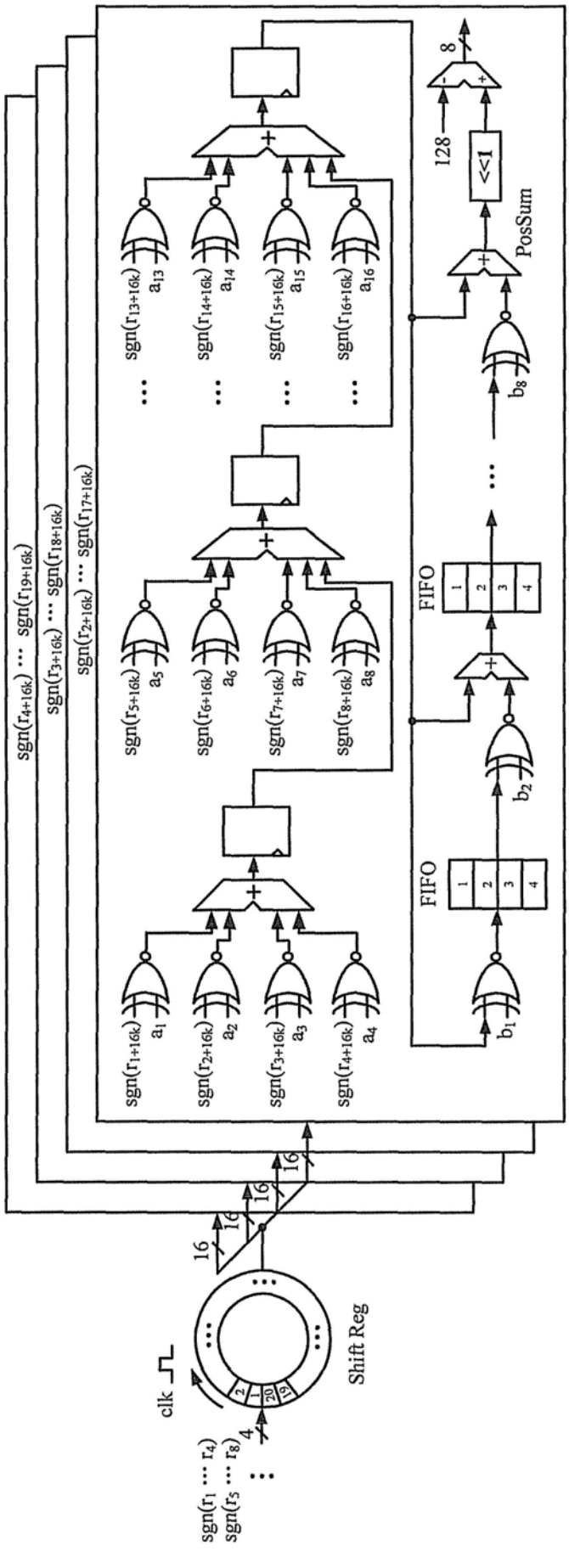


Figure 3.10 Architecture of the matched filter

3.2 Coarse Frequency Synchronization

3.2.1 Effects of carrier frequency offset (CFO)

After DAC, the transmitted signal propagates through the multipath channel with impulse response $h(t)$ and is further corrupted by channel noise $w(t)$. Thus, the received signal can be expressed as

$$r(t) = s(t) \otimes h(t) + w(t) \tag{3.23}$$

where $\otimes$ is the convolution operator. The power of the received signal is normalized by variable gain amplifier and then sampled by ADC. Assume the UWB multipath channel in addictive white Gaussian noise (AWGN) with J paths, the quantized digital signal can be expressed as

$$r[n] = e^{j2\pi n\varepsilon_f} \sum_{m=0}^{J-1} s[n-m]h[m] + w[n] \tag{3.24}$$

In equation (3.24), the normalized CFO $\varepsilon_f = \Delta f / f_s$ is defined as the ratio of CFO and the subcarrier spacing. From (3.24), it is easy to find that even though the value of normalized CFO is very small, the phase distortion of received signal will be serious after a certain period. The received signal in frequency domain with CFO can be expressed as [38]

$$R_{k,l} = S_{k,l} H_{k,l} \frac{\sin(\pi\varepsilon_f)}{N \sin(\pi\varepsilon_f / N)} e^{-j2\pi\varepsilon_f (N-1)/N} + W_{k,l} + W_{ICI} \tag{3.25}$$

where $R_{k,l}$, $S_{k,l}$, $H_{k,l}$, and $W_{k,l}$ stand for the received signal, transmitted signal, channel impulse response and noise in frequency domain at k-th subcarrier and l-th symbol. W_{ICI} is the noise contributed by ICI. Frequency offset will not only destroy the orthogonality of the subcarriers, but also degrade SNR. For relatively small CFO, the SNR degradation can be approximated as [9]

$$D_{SNR} \approx \frac{10}{3\ln 10}(\pi\varepsilon_f)^2 \frac{E_s}{N_o} \tag{3.26}$$

In order to maintain the same bit error rate (BER), E_s/N_o has to be increased essentially equal to D_{SNR}. Therefore, D_{SNR} can also be interpreted as the BER degradation.

3.2.2 Traditional CFO estimation algorithms

The frequency synchronization is also based on AC functions. CFO can be estimated by the phase difference between two AC symbols. The conventional coarse frequency synchronization is Moose algorithm [38]. Morelli [10] improves the traditional algorithm by considering a training symbol composed of more than two identical parts. Minn extended [10] and proposed a robust CFO estimation scheme in [6] by employing a best linear unbiased estimator (BLUE) to improve noise performance.

- **AC**

The estimation of CFO can be obtained simultaneously when the coarse symbol timing is acquired by the AC algorithm mentioned in subsection 3.1.2 . The most straightforward AC algorithm for CFO estimation can be expressed as [5], [38]

$$\hat{\varepsilon}_f = \frac{N}{2\pi L} tan^{-1}\left(\sum_{k=0}^{N-1} r_{n+k}^{*} r_{n+k+L}\right) \tag{3.27}$$

Note that the phase can only be resolved in $[-\pi, \pi]$.

- **BLUE**

If the preamble has U identical repetitions, where $U > 2$, then employing BLUE to exploit the correlation of the repeated segments is possible [6]. Assume that there are M samples in a segment. So totally, there are $N = UM$ samples. The BLUE algorithm starts with several linear AC functions with nM samples of delay.

$$R(n) = \frac{1}{N - nM} \sum_{k=nM}^{N-1} r_{k-nM}^{*} r_k \quad 0 \le n \le K \tag{3.28}$$

where K is a design parameter less than or equal to $U - 1$. Then, the phase differences between all pairs of two AC functions with delay difference of M can be computed as

$$\varphi(n) = [tan^{-1}\{R(n)\} - tan^{-1}\{R(n-1)\}]_{2\pi} \quad 1 \le n \le K \tag{3.29}$$

where $[.]_{2\pi}$ denotes a modulo-2π operation. Note that each $\varphi(n)$ represents an estimation of the CFO, scaled by a constant. The smaller the constant n, the better accuracy it achieves. To gain an effective CFO estimate, the BLUE estimator uses a weighted average of all $\varphi(n)$. And, the coarse frequency offset estimator is given by

$$\hat{\varepsilon}_f = \frac{U}{2\pi}\sum_{n=1}^{K}\varphi(n)\omega(n) \tag{3.30}$$

where

$$\omega(n) = 3\frac{(U-n)(U-n+1) - K(U-K)}{K(4K^2 - 6UK + 3U^2 - 1)} \tag{3.31}$$

The optimal value for K achieving the minimal variance of $\hat{\varepsilon}_f$ is $U/2$. The CFO estimation range is $\pm$ $U/2$ subcarrier spacing for a training symbol having U identical parts.

3.2.3 Proposed CFO correction algorithm

Although BLUE algorithm has better estimation performance, it is quite complicated compared with AC. If apply traditional AC into the MB-OFDM system, the sliding window length (SWL) is 128. The four-parallel structure with the SWL of 128 will be in high complexity as well. In [39], we had provided a simplified solution to shorten the SWL of each symbol to 64. However, it will degrade the estimation performance. To guarantee the system performance, we improve the approach in [39] by making a sum average over three symbols located at three different sub-bands, as expressed in equation (3.32).

$$\begin{aligned}\hat{\varepsilon}_f = \frac{N}{2\pi G_1 M} tan^{-1}\Big(&\sum_{k=1}^{L} r[\frac{N}{L}k + G_1 M]r^*[\frac{N}{L}k] \\ &+\sum_{k=1}^{L} r[\frac{N}{L}k + (G_1+G_2)M]r^*[\frac{N}{L}k + G_2 M] \\ &+\sum_{k=1}^{L} r[\frac{N}{L}k + (G_1+2G_2)M]r^*[\frac{N}{L}k + 2G_2 M]\Big)\end{aligned} \tag{3.32}$$

where L denotes the SWL of each symbol. The value of G depends on TFC. If TFC is {1 2 3 1 2 3} or {1 3 2 1 3 2}, $G_1 = 3$, $G_2 = 1$; otherwise if TFC is {1 1 2 2 3 3} or {1 1 3 3 2 2}, $G_1 = 1$, $G_2 = 2$.

The traditional AC algorithm with SWL 128 is the most complicated scheme to implement, which includes 16 multipliers, 8 adders and the memory size is 40 × 96. If shorten the SWL to 64, the hardware complexity is nearly reduced to half. Eight multipliers, six adders and 20 × 96 memory units are included. However, it will have better performance than the traditional AC scheme with L = 128 after the sum average over three symbols. Because in the case of L = 64, 192 samples are involved in AC, while the conventional algorithm only involves 128 samples in AC. If shorten the sliding window length further, such as L = 32, there will be lower complexity than the case L = 64. Nevertheless, the CFO estimation performance will not be as good as the traditional algorithm even by sum average. That is because the samples involved in AC is only 96 < 128. In this case, to guarantee the CFO estimation performance, sum average should be conducted on more symbols, which will lead to quite long estimation period. Make a tradeoff in complexity, performance and processing period, the proposed CFO estimation scheme with L = 64 is the best choice.

The idea of optimizing CFO compensation part is to take the CFO values on four parallel paths as the same if the differences of the four parallel CFO values are very small [39]. As specified, the center frequency of MB-OFDM UWB is about 4 GHz and the maximum impairment at clock synthesizer is ±20 ppm (parts per million), so the normalized CFO should be less than 0.04. And the maximum CFO difference between two successive samples can be derived as 2.5×10^{-4}, which is small enough. In this case, the CFO compensation scheme can be expressed as

$$\tilde{r}[4(m-1)+q] = r[4(m-1)+q]\exp(-j2\pi 4m\hat{\varepsilon}_f / M)$$
$$m = 1,2,...,\lceil M/4 \rceil, \quad q = 1,2,3,4 \tag{3.33}$$

where $4(m-1)+q$ is the sample index and M is the number of samples each symbol. The optimum CFO compensation algorithm not only reduces the four parallel digital synthesizers to one, but also alleviates the workload of phase accumulator.

3.2.4 Simulation results and comparisons

Fig. 3.11 depicts the estimation performance by mean square error (MSE) with different sliding window length in the condition of 0.01 normalized CFO. The simulation result agrees with the analysis before. The proposed AC algorithm with SWL of 64 has the lowest MSE and the CFO estimation performance of proposed AC scheme with $L = 32$ is worse than the traditional AC scheme with $L = 128$.

The CFO compensation performance is evaluated by error vector magnitude (EVM), which is defined as the ratio of Root Mean Square (RMS) power of the error vector and the RMS power of the reference signal, as shown in (3.34).

$$EVM = \sqrt{\sum_{l=1}^{L} |\hat{X}_l - X_l|^2 \Big/ \sum_{l=1}^{L} |X_l|^2} \tag{3.34}$$

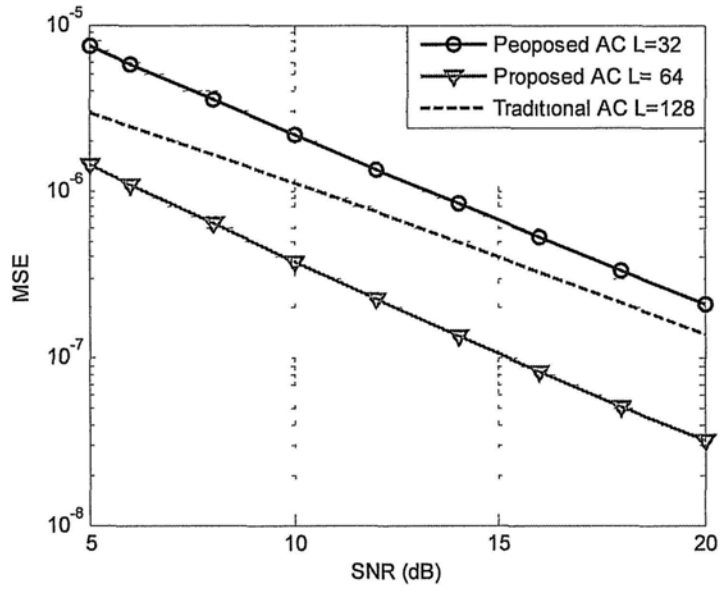


Figure 3.11 MSE performance comparison over different SWL with $\varepsilon_f = 0.01$

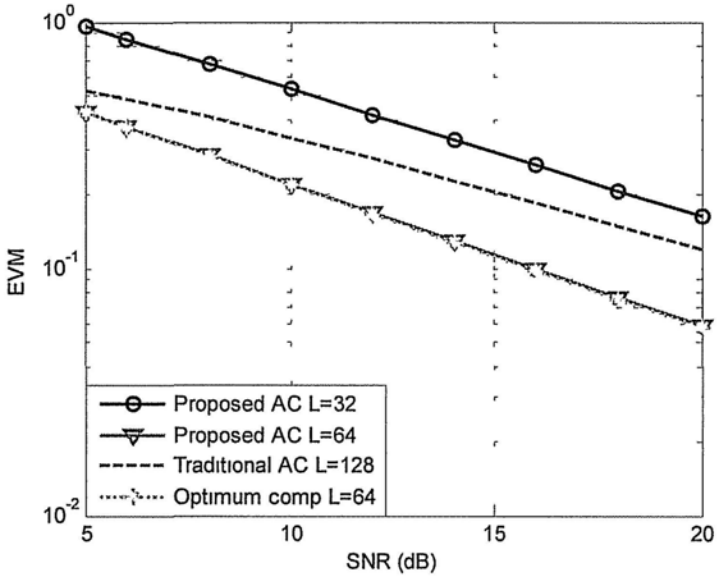


Figure 3.12 EVM performance comparison over different SWL with $\varepsilon_f = 0.01$

Fig. 3.12 shows the EVM performance with different sliding window length and with the optimum CFO compensation algorithm mentioned above. CFO is set to 0.01 as well. It proves again that the proposed CFO estimation scheme with $L = 64$ has the best performance and the optimum CFO compensation algorithm with low complexity will not degrade the EVM performance.

Fig. 3.13 shows the simulation results of the phase error before and after CFO compensation by the proposed AC algorithm with $L = 64$. At 10 dB SNR environment, the signals with 0.01 CFO have serious phase error after 30 preamble sequences. The phase error is limited in the range of $[-\pi, \pi]$. After CFO estimation and compensation, the phase error of the signals is closed to zero. With the growing of the sample index, the phase error of corrected signals deviate from the zero slightly due to the residual CFO, which will be corrected by phase tracking algorithm in frequency domain.

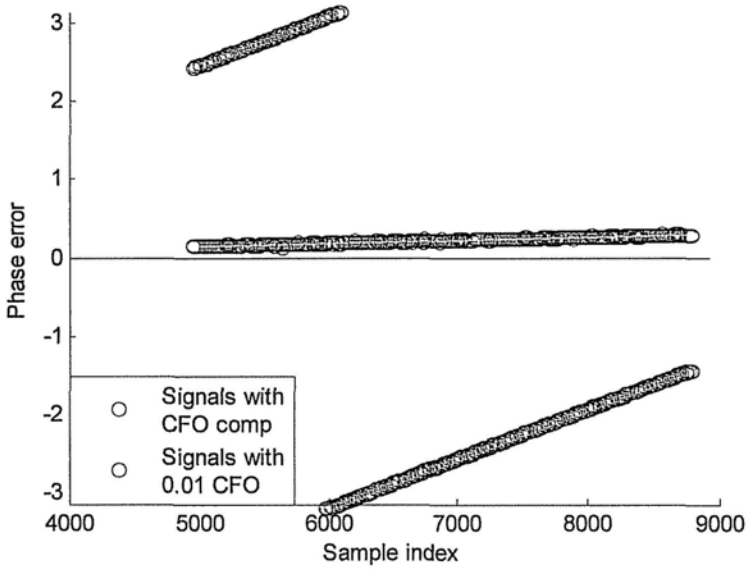


Figure 3.13 Phase tracking before and after CFO compensation with the proposed algorithm

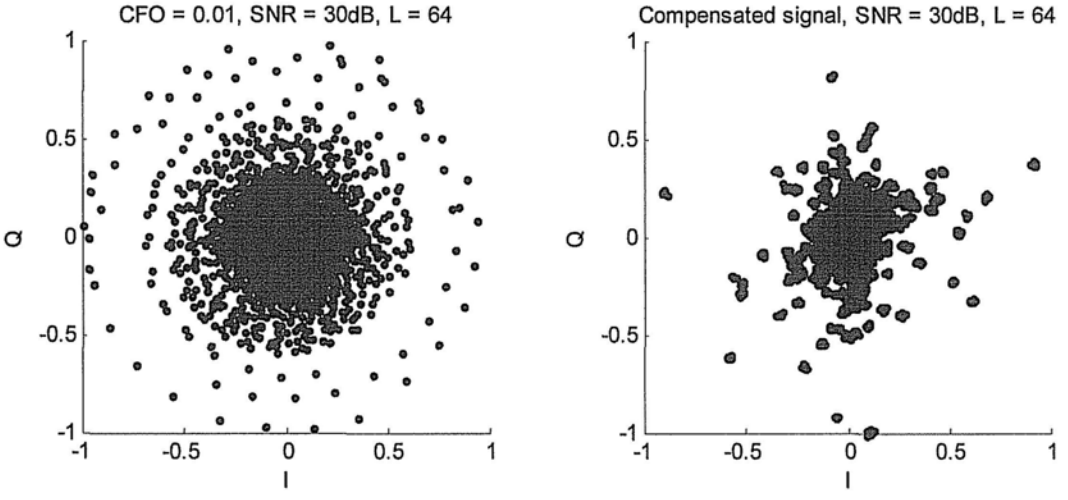


Figure 3.14 Constellation before and after coarse frequency synchronization with the proposed scheme

Fig. 3.14 shows the constellation diagram before and after coarse frequency synchronization with the proposed scheme. The simulation environment is set to 0.01 normalized CFO and 30 dB SNR as well. As shown in Fig. 3.14, before CFO correction, there is an obvious rotation of the quadrature I/Q signal. However, after

frequency synchronization, the rotation phenomenon disappeared.

3.2.5 Implementation of CFO corrector

The design of CFO corrector is divided into two parts. The first part is to estimate the accumulated phase differences between two consecutive preambles in the same band by AC and arctangent calculation. The following part is to compensate the signals by multiplying a complex rotation vector, in which the phase accumulator and sin/cos generator are involved.

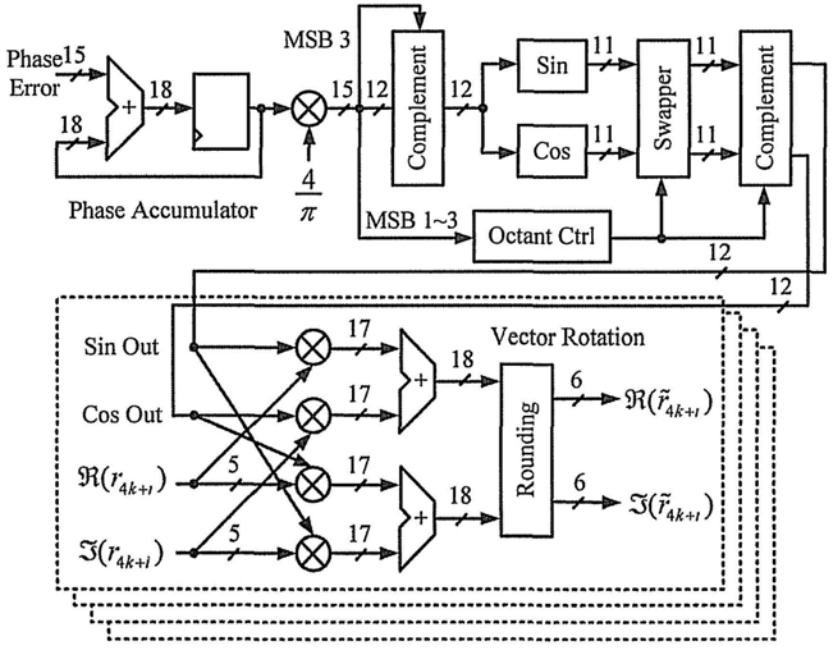


Figure 3.15 Architecture of the CFO compensation block

Fig. 3.15 presents the architecture of CFO compensation block. The phase accumulator produces a digital sweep with a slope proportional to the input phase. The accumulated phase offset is controlled within the range of [0, 2π]. Actually, we just need to calculate sin/cos values with the phase offset range of [0, $\pi/4$]. If the phase offset is in the range of [$\pi/4$, 2π], the sin/cos values can be obtained by the relationship shown in Fig. 3.16. Fifteen bits binary code is used to represent the accumulated phase

offset. Three most significant bits (MSBs) are used to represent integer part and left 12 bits are fraction part. The accumulated phase offset is scaled from [0, 2π] to [0, 8] by multiplying 4/π, so that the three MSBs can be used to control the eight phase offset regions in Fig. 3.15 easily. Otherwise, we have to use 15 bits to control phase range. For instance, the binary code of π/2 is 001.100100100001. Comparing it with the value of phase offset accumulator can decide if the phase offset is in region II or III. Therefore, scaling the phase offset range and using three MSBs to control is a wiser choice than using 15 bits comparator to control the octant. The hardware implementation of octant control includes optional input complement, output complement and output swap.

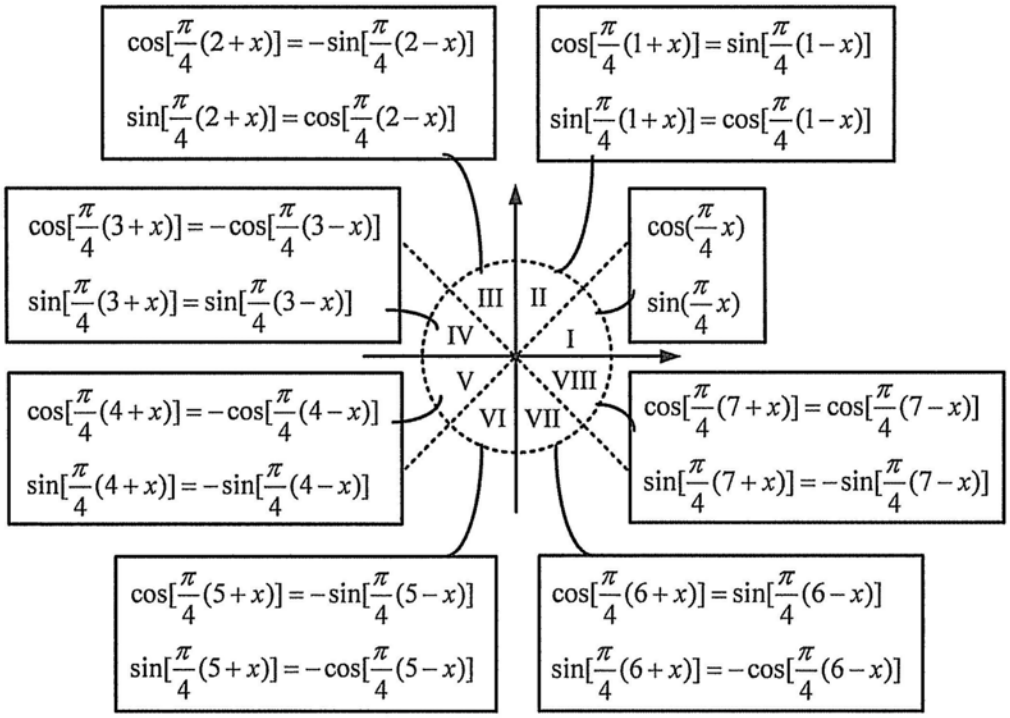


Figure 3.16 Sin/cos calculation in different phase ranges

In CFO corrector, implementation of arctangent and sin/cos blocks is the most critical work. The traditional OFDM-based or CDMA-based systems employed classic coordinate rotation digital computer (CORDIC) algorithm [40] to implement them [41], [42]. In recent years, a novel combined approach, hybrid CORDIC architecture

[43], [44] has been developed to realize numerically controlled oscillator (NCO) for its low complexity and high performance. Besides CORDIC, there are other methods to realized NCO, such as polynomial hyperfolding technique (PHT) [45], piecewise -polynomial approximation (PPA) technique [46]-[48] and multipartite table method (MTM) [49]-[51].

- PHT

The polynomial hyperfolding technique calculates sin/cos functions using an optimized polynomial expression with constant coefficients. The sine and cosine functions in [0, $\pi/4$] can be expressed by polynomial expressions of degree K.

$$\begin{aligned} S(x) &= sin(\frac{\pi}{4}x + \frac{LSB}{2}) \cong a_K x^K + a_{K-1}x^{K-1} + \cdots + a_0 \\ C(x) &= cos(\frac{\pi}{4}x + \frac{LSB}{2}) \cong b_K x^K + b_{K-1}x^{K-1} + \cdots + b_0 \end{aligned} \tag{3.35}$$

where $0 \leq x < 1$ is the scaled input of sine and cosine blocks. Note that half less significant bit (LSB) is added to the phase to compensate the error introduced by one's complementer. It is clear that the introduced error decreases by increasing polynomial degree K. However, hardware complexity increases with K and hence a tradeoff between complexity and performance exists.

In [45], the performance is improved by evaluating the polynomial as a sum of partial products, each one weighted by an appropriate positive or negative number. The technique, names as canonical-signed-digit (CSD) hyperfolding is used to reduce the hardware complexity. Optimization is conducted on two-order ($K = 2$) and three-order ($K = 3$) approximated polynomials, expressed as (3.36) and (3.37) respectively.

$$\begin{aligned} S(x) &\cong -0.004713 + 0.838015x - 2^{-3}x^2 \\ C(x) &\cong 0.9995593 - 0.011408x + (-2^{-2} - 2^{-5})x^2 \end{aligned} \tag{3.36}$$

$$\begin{aligned} S(x) &\cong 0.00015005 + 0.77436217x - 0.00530040x^2 \\ &\quad + (-2^{-2} + 2^{-5})x^3 / 3 \\ C(x) &\cong 0.98423596 + 0.00452969x - 0.32417224x^2 \\ &\quad + (2^{-3} - 2^{-5})x^3 / 3 \end{aligned} \tag{3.37}$$

The two-order PHT can achieve about 60 dBc spurious free dynamic range (SFDR) while the three-order PHT can achieve 80 dBc SFDR.

- PPA

This technique is based on the idea of subdividing the interval in shorter subintervals [46]-[48]. Polynomials of a given degree are used in each subinterval to approximate the trigonometric functions. The signal x represents the input phase [0, $\pi/2$] scaled to a binary fraction in the interval [0, 1]. The range [0, 1] is subdivided in s subintervals, with $s = 2^u$. The u most significant bits (MSBs) of x encode the segment starting point x_k and are used as an address to the small lookup tables that store polynomials coefficients. The remaining bits of x represent the offset $x - x_k$. The quadratic PPA of the sin/cos function can be expressed as follows:

$$\begin{cases} f_s(x) = y_{s_k} + m_{s_k}(x - x_k) - p_{s_k}(x - x_k)^2 \\ f_c(x) = y_{c_k} - m_{c_k}(x - x_k) - p_{c_k}(x - x_k)^2 \end{cases}$$
$$x_k \le x \le x_{k+1}; \quad k = 1, 2, \ldots, s \tag{3.38}$$
$$x_1 = 0; \quad x_{s+1} = 1$$

Use r bits for first-order coefficients quantization and t bits for second-order coefficients quantization. The constant coefficients are quantized with Q −1 bits and the outputs of the sin/cos function are Q bits. The input phase is presented by P bits.

The architecture of sin/cos function implementation with PPA is shown in Fig 3.17. The squarer is fed by the qx MSBs of $(x - x_k)$ whereas its output is rounded to the $qx2$ MSBs. The constant, linear and quadratic coefficients are read from ROMs to conduct polynomial calculation. The partial products are generated by the PP Gen block to compute linear terms. And the carry-save addition tree adds the partial products together after aligning all the bits according to their weights.

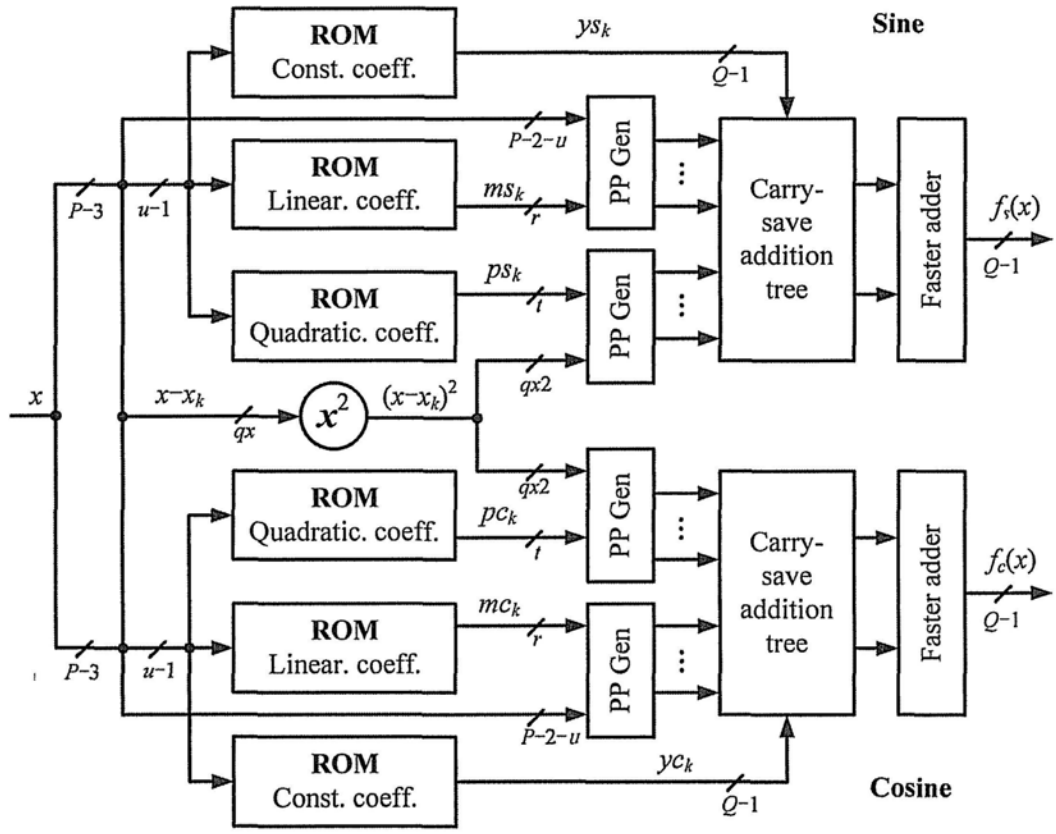


Figure 3.17 Architecture of sin/cos implementation with PPA

- Hybrid CORDIC

This approach splits the phase rotation in three steps. The first two steps are CORDIC-based with computing the rotation directions in parallel. The final step is multiplier based [43], [44].

Suppose the input vector $[X_{in}, Y_{in}]$ and output vector $[X_{out}, Y_{out}]$ word length is 12 and 13 bits respectively. Represent the rotation phase $\varphi \in [0, \pi/4]$ with a binary fractional value in [0, 1] as

$$\frac{4}{\pi}\varphi = f_1 2^{-1} + f_2 2^{-2} + \cdots + f_{13} 2^{-13} \tag{3.39}$$

The LSB of φ has a weight that will be indicated in the following as $\varphi_{LSB} = (\pi/4)2^{-13}$. In the first step, the phase is divided in two sub-words $\varphi = \alpha + \beta$, where

$$\alpha = (f_1 \cdot 2^{-1} + ... + f_3 \cdot 2^{-3} + 2^{-4})\frac{\pi}{4}$$
$$\beta = (-\overline{f_4} \cdot 2^{-4} + f_5 \cdot 2^{-5} + ... + f_{13} \cdot 2^{-13})\frac{\pi}{4} \tag{3.40}$$

and $\overline{f_4}$ is the complement of f_4. The goal of the first stage is to perform a rotation by an angle close to $\alpha + \varphi_{LSB}/2$. To that purpose, the first rotation uses CORDIC algorithm can be described by the following equations:

$$\begin{cases} X_{i+1} = X_i - \sigma_i \cdot 2^{-i} \cdot Y_i \\ Y_{i+1} = Y_i + \sigma_i \cdot 2^{-i} \cdot X_i \\ Z_{i+1} = Z_i - \sigma_i \cdot tan^{-1} 2^{-i} \end{cases} \qquad i = 1,...,4 \tag{3.41}$$

where σ_i is equal to sign of Z_i. The algorithm starts with $X_1 = X_{in}$, $Y_1 = Y_{in}$ and $Z_1 = \alpha + \varphi_{LSB}/2$.

The second and third stages rotate the output vector of the first stage by a phase $\gamma = Z_{residual} + \beta$, which is represented with 11 bits. γ is then split as the sum of two sub-words $\gamma_1 + \gamma_2$, where

$$\gamma_1 = 2^{-3}(-g_0 + g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + 2^{-3})$$
$$\gamma_2 = 2^{-3}(-\overline{g_3} \cdot 2^{-3} + g_4 \cdot 2^{-4} + ... + g_{10} \cdot 2^{-10}) \tag{3.42}$$

The second rotation is aimed to perform the rotation by the phase γ_1. The rotation directions are obtained by the bits of γ_1 as follows:

$$\tau_0 = 2 \cdot \overline{g_0} - 1 \qquad \tau_i = 2 \cdot g_i - 1 \qquad i = 1,2 \tag{3.43}$$

The corresponding CORDIC equations are

$$\begin{cases} X'_{i+1} = X'_i - \tau_i \cdot 2^{-(i+4)} \cdot Y'_i \\ Y'_{i+1} = Y'_i + \tau_i \cdot 2^{-(i+4)} \cdot X'_i \end{cases} \qquad i = 0,1,2 \tag{3.44}$$

The operation to be performed in the final rotation block can be written as

$$\begin{cases} X_{out} = X_{T2} \cos\gamma_2 - Y_{T2} \sin\gamma_2 \\ Y_{out} = X_{T2} \sin\gamma_2 + Y_{T2} \cos\gamma_2 \end{cases} \tag{3.45}$$

where $[X_{T2}, Y_{T2}]$ is the output vector of the second rotation. The absolute value of γ_2 is

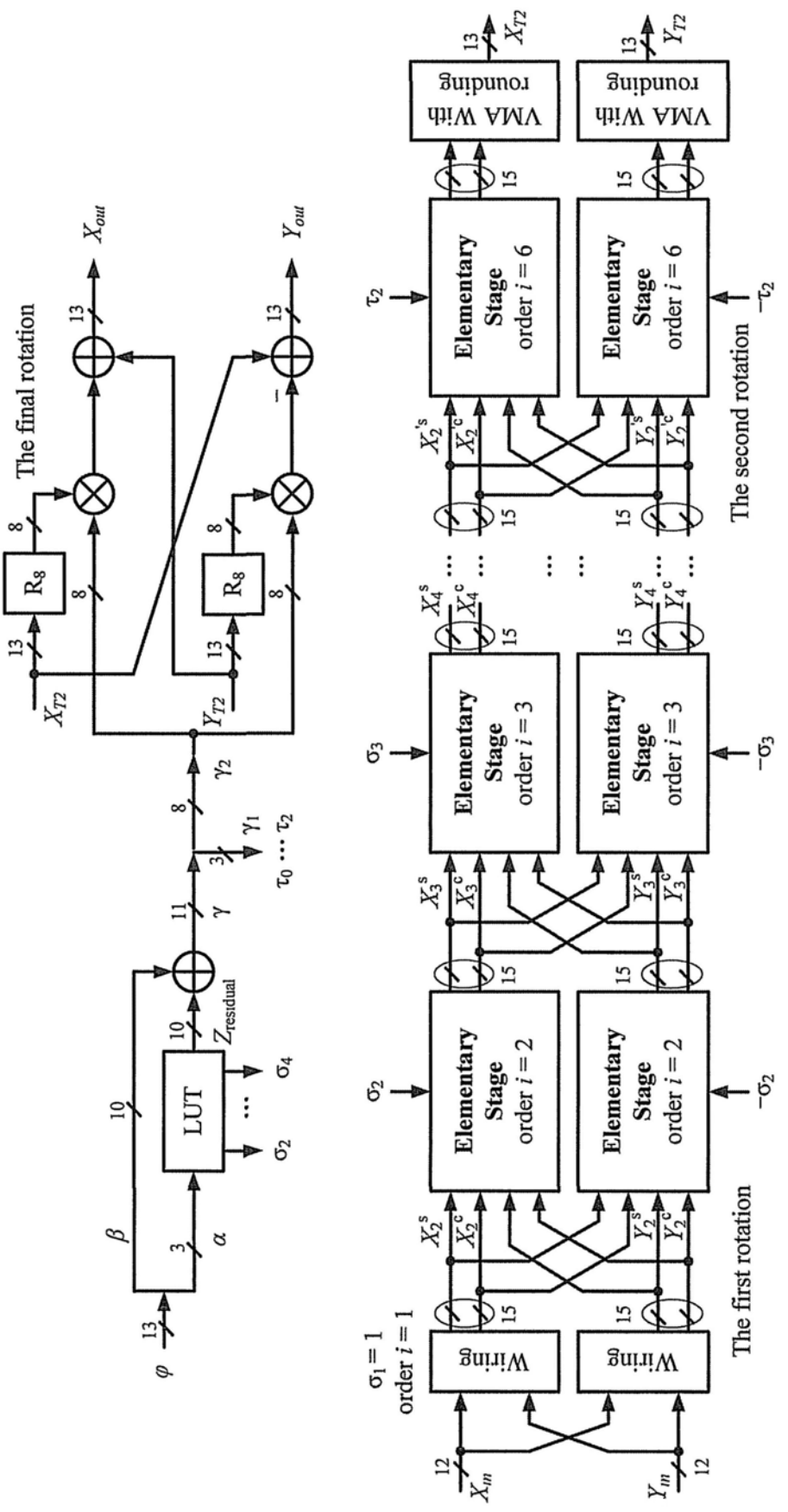


Figure 3.18 Hybrid CORDIC rotator architecture

smaller than 2^{-6}. Therefore, sine and cosine functions can be approximated as $sin\ \gamma_2 \approx \gamma_2$ and $cos\ \gamma_2 \approx 1$.

The architecture of hybrid CORDIC rotator is shown in Fig. 3.18. The elementary stage is composed with adders and shifters. R_8 is the unit rounding to the first 8 MSBs. The two final vector merging adders (VMAs) convert the results to two's complement representation.

- MTM

This is a very effective lookup table compression technique for function evaluation. It has been found ideally suited for high performance synthesizer, requiring both very small ROM size and simple arithmetic circuitry [49]-[51]. The principle of MTM is to decompose Q-bit input signal x in $K + 1$ non-overlapping sub-words: $x_0, x_1, \ldots, x_K$ with lengths of $q_0, q_1, \ldots, q_K$ respectively, where $x = x_0 + x_1 + \ldots + x_K$ and $Q = q_0 + q_1 + \ldots + q_K$. The angle $[0, \pi/4]$ is scaled to a binary fraction in $[0, 1]$. A piecewise linear approximation of $f(x)$ can be expressed as

$$\begin{aligned} f(x) &= f(x_0 + x_1 + \ldots + x_K) \\ &\simeq A(x_0) + B(x_0)(x_1 + \ldots + x_K) \\ &= A(x_0) + B(x_0)x_1 + \ldots + B(x_0)x_K \end{aligned} \tag{3.46}$$

where the interval $[0, 1]$ of x has been divided in 2^{q_0} subintervals. x_0 represents the starting point of each subinterval and $x_1 + \ldots + x_K$ is the offset in each interval between x and x_0.

The term $B(x_0)\ x_1$ can be approximated as $B_1(\alpha_1)\ x_1$, where α_1 is a sub-word of x_0 including its $p_1 \leq q_0$ MSBs. Likewise, the term $B(x_0)\ x_2$ is approximated as $B_2(\alpha_2)\ x_2$, where α_2 is a sub-word of x_0 including its $p_2 \leq p_1$ MSBs. Similar approximation can be conducted on the terms $B(x_0)\ x_i$ $(i = 3 \ldots K)$ and the expression (3.46) becomes

$$f(x) \simeq A(x_0) + B_1(\alpha_1)x_1 + \ldots + B_K(\alpha_K)x_K \tag{3.47}$$

The term $A(x_0)$ can be realized with a ROM, which is named as Table of Initial Values (TIV), with 2^{q_0} entries. And the terms $B(\alpha_i)\ x_i$ $(i = 1 \ldots K)$ can be implemented with K ROMs, which is named as Table of Offsets (TO_i), with $2^{p_i+q_i}$ entries each.

Making the TOs symmetric, the size of ROMs can be reduced by a factor of two. Then, the equation (3.47) becomes

$$f(x) \simeq \tilde{A}(x_0) + B_1(\alpha_1)(x_1 - \frac{\delta_1}{2}) + \cdots + B_K(\alpha_K)(x_K - \frac{\delta_K}{2}) \tag{3.48}$$

where

$$\delta_i = (2^{q_i} - 1)2^{-s_i}; \qquad s_i = \sum_{J=0}^{i} q_J \tag{3.49}$$

The coefficients of TIV and TOs can be calculated by minimizing the maximum approximation error as

$$\begin{aligned} \tilde{A}(x_0) &= \frac{f(x_0) + f(x_0 + \Delta_0)}{2} \\ B_i(\alpha_i) &= \frac{f(\alpha_i + \delta_i) - f(\alpha_i) + f(\alpha_i + \delta_i + \sigma_i) - f(\alpha_i + \sigma_i)}{2\delta_i} \end{aligned} \tag{3.50}$$

where

$$\sigma_i = 2^{-p_i} - 2^{q_i - s_i}; \quad \Delta_0 = \sum_{J=1}^{K} \delta_J = 2^{-q_0} - 2^{-Q} \tag{3.51}$$

$\tilde{A}(x_0)$ is directly the content of the TIV. The content of each TO_i can be computed as

$$TO_i(\alpha_i, x_i) = B_i(\alpha_i)(x_i + 2^{-s_i - 1}) \tag{3.52}$$

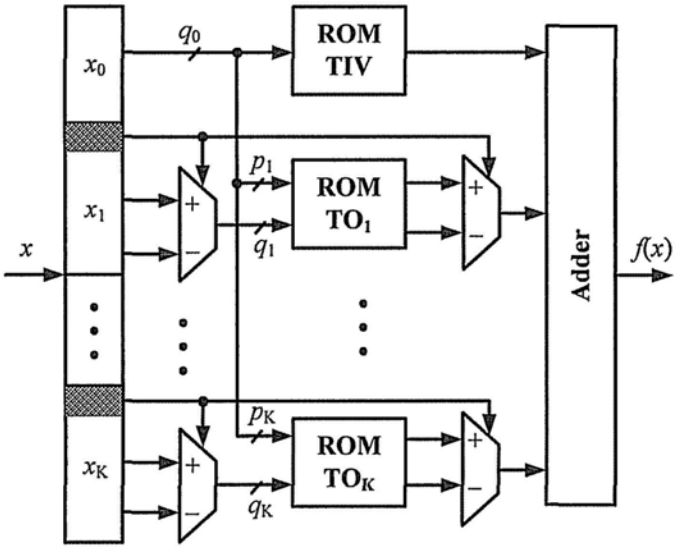


Figure 3.19 Architecture of MTM with symmetric TOs

The architecture of MTM with symmetric TOs is shown in Fig 3.19. The content of the TOs is conditionally added or subtracted from the content stored in TIV. The addition or subtraction of the content in ROMs and complement operation of the inputs are controlled by the MSB of each sub-word.

In order to give a fair comparison of the four function evaluation algorithms, we implement NCO with the techniques of PHT, PPA, hybrid CORDIC and MTM for CFO compensation. In Table 3.3, the parameters of four techniques are listed. We intend to set the parameter to design the NCO in low complexity and make the SFDR of the four methods nearly the same. For all the four algorithms, both the inputs and outputs of the sin/cos block are 12 bits.

Table 3.3 Synthesis performance comparison of CFO compensation with four techniques

Technique	**MTM**	**PPA**	**PHT**	**Hybrid CORDIC**
Parameter	$q_0=4$ $q_1=2$ $q_2=3$ $q_3=3$ $p_1=3$ $p_2=3$ $p_3=1$	$s=64$ $r=6$ $t=7$	$K=3$	(1) 4 rep. (2) 3 rep. (3) $8b \times 8b$
MSE ($\times 10^{-7}$)	2.97	4.91	7.82	5.73
Area (mm^2)	0.018	0.027	0.031	0.146
Power (mW)	0.84	0.88	1.55	13.93
Latency (Clock cycs.)	3	3	4	6

Synthesized with UMC 0.13 um library at 132 MHz clock frequency, the power, area and latency of the four algorithms are compared, as list in Table 3.3. MSE is a statistical value, so it is not easy to set the MSE of the four schemes exactly the same. However, we try to make the MSE of the four techniques to be very closed. With the smallest MSE, MTM outperforms other algorithms in area, power and latency. If improve the performance of other three methods to be as good as MTM, there will be

higher hardware cost and power consumption. Therefore, employing MTM for function evaluation in CFO compensation block makes it faster, have lower complexity and lower power consumption. Since MTM is proved to be an efficient approach for function evaluation from the analysis above, it is also applied for arctangent function in CFO estimation block.

3.3 Fine Frequency Synchronization

Although CFO can be coarsely estimated by frequency synchronizer in time domain by preambles, the residual CFO (RCFO), sampling frequency offset (SFO) and common phase error will lead to accumulated phase shift after a certain period and thus degrade the system performance if they are not carefully tracked. In MB-OFDM UWB systems, pilot subcarriers can help to solve the residual phase distortion issue in frequency domain, which is also called fine frequency synchronization.

3.3.1 Effects of sampling frequency offset (SFO)

The oscillators used to generate the DAC and ADC sampling instants at the transmitter and receiver will never have exactly the same period. Thus, the sampling instants slowly shift relative to each other. The problems have been analyzed by several literatures [9], [52], [53].

The sampling clock error has two main effects: A slow shift of the symbol timing, which rotates subcarriers; and a loss of SNR due to the ICI generated by the slightly incorrect sampling instants, which causes loss of the orthogonality of the subcarriers. Define the normalized sampling error as $\Delta t = (T' - T)/T$, where T' and T are the receiver and transmitter sampling periods respectively. Then the overall effect on the received signal in frequency domain is expressed as

$$R_{k,l} = S_{k,l} H_{k,l} e^{j2\pi k \Delta t l T_s / T_u} sinc(\pi k \Delta t) + W_{k,l} + N_{\Delta t}(k,l) \tag{3.53}$$

where the received signal $R_{k,l}$ is at the l-th symbol and k-th subcarrier. T_s and T_u are the duration of the total symbol and the useful data respectively. $W_{k,l}$ is additive white

noise and the last term $N_{\Delta t}(k, l)$ is the additional interference due to the SFO. The power of the last term is approximated by

$$P_{\Delta t} \approx \frac{\pi^2}{3}(k\Delta t)^2 \tag{3.54}$$

Hence the degradation grows as the square of the produce of the offset Δt and the subcarrier index k. This means that the outermost subcarriers are most severely affected. The degradation can also be expressed directly by SNR loss as

$$D_n \approx 10log_{10}(1+\frac{\pi^2}{3}\frac{E_s}{N_0}(k\Delta t)^2) \ (dB) \tag{3.55}$$

The MB-OFDM system does not have large number of subcarriers and the value of Δt is quite small. So $k\Delta t << 1$, and the interference caused by SFO can usually be ignored.

However, the term showing the amount of rotation angle experienced by the different subcarriers will lead to serious problem. Since the rotated angle depends on both the subcarrier index and symbol index, the angle is the largest for the outermost subcarrier and increases with the consecutive symbols. Although Δt is very small, with the increasing of the symbol index, the phase shift will eventually corrupt the demodulation. In this case, tracking SFO is necessary.

3.3.2 Traditional phase tracking algorithms

Conventionally, the SFO can be estimated by computing a slope from the plot of pilot subcarrier differences versus pilot subcarrier indices [54]. Recently, joint estimation of CFO and SFO has also been studied extensively, such as the linear least squares (LLS) algorithm in [55] and joint weighted least squares (WLS) algorithm in [56]. [57] introduced a simple unbiased joint RCFO and SFO by using continual pilots and [58] proposed the phase tracking method for MB-OFDM system.

- **The algorithm in [54]**

Using FFT to transform signals from time domain to frequency domain and

removing the effect of channel noise, the signals with residual phase distortion can be modeled as

$$\begin{aligned} Z_{k,l} &= S_{k,l}P_{k,l} = S_{k,l}\, exp(j\Phi_{k,l}) \\ &= S_{k,l}\, exp(j(\alpha k + \beta_l)) \end{aligned} \tag{3.56}$$

where $P_{k,\,l}$ is the phase distortion vector and $\Phi_{k,\,l}$ is the residual phase error. The relationship of α, β_l and $\Phi_{k,\,l}$ is shown in Fig. 3.20. α is the slope of the phase distortion and is contributed by SFO. β_l is the intercept of phase distortion and is caused by RCFO of symbol l.

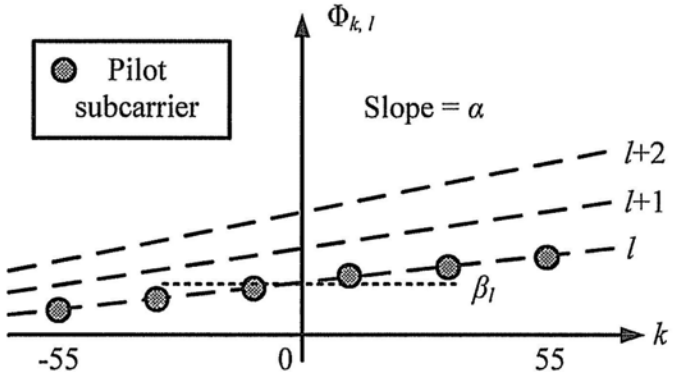


Figure 3.20 The relationship of phase distortion and subcarriers

The basic idea of the conventional phase tracking algorithm is to get the phase differences of pilot subcarriers between two symbols by AC. The pilot subcarriers are divided into two parts, C_1 and C_2. C_1 is on the left of the spectrum, and C_2 is on the right of the spectrum. Then the estimated intercept phase β_l and the slope α are written as

$$\hat{\beta}_l = \frac{1}{2}\left(\sum_{k\in C_1}\Phi^{-}_{k,l} + \sum_{k\in C_2}\Phi^{+}_{k,l}\right) \qquad \hat{\alpha} = \frac{\sum_{k\in C_2}\Phi^{+}_{k,l} - \sum_{k\in C_1}\Phi^{-}_{k,l}}{\sum_{k\in C_2} k - \sum_{k\in C_1} k} \tag{3.57}$$

where

$$\Phi^{-}_{k,l} = tan^{-1}\sum_{k\in C_1} Z_{k,l-1}Z^{*}_{k,l} \qquad \Phi^{+}_{k,l} = tan^{-1}\sum_{k\in C_2} Z_{k,l}Z^{*}_{k,l-1} \tag{3.58}$$

- **LLS**

Applying LLS estimation to (3.58) with K pilots in one symbol, each pilot located at the subcarrier of k_i. The RCFO and SFO estimation yield [55]:

$$\Delta\hat{\theta}_f = \frac{\sum_{i=1}^{K} \Phi_{k_i,l}}{2\pi \frac{MK}{N}} \qquad \hat{\delta} = \frac{\sum_{i=1}^{K} \Phi_{k_i,l} k_i}{2\pi \frac{MK}{N} \sum_{i=1}^{K} k_i^2} \tag{3.59}$$

where

$$\Phi_{k_i,l} = tan^{-1} \sum_{i=1}^{K} Z_{k_i,l} Z^*_{k_i,l-1} \tag{3.60}$$

Such an estimation algorithm that is base on the phase difference between two symbols can remove the common channel fading terms in slow-fade scenarios. Consequently, this estimation scheme can be applied before channel estimation and equalization.

- **WLS**

Though the joint LLS estimation algorithm provides accurate estimation results in the AWGN channel, diverse channel responses on the pilot subcarriers can render its estimation useless. For instance, phase of several deeply faded pilot subcarriers, when employ the estimation of the joint LLS, can lead to a large error in the estimation results. On the other hand, the phases of those subcarriers with little fading are naturally more reliable. Therefore, weighting the subcarrier data is advantageous, and data of serious faded subcarriers should be assigned smaller weights to minimize their adverse effect on estimation accuracy. The WLS algorithm for joint estimation of RCFO and SFO can be expressed as [56]

$$\Delta\hat{\theta}_f = \frac{\sum_{i=0}^{K} \omega_i k_i^2 \sum_{i=0}^{K} \omega_i \Phi_{k_i,l} - \sum_{i=0}^{K} \omega_i k_i \sum_{i=0}^{K} \omega_i \Phi_{k_i,l} k_i}{2\pi \frac{MK}{N} \left[\sum_{i=0}^{K} \omega_i \sum_{i=0}^{K} \omega_i k_i^2 - \left(\sum_{i=0}^{K} \omega_i k_i \right)^2 \right]} \tag{3.61}$$

$$\hat{\delta} = \frac{\sum_{l=0}^{K}\omega_l \sum_{l=0}^{K}\omega_l \Phi_{k_l,l} k_l - \sum_{l=0}^{K}\omega_l k_l \sum_{l=0}^{K}\omega_l \Phi_{k_l,l}}{2\pi \frac{MK}{N}\left[\sum_{l=0}^{K}\omega_l \sum_{l=0}^{K}\omega_l k_l^2 - \left(\sum_{l=0}^{K}\omega_l k_l\right)^2\right]} \tag{3.62}$$

The weight ω_l should be inversely proportional to the variance of phase error, which depends on noise, ICI and the complex channel gain. Usually, the residual synchronization error is so small that the ICI term can be neglected and ω_l only depends on the channel gain of the pilot subcarriers. The disadvantage is this algorithm is very complicated, especially the computation of the parameter of ω_l. Without estimating the ω_l accurately, there will be large error in phase tracking.

3.3.3 The proposed phase tracking algorithms

The conventional phase tracking solutions [54]-[58] require arctangent function block and NCO, both of which are complicated in hardware implementation. The algorithm presented in [59] simplifies the hardware cost significantly compared with the conventional methods, but sacrifices system performance slightly. We try to improve the phase tracking performance with low hardware complexity.

Considering the condition $|\alpha k| << 1$ is satisfied with $k \in [-55, 55]$, the first order approximation can be made as $cos(\alpha k) \approx 1$ and $sin(\alpha k) \approx \alpha k$. Then the phase distortion in (3.56) can be rewritten as

$$P_{k,l} = cos\beta_l - \alpha k \cdot sin\beta_l + j(sin\beta_l + \alpha k \cdot cos\beta_l) \tag{3.63}$$

In (3.63), four parameters are of interests: $sin\beta_l$, $cos\beta_l$, $\alpha \cdot sin\beta_l$ and $\alpha \cdot cos\beta_l$. The former two can be easily obtained by

$$\begin{cases} cos\beta_l = \dfrac{1}{8}\displaystyle\sum_{k=\pm25,\pm35,\pm45,\pm55} \Re\{P_{k,l}\} \\ sin\beta_l = \dfrac{1}{8}\displaystyle\sum_{k=\pm25,\pm35,\pm45,\pm55} \Im\{P_{k,l}\} \end{cases} \tag{3.64}$$

where $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary part respectively. There are 12 pilots

in each symbol of MB-OFDM UWB. Since 1/8 is much easier to implement (3-bit right-shift) than 1/12 and the pilots near DC subcarrier suffer more channel noise than the ones far away from DC subcarrier, we intend to use the pilots outermost as many as possible.

The parameters of $\alpha \cdot cos\beta_l$ and $\alpha \cdot sin\beta_l$ can be calculated by variety of ways. Three of them are presented as follows to give a comparison in complexity and performance.

- ***Method 1***

Utilize the pilots at subcarrier indices of ±5, ±25, ±45, ±55 in (3.63), yielding

$$\begin{cases} 4cos\beta_l + 130\alpha \cdot sin\beta_l = \sum_{k=-55,-45,-25,-5} \Re\{P_{k,l}\} \\ 4sin\beta_l + 130\alpha \cdot cos\beta_l = \sum_{k=55,45,25,5} \Im\{P_{k,l}\} \end{cases} \tag{3.65}$$

Replacing $4cos\beta_l$ and $4sin\beta_l$ with

$$\begin{cases} 4cos\beta_l = \sum_{k=\pm 15,\pm 35} \Re\{P_{k,l}\} \\ 4sin\beta_l = \sum_{k=\pm 15,\pm 35} \Im\{P_{k,l}\} \end{cases} \tag{3.66}$$

Approximating the scaling factor 1/130 to 1/128 to simplify the hardware design, the parameters of $\alpha \cdot cos\beta_l$ and $\alpha \cdot sin\beta_l$ can be expressed as follows

$$\begin{cases} \alpha \cdot sin\beta_l \approx \frac{1}{128}\left(\sum_{k=-55,-45,-25,-5} \Re\{P_{k,l}\} - \sum_{k=\pm 15,\pm 35} \Re\{P_{k,l}\} \right) \\ \alpha \cdot cos\beta_l \approx \frac{1}{128}\left(\sum_{k=55,45,25,5} \Im\{P_{k,l}\} - \sum_{k=\pm 15,\pm 35} \Im\{P_{k,l}\} \right) \end{cases} \tag{3.67}$$

- ***Method 2***

Utilize the pilots at subcarrier indices of ±15, ±25, ±35, ±55 in (3.63). Similarly, the parameters of $\alpha \cdot cos\beta_l$ and $\alpha \cdot sin\beta_l$ can also be obtained.

$$\begin{cases} \alpha \cdot sin\beta_l \approx \frac{1}{128}\left(\sum_{k=-55,-35,-25} \Re\{P_{k,l}\} - \sum_{k=\pm 45,15} \Re\{P_{k,l}\} \right) \\ \alpha \cdot cos\beta_l \approx \frac{1}{128}\left(\sum_{k=55,35,25} \Im\{P_{k,l}\} - \sum_{k=\pm 45,-15} \Im\{P_{k,l}\} \right) \end{cases} \tag{3.68}$$

- ***Method 3***

This scheme uses the same pilot subcarriers as *Method 2* in (3.63). However, $4cos\beta_l$ and $4sin\beta_l$ are expressed in different forms, as shown in (3.69).

$$\begin{cases} 4cos\beta_l = \frac{1}{2} \sum_{k=\pm15,\pm25,\pm35,\pm55} \Re\{P_{k,l}\} \\ 4sin\beta_l = \frac{1}{2} \sum_{k=\pm15,\pm25,\pm35,\pm55} \Im\{P_{k,l}\} \end{cases} \tag{3.69}$$

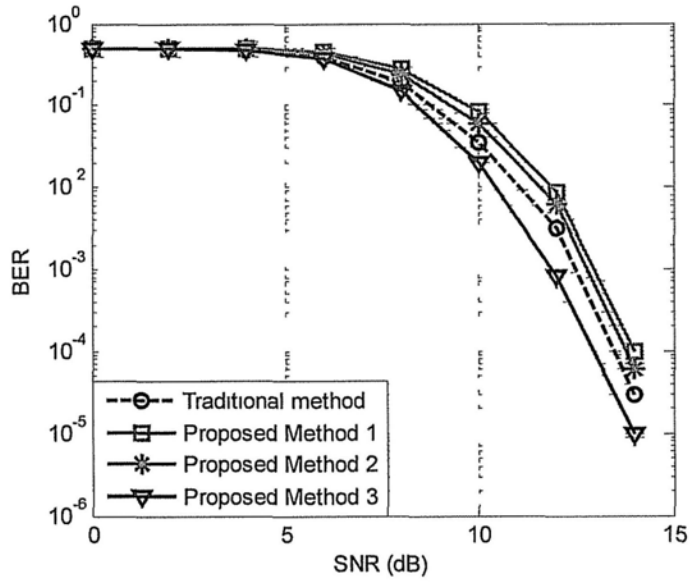
Approximating the scaling factor 1/260 to 1/256, which can be implemented with 8-bit right-shift, the parameters of $\alpha \cdot cos\beta_l$ and $\alpha \cdot sin\beta_l$ are given by

$$\begin{cases} \alpha \cdot \sin\beta_l \approx \frac{1}{256}\left(\sum_{k=-55,-35,-25,-15} \Re\{P_{k,l}\} - \sum_{k=15,25,35,55} \Re\{P_{k,l}\} \right) \\ \alpha \cdot \cos\beta_l \approx \frac{1}{256}\left(\sum_{k=55,35,25,15} \Im\{P_{k,l}\} - \sum_{k=-15,-25,-35,-55} \Im\{P_{k,l}\} \right) \end{cases} \tag{3.70}$$

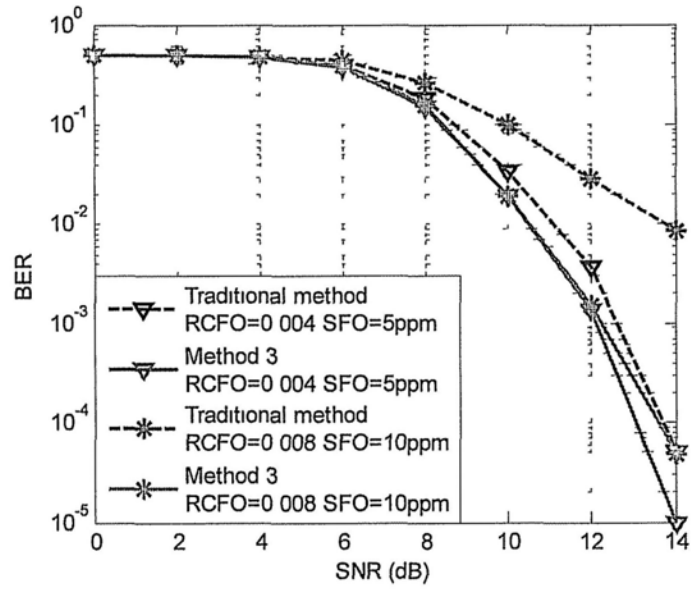
From the expressions of (3.67), (3.68) and (3.70), it is easy to find that *Method 1* and *Method 3* have nearly the same hardware complexity. *Method 2* saves four adders in architecture compared with *Method 1* and *Method 3*. *Method 2* may outperform *Method 1* due to avoiding the use of the pilots on ±5 subcarriers, which are affected by channel noise most severely. During the approximation, both *Method 1* and *Method 2* introduce an error of 1.2×10^{-4} while *Method 3* only introduces an error of 6×10^{-5}. Therefore, *Method 3* should have the best phase tracking performance in the three proposed schemes. A tradeoff between hardware complexity and system performance will be made to determine the most efficient scheme according to simulation results.

3.3.4 Simulation results and comparisons

In the traditional algorithms, although LLS and WLS has better phase tracking performance than the method in [54], they have very high complexity for practical application. For hardware implementation, the algorithm in [54] is in low complexity and moderate phase correction performance. Therefore, we just compare the proposed three methods with the algorithm in [54].



(a) BER comparison in three proposed methods and the traditional algorithm



(b) BER performance comparison between *Method 3* and the conventional method for in different phase error conditions

Figure 3.21 BER performance comparison

The traditional [54] and three proposed algorithms for phase tracking are simulated in CM1 channel environment at the data rate of 200 Mbps. In Fig. 3.21 (a) and Fig. 3.22, BER and EVM performance of the four methods are compared, in which the SFO is set to 10 parts per million (ppm) and the normalized RCFO is set to

0.004. The simulation results agree with the analysis before. *Method 3* has the best BER and EVM performance. *Method 2* has better performance than *Method 1*. Although *Method 2* can reduce hardware cost with adders, it does not weight much in the whole phase tracking circuit design. In addition, *Method 3* is the only one outperforms traditional scheme and has low complexity structure as well, it is the best choice for practical phase tracking circuit design.

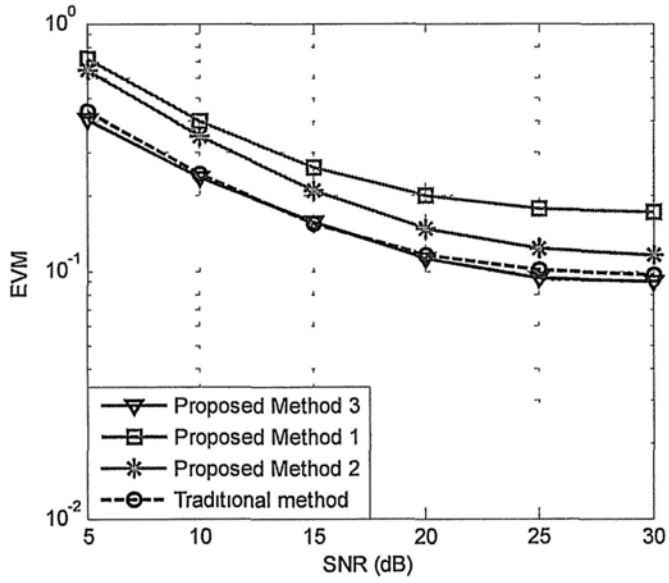


Figure 3.22 EVM performance comparison in three proposed methods and the conventional method

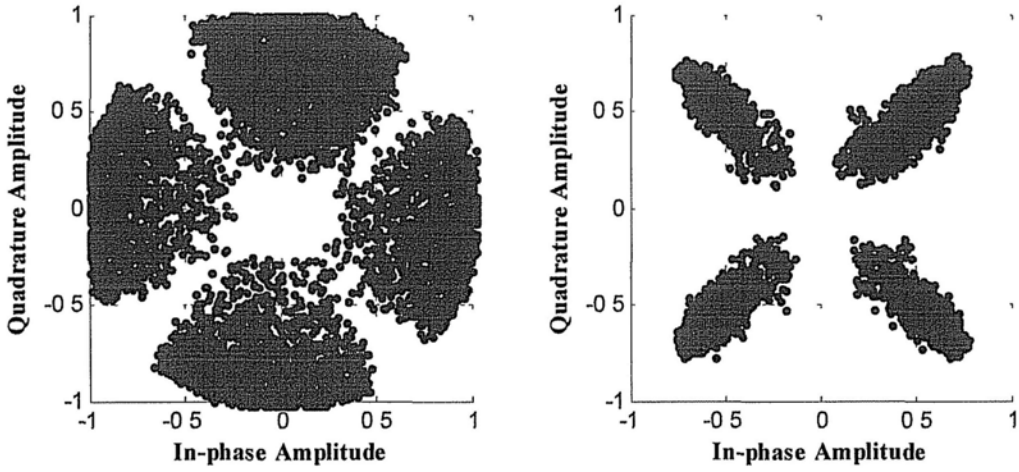


Figure 3.23 QPSK constellation diagram before and after phase tracking with *Method 3*

In Fig. 3.21 (b), we compare the traditional algorithm [54] and *Method 3* in different phase error conditions. Another advantage of *Method 3* is with the increasing of phase error, it does not degrade as severe as the traditional method [54].

The QPSK constellation with residual phase distortion and after phase correction by *Method 3* is shown in Fig. 3.23. The phase shift is almost eliminated by phase tracking.

3.3.5 Architecture design of phase tracking block

Fig. 3.24 shows the architecture of phase tracking block. The signals after channel equalization are stored in pilots buffer and data buffer separately. Considering that the transmitted pilots are known and have the modulus of 1, the phase error vector of the pilots can be derived by multiplying the conjugation of transmitted pilots. Phase error estimator is the critical unit to determine the parameters of phase error vector in the proposed phase tracking algorithm. As shown in Fig. 3.24, it is composed by 8 complex adders and 2 complex shifters.

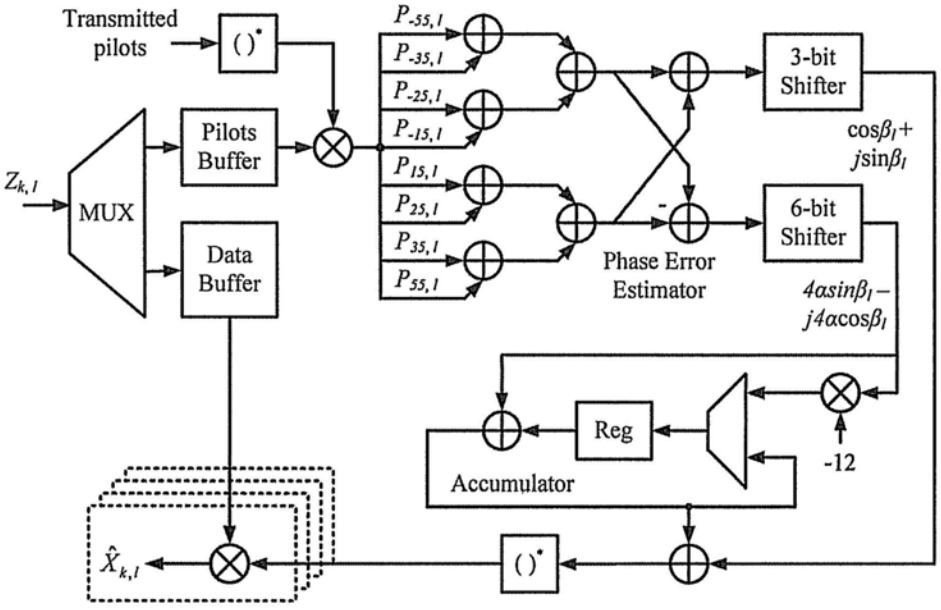


Figure 3.24 Architecture of phase tracking block

The values of parameters $\alpha \cdot sin\beta_l$ and $\alpha \cdot cos\beta_l$ are very small, so the phase errors contributed by SFO of four parallel data can be approximately thought the same, rewritten as $\alpha\lceil k/4 \rceil sin\beta_l$ and $\alpha\lceil k/4 \rceil cos\beta_l$ ($\lceil k/4 \rceil \in [-12, 12]$). Calculating the parameters of $4\alpha \cdot sin\beta_l$ and $4\alpha \cdot cos\beta_l$ instead of $\alpha \cdot sin\beta_l$ and $\alpha \cdot cos\beta_l$ further simplifies the architecture of phase tracking block.

Chapter 4 FFT

Fast Fourier Transform (FFT) is the most important demodulation kernel in OFDM-based receiver systems to transform signal from time domain to frequency domain. FFT operation has been proven to be both computational intensive, in terms of arithmetic operations, and communicational intensive, in terms of data swapping or exchanging in the storage.

4.1 FFT Algorithms

The N-point discrete Fourier transform (DFT) is defined by [61]

$$X[k]=\sum_{n=0}^{N-1}x[n]W_N^{nk} \qquad 0\le k<N \tag{4.1}$$

where W_N, the so called "twiddle factor", denotes the N-th primitive root of unity, with its exponent evaluated modulo N is given by

$$W_N = e^{-j2\pi/N} \tag{4.2}$$

Direct implementation of the above formula requires $O(N^2)$ arithmetic complexity. Cooley and Tukey proposed an FFT algorithm that decomposes the N-point DFT into recursive 2-point DFT operations, known as the radix-2 FFT, which requires only $O(N \log N)$ computations.

Usually, FFT algorithms can be categorized into two types: decimation-in-time (DIT) and decimation-in-frequency (DIF). In DIT FFT algorithms, the time domain signals are not processed sequentially, while in DIF FFT algorithms, the frequency domain signals are not generated sequentially. In the following, both DIT and DIF algorithms of different radixes will be introduced.

4.1.1 Radix-2

The basic concept underlying the radix-2 FFT algorithm is the use of symmetry between W_N^{nk} and $W_N^{nk+N/2}$.

In DIT FFT algorithm, time domain signals are first partitioned into even samples and odd samples. Then, we can get

$$X[k]=\sum_{m=0}^{N/2-1}x[2m]W_N^{2mk}+\sum_{m=0}^{N/2-1}x[2m+1]W_N^{(2m+1)k} \qquad k=0,1,\cdots,N-1 \tag{4.3}$$

Since $W_N^{2mk}=W_{N/2}^{mk}$, for the first half of $X[k]$ ($0\le k<N/2$), equation (4.3) becomes

$$X[k]=\sum_{m=0}^{N/2-1}x[2m]W_{N/2}^{mk}+W_N^k\sum_{m=0}^{N/2-1}x[2m+1]W_{N/2}^{mk}, \qquad k=0,1,\cdots,N/2-1 \tag{4.4}$$

Due to the symmetry of $W_N^{k+N/2}=-W_N^k$, the second half can be given by

$$\begin{aligned} X[k+\frac{N}{2}] &= \sum_{m=0}^{N/2-1}x[2m]W_{N/2}^{m(k+N/2)}+W_N^{k+N/2}\sum_{m=0}^{N/2-1}x[2m+1]W_{N/2}^{m(k+N/2)} \\ &= \sum_{m=0}^{N/2-1}x[2m]W_{N/2}^{mk}-W_N^k\sum_{m=0}^{N/2-1}x[2m+1]W_{N/2}^{mk} \\ & k=0,1,\cdots,N/2-1 \end{aligned} \tag{4.5}$$

So the N-point FFT can be regarded as a combination of $N/2$-point FFT of even numbered time domain signals $x[2m]$ and $N/2$-point FFT of odd numbered time domain signals $x[2m+1]$ multiplied by W_N^k, as shown in Fig. 4.1. The basic arithmetic module is called "butterfly" operation, with one addition and one subtraction. Similarly, a $(N/2)$-point FFT can be further decomposed into two $(N/2)$-point FFTs and so on, until 2-point FFTs.

For the DIF FFT, consider only even frequency domain signals, we can get

$$X[2m]=\sum_{n=0}^{N-1}x[n]W_N^{2mn}, \qquad m=0,1,\cdots,N/2-1 \tag{4.6}$$

By partitioning the sum into two halves, the equation (4.6) takes the form of

$$\begin{aligned} X[2m] &= \sum_{n=0}^{N/2-1} x[n] W_N^{2mn} + \sum_{n=N/2}^{N-1} x[n] W_N^{2mn} \\ &= \sum_{n=0}^{N/2-1} x[n] W_N^{2mn} + \sum_{n=0}^{N/2-1} x[n+N/2] W_N^{2m(n+N/2)} \end{aligned} \tag{4.7}$$

Since we have $W_N^{2mn} = W_{N/2}^{mn} = W_N^{2m(n+N/2)}$, the equation (4.7) becomes

$$X[2m] = \sum_{n=0}^{N/2-1} (x[n] + x[n+N/2]) W_{N/2}^{nm}, \qquad m = 0,1,\cdots,N/2-1 \tag{4.8}$$

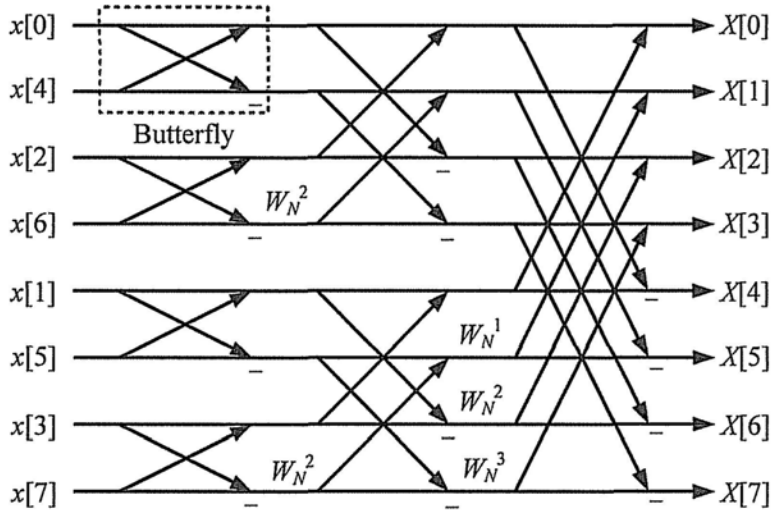


Figure 4.1 Signal flow graph of radix-2 DIT decomposition in 8-point FFT

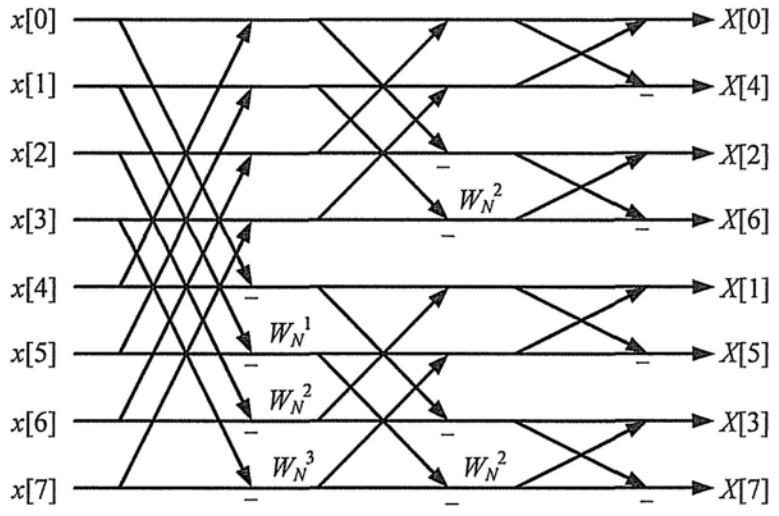


Figure 4.2 Signal flow graph of radix-2 DIF decomposition in 8-point FFT

From (4.8), all $X[2m]$ can be computed from a $(N/2)$-point FFT of the sequence $x[n] + x[n+N/2]$. Similarly, the odd frequency domain signals are given by

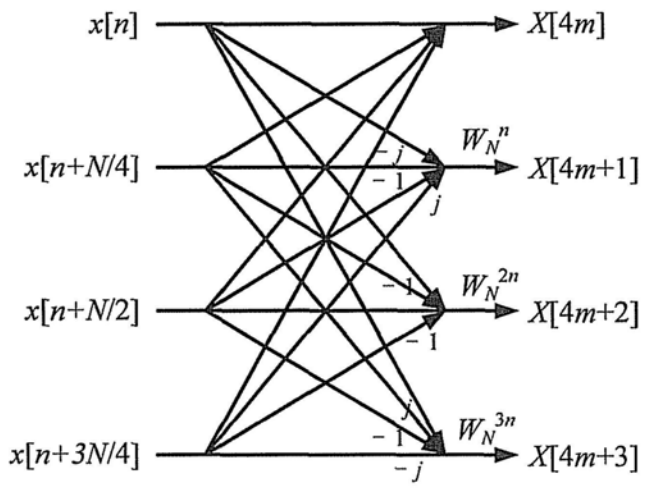
$$\begin{aligned} X[2m+1] &= \sum_{n=0}^{N/2-1} x[n] W_N^{n(2m+1)} + \sum_{n=0}^{N/2-1} x[n+N/2] W_N^{(n+N/2)(2m+1)} \\ &= \sum_{n=0}^{N/2-1} (x[n]-x[n+N/2]) W_N^{n(2m+1)} \\ &= \sum_{n=0}^{N/2-1} (x[n]-x[n+N/2]) W_N^n W_{N/2}^{mn}, \qquad m=0,1,\cdots,N/2-1 \end{aligned} \tag{4.9}$$

Fig. 4.2 shows the signal flow graph of radix-2 DIF decomposition in 8-point FFT. It is clear that DIT and DIF algorithms have the same arithmetic complexity. There are $\log_2 N$ stages and each stage has N complex multiplications and N complex additions. Accordingly, radix-2 FFT algorithm has $O(N \log_2 N)$ computational complexity. Since the complexity of DIT and DIF are the same, only the DIF FFT algorithms for high-radix will be illustrated in the following.

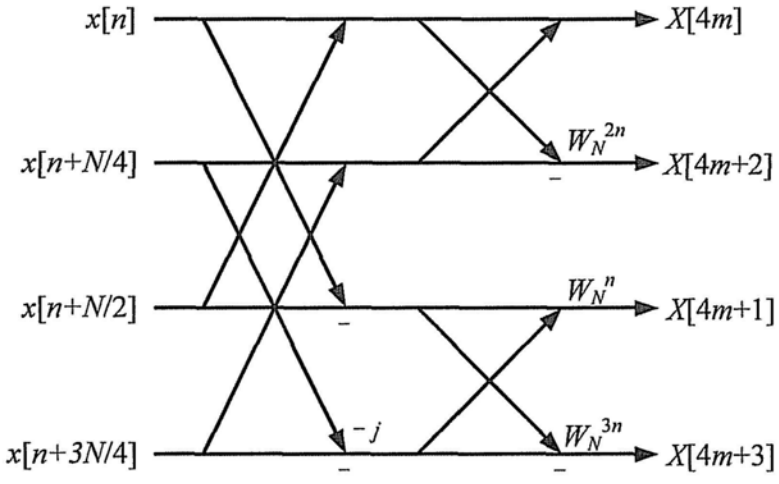
4.1.2 Radix-4

Four-way symmetry of W_N^{nk} ($W_N^{nk+N/4} = -W_N^{nk+3N/4} = -jW_N^{nk}$) has been utilized to minimize the number of complex multiplications in the radix-4 algorithm. For m = 0, 1, …, $N/4$ − 1, its formula can be derived as

$$\begin{aligned} X[4m] &= \sum_{n=0}^{N/4-1} \{x[n]+x[n+\tfrac{N}{4}]+x[n+\tfrac{N}{2}]+x[n+\tfrac{3N}{4}]\} W_{N/4}^{nm} \\ X[4m+1] &= \sum_{n=0}^{N/4-1} \{x[n]-jx[n+\tfrac{N}{4}]-x[n+\tfrac{N}{2}]+jx[n+\tfrac{3N}{4}]\} W_N^n W_{N/4}^{nm} \\ X[4m+2] &= \sum_{n=0}^{N/4-1} \{x[n]-x[n+\tfrac{N}{4}]+x[n+\tfrac{N}{2}]-x[n+\tfrac{3N}{4}]\} W_N^{2n} W_{N/4}^{nm} \\ X[4m+3] &= \sum_{n=0}^{N/4-1} \{x[n]+jx[n+\tfrac{N}{4}]-x[n+\tfrac{N}{2}]-jx[n+\tfrac{3N}{4}]\} W_N^{3n} W_{N/4}^{nm} \end{aligned} \tag{4.10}$$



(a) Radix-4 butterfly



(b) Radix-2^2 butterfly

Figure 4.3 Signal flow graph of radix-4 and radix-2^2 butterfly operation

Fig. 4.3 (a) depicts the signal flow graph of radix-4 FFT butterfly operation. The radix-4 butterfly can be further decomposed and implemented by cascading two radix-2 stages, known as radix-2^2 algorithm. The signal flow graph of such an algorithm is shown in Fig. 4.3 (b).

4.2 FFT Architectures

Pipeline FFT processor is a class of architectures for application specific FFT computation utilizing fast algorithms. It is characterized by non-stopping processing on a clock frequency of the input data. It is also the solution for high speed processing and low power consumption. In the following, we will introduce some classical pipeline architectures.

4.2.1 Multi-path Delay Commutator (MDC)

Radix-2 multi-path delay commutator (MDC) [62] was probably the most classical approach for pipeline implementation of radix-2 FFT algorithm. The input sequence has been broken into two parallel data stream flowing forward, with correct distance between the data elements entering the butterfly scheduled by proper delays. Fig. 4.4 shows the radix-2 MDC FFT architecture. Totally, there are ($\log_2 N$ − 2) complex multipliers, $\log_2 N$ butterfly units and ($3N/2$ − 2) registers. Both butterfly units and multipliers can work in 50% utilization.

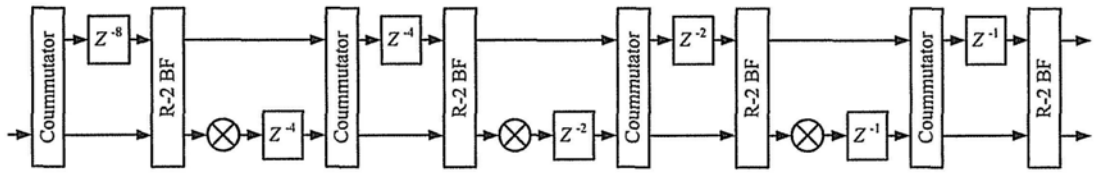


Figure 4.4 Radix-2 MDC FFT architecture (N = 16)

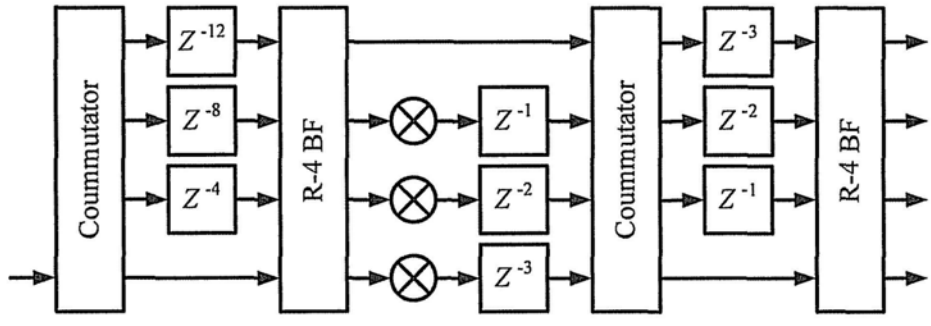


Figure 4.5 Radix-4 MDC FFT architecture (N = 16)

Fig. 4.5 shows the radix-4 MDC architecture. It has been used as the architecture for the initial VLSI implementation of pipeline FFT processor [65] and massive wafer scale integration [66]. It can be implemented in a similar way as the radix-2 MDC architecture, except that four parallel data streams are processed simultaneously and the butterfly units must handle radix-4 butterfly operation. Radix-4 MDC architecture requires $3(\log_4 N - 1)$ multipliers, $\log_4 N$ full radix-4 butterflies and $(5N/2 - 4)$ registers.

4.2.2 Single-path Delay Feedback (SDF)

The radix-2 single-path delay feedback (SDF) [63] utilizes the registers more efficiently by storing the butterfly output in feedback shift registers, as shown in Fig. 4.6. A single data stream goes through the multiplier at every stage. This architecture has the same number of butterfly units and multipliers as those in the radix-2 MDC architecture, but much reduces the memory size: with only $N - 1$ delay elements.

Fig. 4.7 shows the radix-4 SDF FFT architecture [64] employing CORDIC iterations. The utilization of multipliers has been increased to 75% by storing 3 out of 4 radix-4 butterfly outputs. However, the utilization of the radix-4 butterfly, which is fairly complicated and contains at least 8 complex adders, is drooped to 25%. It requires $(\log_4 N - 1)$ multipliers, $\log_4 N$ full radix-4 butterflies and storage of size $N - 1$.

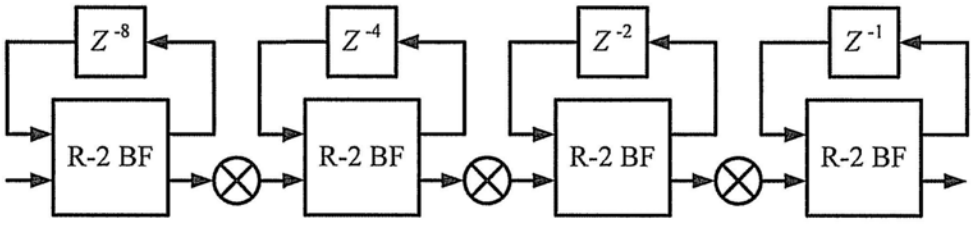


Figure 4.6 Radix-2 SDF FFT architecture ($N = 16$)

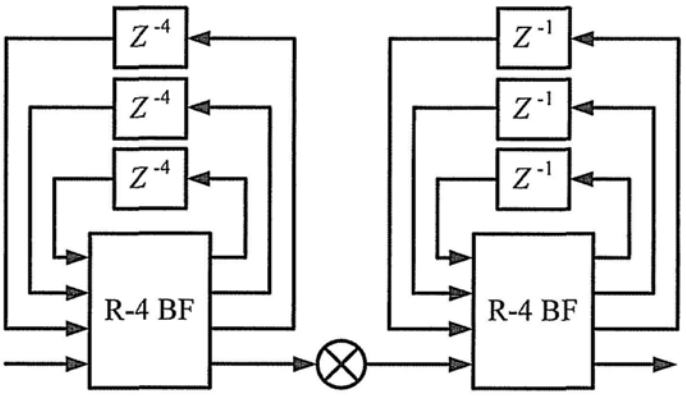


Figure 4.7 Radix-4 SDF FFT architecture ($N = 16$)

4.2.3 Single-path Delay Communtator (SDC)

Radix-4 single-path delay commutator [67] uses a modified radix-4 algorithm with programmable 1/4 radix-4 butterflies to achieve higher utilization of multiplier (75%). A multiplexed delay commutator also reduced the memory size to $2N - 2$. The butterfly units and delay commutators become complicated due to programmability requirement. The architecture of radix-4 SDC is shown in Fig. 4.8.

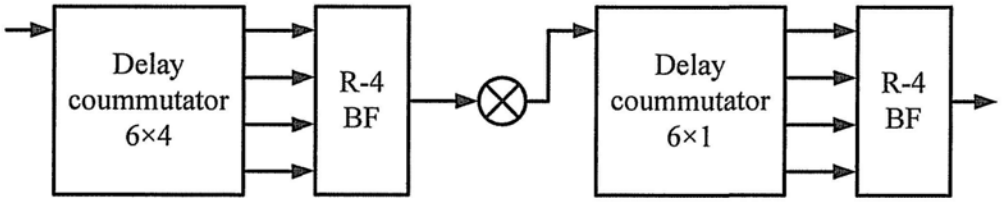


Figure 4.8 Radix-4 SDC FFT architecture ($N = 16$)

The hardware complexity of MDC, SDF and SDC architectures are compared in Table 4.1. The number of complex multipliers, adders, buffer size and the complexity of control logic are listed. For easy reading, base-4 logarithm is used whenever applicable. Due to efficient use of delay buffers, SDF architectures have the minimum memory size. In SDF architectures, the higher radix algorithms reduce the number of complex multipliers. On the other hand, the higher radix butterfly units need complicated adders. The Table 4.1 indicates that R-2^2 SDF has reach the minimum

requirement for both multipliers and storages, and only second to R-4 SDC for adders. This makes it an idea architecture for VLSI implementation of pipeline FFT processor.

Table 4.1 Complexity comparison of pipelined FFT architecture

	R-2 MDC	R-4 MDC	R-2 SDF	R-4 SDF	R-2^2 SDF	R-4 SDC
Complex multipliers	$2(\log_4 N-1)$	$3(\log_4 N-1)$	$2(\log_4 N-1)$	$\log_4 N-1$	$\log_4 N-1$	$\log_4 N-1$
Complex adders	$4\log_4 N$	$8\log_4 N$	$4\log_4 N$	$8\log_4 N$	$4\log_4 N$	$3\log_4 N$
Delay buffer	$3N/2-2$	$5N/2-4$	$N-1$	$N-1$	$N-1$	$2N-2$
Control logic	Simple	Simple	Simple	Medium	Simple	Complex

4.3 Proposed FFT Design

For MB-OFDM UWB system, the processing time of the 128-point FFT is only 312.5 ns, which means a high throughput and low complexity FFT is required. As comparison and analysis of pipelined FFT architectures before, the architecture of SDF has the lowest complexity. The throughput is improved by employing mixed high radix algorithm. Therefore, the proposed FFT is designed with radix-2^4 cascaded with radix-2^3 SDF architecture.

4.3.1 Algorithm of proposed FFT

Decompose the 128-point FFT to a two-dimensional DFT. One is 16-point and the other one is 8-point. The equation (4.1) can be expressed as

$$
\begin{aligned}
X(k_1+16k_2) &= \sum_{n_2=0}^{7}\sum_{n_1=0}^{15} x(8n_1+n_2)\cdot W_{128}^{(8n_1+n_2)(k_1+16k_2)} \\
&= \underbrace{\sum_{n_2=0}^{7}\left\{\underbrace{\sum_{n_1=0}^{15} x(8n_1+n_2)W_{16}^{n_1k_1}}_{16-point\ DFT}\cdot \underbrace{W_{128}^{n_2k_1}}_{twiddle\ factor}\right\}\cdot W_8^{n_2k_2}}_{8-point\ DFT} \\
&= \sum_{n_2=0}^{7} BF_{16}(k_1,n_2)\cdot W_{128}^{n_2k_1}\cdot W_8^{n_2k_2}
\end{aligned}
\tag{4.11}
$$

where

$$
\begin{aligned}
n &= 8n_1+n_2, \quad \begin{cases} n_1 = 0\cdots 15 \\ n_2 = 0\cdots 7 \end{cases} \\
k &= k_1+16k_2, \quad \begin{cases} k_1 = 0\cdots 15 \\ k_2 = 0\cdots 7 \end{cases}
\end{aligned}
\tag{4.12}
$$

Then using radix-2^3 algorithm, further decompose the butterfly of radix-8 FFT algorithm into three-dimensional radix-2 DFT, n_2 and k_2 can be defined as

$$
\begin{cases} n_2 = 4p_1+2p_2+p_3 \\ k_2 = q_1+2q_2+4q_3 \end{cases} \quad \begin{cases} p_1,p_2,p_3 = 0,1 \\ q_1,q_2,q_3 = 0,1 \end{cases}
\tag{4.13}
$$

By means of (4.13), (4.11) takes the form of

$$
\begin{aligned}
&X(k_1+16(q_1+2q_2+4q_3)) = \\
&\underbrace{\underbrace{\underbrace{\sum_{p_3=0}^{1}\sum_{p_2=0}^{1}\sum_{p_1=0}^{1} BF_{16}(k_1,4p_1+2p_2+p_3)W_2^{p_1q_1}\underbrace{W_4^{p_2q_1}W_2^{p_2q_2}}_{TF}\underbrace{W_8^{p_3(q_1+2q_2)}}_{TF}W_2^{p_3q_3}}_{step\,1}}_{step\,2}}_{step\,3}
\end{aligned}
\tag{4.14}
$$

The eight-point DFT can be divided into three steps by mapping radix-2 index, as shown in (4.14). And similarly employing radix-2^4 algorithm, further decompose the butterfly of radix-16 to four-dimensional radix-2 DFT, n_1 and k_1 can be defined as

$$
\begin{cases} n_1 = 8\alpha_1+4\alpha_2+2\alpha_3+\alpha_4 \\ k_1 = \beta_1+2\beta_2+4\beta_3+8\beta_4 \end{cases} \quad \begin{cases} \alpha_1,\alpha_2,\alpha_3,\alpha_4 = 0,1 \\ \beta_1,\beta_2,\beta_3,\beta_4 = 0,1 \end{cases}
\tag{4.15}
$$

The radix-16 butterfly unit BF_{16} can be divided into four steps by mapping radix-2 index, and is expressed as

$$BF_{16} = \sum_{\alpha_4=0}^{1}\sum_{\alpha_3=0}^{1}\sum_{\alpha_2=0}^{1}\underbrace{\sum_{\alpha_1=0}^{1} x\, W_2^{\alpha_1\beta_1}\, \underbrace{W_4^{\alpha_2\beta_1}}_{TF}}_{step\,1} W_2^{\alpha_2\beta_2}\, \underbrace{W_8^{\alpha_3(\beta_1+2\beta_2)}}_{TF}}_{step\,2} W_2^{\alpha_3\beta_3}\, \underbrace{W_{16}^{\alpha_4(\beta_1+2\beta_2+4\beta_3)}}_{TF}}_{step\,3} W_2^{\alpha_4\beta_4} \quad (4.16)$$

Fig. 4.9 shows the signal flow graph of the proposed 128-point mixed radix FFT. It is divided into two stages, seven steps. Radix-2^4 FFT algorithm is used in the first stage and further decomposed into four steps while radix-2^3 FFT algorithm is used in the second stage and further decomposed into three steps. There are 120 twiddle factors between the first stage and the second stage, however, just 31 of them are actually used in multiplications and others can be implemented by swap or complement operations.

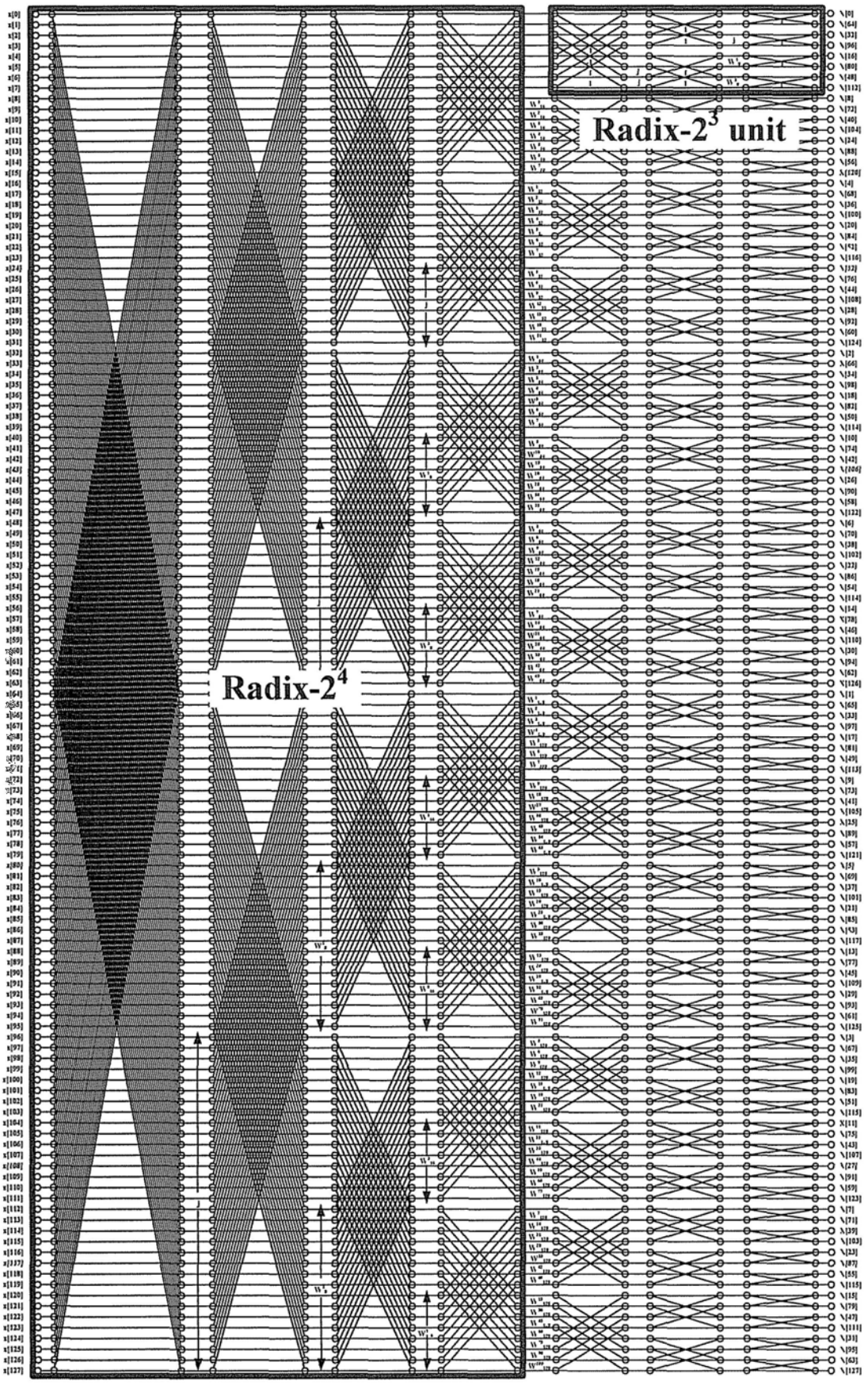


Figure 4.9 Signal flow graph of the proposed 128-point mixed radix FFT

4.3.2 Architecture of proposed FFT

Most of the FFT designs applied in UWB [68]-[71] have the word length of 10 bits in both the real and imaginary parts to keep the quantization noise to the least value and to minimize the hardware complexity. The word length is fixed in each butterfly step in their designs, as shown in Fig. 4.10. Decompose the word into two parts, integer part and fraction part. In each step, the word length of integer part should be 1 bit longer than last step to avoid overflow. Accordingly, for fixed word length method, the word length of fraction part in each step should be 1 bit shorter than last step. The signal-to-quantization noise ratio (SQNR) is usually used to measure the accuracy of

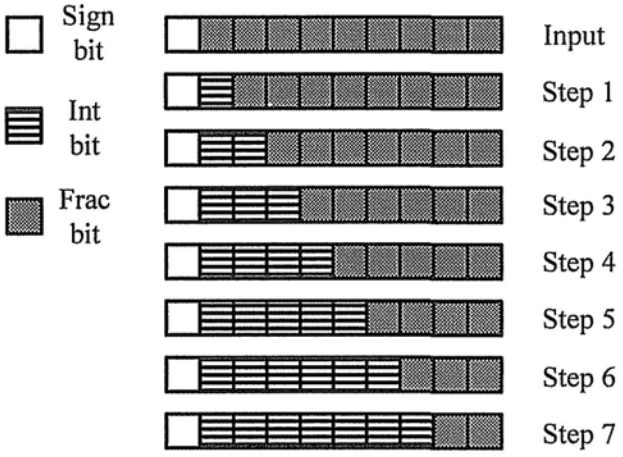


Figure 4.10 Fixed word length method in [68]-[71]

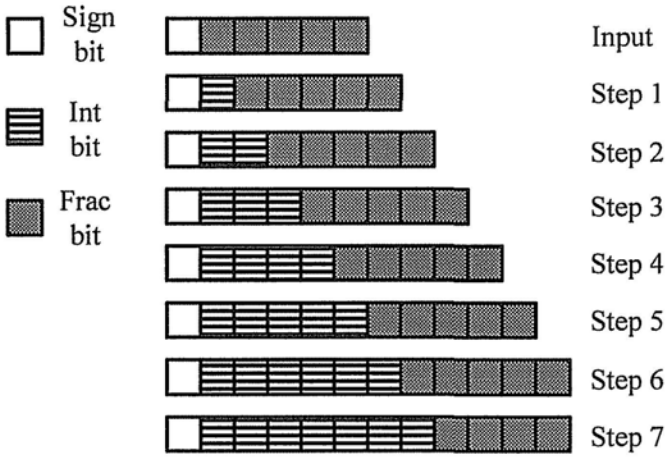


Figure 4.11 The proposed variable word length method

the design. Actually, SQNR is just determined by the word length of fraction part, but is not related to the integer part.

Therefore, the proposed FFT architecture is designed to guarantee the word length of fraction part and the whole word length is variable, as shown in Fig. 4.11. The SQNR simulation result of the proposed variable word length method is 39.3 dB while the 10-bit fixed word length method is 34.7 dB. The variable word length not only has better SQNR performance than the fixed word length method, but also saves the memory size.

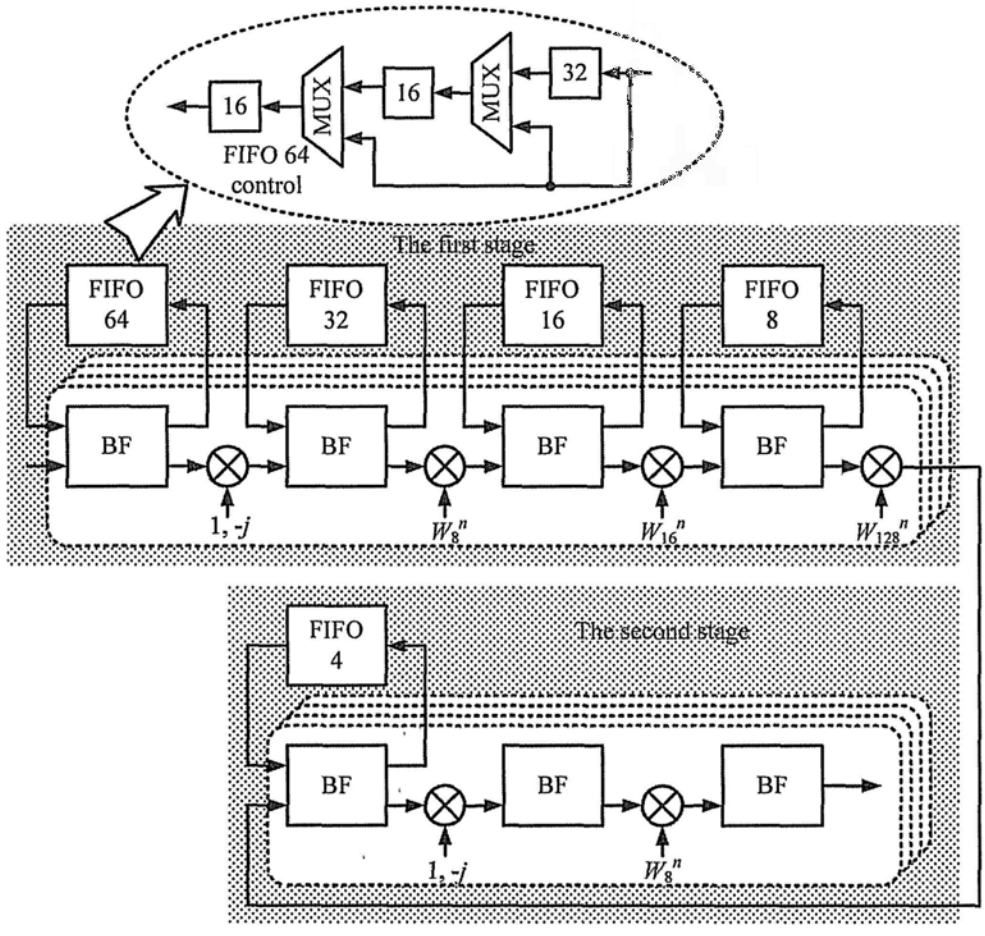


Figure 4.12 Block diagram of the proposed 128-point FFT

The architecture of the proposed FFT engine is shown in Fig. 4.12. The depth of each FIFO is illustrated. Obviously, the four steps in the first stage requires more than 90% of the memory depth. The proposed variable word length method has much

shorter word length in the first stage than the second stage and the word length slowly increased each step, which is the optimum scheme for using least memory units. The memory size of the proposed FFT is 1944 bits but the memory size of the fixed word length method is 2480 bits.

The control circuit of the FIFO with depth of 64 is shown in Fig. 4.12. Among the multipliers, only the forth step in the first stage adopts complex multipliers. And in other steps, the multiplications with trivial twiddle factors can be implemented with simple constant multipliers and multiplexers.

4.3.3 Architecture of output reorder buffer

The 128-point FFT in the proposed MB-OFDM UWB receiver is developed with DIF algorithm and the output data of which are not arranged into the normal sequence. To arrange the output data into the normal order, output reorder buffer is necessary. If the input data of FFT module are in the sequence of (y[0], y[1], y[2], y[3], ...), the output data will be in the order of (Y[0], Y[64], Y[32], Y[96], ...). Represent the address of output data in binary mode, it is easy to find that the address of output data can be obtained by reversing the corresponding address of the input data. Take the output data Y[64] for example, its address can be expressed as binary data 1000000, which is exactly the reversing style of the address of corresponding input data y[1]. In this case, the reordering of the FFT output data can be processed by writing the data to the buffer with the address sequence of (0, 64, 32, 96, ...) and reading the data from the buffer with the address sequence of bit-reversing style, (0, 1, 2, 3, ...).

Designed with four parallel architecture, the interval between two FFT symbols is 9 or 10 clock cycles. In order to avoid writing or reading conflicts, the FFT output reorder buffer should be with the depth of 256. In UWB specification, the FFT inputs (1 ... 128) are mapped to the outputs (65 ... 128, 1 ... 64). To conduct this mapping, another buffer with depth of 64 is necessary. Since buffer is the unit consumes much power and covers large area, we explore a novel address coding method to reduce the depth of the buffer to 128.

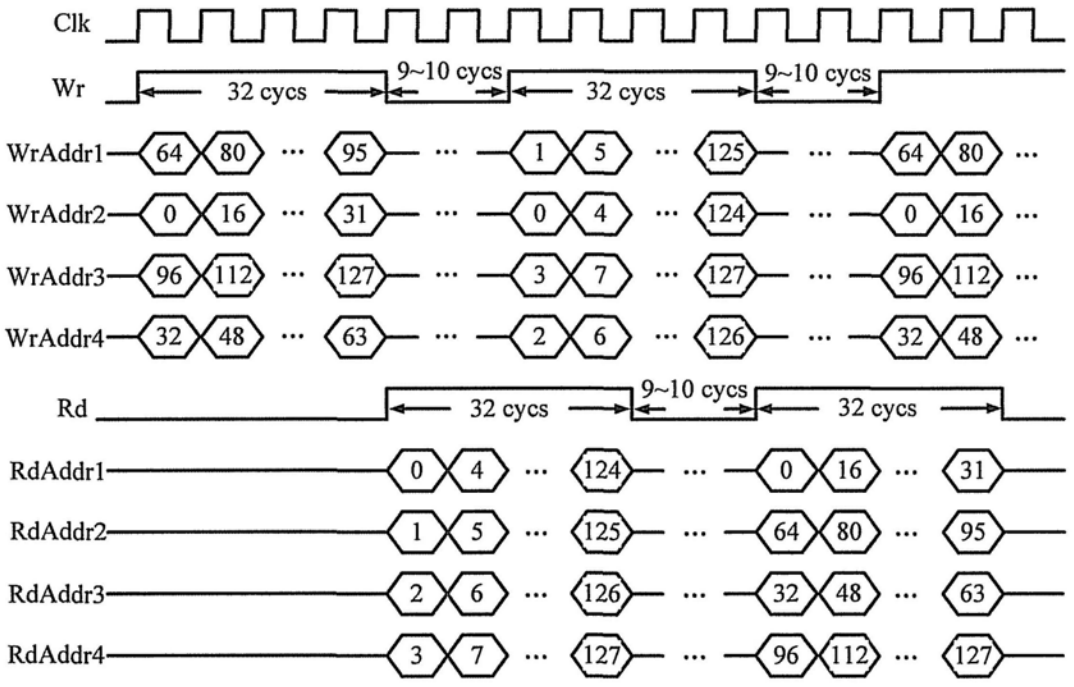


Figure 4.13 Timing diagram of data reorder in FFT output buffer

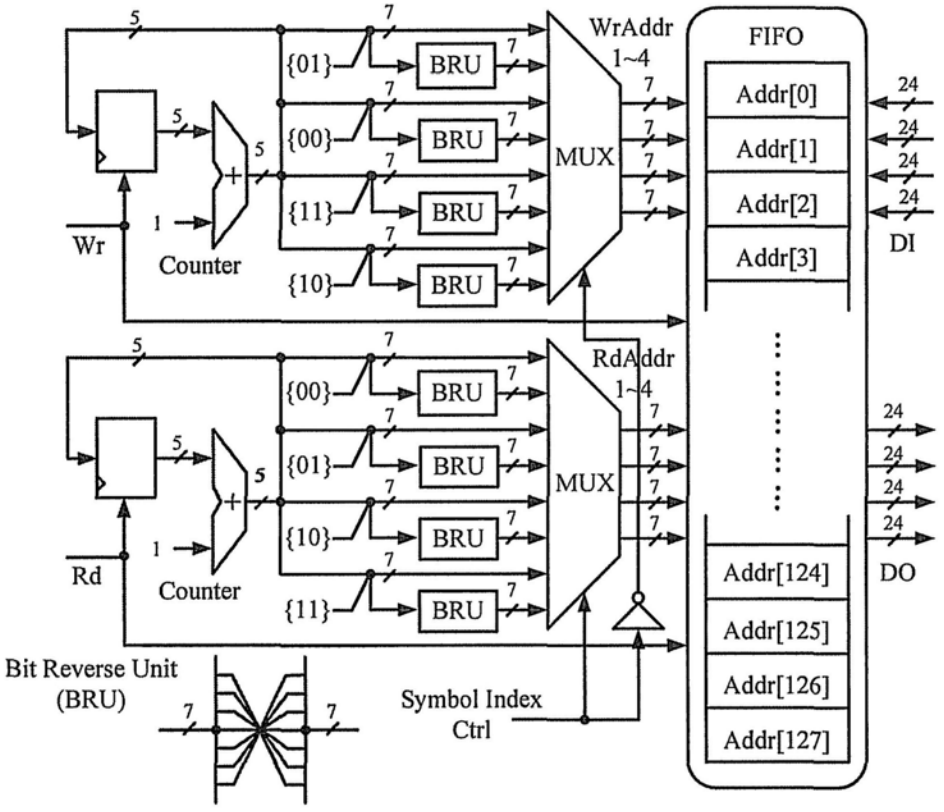


Figure 4.14 The architecture of FFT output reorder buffer

The timing diagram of the proposed reorder buffer is shown in Fig. 4.13. FFT outputs of the first symbol are stored in the memory with the bit-reversing sequence as mentioned above, meanwhile exchange *WrAddr1* with *WrAddr2* and *WrAddr3* with *WrAddr4*. The writing address sequence is (64, 0, 32, 96, ...) and read the data out of buffer with normal order. This operation is equivalent to reordering the FFT outputs and mapping the inputs (1 ... 128) to the outputs (65 ... 128, 1 ... 64). When the next symbol is coming, store the data in the memory with the order of (1, 0, 3, 2, ...) and read the data out of buffer with bit-reversing order (0, 64, 32, 96, ...). In general case, process the odd symbols as the first symbol and the even symbols as the second symbol. As illustrated in Fig. 4.13, this symbol interleaving reorder method avoids reading and writing conflicts and saves the buffer depth of 192.

Fig. 4.14 shows the architecture of the FFT output reorder buffer. The writing addresses for odd symbols can be obtained by attaching the 2-bit sequence {01 00 11 10} to a 5-bit counter. For even symbols, the MUX select the data by bit-reversing units. And the reading addresses for even symbols can be encoded by attaching the 2-bit sequence {00 01 10 11} to a 5-bit counter. For odd symbols, the MUX select the data by bit-reversing units.

Chapter 5 Channel Estimation

Channel estimation is the task of estimating the frequency response of the radio channel the transmitted signal travels before reaching the receiver antenna. The impulse response of a time varying radio channel is usually represented as a discrete time finite impulse response (FIR) filter. In order to mitigate hostile channel effects on the received signal, precise channel estimation is required to make the signal reliable for further processing.

Channel estimation can be categorized as non-data-aided and data-aided. Non-data-aided or blind channel estimation estimates channel response by statistics of the received signals. No specialized training signals are needed and the transmission efficiency is retained for employing blind estimation scheme. However, without precise knowledge of the transmitted signals, a large number of data must be collected in order to obtain reliable estimation. On the other hand, the data-aided channel estimation requires training signals to be transmitted. Rapid and accurate channel estimation can be achieved by comparing the received and transmitted reference signals. A sufficient number of such reference signals must be inserted according to the degree of channel variation, namely coherent time and coherent bandwidth of channel under estimation.

To obtain precise channel estimation for equalizing each subcarrier, MB-OFDM standard provides reference signals, namely pilots. This chapter will focus on data-aided channel estimation algorithms for MB-OFDM UWB communications.

5.1 Pilot Patterns

In data-aided channel estimation, known information to the receiver is inserted in symbols so that the current channel can be estimated. Two techniques are commonly used: sending known information over one or more symbols with no data being sent, or sending known information with data. The previous arrangement is usually called

channel estimation with training symbols while the latter is called pilots aided channel estimation [72]. Fig. 5.1 and Fig. 5.2 shows the typical training symbols and pilot subcarriers arrangement respectively. Channel estimation employing training symbols periodically sends training symbols so that the channel estimates are updated. In the pilots aided channel estimation, the pilots are multiplexed with the data.

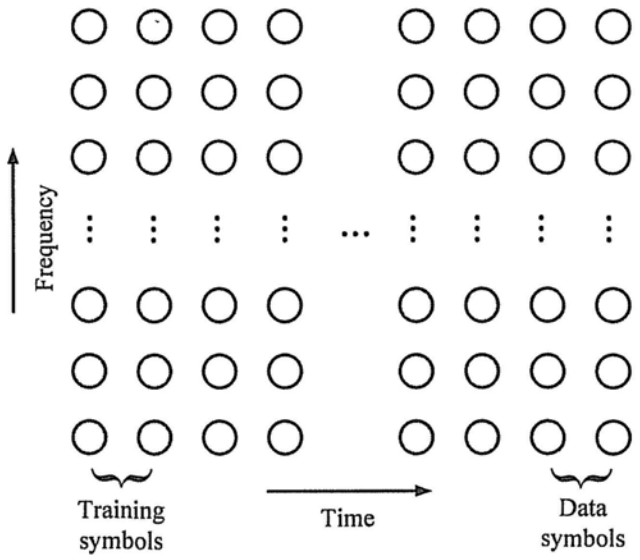


Figure 5.1 Training symbols arrangement

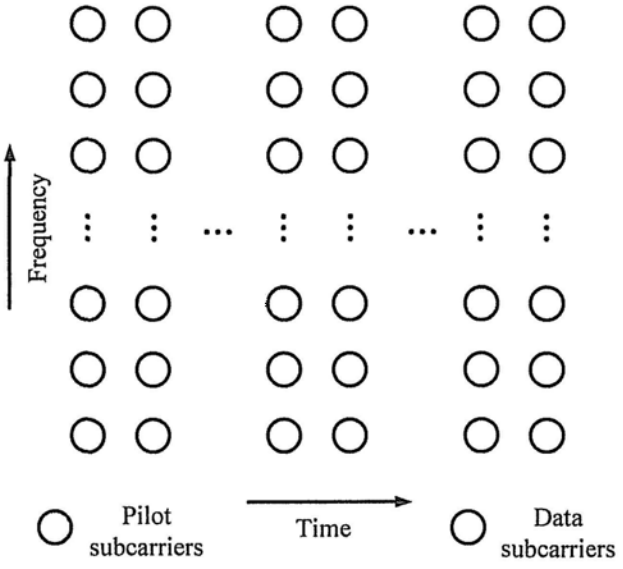


Figure 5.2 Pilot aided subcarriers arrangement

5.2 Traditional Channel Estimation Techniques

Before going into the details of the estimation techniques, it is necessary to give the matrix notations of the received signal as

$$\mathbf{Y} = \mathbf{XH} + \mathbf{W} = \mathbf{XFh} + \mathbf{W} \tag{5.1}$$

where

$$\begin{aligned}
&\mathbf{X} = diag[X_{-N/2}\ X_{-N/2+1} \cdots X_{-N/2-1}] \\
&\mathbf{Y} = [Y_{-N/2}\ Y_{-N/2+1} \cdots Y_{-N/2-1}]^T \\
&\mathbf{H} = [H_{-N/2}\ H_{-N/2+1} \cdots H_{-N/2-1}]^T \\
&\mathbf{h} = [h_0\ h_1 \cdots h_{N-1}]^T \\
&\mathbf{W} = [W_{-N/2}\ W_{-N/2+1} \cdots W_{-N/2-1}]^T \\
&\mathbf{F} = \begin{bmatrix} 1 & e^{-j2\pi(-N/2)/N} & \cdots & e^{-j2\pi(-N/2)(N-1)/N} \\ \vdots & \vdots & & \vdots \\ 1 & e^{-j2\pi(N/2-1)/N} & \cdots & e^{-j2\pi(N/2-1)(N-1)/N} \end{bmatrix}
\end{aligned} \tag{5.2}$$

In (5.1) and (5.2), **F** is the unitary FFT matrix; **X** is the diagonal elements of transmitted signal at each subcarrier; **Y**, **h** and **W** are the column vectors of the received signal, channel transfer function and noise at each subcarrier respectively. In the following, two classical channel estimation techniques will be introduce, least squares (LS) [73]-[75], maximum likelihood (ML) [77]-[78] and minimum mean square error (MMSE) [79], [80].

5.2.1 LS estimation

LS method, in general, is utilized to get initial channel estimation at the pilot subcarriers , which are then further improved via different methods. In LS method, assumes the time domain channel impulse response is deterministic and tries to find $\mathbf{h}_{LS}$ to minimize $(\mathbf{Y} - \mathbf{XFh})^H(\mathbf{Y} - \mathbf{XFh})$, where $(\cdot)^H$ is conjugate transpose operation. The LS solution is then given by

$$\hat{\mathbf{H}}_{LS} = (\mathbf{F}^H\mathbf{X}^H\mathbf{XF})^{-1}\mathbf{F}^H\mathbf{X}^H\mathbf{Y} = \mathbf{X}^{-1}\mathbf{Y} \tag{5.3}$$

Without using any knowledge of the statistics of the channels, LS estimation is with low complexity, but suffers from high MSE. So in general, it is utilized to get initial channel estimation.

5.2.2 ML estimation

If the number of pilots is greater than the channel length and the noise, the LS estimation is equivalent to ML estimation. With the assumption of L channel taps and N_p number of pilot subcarriers, the ML estimation of the channel coefficients is expressed as

$$\hat{\mathbf{H}}_{ML} = (\mathbf{F}_p^H \mathbf{F}_p)^{-1} \mathbf{F}_p^H \mathbf{X}^H \mathbf{Y} \tag{5.4}$$

where $\mathbf{F}_p$ is $N_p \times L$ truncated unitary Fourier matrix. For the sake of simplicity, it is assumed the transmitted signal matrix meets $\mathbf{X}^H\mathbf{X} = \mathbf{I}_K$, and it does not appear in the parenthesis for the inverse operation.

5.2.3 MMSE estimation

MMSE uses additional information like the operating SNR and the other channel statistics. However, the computational complexity is very high due to extra information incorporated in the estimation technique. Assume the time domain channel impulse response is a random vector with Gaussian distribution and is uncorrelated with noise. Then the MMSE estimator that minimizes $E\{(\hat{\mathbf{h}}-\mathbf{h})^H(\hat{\mathbf{h}}-\mathbf{h})\}$ takes the form of

$$\hat{\mathbf{h}}_{MMSE} = \mathbf{R}_{hy}\mathbf{R}_{yy}^{-1}\mathbf{Y} \tag{5.5}$$

where

$$\mathbf{R}_{hy} = E\{\mathbf{h}\mathbf{Y}^H\} = \mathbf{R}_{hh}\mathbf{F}^H\mathbf{X}^H \tag{5.6}$$

is the cross-covariance matrix between $\mathbf{h}$ and $\mathbf{Y}$.

$$\mathbf{R}_{yy} = E(\mathbf{Y}\mathbf{Y}^H) = \mathbf{X}\mathbf{F}\mathbf{R}_{hh}\mathbf{F}^H\mathbf{X}^H + \sigma_w^2\mathbf{I}_N \tag{5.7}$$

is the auto-covariance matrix of **Y**; $\mathbf{R}_{hh}$ is the auto-covariance matrix of **h**. σ_w^2 is the variance of noise, and $\mathbf{I}_N$ is the identity matrix. As a result, the MMSE frequency domain channel response is given by

$$\hat{\mathbf{H}}_{MMSE} = \mathbf{F}\hat{\mathbf{h}}_{MMSE} \tag{5.8}$$

The complexity of MMSE estimation can be significantly reduced if it is independent of the transmitted signals. Assuming the transmitted symbols use the same constellation, rewritten (5.8) as

$$\hat{\mathbf{H}}_{MMSE} = \mathbf{R}_{H_pH_p}(\mathbf{R}_{H_pH_p} + \frac{\beta}{SNR}\mathbf{I}_{N_p})^{-1}\hat{\mathbf{H}}_{\mathrm{LS}} \tag{5.9}$$

where $\beta = E\{|X_k^2|\}\ E\{1/|X_k^2|\}$; $\mathbf{R}_{HpHp}$ represents the auto-correlation between the pilot subcarriers and

$$\frac{\beta}{SNR}\mathbf{I}_{N_p} = E\{(\mathbf{X}\mathbf{X}^H)^{-1}\} \tag{5.10}$$

Optimum rank reduction is achieved by using Singular Value Decomposition (SVD) [74]. The SVD of the channel auto-covariance matrix is

$$\mathbf{R}_{H_pH_p} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H \tag{5.11}$$

where **U** is a unitary matrix and $\mathbf{\Lambda}$ is a diagonal matrix with singular values $\lambda_0, \lambda_1, \ldots \lambda_{Np-1}$ in descending orders. Then the channel can be estimated as

$$\hat{\mathbf{H}}_{MMSE} = \mathbf{U}\mathbf{\Delta}\mathbf{U}^H\hat{\mathbf{H}}_{LS} \tag{5.12}$$

where $\mathbf{\Delta}$ is a diagonal matrix with entries

$$\delta_i = \frac{\lambda_i}{\lambda_i + \frac{\beta}{SNR}} \qquad i = 0,1,\ldots N_p - 1 \tag{5.13}$$

In MMSE, the statistic about the random vector, such as the covariance matrix and the SNR can help the estimation. However, it also necessitates additional time and complexity. And in high SNR scenario, MMSE boils down to the LS estimator. In LS and ML estimation, the channel response is regarded as a deterministic unknown

vector. This assumption basically uses a snap shot to represent the slow varying random process. ML estimator eliminates the noisy part of the estimated channel impulse response, so it gets better estimation performance than LS.

The equalized signal can thus be obtained by $\hat{\mathbf{X}} = \mathbf{Y} / \hat{\mathbf{H}}$.

5.3 Channel Estimation for MB-OFDM UWB

Since the MB-OFDM UWB system has training mode and the channels of which is stable and varying slowly, the proposed channel estimator design is based on LS algorithm. Rewrite the expression of channel transfer function as

$$\hat{\mathbf{H}}_{LS} = \frac{\mathbf{Y}}{\mathbf{X}} = \frac{\mathbf{Y}\mathbf{X}^*}{\mathbf{X}\mathbf{X}^*} = \frac{\mathbf{Y}\mathbf{X}^*}{\|\mathbf{X}\|^2} \tag{5.14}$$

The channel estimation training sequence $\mathbf{X}$ in frequency domain is specified as $(\pm 1 \pm j)/\sqrt{2}$. Hence $\mathbf{X}$ has the modulus of 1 ($\|\mathbf{X}\|^2 = 1$). The expression (5.14) can be simplified as $\hat{\mathbf{H}}_{LS} = \mathbf{Y}\mathbf{X}^*$. In this case, the complex divider is replaced by a complex multiplier and the hardware complexity is reduced.

The expression of equalized signals can be rewritten as

$$\hat{\mathbf{X}} = \mathbf{Y}\hat{\mathbf{H}}^* / \| \hat{\mathbf{H}} \|^2 \tag{5.15}$$

Considering the modulation scheme of MB-OFDM UWB is QPSK, only phase is corrected during signal equalization, which will not affect the performance. In this case, division operation is not necessary for data signals and the hardware complexity is simplified. However, for twelve pilots in each symbol, division based channel correction is necessary due to the proposed phase tracking scheme.

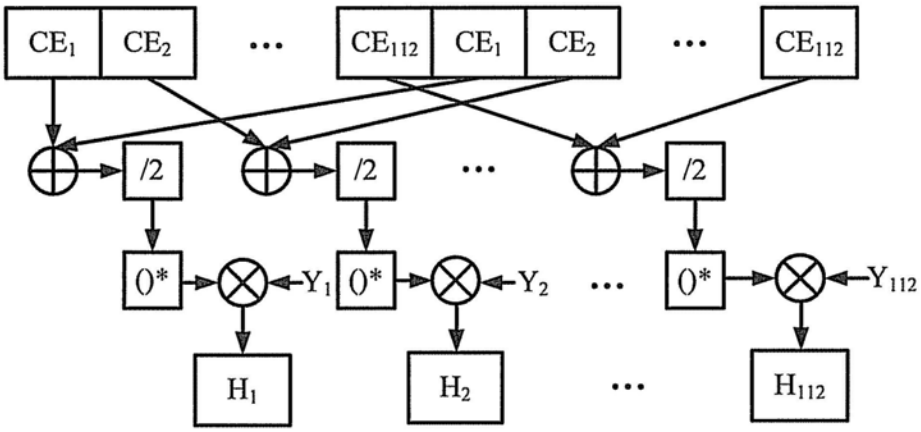


Figure 5.3 The block diagram of the proposed channel estimation

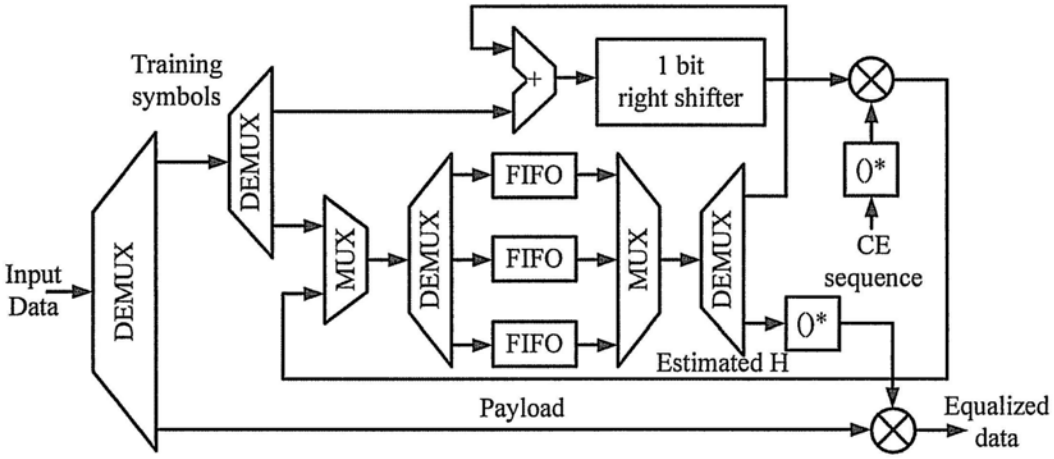


Figure 5.4 The architecture of the proposed channel estimator

The block diagram and architecture of channel estimation and signal equalization are shown in Fig 5.3 and Fig 5.4 respectively. Six training symbols are used for channel estimation. The signals of two symbols in the same band and the same subcarriers are averaged for noise reduction. Memory circuits are divided into three parts for three bands respectively. In this case, only one third of the register files are active to read/write signals of each band, which saves power consumption significantly.

Chapter 6 System Implementation

System design is a series of mapping processes. As illustrates in Fig. 6.1, a design starts from s clear system specification, and them descends gradually from higher level to lower level. Checked points are specified between successive phases to make sure that design consistency is always maintained and target specifications are met. If an intermediate design fails to meet the target function and algorithm specifications or the system behaviors are not consistent with those from higher level models, the design should be revised for a better solution.

During the initial design phase, system algorithm is developed by constructing a floating-point function model. For receiver algorithm exploration, it is also necessary to analyze and model the channel effects, non-ideal characteristics and impairments.

After completing the functional model design, the fixed-point bit-true model should be checked for hardware realization. In the next phase, the task depend on how the design is to be implemented. The fixed-point arithmetic model will be translated to hardware description language (HDL) for dedicated IC [81]-[83].

In this chapter, we implement the proposed packet detection, CFO correction, FFT, channel estimation and phase tracking algorithms and integrate them into a baseband UWB receiver. System is optimized by compromising among cost, performance, power consumption and hardware constrains of building blocks to obtain acceptable designs meeting the overall system performance specifications.

In algorithm design phase, the signal quality degradation caused by various impairment sources should be considered. In general, these impairments may be from interference sources, analog front end, estimation errors in synchronization and gain control loops, and the finite precision effect. The parameter estimation accuracy is usually determined by the estimation algorithms, while the finite precision effect can be reduced simply by increasing the signal word length.

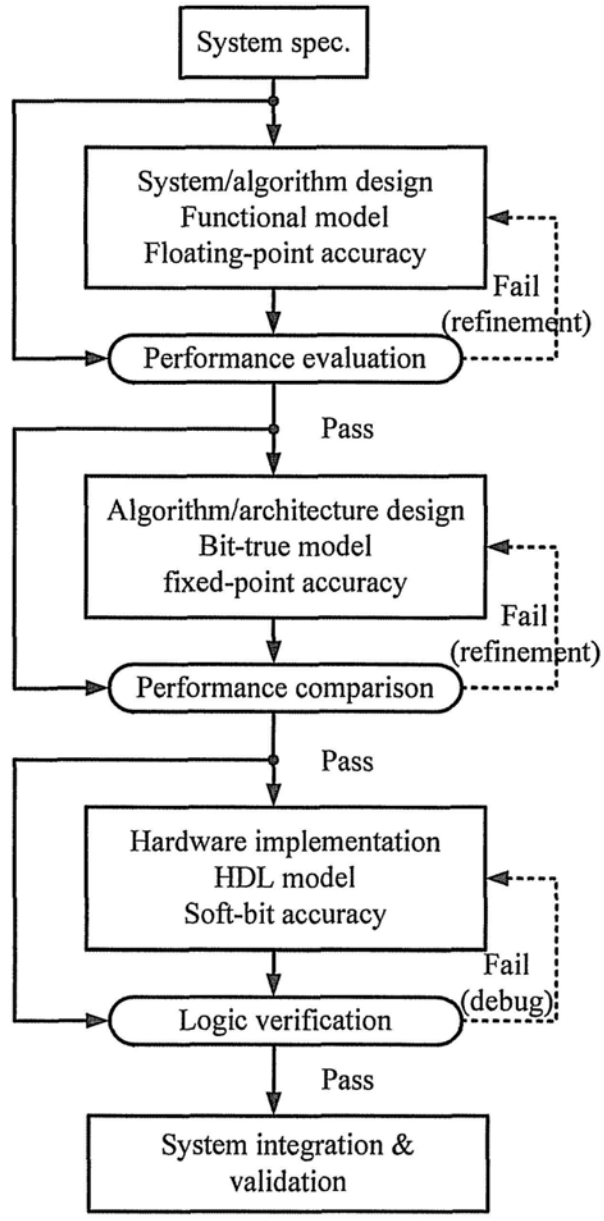


Figure 6.1 System design flow illustration

6.1 Fixed-Point Error Models

Digital baseband processing mainly involves arithmetic functions such as addition, subtraction, multiplication can lookup table (LUT). In the following, fixed-point data formats and error models of these functions will be introduced.

A fixed-point number is a real data type for a number that has a fixed number of digits after the decimal point. It can be specified in three parts, the total word length (WL), the word length of integer part (IWL) and the sign flag. A more elaborate

representation uses four parts that also specifies the quantization or rounding operation.

In a fixed-point system, the errors of truncation, clipping and propagation will be introduced during signal processing.

- Truncation error

Truncation error is introduced when the fraction part of a fixed-point number is truncated. Generally speaking, truncation is inevitable since the operations like multiplications or filtering increase the word length of the output signal. To prevent the word length from growing indefinitely, several LSBs in the fraction part can be discarded.

- Clipping error

Clipping error happens when the output of an arithmetic operation exceeds the range of the fixe-point representation. With proper range estimation and sufficient IWL setting, this kind of error can be avoided. However, in some cases, reserving sufficient IWL may lead to low hardware efficiency. Therefore, it is recommended to adopt clipping protection to avoid overflow signals propagation and crashing the whole system design.

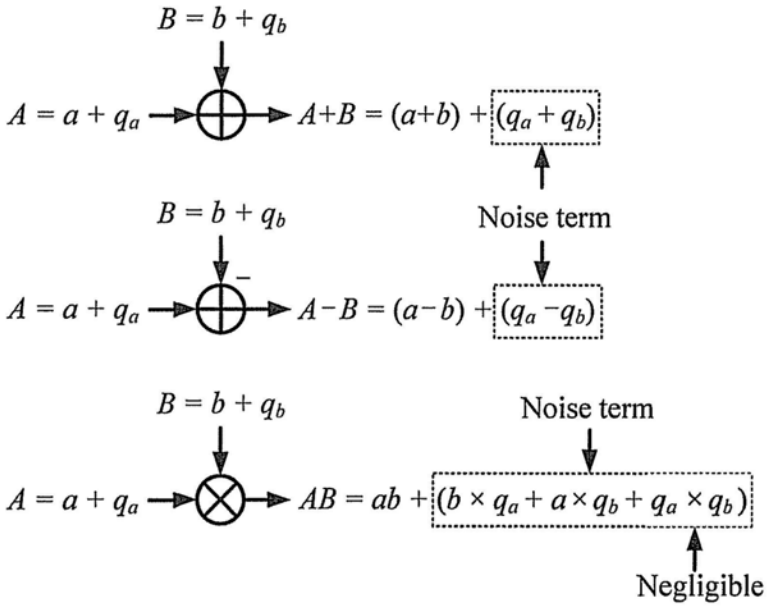


Figure 6.2 Propagation error model for a fixed-point adder, subtractor and multiplier

- Propagation error

Error terms generated in arithmetic operations will propagate along paths of the design. Fig. 6.2 illustrates the propagation error generated by a fixed-point adder, subtractor and multiplier. A and B denote the two fixed-point variables; a and b denote the original floating-point signals; q_a and q_b denote the truncation errors generated by earlier arithmetic operations. Obviously, the effect of propagation error in multiplier is more serious than other operations.

6.2 Implementation Results

For MB-OFDM UWB system, fixed-point implementation will lead to high complexity and high power consumption. Hence, the word length of each block in the receiver should be optimized, which will not only reduce the hardware complexity, but also guarantee acceptable system performance. According to the criteria of the system design, SNR loss due to quantization error should be 1~2 dB at BER of 10^{-5}. After tradeoff between complexity and performance, the optimum word length of the key blocks can be derived, as listed in Table 6.1.

Table 6.1 Signal word length of major blocks

Signal	WL (bit)	Signal	WL (bit)
ADC output	5	MTM NCO I/O	12/12
PD CC output	8	FFT twiddle factor	10
PD AC output	21	FFT reorder I/O	12/12
CFO I/O	5/6	Channel est. I/O	12/12
MTM arc tan T/O	15/15	Phase tracking I/O	12/12

The downlink baseband receiver for MB-OFDM UWB was integrated and implemented in 0.13-μm one-poly and eight-metal CMOS process. The estimated gate count after being synthesized by high-speed library is 404 K. Seven SRAM modules (17 K bits) and 5 K register files are used. To achieve the 528 MSamples/s throughput at 132 MHz clock frequency, the parallel factor of the proposed system is four.

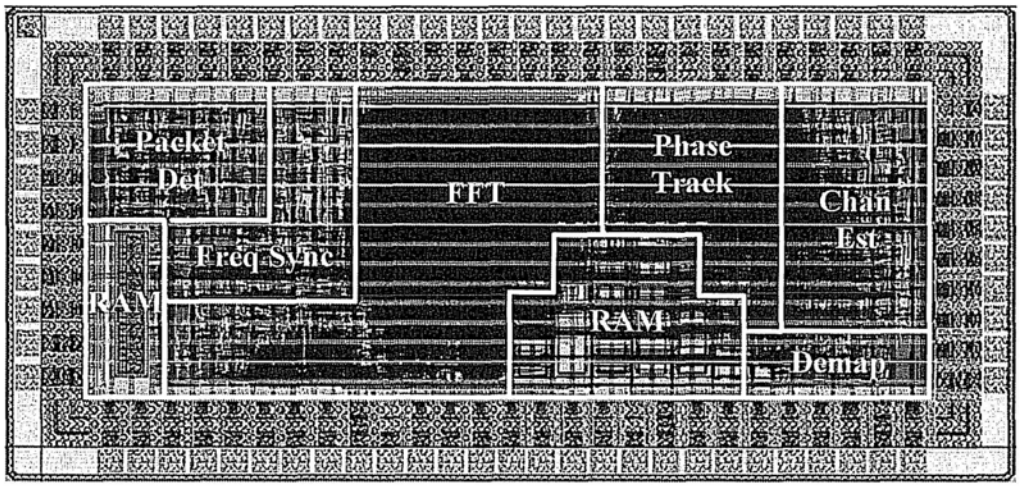


Figure 6.3 Layout of the proposed downlink baseband MB-OFDM receiver

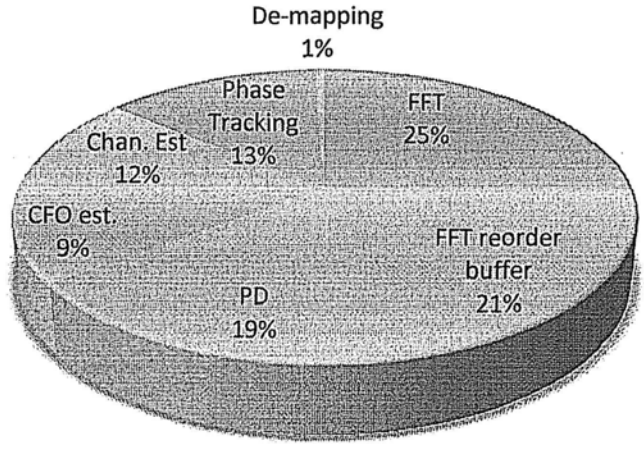


Figure 6.4 Power consumption distribution of the MB-OFDM UWB baseband receiver

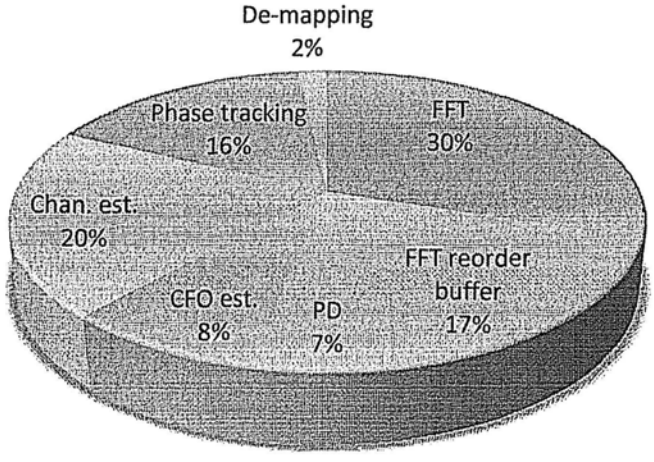


Figure 6.5 Area distribution of the MB-OFDM UWB baseband receiver

Fig. 6.3 shows the layout of the presented MB-OFDM UWB baseband receiver. The core area is 2.66 mm × 0.94 mm. The estimated power consumption after placement and routing is 170 mW at 132 MHz frequency, which is equivalent to 88 pJ/b energy efficiency at 480 Mbps data rate. Table 6.2 lists the summary of the post-layout implementation results.

Table 6.2 Summary of the post-layout implementation results

Technology	0.13 μm 1p8m CMOS
Core area	2.66 mm × 0.96 mm
Clock frequency	132 MHz
Max operation frequency	500 MHz
Logic gate count	404 K
SRAM size	17 K
Register file size	5.016 K
Max data rate	480 Mbps
Power consumption	170 mW @ 132 MHz
Energy per bit	88 pJ/b

In the worst case, the proposed design can work normally at 170 MHz frequency. While in the best case, the system frequency can achieve as high as 500 MHz. The latency from the packet arriving to it being detected is 2.33 μs. CFO estimation and compensation requires 3.96 μs. And the phase tracking delay is 0.27 μs.

Fig. 6.4 and Fig. 6.5 show the power and area distribution of each block in the MB-OFDM UWB system respectively. FFT block and FFT output reorder buffer are the most complicated components, which consumes 46% power and covers 47% area of the downlink baseband receiver system. The synchronization blocks, including packet detector, CFO estimator and phase tracking block, consume 41% power and occupy 31% area. The channel estimator, which includes large size of memory units, consumes 12% power and covers 20% area.

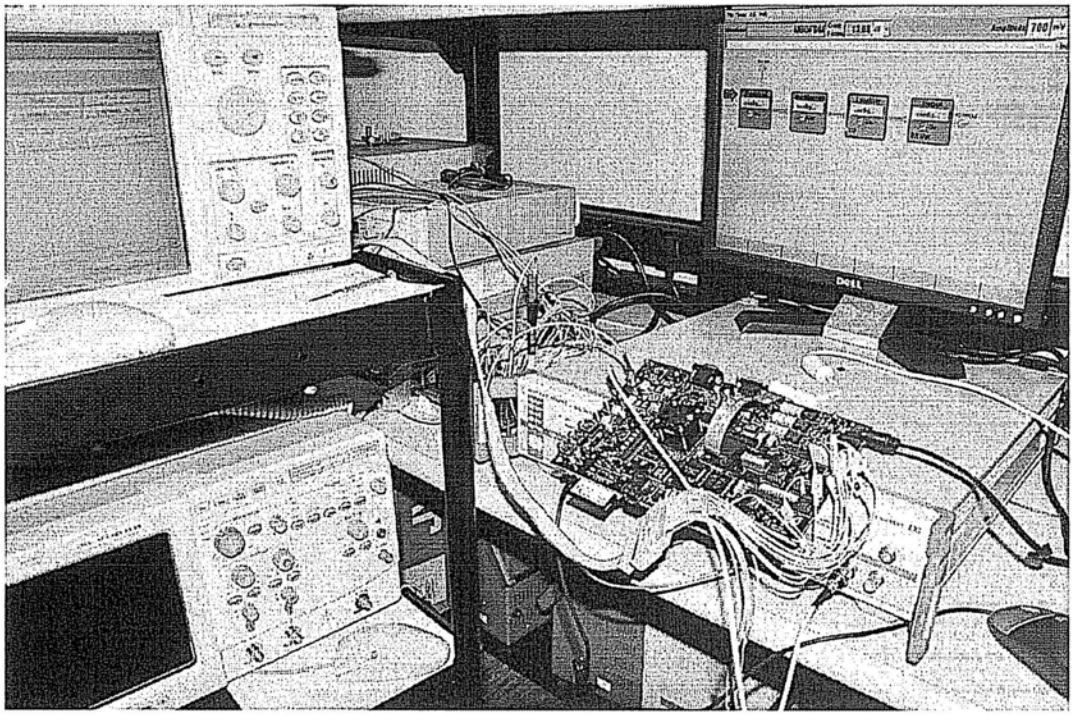
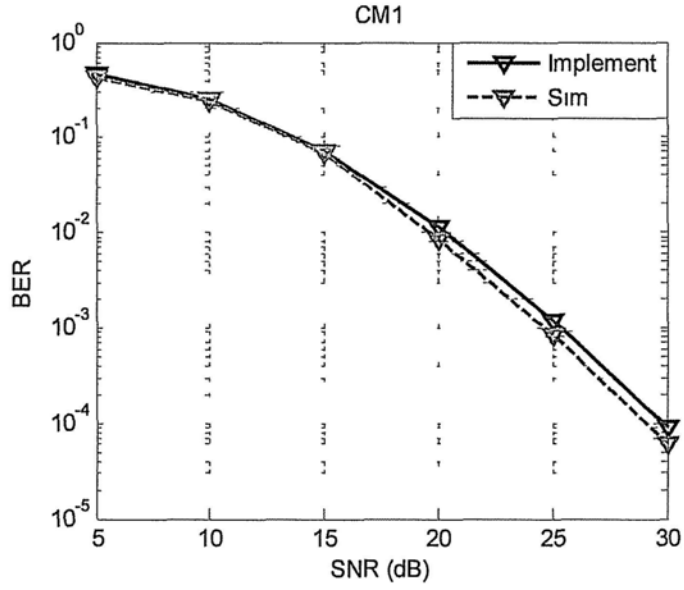
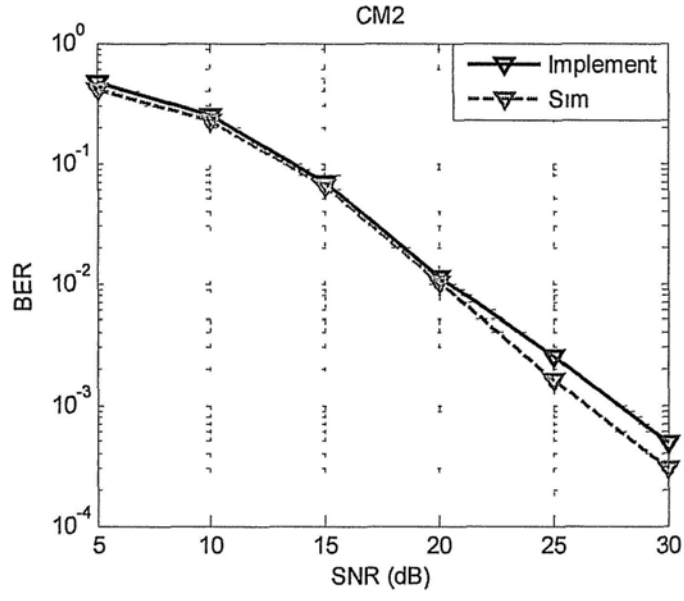


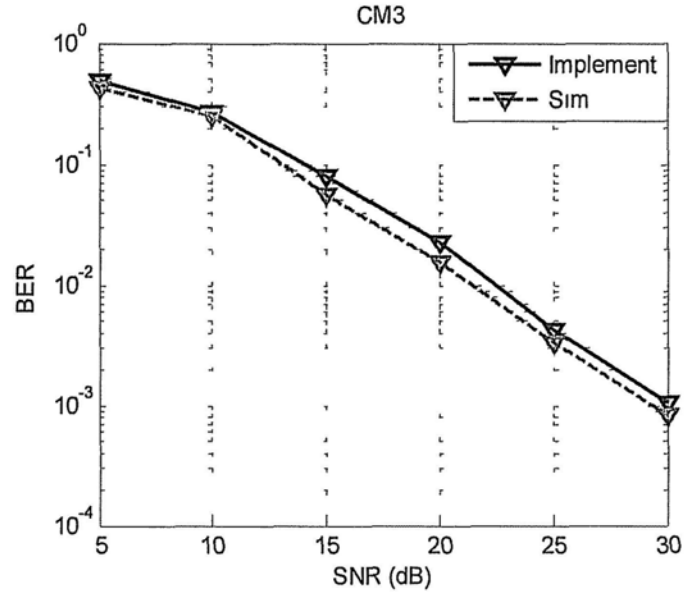
Figure 6.6 FPGA-based fast prototyping demonstration

We use Altera Stratix II EP2S60 DSP development board to emulate the uncoded BER of the proposed baseband receiver system, as shown in Fig. 6.6. The I/Q input signals are generated by R&S AFQ 100B generator and passed through two ADCs on development board. Output signals are measured by Logic Analyzer through GPIO.

Fig. 6.7 illustrates the uncoded BER performance comparison of the proposed baseband receiver system between the fixed-point simulation results by MatLab and implementation results by FPGA in the channel mode of CM1~CM4. Timing offset is set to 1000 samples and the normalized CFO is set to 0.04. Compared with the fixed-point simulation results, there is a slight SNR degradation of the implementation results by FPGA due to quantization noise. Because of the memory depth limitation, the implementation results can only be above to 10^{-5}. Fig 6.7 indicates that the proposed design meets the design criteria of 1~2 dB SNR degradation at 10^{-5} BER.


CM1
Implement
Sim
BER
SNR (dB)


CM2
Implement
Sim
BER
SNR (dB)


CM3
Implement
Sim
BER
SNR (dB)

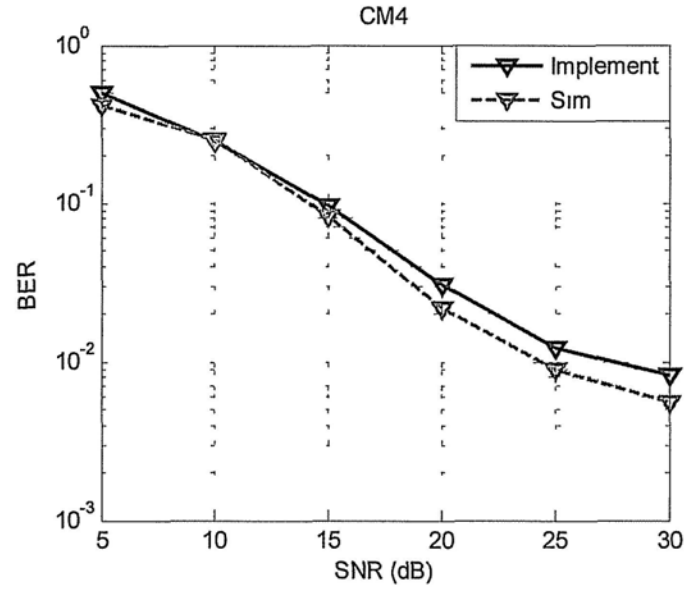


Figure 6.7 Function simulation and FPGA implementation BER performance of the proposed design

Fig. 6.8 and Fig 6.9 show the reduction effects of hardware and power consumption by employing low-complexity optimization techniques in each block. Compared with directly using the traditional CC in 128 taps, the techniques of sign-bit CC and decomposition CC taps in 16 taps cascaded with 8 taps in matched filter saves 8.5% hardware complexity and 17.1% power. Although the cascaded auto-correlator in packet detector increases 1.5% hardware complexity and 4.3% power, it improves the system performance significantly as shown in Chapter 3. In CFO corrector, shortening SWL can save 6.7% hardware and 9.2% power consumption. And employing MTM to implement arctangent and sin/cos function instead of CORDIC algorithm reduces 8.6% complexity and 12.4% power. The proposed address encoding method in FFT output reorder block has the greatest contribution to low-cost and low-power design. It saves 15.3% hardware cost and 23.7% power consumption. Finally, the proposed novel phase tracking approach reduces 7.5% area and 8.2% power compared with the conventional approach with arctangent and NCO. Totally, the proposed downlink baseband receiver for MB-OFDM UWB occupies only 54.9% area and consumes 33.7% power of the traditional designed system without any optimization.

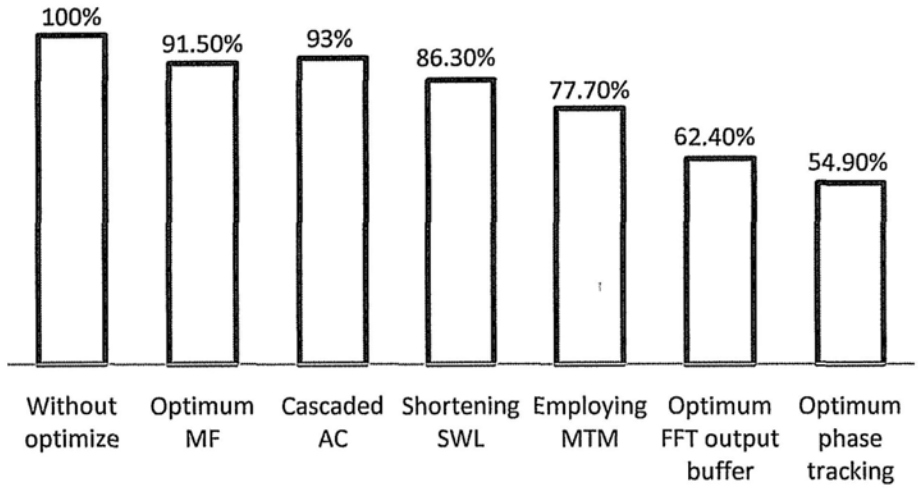


Figure 6.8 Summary of the hardware reduction for the proposed baseband receiver

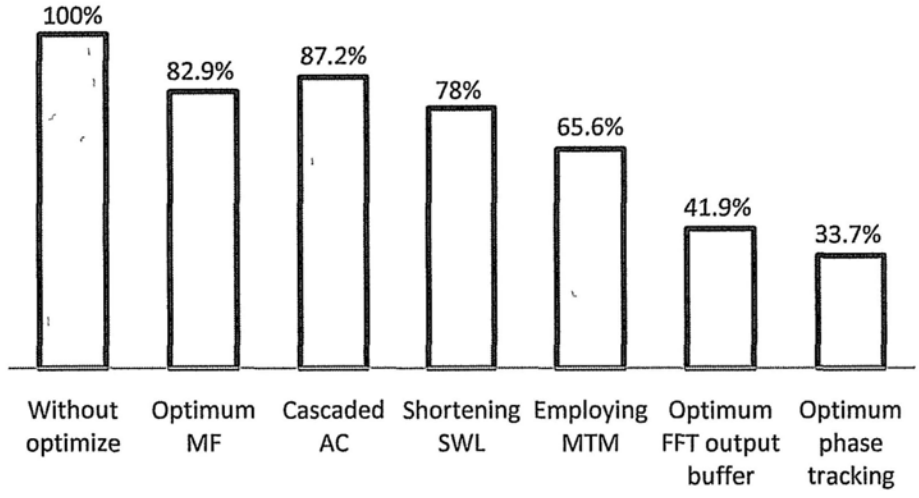


Figure 6.9 Summary of the power reduction for the proposed baseband receiver

The implementation results after placement and routing are listed in Table 6.3 and compared to the state-of-the-art baseband receivers. There is rarely circuit design of MB-OFDM UWB baseband receiver, so a fair comparison is hard to conduct. The coded OFDM (COFDM)-based baseband solutions for 528 MHz UWB in [84] is the most similar work. However, the transceiver design includes the transmitter and LDPC, so the area and power of the design in [84] cannot be directly compared with the

proposed design. The multicarrier-CDMA (MC-CDMA) downlink baseband receiver in [41] has a very low power consumption of 9.9 mW at 5.76 MHz frequency. Taking the clock frequency and throughput into account, we use the parameter energy per bit to evaluate the energy efficiency. From Table 6.3, it is easy to find although the power consumption of the design in [41] is very low, its energy is not as efficient as our work. In [42], an inner receiver for OFDM-based WLAN system is proposed. Its energy efficiency is much higher than our design. In order to compare the area fairly in different technologies, we scale the area of different designs to 0.13 μm process. Due to the low-complexity scheme in the proposed design, it has much smaller area compared with [41] and [42].

Table 6.3 Post-layout implementation results and comparison to the measurement results of other works

	This work	[41]	[42]
Technique	MB-OFDM UWB	MC-CDMA	OFDM
Technology	0.13 μm 1p8m CMOS	0.18 μm 1p6m CMOS	0.25 μm 1p5m CMOS
Core power supply	1.2 V	1.1 V	2.5 V
Clock frequency	132 MHz	5.76 MHz	20/40/80 MHz
Parallel factor	4	1	1
Data rate	480 Mbps	21.7 Mbps	54 Mbps
Power consumption	170 mW	9.9 mW	183 mW
Energy per bit	88 pJ/b	460 pJ/b	3.4 nJ/b
Area Scale to 0.13 μm	2.50 mm^2	6.76 mm^2 4.82 mm^2	23.60 mm^2 12.27 mm^2

Chapter 7 Conclusions

This paper presents algorithm and architecture design for MB-OFDM UWB baseband system.

The contributions on synchronization methods are illustrated in Chapter 3. DT judgment strategy in packet detector improves the detection probability significantly compared with conventional CC algorithm with just increasing 1.5% hardware and 4.3% power consumption. Shortening the SWL for CFO estimation is an effective approach for low complexity with slightly performance degradation. However, it can be overcome by sum average over different symbols. Just average on three symbols located at three subbands, the SWL of 64 can outperforms the traditional estimation method with SWL of 128. The provision of a novel phase tracking approach reduces 7.5% hardware cost and 8.2% power consumption of the baseband system and even has better performance than the traditional approach. This phase tracking method is designed without arctangent block and NCO, just adders and shifters are included.

The contributions on architecture design include architecture optimization on matched filter, employing MTM in coarse frequency synchronization instead of CORDIC and encoding address of FFT output reorder buffer in a new way (Chapter 4). The optimum matched filter architecture saves 8.5% area and 17.1% power. Using MTM in CFO correction saves 8.6% hardware and 12.4% power. And the design of FFT output reorder buffer reduces 15.3% area and 23.7% power.

The proposed baseband receiver is implemented with UMC 0.13 μm process. The power consumption is 170 mW at 132 MHz frequency by post-layout power estimation. Due to the optimization on algorithm and architecture, the power consumption saves 66.3%. The equivalent energy efficiency of 88 pJ/b at 480 Mbps data rate. The core area of the layout is 2.5 mm^2, which occupies only 54.9% of the area without any optimization. The implementation results indicate the proposed baseband receiver for MB-OFDM UWB is robust, low complexity and energy

efficient for practical application.

The future work is summarized as follows.

- Complete the whole MB-OFDM UWB baseband receiver system, including the blocks of de-interleaver, Viterbi decoder and descrambler. And the coded BER performance can be measured.
- Integrate with the RF front-end and transmitter, to build a MB-OFDM UWB transceiver system. In this case, the received signal will be generated by the transmitter rather than arbitrary waveform generator or software and there will be the real environment.
- I/Q mismatch issue is not taken into account in the proposed system because it will not degrade the system performance significantly if the mismatch is not serious enough. In future work, this issue can be considered in case the I/Q mismatch is severe.
- The proposed synchronization methods can be further applied in other similar and compatible systems, such as CDMA-based receiver and OFDM-based receiver.

References

[1] A. Batra et al, *Multi-band OFDM Physical Layer Proposal for IEEE 802.15 Task Group 3a.* IEEE P802.15-03/268r2, 2003.
(Available online at http://grouper.ieee.org/groups/802/15)

[2] J. H. Reed, *An Introduction to Ultra Wideband Communication Systems*, Prentice Hall, 2005.

[3] R. Gharpure and P. Kinget, *Ultra Wideband: Circuits, Transceivers, and Systems*, Springer, 2008.

[4] S. Wood and R. Aiello, *Essential of UWB (The Cambridge Essentials Series)*, Cambridge University Press, 2008.

[5] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE trans. Commun.*, vol. 45, pp. 1613-1621, Dec. 1997.

[6] H. Minn, V. K. Bhargava and K. B. Letaief, "A robust timing and frequency synchronization for OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 2, pp. 822-839, Jul. 2003.

[7] L. Hazy and M. El-Tanany, "Synchronization of OFDM systems over frequency selective fading channels," *in Proc. IEEE Vehic. Tech. Conf.*, Phoenix, AZ, May 1997, pp. 2094-2098.

[8] F. Tufvesson, O. Edfors and M. Faulkner, "Time and frequency synchronization for OFDM using PN-sequency preambles," *in Proc. IEEE Vehic. Tech. Conf.*, Sep. 1999, vol. 4, pp. 2203-2207.

[9] T. Pollet, M. van Bladel, M. Moeneclaey, "BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise," *IEEE Trans. Comm.*, vol. 43, issue 2, part 3, pp. 191-193, February, March, April 1995.

[10] M. Morelli and U. Mengali, "An improved frequency offset estimator for OFDM applications," *IEEE Comm. Letter*, vol. 3, pp. 75-77, Mar. 1999.

[11] M. H. Hsieh and C. H. Wei, "A low-complexity frame synchronization and frequency offset compensation scheme for OFDM Systems over fading channels," *IEEE Trans. Vehic. Tech.*, Vol. 48, No. 5, pp. 1596-1609, Sep. 1999.

[12] K. Shi, E. Serpedin and P. Ciblat, "Decision-directed fine synchronization in

OFDM systems," *IEEE Trans. Commun.*, vol. 53, no. 3, pp. 408-412, Mar. 2005.

[13] L. Kuang, Z. Ni and J. Lu et al., "A time-frequency decision-feedback loop for carrier frequency offset tracking in OFDM systems," *IEEE Trans. Commun.*, vol. 4, no. 2, pp. 367-373. Mar. 2005.

[14] Y-H You, J-B Kim and H-K Song, "Pilot-assisted fine frequency synchronization for OFDM-based DVB receivers," *IEEE Trans. Broadcasting*, vol. 55, no. 3, pp. 674-678, Sep. 2009.

[15] Z. Wang, Y. Xin and G. Mathew et al., "Efficient phase-error suppression for multiband OFDM-based UWB systems," *IEEE Trans. Vehic. Tech.*, vol. 59, no. 2, pp. 766-778, Feb. 2010.

[16] A. M. Karim and M. Othman, "Improved fine CFO synchronization for MB-OFDM UWB," *IEEE Commun. Letters*, vol. 14, no. 4, pp. 351-353, Apr. 2010.

[17] Z-Z Ye, C-J Duan and P. V. Orlik et al., "A synchronization design for UWB-Based wireless multimedia systems," *IEEE Trans. Broadcast.*, vol. 56, no. 2, pp. 211-225, Jun. 2010.

[18] Y-H Li, H. Minn and T. Jacobs et al., "Freuqency offset estimation for MB-OFDM-based UWB systems," *IEEE Trans. Commun.*, vol. 56, no. 6, pp. 968-979, Jun. 2008.

[19] J-T Lai, A-Y Wu and W-C Chen, "A systematic design approach to the band-tracking packet detector in OFDM-based ultra wideband systems," *IEEE Trans. Vehic. Tech.*, vol. 56, no. 6, pp. 3791-3806, Nov. 2007.

[20] C. R. Berger, S-L Zhou and Z. Tian et al., "Performance analysis on an MAP fine timing algorithm in UWB multiband OFDM," *IEEE Trans. Commun.*, vol. 56, no. 10, pp. 1606-1611, Oct. 2008.

[21] X-Y Wang, Z-C Zhang and C-Y Chen, "A robust timing synchronization scheme for MB-OFDM UWB system," *in proc. IEEE ICSPS 2010*, pp. 529-532.

[22] M. Verhelst and W. Dehaene, "A flexible, ultra-low-energy 35 pJ/pulse digital back-end for a QAC IR-UWB receiver," *IEEE J. Solid-State Circuits*, vol. 43, pp. 1677-1687, Jul. 2008.

[23] D. Barras, R. Meyer-Piening and G. von Bueren et al., "A low-power baseband ASIC for an energy-collection IR-UWB receiver," *IEEE J. Solid-State Circuits*, vol. 44, no. 6, pp. 1721-1733, Jun. 2009.

[24] N. V. Helleputte, M. Verhelst and W. Dehaene et al., "A reconfigurable, 130 nm

CMOS 108 pJ/pulse, fully integrated IR-UWB receiver for communication and precise ranging," *IEEE J. Solid-State Circuits*, vol. 45, no. 1, pp. 69-83, Jan. 2010.

[25] P. P. Mercier, M. Bhardwaj and D. C. Daly et al., "A low-voltage energy sampling IR-UWB digital baseband employing quadratic correlation," *IEEE J. Solid-State Circuits*, vol. 45, no. 6, Jun. 2010.

[26] C-H Yang, K-H Chen and T-Z Chiueh, "A 1.2V 6.7mW impulse-radio UWB baseband transceiver," *in proc. IEEE ISSCC 2005*, pp. 442-443, 608.

[27] M. Verhelst, N. van Helleputte and G. Gielen et al., "A reconfigurable, 0.13um CMOS 110pJ/pulse, fully integrated IR-UWB receiver for communication and sub-cm ranging," *in proc. IEEE ISSCC 2009*, pp. 250-251.

[28] P. P. Mercier, M. Bhardwaj and D. C. Daly et al., "A 0.55V 16Mb/s 1.6mW non-coherent IR-UWB digital baseband with ±1ns synchronization accuracy," *in proc. IEEE ISSCC 2009*, pp. 252-253.

[29] W. P. Siriwongpairat and K. J. Ray Liu, *Ultra-Wideband Communications Systems*: *Multiband OFDM Approach*, Wiley, 2007.

[30] H. Arslan, Z-N Chen and M-G Di Benedetto, *Ultra Wideband Wireless Communication*, Wiley, 2006.

[31] T-D Chiueh and P-Y Tsai, *OFDM Baseband Receiver Design for Wireless Communications*, Wiley, 2007.

[32] M. Speth, F. Classen and H. Meyr, "Frame synchronization of OFDM systems in frequency selective fading channels," in *Proc. IEEE VTC,* Phoenix Arizona, USA, 1997, pp. 1807-1811.

[33] J. J. van de Beek, M. Sandell and P. O. Borjesson, "ML estimation of time and frequency offset in OFDM systems," *IEEE Trans. Signal Processing,* vol. 45, no. 7, pp. 1800-1805, Jul. 1997.

[34] A. J. Coulson, "Maximum likelihood synchronization for OFDM using a pilot symbol: algorithms," *IEEE J. Select. Areas in Comm.,* vol. 19, no. 12, pp. 2486-2494, Dec. 2001.

[35] A. Fort, J-W Weijers, V. Derudder, W. Eberle and A. Bourdoux, "A performance and complexity comparison of auto-correlation and cross-correlation for OFDM burst synchronization," in *Proc. IEEE ICASSP,* Jun. 2003, pp. 341-344.

[36] W. Fan, C-S Choy and K-N Leung, "Robust and low complexity packet detector design for MB-OFDM UWB," in *Proc. IEEE ISCAS,* Taipei, 2009, pp. 693-696.

[37] S. M. Kay, *Fundamentals of Statistical Signal Processing II: Detection Theory*, Prentice Hall, 1998.

[38] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Trans. Comm.*, vol. 42, no. 10, Oct. 1994, pp. 2908-2914.

[39] W. Fan and C-S Choy, "Power efficient and high speed frequency synchronizer design for MB-OFDM UWB," in *Proc. IEEE ICUWB*, Vancouver, CA, 2009, pp. 669-673.

[40] Y. H. Hu, "The quantization effects of the CORDIC algorithm," *IEEE Trans. Signal Processing,* vol. 40, no. 4, pp. 834-844, Apr. 1992.

[41] P-Y Tsai and T-D Chiueh, "A low-power multicarrier-CDMA downlink baseband receiver for future cellular communication systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 10, pp. 2229-2239, Oct 2007.

[42] A. Troya, K. Maharatna, and M. Krstic et al., "Low-power VLSI implementation of the inner receiver for OFDM-based WLAN systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 2, pp. 672-686, Mar. 2008.

[43] D. D. Caro, N. Petra and A. G. M. Steollo, "A 380 MHz direct digital synthesizer/mixer with hybrid CORDIC architecture in 0.25 um CMOS," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 151-160, Jan. 2007.

[44] D. D. Caro, N. Petra and A. G. M. Steollo, "Digital synthesizer/mixer with hybrid CORDIC-multiplier architecture: error analysis and optimization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 2, pp. 364-373, Feb. 2009.

[45] D. D. Caro, E. Napoli and A. G. M. Steollo, "Direct digital frequency synthesizer with polynomial hyperfolding technique," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 51, no. 7, pp. 337-344, Jul. 2004.

[46] D. D. Caro and A. G. M. Steollo, "High-performance direct digital frequency synthesizer in 0.25 um CMOS using dual-slope approximation," *IEEE J. Solid-State Circuits*, vol. 40, no. 11, pp. 2220-2227, Nov. 2005.

[47] J. M. P. Langlois and D. Al-Khalili, "Novel approach to the design of direct digital frequency synthesizer based on linear interpolation," *IEEE Trans. Circuits Syst. II, Analog and Digital Signal Processing*, vol. 50, no. 9, pp. 567-578, Sep.

2003.

[48] D. D. Caro and A. G. M. Steollo, "High-performance direct digital frequency synthesizer using piecewise-polynomial approximation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 2, pp. 324-337, Feb. 2005.

[49] F. de Dinechin and A. Tisserand, "Multipartite table methods," *IEEE Trans. Computers*, vol. 54, no. 3, pp. 319-330, Mar. 2005.

[50] A. G. M. Strollo, D. D. Caro and N. Petra, "A 630 MHz, 76 mW direct digital frequency synthesizer using enhanced ROM compression technique," *IEEE J. Solid-State Circuits*, vol. 42, no. 2, pp. 350-360, Feb. 2007.

[51] D. De Caro, N. Petra and A. G. M. Strollo, "Reducing lookup-table size in direct digital frequency synthesizers using optimized multipartite table method," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 7, pp. 2116-2127, Aug. 2008.

[52] S. A. Fechtel, "OFDM Carrier and Sampling Frequency Synchronization and its Performance on Stationary and Mobile Channels," *IEEE Trans. Consumer Electronics*, Vol. 46, No. 3, pp. 438–441, Aug. 2000.

[53] D. K. Kim, S. H. Do, H. B. Cho, H. J. Choi, K. B. Kim, "A New Joint Algorithm of Symbol Timing Recovery and Sampling Clock Adjustment for OFDM Systems," *IEEE Trans. Consumer Electronics*, Vol. 44, No. 3, pp. 1142–1149, Aug. 1998.

[54] M. Speth, S. Fechtel and G. Fock et al., "Optimum receiver design for OFDM-based broadband transmission-part II: A case study," *IEEE Trans. Commun.*, vol. 49, no. 4, pp. 571-578, Apr. 2001.

[55] S. Y. Liu and J. W. Chong, "A study of joint tracking algorithms of carrier frequency offset and sampling clock offset for OFDM-based WLANs," *in Proc. IEEE Int. Conf. Commun., Circuits and Sys. and West Sino Expositions*, Chengdu, China, Jun. 2002, pp. 109-133."

[56] P. Y. Tsai, H. Y. kang and T. D. Chiueh, "Joint weighted least squares estimation of carrier frequency offset and timing offset for OFDM systems over multipath fading channel," *IEEE Trans. Vehicular technol.*, vol. 54, no. 1, pp. 211-224, Jan. 2005.

[57] Y-H. You, J-B. Kim and H-K. Song, "Pilot-assisted fine frequency synchronization for OFDM-based DVB receivers," *IEEE Trans. Broadcasting*, vol. 55, no. 3, pp. 674-678, Jul. 2009.

[58] Z-J. Wang, Y. Xin, G. Mathew and X-D. Wang, "Efficient phase-error suppression for multiband OFDM-based UWB systems," *IEEE Trans. Vehicular Tech.*, vol. 59, no. 2, pp. 766-778, Feb. 2010.

[59] A. Troya, K. Maharatna and M. Krstic et al, "Efficient inner receiver design for OFDM-based WLAN systems: algorithm and architecture," *IEEE Trans. Wireless Comm.,* vol. 6, no. 4, pp. 1374-1385, Apr. 2007.

[60] W. Fan and C-S Choy, "Efficient and low complexity phase tracking method for MB-OFDM UWB receiver," in *Proc. IEEE MWSCAS*, Seattle, USA, 2010, pp. 221-224.

[61] S-S He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation," in *Proc. IEEE ISSSE 98,* Pisa, Italy, pp. 257-262.

[62] L. R. Rabiner and B.Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall, Inc., 1975.

[63] E. H. Wold and A. M. Despain, "Pipeline and parallel-pipeline FFT processors for VLSI implementation," *IEEE Trans Comput.*, vol. C-33, no. 5, pp. 414-426, May 1984.

[64] A. M. Despain, "Fourier transform computer using CORDIC iterations," *IEEE Trans. Comput.*, vol. C-23, no. 10, pp. 993-1001, Oct. 1974.

[65] E. E. Swartzlander, W. K. W. Young and S. J. Joseph, "A radix 4 delay commutator for fast Fourier transform processor implementation," *IEEE J. Solid-State Circuits*, vol. SC-19, no. 5, pp. 702-709, Oct. 1984.

[66] E. E. Swartzlander, V. K. Jain and H. Hikawa, "A radix 8 wafer scale FFT processor," *J. VLSI Signal Processing*, vol. 4, no. 2-3, pp. 165-176, May 1992.

[67] G. Bi and E. V. Jones, "A pipelined FFT processor for word sequential data," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 12, pp. 1982-1985, Dec. 1989.

[68] Y-W Lin, H-Y Liu and C-Y Lee, "A 1-GS/s FFT/IFFT Processor for UWB Applications," *IEEE J. Solid-State Circuits*, Vol. 40, No. 8, pp. 1726-1735, Aug. 2005.

[69] Y-W Lin and C-Y Lee, "Design of an FFT/IFFT Processor for MIMO OFDM Systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, Vol. 54, No. 4, pp. 807-815, Apr. 2007.

[70] J. Lee, H. Lee and S. Cho et al., "A High-Speed, Low-Complexity Radix-2^4 FFT Processor for MB-OFDM UWB Systems," *IEEE Int. Sym. Circuits and Systems*, pp. 4719-4722, May 2006.

[71] S-I Cho, K-M Kang and S-S Choi, "Implementation of 128-Point Fast Fourier Transform Processor for UWB Systems," *Int. Wireless Commun. and Mobile Computing conf.*, pp. 210-213, Aug. 2008.

[72] M. K. Ozdemir, "Channel estimation for wireless OFDM systems," *IEEE Commun. Surveys & Tutorials*, vol. 9, no. 2, pp. 18-48, 2nd Quarter, 2007.

[73] J. J. van de Beek, O. Edfors and M. Mersereau et al., "On channel estimation in OFDM systems," *in Proc. IEEE Vehic. Tech. Conf.*, Jul. 1995, pp. 815-819.

[74] O. Edfors, M. Sandell and J. J. van de Beek et al., "OFDM channel estimation by singular value decomposition," IEEE Trans. Commun., vol. 46, no. 7, pp. 931-939, Jul. 1998.

[75] S. Lasaulce, N. Sellami and Y. Yi et al., "Performance study of least-squares channel estimation based on hard decisions," *in Proc. IEEE Vehic. Tech. Conf.*, vol. 4, Jul. 2003, pp. 2570-2574.

[76] M. Morelli and U. Mengali, "A comparison of pilot-aided channel estimation methods for OFDM systems," *IEEE Trans. Signal Processing*, vol. 49, no. 12, pp. 3065-3073, Dec. 2001.

[77] E. G. Larsson, G-Q Liu and J. Li et al., "Joint symbol timing and channel estimation for OFDM based WLANs," *IEEE Commun. Letters*, vol. 5, no. 8, pp. 325-327, Aug. 2001.

[78] E. Ertin, U. Mitra and S. Siwamogsatham, "Maximum-likelihood-based multipath channel estimation for code-division multiple-access systems," *IEEE Trans. Commun.*, vol. 49, no. 2, pp. 290-302, Feb. 2001.

[79] V. Srivastava, C-K Ho and H-W Fung et al., "Robust MMSE channel estimation in OFDM systems with practical timing synchronization," *IEEE Wireless Commun. Networking Conf.*, pp. 711-716, Mar. 2004.

[80] C. R. N. Athaudage and A. D. S. Jayalath, "Enhanced MMSE channel estimation using timing error statistics for wireless OFDM systems," *IEEE Trans. Broadcasting*, vol. 50, issue. 4, pp. 369-376, Dec. 2004.

[81] K. K. *Parhi, VLSI Digital Signal Processing Systems: Design and Implementation*, New York: John Wiely & Son, 1999.

[82] W. Sung and K-I Kum, "Simulation-based word-length optimization method for fixed-point digital signal processing systems," *IEEE Trans. Signal Processing*, vol. 43, no. 12, pp. 3087-3090, Dec. 1995.

[83] G. A. Constantinides, Y. K. Cheung and Y. Luk, "Wordlength optimization for linear digital signal processing," vol. 22, no.10, pp. 1432-1442, Oct. 2003.

[84] H-Y Liu et al., "A 480 Mb/s LDPC-COFDM-based UWB baseband transceiver," *in Dig. Tech. Papers, IEEE ISSCC*, col. 1, pp. 444-445, 609, Feb. 2005.