An Equalization Technique for High Rate OFDM Systems

A Thesis Submitted to the College of

Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in the Department of Electrical Engineering

University of Saskatchewan

Saskatoon

By

Naihua Yuan

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a degree of Master of Science from the University of Saskatchewan, the author agrees that the libraries of this University may make it freely available for inspection. The author further agrees that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised this thesis work or, in their absence, by the Head of the Department or the Dean of the College of the Graduate Studies and Research. It is understood that any copying or publication or use of this thesis or parts for financial gain shall not be allowed without the author's written permission. It is also understood that recognition shall be given to the author and to the University of Saskatchewan in any scholarly use which may be made of any material in this thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Electrical Engineering,
57 Campus Drive,
University of Saskatchewan,
Saskatoon, Saskatchewan,
Canada S7N 5A9

ABSTRACT

In a typical orthogonal frequency division multiplexing (OFDM) broadband wireless communication system, a guard interval using cyclic prefix is inserted to avoid the inter-symbol interference and the inter-carrier interference. This guard interval is required to be at least equal to, or longer than the maximum channel delay spread. This method is very simple, but it reduces the transmission efficiency. This efficiency is very low in the communication systems, which inhibit a long channel delay spread with a small number of sub-carriers such as the IEEE 802.11a wireless LAN (WLAN).

To increase the transmission efficiency, it is usual that a time domain equalizer (TEQ) is included in an OFDM system to shorten the effective channel impulse response within the guard interval. There are many TEQ algorithms developed for the low rate OFDM applications such as asymmetrical digital subscriber line (ADSL). The drawback of these algorithms is a high computational load. Most of the popular TEQ algorithms are not suitable for the IEEE 802.11a system, a high data rate wireless LAN based on the OFDM technique. In this thesis, a TEQ algorithm based on the minimum mean square error criterion is investigated for the high rate IEEE 802.11a system. This algorithm has a comparatively reduced computational complexity for practical use in the high data rate OFDM systems. In forming the model to design the TEQ, a reduced convolution matrix is exploited to lower the computational complexity. Mathematical analysis and simulation results are provided to show the validity and the advantages of the algorithm. In particular, it is shown that a high performance gain at a data rate of 54Mbps can be obtained with a moderate order of TEQ finite impulse response (FIR) filter. The algorithm is implemented in a field programmable gate array (FPGA). The characteristics and regularities between the elements in matrices are further exploited to

reduce the hardware complexity in the matrix multiplication implementation. The optimum TEQ coefficients can be found in less than 4μs for the 7^{th} order of the TEQ FIR filter. This time is the interval of an OFDM symbol in the IEEE 802.11a system. To compensate for the effective channel impulse response, a function block of 64-point radix-4 pipeline fast Fourier transform is implemented in FPGA to perform zero forcing equalization in frequency domain. The offsets between the hardware implementations and the mathematical calculations are provided and analyzed. The system performance loss introduced by the hardware implementation is also tested. Hardware implementation output and simulation results verify that the chips function properly and satisfy the requirements of the system running at a data rate of 54 Mbps.

ACKNOWLEDGEMENTS

I am deeply grateful to my supervisors, Dr. A. Dinh and Dr. Ha H. Nguyen, for their guidance, support, and patience during my graduate program. They have been an invaluable source of knowledge and have certainly helped inspire many of the ideas expressed in this thesis.

I would like to thank the Telecommunication Research Laboratory (TRLab), Saskatoon for its funding support in this research. I also want to show my thanks for the funding from the Barberhold Chair of Information and Technology.

I would like to thank Mr. Jack Hanson, Mr. Trevor Hamm, staffs and students of TRLabs in Saskatoon, and all the other faculties, staffs for their support and help.

Also, I would like to thank the Department of Electrical Engineering and the College of Graduate Studies and Research, University of Saskatchewan, for providing support during this research.

At last, I am very appreciated my wife and my parents-in-law for their patience, endless support during my study here.

DEDICATION

To my lovely daughter Xinyi Yuan

To my father Peisong Yuan and mother Fengyin Zhou.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADSL	Asymmetrical Digital Subscriber Line
AGC	Automatic Gain Control
ARMA	Autoregressive Moving Average
AWGN	Additive White Guassian Noise
BPSK	Binary Phase Shift Keying
DAB	Digital Audio Broadcasting
DFT	Discrete Fourier Transform
DIF	Decimation in Frequency
DIT	Decimation in Time
DSP	Digital Signal Processing
DVB-T	Digital Video Terrestrial Broadcasting
FDM	Frequency Division Multiplexing
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FIR	Finite Impulse response
FPGA	Field Programmable Gate Array
HF	High Frequency

ICI	Inter-carrier Interference
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
IIR	Infinite Impulse Response
ISI	Inter-symbol Interference
ISM	Industrial, Scientific, and Medical
LE	Logical Element
LSB	Least Significant Bit
MAC	Media Access Control
MGSNR	Maximum Geometric Signal to Noise Ratio
MSB	Most Significant Bit
MSE	Mean Square Error
MMSE	Minimum Mean Square Error
MPDU	Media Access Control (MAC) Protocol Data Unit
MSSNR	Minimum Shortening Signal to Noise Ratio
OFDM	Orthogonal Frequency Division Multiplexing
PER	Packet Error Rate
PHY	Physical Layer
PLCP	Physical Layer Convergence Procedure
PMD	Physical Medium Dependent
PPDU	Physical Layer Convergence Procedure (PLCP) Protocol Data Unit
PSDU	Physical Sub-layer Service Data Unit

QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
R2MDC	Radix-2 Multi-path Delay Commutator
R4MDC	Radix-4 Multi-path Delay Commutator
R2SDF	Radix-2 Single-path Delay Feedback
R4SDC	Radix-4 Single-path Delay Commutator
ROM	Read Only Memory
SNR	Signal to Noise Ratio
SOPC	System on a Programmable Chip
SSNR	Shortening Signal to Noise Ratio
TIR	Target Impulse Response
TEQ	Time Domain Equalizer
U-NII	Unlicensed National Information Infrastructure
VLSI	Very Large Scale Integration
V-OFDM	Vector Orthogonal Frequency Division Multiplexing
ZFE	Zero Forcing Equalization
WCDMA	Wideband Code Division Multiple Access

Chapter 1 Introduction

In this chapter, the history of OFDM and its applications are introduced. After the discussion of the basic concept of OFDM, the motivation of the thesis is proposed.

1.1 Brief History of OFDM Technology

The rapid advances in multimedia applications involve more and more transmissions of graphical data, video and audio messages. The demand for high capacity broadband transmission links continues to increase rapidly. In the recent years, wireless communications have drawn remarkable attractions for their flexibility. The market for wireless communications has enjoyed a tremendous growth. Wireless technology now reaches or is capable of reaching virtually every location on the surface of the earth. Hundreds of millions of people exchange information every day in their personal or business activities using pagers, cellular telephones and other wireless communication products. It seems that it is a feasible solution and market competitive for the broadband wireless communication systems to replace, or to complement the traditional fixed copper communication systems. Generally wireless communication environment is more adverse than fixed wired communication; it requires a reasonable system design or architecture to provide a reliable system performance. In the third generation (3G) wireless communication system, although the maximum data rate can be 2Mbps, the typical data rate is around 384kbps. To achieve the goals of broadband cellular service,

it is very appealing to leap to the fourth generation (4G) networks. In the last few years, the orthogonal frequency division multiplexing (OFDM) broadband wireless communication system has attracted much interest for its advantages. It is also considered as one of the promising candidates for wireless communication standard. The world standard bodies such as IEEE and ETSI are selecting the OFDM as the physical layer technique for the next generation of wireless systems.

OFDM is an old concept. It was first introduced in the late 60's, based on the multi-carrier modulation technique used in the high frequency military radio. Many years after its introduction, however, OFDM technique has not become popular. This is due to the requirements for large arrays of the sinusoidal generators and coherent demodulators in the transceiver. These are too expensive and too complex for the practical deployment. Eventually the discrete Fourier transfer (DFT) and inverse DFT (IDFT) were introduced as an effective solution to the arrays of sinusoidal generators and demodulators, which lower the system complexity. With the constant advances in the technology of very large scale integration (VLSI), the implementation of fast Fourier transform (FFT) and inverse FFT (IFFT) become possible and economical.

In 1971 Weinstein and Ebert proposed the use of IFFT/FFT as an efficient way to realize the OFDM function and the concept of the guard interval to avoid the inter-symbol interference (ISI) and inter-carrier interference (ICI). This proposal opened a new era for OFDM. The technique began to attract more and more attention and became very popular. OFDM has already been successfully adopted in many applications, such as digital audio broadcasting (DAB) and digital video terrestrial broadcasting (DVB-T). A brief history of OFDM technique and its applications are listed in Table 1.1. In the past few years, applying the OFDM technique in the wireless LAN (WLAN) has

received a considerable attention. OFDM technique has been adopted in the WLAN standards of IEEE 802.11a in North America and HIPERLAN/2 in Europe. It was considered for the IEEE 802.11g and the IEEE 802.16 WLAN standards.

Table 1.1 A history of OFDM technique and its applications

1957	Kineplex, multi-carrier high frequency (HF) modem
1966	R. W. Chang, Bell Labs, OFDM paper+patent
1971	Weinstein & Ebert proposed the use of FFT and guard interval
1985	Cimini described the use of OFDM for mobile communications
1987	Alard & Lasalle proposed the OFDM for digital broadcasting
1995	ETSI established the first OFDM based standard, digital audio broadcasting (DAB) standard
1997	Digital video terrestrial broadcasting (DVB-T) standard was adopted.
1997	Broadband internet with asymmetrical digital subscriber line (ADSL) was employed
1998	Magic WAND project demonstrated OFDM modems for wireless LAN
1999	IEEE 802.11a and HIPERLAN/2 standards were established for wireless LAN (WLAN)
2000	Vector OFDM (V-OFDM) for a fixed wireless access
2001	OFDM was considered for the IEEE 802.11g and the IEEE 802.16 standards

1.2 Objective of the Thesis

IEEE 802.11a can provide a variable data transmission rate from 6Mbps up to

54Mbps. Using IEEE 802.11a to build WLAN that offers up to 54Mbps data transmission rate is very appealing. It is well known that in many communication systems, it is important to deal with the multi-path fading channels. In the IEEE 802.11a system, OFDM technique is adopted as the modulation and demodulation technique in the physical layer. In an OFDM communication system, the broadband is partitioned into many orthogonal sub-carriers, in which data is transmitted in a parallel fashion. Thus the data rate for each sub-carrier is lowered by a factor of N in a system with N sub-carriers. By this method, the channel is divided into many narrowband flat fading sub-channels. This makes the OFDM system more resistant to the multi-path frequency selective fading than the single carrier communication system. The sub-carriers are totally independent and orthogonal to each other. The sub-carriers are placed exactly at the nulls in the modulation spectral of one another. At the peak point of one sub-carrier waveform, the sample values of other sub-carriers at the nulls are zeros and thus contribute no ISI to the sampled sub-carrier. This is where the high spectral efficiency of OFDM comes from. It can be shown that keeping the orthogonality of the sub-carriers is very critical for an OFDM system to be free from inter-carrier interference.

In a typically OFDM system, a cyclic prefix is used as a guard interval to avoid the ISI and ICI. The cyclic prefix is the insertion of the last N_g samples to the original sample sequence, where N_g is the length of the guard interval. This guard interval is required to be at least equal to or longer than the maximum channel delay spread of the system. Using a cyclic prefix as the guard interval is a simple way to reduce the ISI and ICI, however it also reduces the transmission efficiency of the system. The reduction

factor is $\frac{N}{N+N_g}$, where N is the number of the sub-carriers. In the IEEE 802.11a system, the number of sub-carriers is set to be N=64. For the IEEE 802.11a multi-path channel model, when the maximum channel delay spread is very long, the data transmission efficiency is significantly reduced. To maintain a high level of efficiency, a pre-FFT time domain equalizer (TEQ) is typically introduced in cascade with the original channel to shorten the effective channel impulse response to be in the range of the guard interval. This technique eliminates the necessary in extending the range of the cyclic prefix to obtain a comparatively high transmission efficiency. Many TEQ algorithms have been proposed based on different optimum criteria. However most popular algorithms [6-12] were developed for low data rate OFDM systems, such as ADSL applications, and their computational complexities are also high. In ADSL applications, before data transmission an initialization process is carried out for handshake and identification of the network equipment and the underlying physical infrastructure. The IEEE 802.11a OFDM system does not have an initialization process. In the IEEE 802.11a OFDM system, the training sequence is used to perform channel estimation, automatic gain control (AGC), coarse carrier frequency estimation and fine frequency tuning, etc. The total time interval for this training sequence is 16μs and its OFDM symbol interval is 4.0 μs. The TEQ algorithm for the high rate IEEE 802.11a OFDM system should have a reasonable computational complexity and should not take long time to fulfill. Millisecond-solutions for the TEQ coefficients are acceptable for low data rate systems. However for the 54Mbps IEEE 802.11a OFDM system, most of the popular TEQ algorithms are not suitable.

From the above discussion, it is necessary to investigate a TEQ algorithm with a

comparatively low computational complexity to be used in the IEEE 802.11a OFDM system. This thesis aims at achieving the followings:

1) Propose a time domain equalization technique for the high rate IEEE 802.11a OFDM system, which has a comparatively reduced computational complexity for a practical use.
2) Introduce the effective channel compensation function block for the quadrature amplitude modulation (QAM) after the insertion of the TEQ.
3) Setup the OFDM system according to the IEEE 802.11a standard in Simulink® including the TEQ and channel compensation function blocks, and test the functionality of the algorithm and the overall system performance.
4) Design the hardware model according to the proposed algorithm and implement the function blocks in a field programmable gate array (FPGA).
5) Test the FPGA chips to verify the algorithm and the functionalities, and analyze the difference between the hardware implementation and mathematical calculations.
6) Test the system performance using the hardware implementation output, and analyze system performance loss introduced by the hardware implementation.

When it is necessary to shorten the long channel's delay spread, the proposed algorithm is very attractive for a practical use. The simplicity of the technique makes its hardware implementation possible in FPGA for the high rate OFDM systems.

1.3 Thesis Organization

In Chapter 2, the OFDM system model based on the IEEE 802.11a standard is provided and discussed. Chapter 3 first describes some popular TEQ algorithms for

OFDM system to provide the background information. Then a reduced computational complexity TEQ algorithm is proposed and analyzed. The computation complexity of a popular algorithm and the proposed algorithm is analyzed and compared. In Chapter 4, examples of the functionality of the algorithm are demonstrated to show the validity of the algorithm. System performance through simulations is analyzed and the effect of the TEQ FIR filter with different orders on the system performance is also tested. In Chapter 5, FPGA hardware design model and the discrete implementation of the algorithm are discussed. To realize the function of the zero forcing equalization, a 64-point radix-4 pipeline FFT is also designed. In Chapter 6, the detailed information on hardware implementation is provided and the system performance based on the hardware implementation is tested. Simulation results are provided to verify the functionality of the chips and to study the system performance loss introduced by the hardware implementation. Finally Chapter 7 gives the conclusions and the suggestions for the further study.

Chapter 2 OFDM System Model

This chapter provides the background information about the IEEE 802.11a OFDM system. Specifically a top-level OFDM system model based on the IEEE 802.11a standard is introduced and analyzed.

2.1 Top-level OFDM System Model

2.1.1 Overview of the IEEE 802.11a

High data rate WLAN systems are being built rapidly around the world. The IEEE 802.11a systems are available in North America. Other systems currently under development include HIPERLAN in Europe, MMAC in Japan, and recently the IEEE 802.11g & h radio standards. While the 2.4GHz IEEE 802.11b WLAN moderately deployed around the world, and the associated ICs predominantly supplied by just a few vendors, OFDM in one form or another has won the endorsement from many major telecommunication equipment manufacturers, such as Lucent, Cisco, Philips Semiconductors and Nokia. IEEE 802.11a presents an immediate opportunity for new WLAN market entry. Numerous venture-capital-funded IC design groups are developing IEEE 802.11a multi-mode radios, some are acquired and supported by established companies. This urgent IEEE 802.11a development activity has created a tremendous demand for innovative and accurate design and verification solutions to help accelerate

high data rate WLAN product development.

The original standard family IEEE 802.11 was defined in 1997; and the IEEE 802.11a was defined in 1999. Since then, improvements had been proposed and adopted; and the standard was updated in 2002 to create a standard technology that could span multiple physical encoding types, frequencies and applications. The IEEE committee intends to setup the IEEE 802.11a in the same way as that of the popular IEEE 802.3 Ethernet standard, which has been successfully applied to 10, 100 and 1000 Mbps technology over fiber and various kinds of copper.

The OFDM physical layer (PHY) operates at a carrier frequency of 5GHz in the industrial, scientific, and medical (ISM) frequency bands. The radio frequency for the IEEE 802.11a OFDM layer is initially falling into the three 100MHz unlicensed national information infrastructure (U-NII) bands, 5.15-5.25, 5.25-5.35 and 5.725-5.825GHz. The centers of the outmost channels shall be at a distance of 30MHz from the band's edges for the lower and middle U-NII bands, and 20MHz for the upper U-NII band [1]. The spectrum allocation is subject to the authorities responsible for geographic specific regulatory domains. In Canada, it is regulated by the License Exempt Local Area NetworkCode (LELAN). Table 2.1 shows the channel allocation scheme for this standard. There are twelve 20MHz channels, and each band has a different output power limit. The first 100MHz in the lower section is restricted to a maximum power output of 40mW. The second 100MHz has a higher limit of 200mW, while the top 100MHz is dedicated for outdoor applications, with a maximum of 800mW power output. The IEEE 802.11a standard requires the receivers to have a minimum sensitivity ranging from -82 to -65dBm, depending on the chosen data rate shown in Table 2.2. The packet error rate

Table 2.1 OFDM operating bands and channels

Band	Channel numbers	Frequency (MHz)	Maximum output power (Up to 6 dBi antenna gain)
U-NII lower band 5.15 to 5.25 MHz	36	5180	40mW (2.5mW/MHz)
	40	5200	
	44	5220	
	48	5240	
U-NII middle band 5.25 to 5.35 MHz	52	5260	200mW (12.5mW/MHz)
	56	5280	
	60	5300	
	64	5320	
U-NII lower band 5.725 to 5.825 MHz	149	5745	800mW (50mW/MHz)
	153	5765	
	157	5785	
	161	5805	

(PER) is required to be less than 10% at a physical sub-layer service data units (PSDU) length of 1000 bytes with the input levels shown in the Table 2.2. When the PER is 10%, the power difference between the adjacent channel and desired channel is referred to as adjacent channel rejection. The power difference between the non-adjacent channel and desired channel is referred to as alternate adjacent channel rejection. These requirements are also indicated in Table 2.2. In a multiple cell network topology, overlapping and/or adjacent cells using different channels can operate simultaneously.

The IEEE 802.11a OFDM system can provide a variable data transmission rate of 6, 9, 12, 18, 24, 36, 48 and 54Mbps. Among these, the support of transmitting and receiving data rate of 6, 12 and 24Mbps is mandatory. The IEEE 802.11a system uses a training sequence for its synchronization. The training sequence comprises of 10 short symbols and 2 long symbols with the total training length of 16μs. The sub-carriers are modulated using binary phase shift keying (BPSK), quadrature PSK (QPSK), 16-quadrature amplitude modulation (16-QAM), or 64-QAM, depending on the data transmission rate.

Table 2.2 Receiver performance requirements

Data rate (Mbps)	Minimum Sensitivity	Adjacent channel rejection (dB)	Alternate adjacent channel rejection (dB)
6	-82	16	32
9	-81	15	31
12	-79	13	29
18	-77	11	27
24	-74	8	24
36	-70	4	20
48	-66	0	16
54	-65	-1	15

Convolution codes or punctured convolution codes are used as the forward error correction (FEC) to improve the system performance. Double block interleaving (outer block interleaving and inner block interleaving) is employed to combat the burst channel

interference. The 64-point IFFT and FFT are employed to realize the functions of the OFDM modulation and demodulation. A cyclic prefix is pre-pended as guard interval to avoid ISI and ICI. The total interval of an OFDM symbol is 4.0μs, including 3.2 μs for the data samples and 0.8μs for the guard interval. The IEEE 802.11a specifies the OFDM physical layer (PHY) characteristics. It loads the information signal into its 52 sub-carriers, among which 48 of the sub-carriers are used for data transmission and 4 remaining sub-carriers are used as the pilot sub-carriers. The pilot signal can be used as a reference to disregard frequency or phase shift of the signal during transmission. The sub-carrier frequency spacing is 0.3125MHz and the occupied bandwidth for each sub-carrier is 16.6MHz.

2.1.2 OFDM System Model

A top-level block diagram of the base-band high rate OFDM system is shown in Fig. 2.1. This model is based on the parameters defined in the IEEE 802.11a standard and includes the TEQ and the effective channel compensation function blocks. The function blocks shown in the diagram will be discussed in detail.

From the diagram the data flow can be described as follows. When the data transmission rate is 48 Mbps or 54 Mbps, the punctured convolution code with coding rate $R = 2/3$ or $3/4$ is adopted respectively. The input stream is first fed into the punctured convolution encoder. The coded bit stream is buffered and block interleaved. After that the binary bits are mapped into QAM signals according to the QAM constellation map. These complex numbers are then buffered to a multiplication of 64 samples, employs a 64-point IFFT operation to generate an OFDM symbol. The output data is then converted from parallel version to serial data, and the cyclic prefix is added.

The block inside the dotted line on the upper branch realizes the OFDM modulation.

The serial data stream is fed into the multi-path fading channel with additive white Gaussian noise (AWGN). At the receiver the inverse operations are employed. The corrupted signal is first passed to the TEQ finite impulse response (FIR) filter. The output signal is then converted to the parallel version after discarding the interfered cyclic prefix. A 64-point FFT is used to transfer the signal back to the base band frequency domain. The OFDM demodulator is also indicated in the dotted line box in the diagram (lower branch). Then the effective channel is compensated. After QAM demodulation, de-interleaving, Veterbi decoding, the approximated signal $d'(n)$ is recovered.

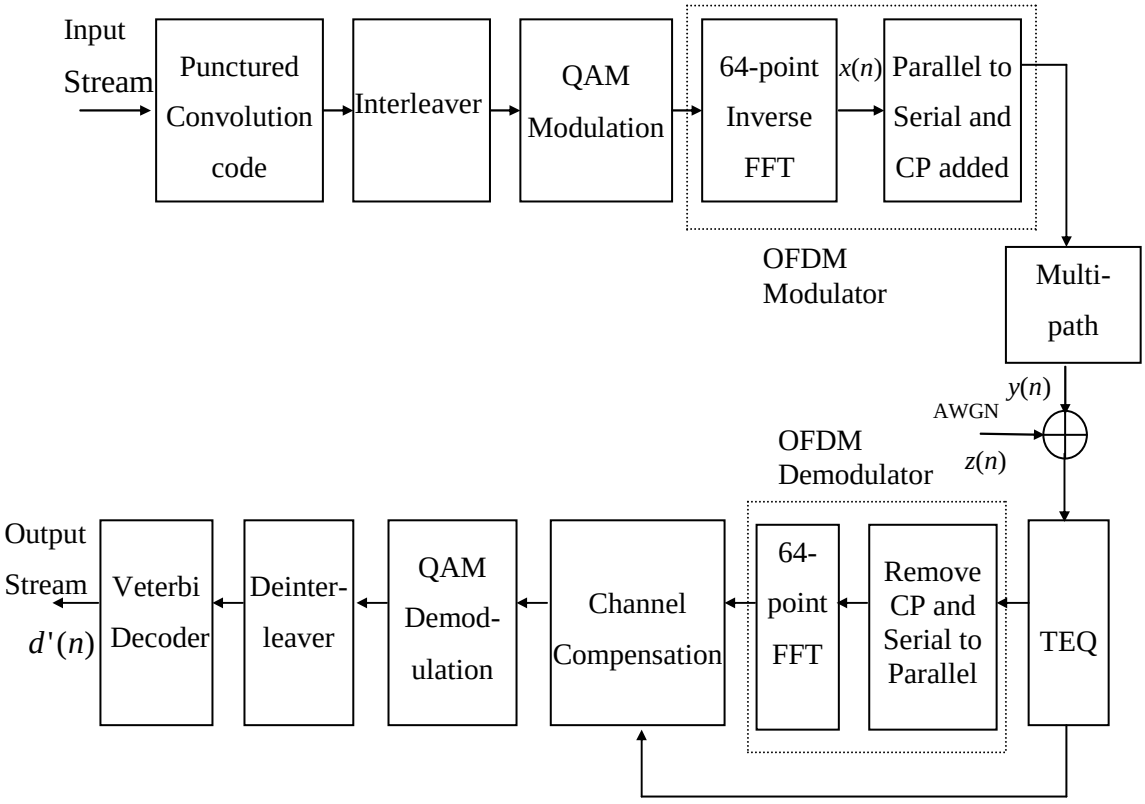


Fig. 2.1 A block diagram of the base-band OFDM

2.2 Frame format of the IEEE 802.11a

According to the IEEE 802.11a, the primary function of the OFDM PHY layer is to transmit media access control (MAC) protocol data units (MPDUs) as directed by the 802.11 MAC layer. The OFDM PHY layer consists of two protocol functions [1]:

1) A PHY convergence function, which adapts the capabilities of the physical medium dependent (PMD) system to PHY services.

2) A PMD system whose function defines the characteristics and methods of transmitting and receiving data through the wireless medium.

During transmission, PHY sub-layer service data unit (PSDU) is provided with a physical layer convergence procedure (PLCP) preamble and a header to create the PLCP protocol data unit (PPDU). The frame format of PPDU is shown in Fig. 2.2.

The frame of PPDU includes a 12-symbol PLCP preamble, PLCP header, PSDU, tail bits and pad bits. The fields of RATE, a reserved bit, LENGTH, an even parity bit and tail bits constitute a separate single OFDM symbol, i.e. the SIGNAL symbol. In the PLCP header, 6 tail bits are inserted to facilitate a reliable and timely detection of the RATE and LENGTH fields. It is important for this field to be correctly transmitted and detected, because it is used for the demodulation of the rest of the packet. It is transmitted with the most robust combination of the BPSK modulation and the convolution code with coding rate of $1/2$. The SERVICE, PSDU, tail bit and pad bits parts are also convolution encoded and the code rate depends on the required data transmission rate parameters listed in Table 2.3.

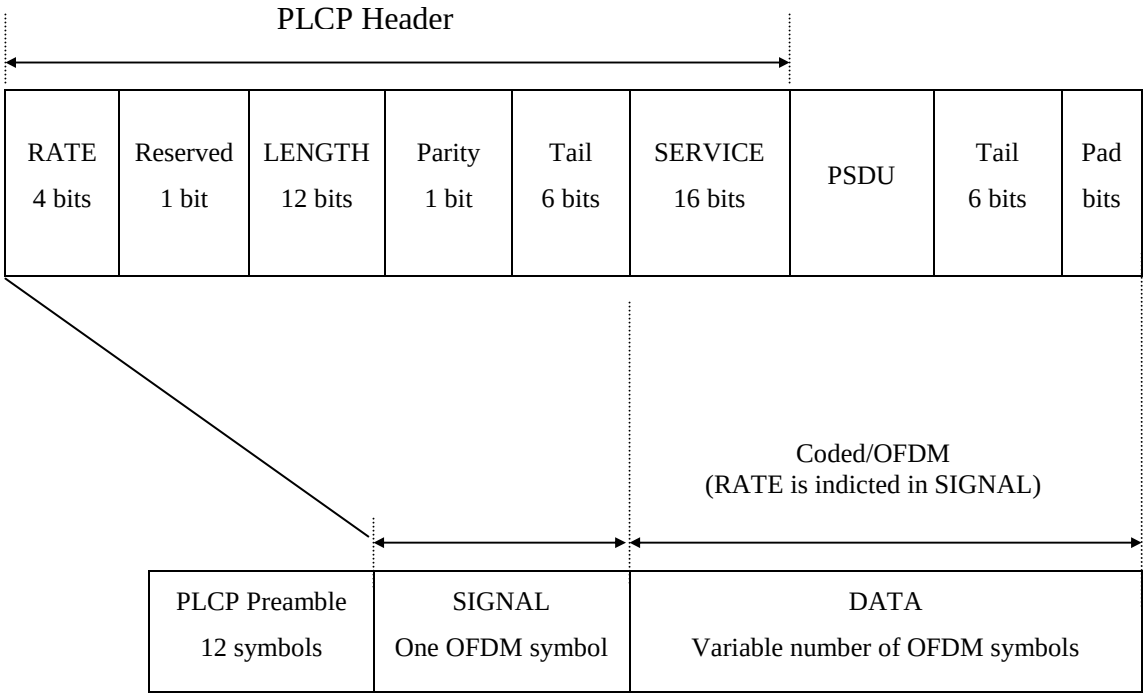


Fig. 2.2 PPDU frame format

When the data transmission rate is 54Mbps, it involves two types of the convolution coding rate, the SIGNAL part employs a (2, 1, 7) convolution code with a coding rate R=1/2; the data field shall be convolutional encoded with a coding rate R=3/4.

2.3 Convolution Encoder, Punctured Convolution Encoder and Viterbi Decoder

In this section the function blocks in the diagram will be discussed in detail. The forward error correction technique used in IEEE 802.11a is based on the convolution coding or the punctured convolution coding depending on the data transmission rate. Among the channel coding techniques, convolution coding has received much attention and is good for coded modulation implementation. It is often used in the digital communication system when the signal to noise ratio (SNR) is low. The code improves

system performance by adding redundant bits to the source information data. The choice of convolution codes depends on the applications.

Table 2.3 Rate dependent parameters

Data rate (Mbps)	Modulation	Coding rate	Coded bits per sub-carrier	Coded bits per OFDM symbols	Data bits per OFDM symbol
6	BPSK	1/2	1	48	24
9	BPSK	3/4	1	48	36
12	QPSK	1/2	2	96	48
18	QPSK	3/4	2	96	72
24	16-QAM	1/2	4	192	96
36	16-QAM	3/4	4	192	144
48	64-QAM	2/3	6	288	192
54	64-QAM	3/4	6	288	216

A convolution code is generated by feeding the source binary bits to a linear finite state shift registers. Generally a (*n, k, K*) convolution code can be implemented with a *k*-bit input, *n*-bit output linear sequential circuit with a memory length of *kK*-bits. The parameter *K* is called the constraint length of the convolution code [2]. The coding rate is defined as the ratio of the number of input bits into the convolution encoder to the number of output bits generated by the convolution encoder, i.e., $R = k / n$. Typically *n* and *k* are small integers with $k < n$, but the constrain length *K* should be large to achieve a low error probability. At each sampling clock, the input data is shifted *k* bits a

time into and along the shift registers that are *K* flip/flop long, and the oldest *k* bits are dropped out. After *k* bits have entered the shift registers, *n* linear combination of the current *kK* memory elements are computed and used to generate the encoded output bits. From the above encoding procedure, it is obvious that the *n*-bit encoded output not only depends on the most recent *k* bits but also on the previous $(K-1)k$ bits.

The convolution encoder with coding rate *R*=1/2 used in IEEE 802.11a can be expressed using the following industry standard generator polynomials:

$$g_0 = [1011011]$$

$$g_1 = [1111001]$$

The block diagram of the convolution encoder is shown in the Fig. 2.3.

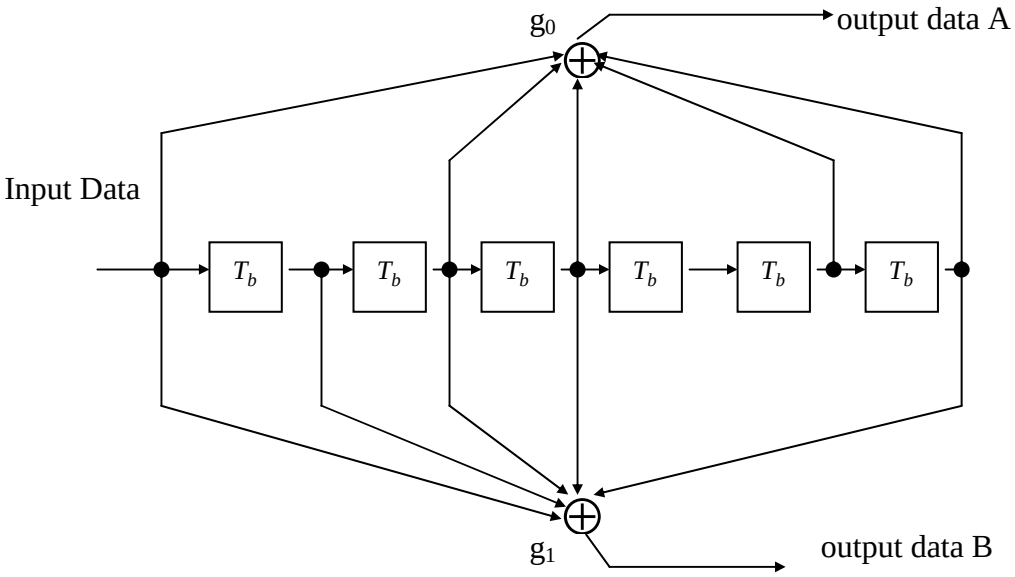


Fig. 2.3 Convolution encoder (*K*=7, *R*=1/2)

The bit denoted as the output data A is output from the encoder before the bit denoted as the output data B. The value of "1" in the polynomials means the connection to the modulo-2 adders, while "0" means no connection to the modulo-2 adder. Before

encoding, the shift registers are assumed to be in the all-zero state.

There are a number of techniques to decode the convolution code. The Viterbi decoder is the most popular method and it is commonly used to decode the bit stream coded by the convolution encoder. The Viterbi algorithm was first proposed in 1967 by A. Viterbi. The algorithm operates on the trellis structure of the code and determines the maximum-likelihood estimate of the transmitted sequence that has the largest metric. This rule maximizes the probability of a correct decision, i.e. it minimizes the error probability of the information bit sequence. If the channel is binary symmetric, a maximum-likelihood decoder is equivalent to a minimum distance decoder.

When decoding a long information bit sequence, the decoding delay usually is too long for most practical applications. Furthermore, the storage required to store the entire length of surviving paths is too large and expensive. Thus generally some compromises must be made [2]. The usually taken approach is to modify the Viterbi algorithm to obtain a fixed decoding delay without significantly affecting the optimum performance of the algorithm, thus to truncate the path memory of the decoder. The decoding decision made in this way is no longer the truly maximum likelihood, but it can obtain almost the same good performance, provided that the decoding window is long enough. Experience and analysis have shown that a decoding delay on the order of 5-7 times or more of the constrain length K results in negligible degradation in the performance compared with the optimum Viterbi algorithm. Detailed discussion of the algorithm can be found in [2].

In the IEEE 802.11a system, when a higher coding rate such as $R = 2/3$ or $3/4$ is desired, the punctured convolution code is employed. The punctured convolution code

starts with a lower coding rate of $R = 1/n$ code. In the IEEE 802.11a system it starts with a $R = 1/2$ code, then puncturing is used to create the needed higher coding rate. Puncturing procedure is to erase some of the encoded bits according to the punctured pattern defined in the transmission, which also reduces the number of transmitted bits and increases the coding rate. At the receiver the erasure bits must be inserted into the punctured data stream to make the coding rate back to $1/n$. These erasure bits are binary 0's inserted to the desired positions that were deleted by the puncturing operation in the transmitter.

By puncturing and insertion of the erasure bits, the Viterbi decoder operates on the metric of one input bit per encoded symbol instead of the higher numbers needed for higher rate codes. This avoids the computational complexity inherent in the implementation of a decoder of the high rate convolution code. Puncturing a code reduces the free distance of the rate $1/n$. In general the free distance is either equal to or 1 bit less than that of the best convolution code the same rate obtained directly without puncturing [2]. For the punctured convolution code system, sometimes the trace back length has to be extended to compensate for the addition of these dummy bits. Generally the decoding delay is longer than five times the constraint length of the convolution code before making decision.

2.4 Interleaving, Deinterleaving and Signal Mapping

Most coding techniques are devised to correct the error in the transmission of the information bits over the AWGN channels. The transmitted bits are affected randomly by the noise, thus the induced bit errors occur independently of bit positions. However there are many cases the interference will cause a burst of errors. One example is a

stroke of lighting or a human-made electrical disturbance. Another important example is the communication channel, which can cause burst transmission errors, like the multi-path fading channel. Fading caused by the time variant multi-path channel makes the SNR of the received signal to fall below a certain limit, disrupting a number of sub-carriers. The disruption causes a block or blocks of erroneous bits at the receiver. In general, codes designed for correcting statistically independent bit errors are not effective to correct burst errors.

The technique of interleaving is very effective to deal with burst errors. In the transmitter, the coded data bits are interleaved according to the designed interleaving pattern. At the receiver the deinterleaving operation is applied to convert the data bits back to their original indices. By interleaving the burst errors are spread to random positions and are transformed into random errors. These random errors can be effectively corrected by the codes designed for statistically independent errors.

Further more, for transmission in a multi-path channel environment, interleaving can provide time diversity against the fading. Generally it is desired that the ratio of interleaving interval to the coherent time is as large as possible. But as the ratio is big, the introduced time delay is also increased. In practical, the ratio is about 10 as long as we can implement the interleaving without suffering an excessive delay [2].

There are two structures of interleaver: block interleaver and convolution interleaver. In the IEEE 802.11a system, the encoded data bits are interleaved using block interleavers with a block size corresponding to the number of coded bits in a single OFDM symbol, N_C. The whole interleaver is divided into two parts: outer block interleaver and inner block interleaver. The outer interleaver ensures that adjacent coded

bits are mapped onto nonadjacent sub-carriers. The inner interleaver ensures that adjacent coded bits are mapped alternately onto less and more significant bits of the constellation and therefore long runs of low reliability bits are avoided.

The outer interleaving is defined by the following rule [1]:

$$i = (N_C / 16)(k \bmod 16) + \lfloor (k / 16) \rfloor, \qquad k = 0,1,\cdots,(N_C - 1) \tag{2.1}$$

Here k is the index of the coded bit before outer interleaving, i is the index after the outer interleaving, $\lfloor \ \rfloor$ is the function to find the maximum integer less than the number.

The inner interleaving is defined by the following rule [1]:

$$j = s \times \lfloor (i / s) \rfloor + (i + N_C - \lfloor (16 \times i / N_C) \rfloor) \bmod s, \quad i = 0,1,\cdots,(N_C - 1) \tag{2.2}$$

Here j is the index after the inner interleaving, and the value of s is determined by the number of coded bits per sub-carrier, N_{BS}, according to [1]:

$$s = \max(\frac{N_{BS}}{2}, 1) \tag{2.3}$$

The deinterleaving, which performs the inverse operation, is also defined by two rules. The inner deinterleaving rule is [1]:

$$i = s \times \lfloor (j / s) \rfloor + (j + \lfloor (16 \times j / N_C) \rfloor) \bmod s, \qquad j = 0,1,\cdots,(N_C - 1) \tag{2.4}$$

Here j is the index of the original received bit before deinterleaving, i is the index after the inner deinterleaving and s is defined in Equation (2.3). The outer deinterleaving rule is defined as [1]:

$$k = 16 \times i - (N_C - 1) \times \lfloor (16 \times i / N_C) \rfloor, \quad i = 0,1,\cdots,(N_C - 1) \tag{2.5}$$

Here k is the index after the outer deinterleaving.

After coding and interleaving, the bits stream is modulated by BPSK, QPSK, 16-QAM or 64-QAM according to the RATE field in the PLCP header in Fig. 2.2. The

serial input stream shall be divided into groups of N_{BS} (1, 2, 4, or 6) bits and mapped into modulated signals according to BPSK, QPSK, 16-QAM or 64-QAM constellation. In this thesis the focus is on the 54Mbps data transmission rate, at which the 64-QAM is employed. In this arrangement, every 6 input bits are mapped into one 64-QAM complex number according to the constellation shown in Fig. 2.4. Among the 6 input bits, the 3 least significant bits (LSB) $b_0 b_1 b_2$ determine the imaginary value of I and the most significant bits (MSB) $b_3 b_4 b_5$ determine the real value of Q, which is illustrated in Table 2.4.

Table 2.4 64-QAM encoding table

Input bits ($b_0 b_1 b_2$)	*I*-out	Input bits ($b_3 b_4 b_5$)	*Q*-out
000	-7	000	-7
001	-5	001	-5
011	-3	011	-3
010	-1	010	-1
110	1	110	1
111	3	111	3
101	5	101	5
100	7	100	7

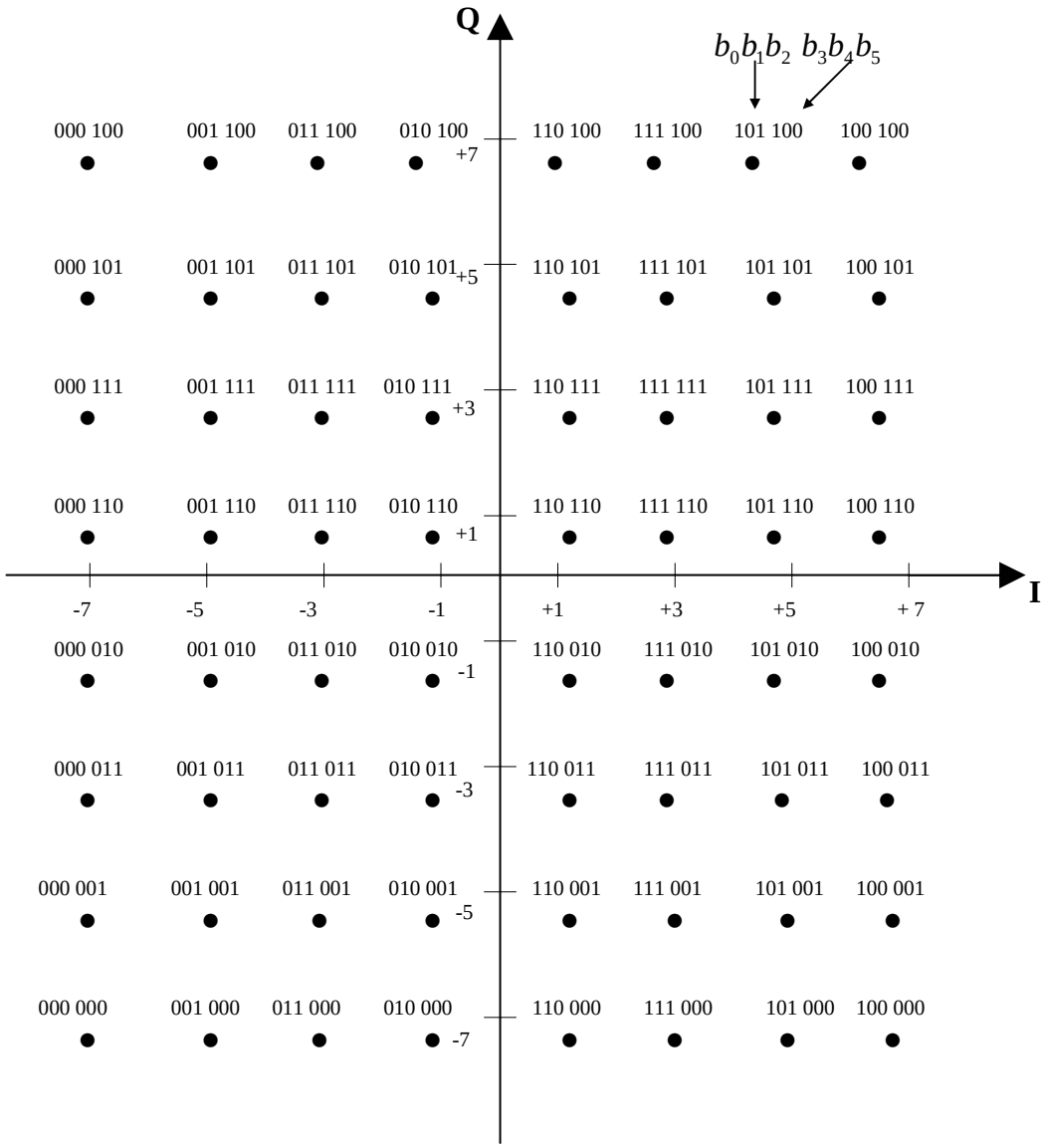


Fig. 2.4 64-QAM constellation bit encoding

2.5 FFT and IFFT

The IFFT/FFT is the most critical part of the OFDM system. FFT is an efficient way to calculate the discrete Fourier transform (DFT) to find the signal spectra. Since the DFT and inverse DFT (IDFT) basically involve the same type of computations, discussions of an efficient computational algorithm for the DFT also apply to the efficient computation of IDFT. The concept of the DFT is discussed first.

2.5.1 DFT/IDFT

The DFT of an *N*-point sequence $\{x(n)\}, 0 \le n \le N-1$ is calculated as [3]:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0, 1, \ldots, N-1 \tag{2.6}$$

Here $X(k)$ denotes the k^{th} discrete spectral sample and W_N is defined as:

$$W_N = e^{-j\frac{2\pi}{N}} \tag{2.7}$$

So the twiddle factor W_N^{kn} can be written as:

$$W_N^{kn} = e^{-j\frac{2\pi}{N}kn} \tag{2.8}$$

The IDFT of an *N*-point sequence $\{X(k)\}, 0 \le k \le N-1$ is similarly defined as:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \qquad n = 0, 1, \ldots, N-1 \tag{2.9}$$

The sequence $\{x(n)\}$ contains *N* samples in the time domain and the sequence $\{X(k)\}$ contains *N* samples in the frequency domain. The sampling points in the frequency domain occur at the *N* equally spaced frequencies $w_k = 2\pi k / N,\ k = 0, 1, \ldots, N-1$. With these sampling points, $\{X(k)\}$ uniquely represents the sequence of $\{x(n)\}$ in the frequency domain. Some important properties of the DFT, which can be exploited in the calculation, are introduced below.

It can be seen that W_N^{kn} is periodic with the period of *N*, i.e.,

$$W_N^{(n+mN)(k+lN)} = W_N^{nk},\ m, l = 0, \pm 1, \cdots \tag{2.10}$$

And it is easy to observe that the twiddle factor is inversely symmetrical stated as follows:

$$W_N^{k+N/2} = -W_N^k \tag{2.11}$$

These properties can be shown graphically on the unit circle as indicated in Fig. 2.5 for N=8, in which the twiddle factor is represented as a vector.

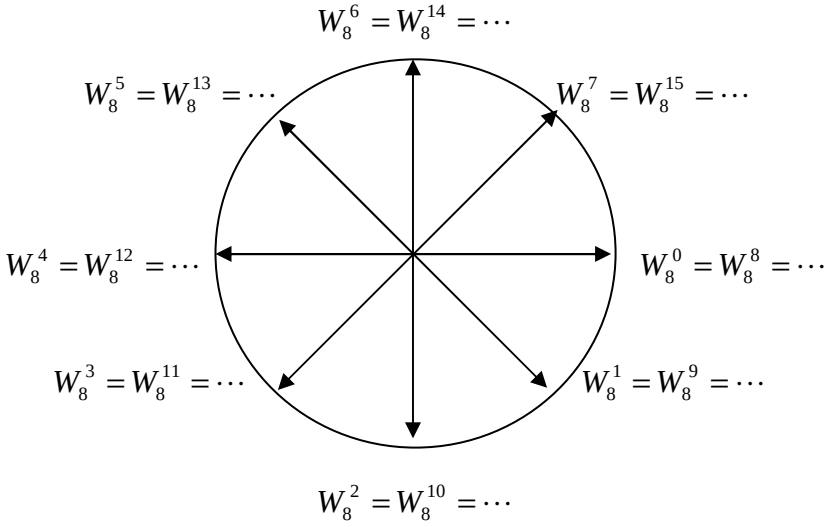


Fig. 2.5 Characteristics of twiddle factor

When $\{x(n)\}$ is a real-valued sequence, its DFT output is symmetrical. The DFT of a real sequence has the following property:

$$X(0) = X^*(0) \tag{2.12}$$

$$X(N-k) = X^*(k), \qquad k = 1, \ldots, N-1 \tag{2.13}$$

where "*" denotes complex conjugate. By the uniqueness of the DFT, the inverse is also true, that is if equations (2.12) and (2.13) are true then the IDFT of $\{X(k)\}$ produces a real sequence. This property can be exploited to generated real signal.

It can be observed from equation (2.6) that when $\{x(n)\}$is a complex sequence, a complete direct calculation of a N-point DFT requires $(N-1)^2$ complex multiplications and $N(N-1)^2$ complex additions. It can be seen that the computational complexity is in

the order of N^2, namely $O(N^2)$. For large values of *N*, direct calculation of the DFT is too computational intensive and not practical for implementation in hardware. So the idea of FFT is brought forward.

2.5.2 FFT/IFFT

The FFT algorithm is well known and widely used in digital signal processing for its efficient evaluation of the DFT. FFT/IFFT is one of most important feature in the OFDM communication system. In this thesis the IFFT/FFT is used for OFDM modulation and OFDM demodulation, it is also used in the FFT function block to realize the zero forcing equation to compensate the effective channel in frequency domain

The set of algorithms of FFT consists of various methods to reduce the computation time required to evaluate the DFT. The basic idea of FFT algorithm can be derived by decimating the original sequence into smaller sets either in time domain (DIT) or in frequency domain (DIF), then performs the DFT on each sub-set. The decimation process continues till the desired number of samples, which can be used to calculate the DFT easily and simply. There are many radices used in the decimation process. Among the numerous FFT algorithms, the radix-2 decimation in time (DIT) and decimation in frequency (DIF) algorithms are the most fundamental methods.

In the radix-2 algorithm, the length of the data sequence, $\{x(n)\}$, $n = 0, 1, \ldots, N-1$, is chosen to be a power of 2, i.e., $N = 2^p$, where *p* is a positive integer. Define two (*N*/2)-point sub-sequences $x_1(n)$ and $x_2(n)$ as the even and odd index values of $x(n)$, i.e.,

$$x_1(n) = x(2n), \qquad n = 0,1,\cdots,\frac{N}{2}-1 \tag{2.14}$$

$$x_2(n) = x(2n+1), \qquad n = 0,1,\cdots,\frac{N}{2}-1 \tag{2.15}$$

Then the *N*-point DFT in (2.6) can be expressed as:

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{kn} \\ &= \sum_{n=0}^{(N/2)-1} x(2n) W_N^{2kn} + \sum_{n=0}^{(N/2)-1} x(2n+1) W_N^{k(2n+1)} \end{aligned} \tag{2.16}$$

As $W_N^2 = [e^{j(2\pi/N)}]^2 = e^{j(2\pi/(N/2)} = W_{N/2}$, the above equation can be simplified as:

$$X(k) = \sum_{n=0}^{(N/2)-1} x_1(n) W_{N/2}^{kn} + W_N^k \sum_{n=0}^{(N/2)-1} x_2(n) W_{N/2}^{kn} \tag{2.17}$$

$$\text{or} \quad X(k) = X_1(k) + W_N^k X_2(k) \tag{2.18}$$

Here the $X_1(k)$ and $X_2(k)$ are the (*N*/2)-point DFT of $x_1(n)$ and $x_2(n)$, respectively, thus the *N*-point DFT $X(k)$ can be decomposed into two (*N*/2)-point DFT of $X_1(k)$ and $X_2(k)$, $0 \le k \le (N/2)-1$. If the (*N*/2)-point DFT is calculated directly, each (*N*/2)-point DFT requires $(N/2)^2$ complex multiplications, plus the $(N/2)$ complex multiplications with W_N^k, then the total number of complex multiplications required for computing $X(k)$ is $2(N/2)^2 + (N/2) = (N^2/2) + (N/2)$. This results in a reduction number of complex multiplication from N^2 to $(N^2/2) + (N/2)$. In the case of large value of *N*, it is almost a saving of 50% in calculation. This process to calculate the *N*-point DFT from the even and odd sequences of (*N*/2)-point DFT can be repeated until it reaches the stage of calculating the last 2-point DFT. The number of the stages for radix-2 *N*-point DFT calculation is therefore $p = \log_2 N$. The total number of complex multiplications is reduced from $(N-1)^2$ to $\frac{N}{2}\log_2 N$. For the 64-point DFT, the number

of complex multiplication is reduced from 3969 to 192, about 20 times reduction. For the 128-point DFT, the number of complex multiplication is reduced from 16129 to 448, about 36 times reduction. It can be seen that FFT is very efficient in the evaluation the DFT. As the value of *N* increases, the complex multiplication reduction also increases. Actually it can be seen that the multiplication by the twiddle factors such as $W_N^0, W_N^{N/4}, W_N^{N/2}, W_N^{3N/4}$ is equivalent to multiplication with 1, -j, -1 and j, respectively. They are just complex additions, subtractions or swap of the imaginary and real parts, which can be exploited to reduce the computation further.

An 8-point radix-2 FFT decimation in time process is shown in Fig. 2.6 [4]. It consists of three stages. The first stage can be realized solely with real additions and subtractions, which leads to an easy arithmetic element design. Also it can be seen from the figure that in order for the output sequence to be in the normal index, the input sequence is arranged in an order generally called bit-reversal. The rule is defined as the follows: if one string of *p* bits represents the normal index of the input sequence, then reverse the bits, the resulting bit string represent the index of the actually input sequence. For the 8-point FFT, the rule is shown in Table 2.5, the input index is arranged as $x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)$. This rule can be extended to the higher point of *N*, the input index can be calculated easily according to the bit-reserved rule. Alternatively, the input sequence can be arranged in the normal index, and a shuffler is employed to convert the bit reverse sequence back to its normal index at the output.

Table 2.5 Bit-reverse for 8-point FFT input sequence

Index	Initial bits	Reversed bits	Bit-reversed index
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

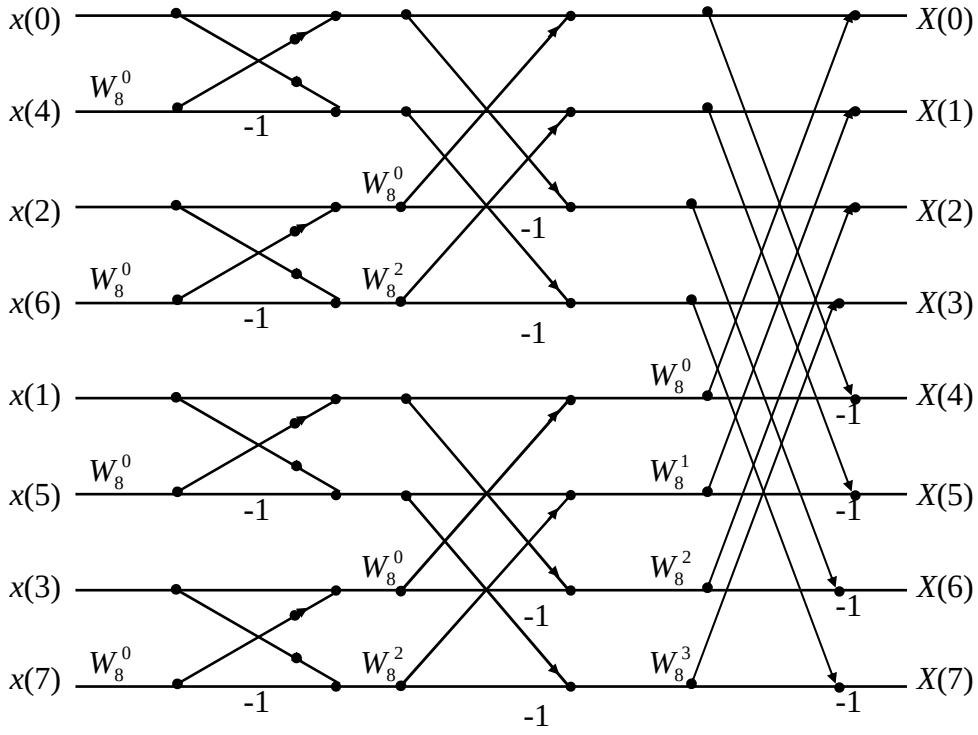


Fig. 2.6 Decimation in time of 8-point FFT.

For each stage the twiddle factors are also illustrated in the figure, which can be calculated according to the Equation (2.18).

Under different cases, the DFT calculation can be decimated other than by radix-2. FFT algorithms were proposed that decimate the input sequence based on different radices or mixed radices. An important one is the radix-4 FFT algorithm, which requires the number of data samples N to be a power of 4, i.e., $N = 4^p$, where p is a positive integer. In the hardware implementation of the zero forcing equalization, a 64-point radix-4 pipeline FFT is designed. The detailed discussions of this algorithm are presented in Chapter 4.

2.6 OFDM Transceiver

This section discusses the basic principles of the OFDM and its advantages.

2.6.1 OFDM Modulation and Demodulation Technique

OFDM is originated from the multi-carrier modulation and demodulation technique. A simple multi-carrier communication system is the frequency division multiplexing (FDM) or multi-tone. The broad transmission bandwidth is divided into many narrow non-overlapping sub-carriers, in which the data is transmitted in a parallel fashion. Ideally each sub-carrier is narrow enough so that the sub-carrier channel can be considered to be slow, flat fading to reduce the effect of ISI. The fundamental structure of a multi-carrier system is depicted in Fig. 2.7. The data stream is mapped to the desired waveform, filter banks are used to limit the signal bandwidth. After modulated by separate center frequencies, these signals are multiplexed and transmitted. At the receiver the frequency multiplexed signal is down converted to different channels by

multiplication with separate center frequencies, filtered by the filter banks to get the baseband multi-carrier signal for further processing.

The spectrum allocation for sub-carriers in a FDM system is shown as in Fig. 2.8, where f_0, f_1, ···, f_N are the center frequencies of the sub-carriers. This modulation has the following disadvantages:

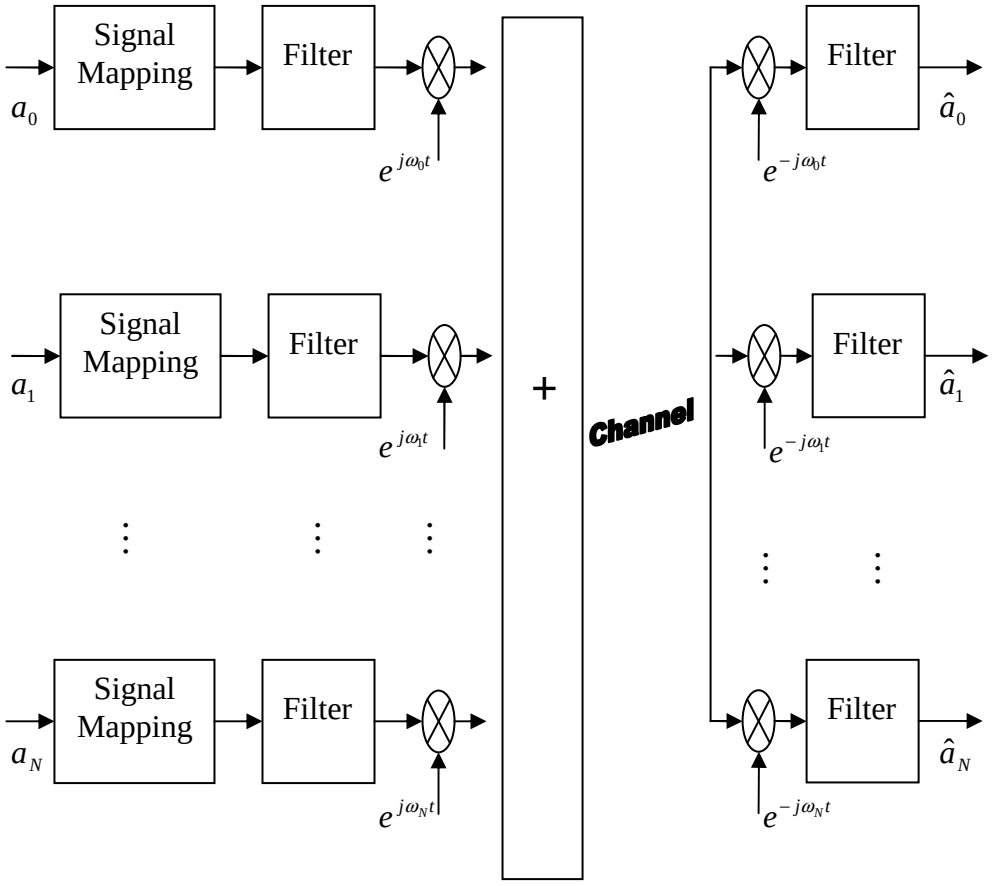


Fig. 2.7 Fundamental transceiver structure of a multi-carrier system

1) Since the sub-carriers are not overlapped with each other, the wide spacing between the sub-bands means a lower spectrum efficiency.

2) The filter banks are required both at the transmitter and the receiver, which make

the system more complicated.

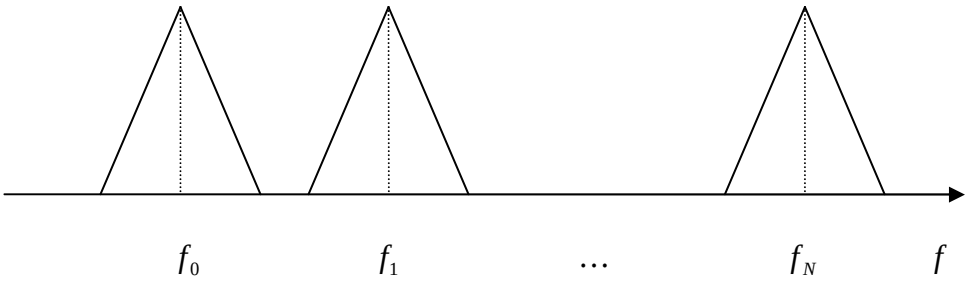


Fig. 2.8 FDM sub-band spectrum distribution.

With the input sequence $\{a[k]\}, 0 \le k \le N-1$, the frequency spacing Δf between the difference sub-carriers and the symbol interval T_s, the transmitted signal $x_a(t)$ can be expressed as:

$$x_a(t) = \sum_{k=0}^{N-1} a[k] e^{j2\pi k \Delta f t}, \quad 0 \le t \le T_s \tag{2.19}$$

If the signal is sampled at a rate T_s / N, then the above equation can be rewritten as:

$$x_a[n] = x_a(\frac{n}{N} T_s) = \sum_{k=0}^{N-1} a[k] e^{j2\pi n k \Delta f T_S / N} \tag{2.20}$$

If the following equation:

$$\Delta f T_s = 1 \qquad (\Delta f = \frac{1}{T_s}) \tag{2.21}$$

is satisfied, then the multi-carriers are orthogonal to each other and equation (2.20) can be rewritten as:

$$x_a[n] = \sum_{k=0}^{N-1} a[k] e^{j2\pi n k / N} = N \cdot IDFT\{a[k]\} \tag{2.22}$$

The above is just the IDFT expression of the input signal stream $\{a[k]\}$ with a difference of the gain factor 1/*N*. At the receiver the DFT implementation to find the approximate

signal $\hat{a}[k]$ can be written as:

$$\hat{a}[k] = DFT\{x_a[n]\} \tag{2.23}$$

$$= \sum_{n=0}^{N-1} x_a[n] e^{-i2\pi nk/N}$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} a[m] e^{j2\pi n(m-k)/N}$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} a[m] \sum_{n=0}^{N-1} e^{j2\pi n(m-k)/N}$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} a[m] N\delta[m-k]$$

$$= a[k]$$

Here $\delta[m-k]$ is the delta function defined as:

$$\delta[n] = \begin{cases} 1, & if\ n = 0 \\ 0, & otherwise \end{cases}$$

From the derivation above, it can be observed that there are two most important features of the OFDM technique, which are different from the traditional FDM systems.

1) Each sub-carrier has a different center frequency. These frequencies are chosen so that the following integral over a symbol period is satisfied:

$$\int_0^{Ts} a_m e^{j\omega_m t} a_l e^{j\omega_l t} dt = 0, \quad m \neq l$$

The sub-carrier signals in an OFDM system are mathematically orthogonal to each other. The sub-carrier pulse used for transmission is chosen to be rectangular so that the IDFT and DFT can be implemented simply with IFFT and FFT. The rectangular pulse leads to a $\frac{\sin(x)}{x}$ type of spectrum shape. The spectrum of the OFDM sub-carriers is illustrated in Fig. 2.9. The spectrum of the sub-carriers is overlapped to each other, thus

the OFDM communication system has a high spectrum efficiency. Maintenance of the orthogonality of the sub-carriers is very important in an OFDM system, which requires the transmitter and receiver to be in the perfect synchronization.

2) IDFT and DFT functions can be exploited to realize the OFDM modulation and demodulation instead of the filter banks in the transmitter and the receiver to lower the system implementation complexity and cost. This feature is attractive for practical use.

As it is already discussed, the IFFT and FFT algorithms can be used to calculate the IDFT and DFT efficiently. IFFT and FFT are used to realize the OFDM modulation and demodulation to reduce the system implementation complexity and to improve the system running speed.

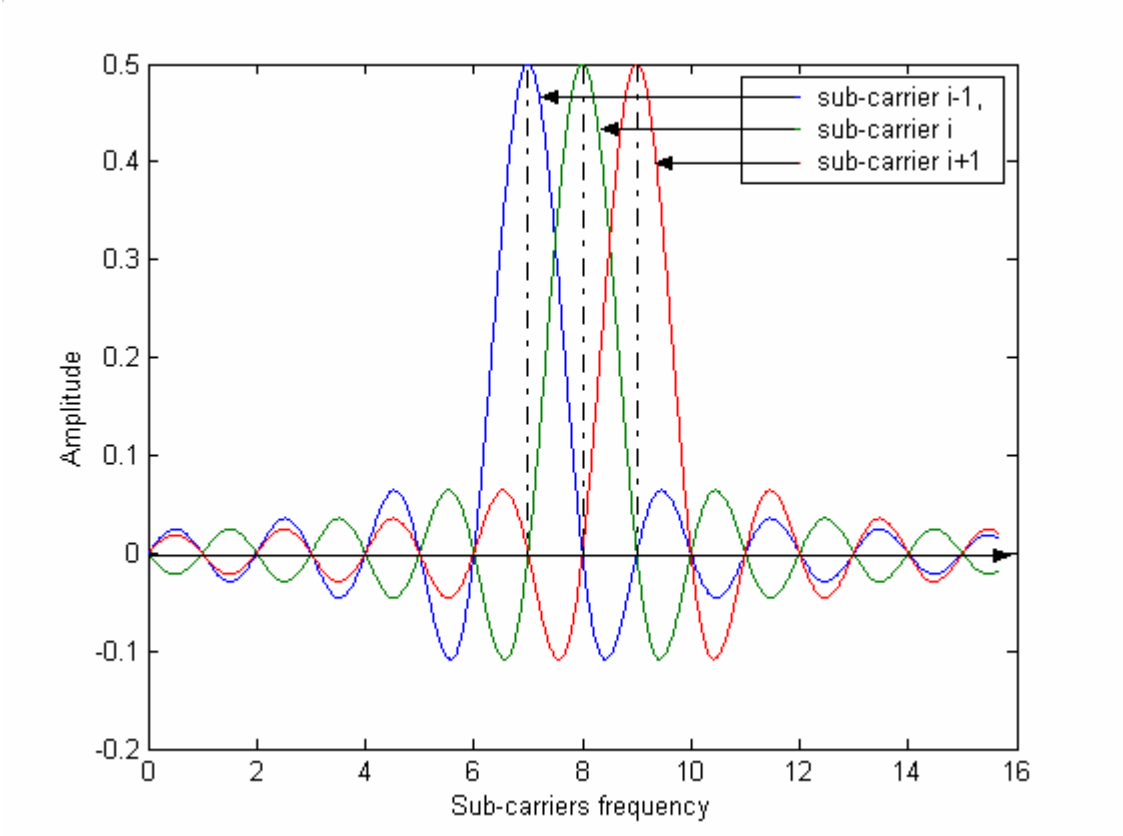


Fig. 2.9 Orthogonality principle of OFDM

2.6.2 Cyclic Prefix

It is known that in multi-path fading channel environment, channel dispersion cause the consecutive blocks to overlap, creating ISI/ICI. This degrades the system performance. In order for the orthogonality of the OFDM sub-carriers to be preserved, typically in an OFDM system, a guard interval is inserted. Actually the guard interval can be realized by the insertion of zeros, but using the cyclic prefix as guard interval can transform the linear convolution with the channel into circular convolution [14]. The insertion of cyclic prefix is very simple. Assume the length of the guard interval is v, it is just pre-pended the last v samples to the original OFDM sample sequence at the transmitter. At the receiver the so-called guard interval is removed. The process is shown in Fig. 2.10. The length of the cyclic prefix is required to be equal to or longer than the maximum channel delay spread to be free from ISI/ICI. As already mentioned, this is simple, but it reduces the transmission efficiency of the information bits.

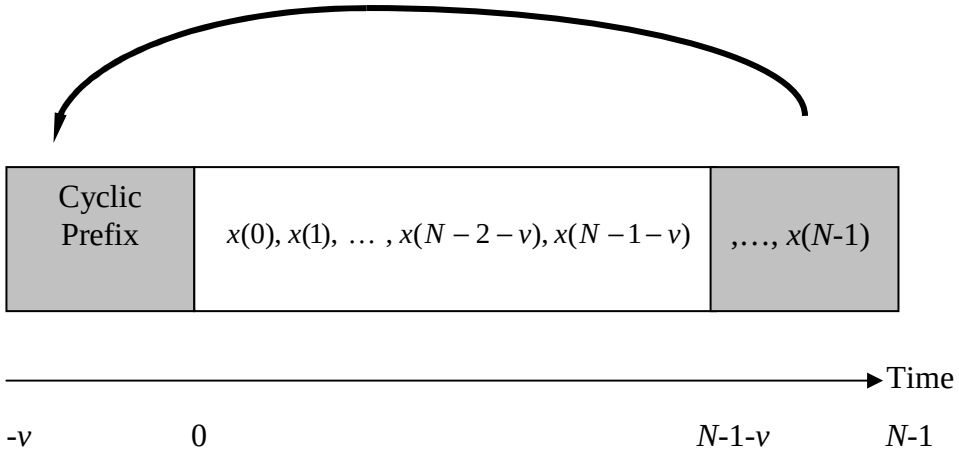


Fig. 2.10 The structure of cyclic prefix

The counteract of the cyclic prefix against the multi-path channel is shown in Fig. 2.11.

Assume that channel impulse is shown as $|h(t)|$, the maximum delay spread is shorter than the guard interval. The ith received OFDM symbol is only disrupted by the (i-1)th symbol. The “fading in” part of the received symbol, i.e., the corrupted guard cyclic prefix, is discarded, thus to provide a mechanism to suppress the ISI and ICI.

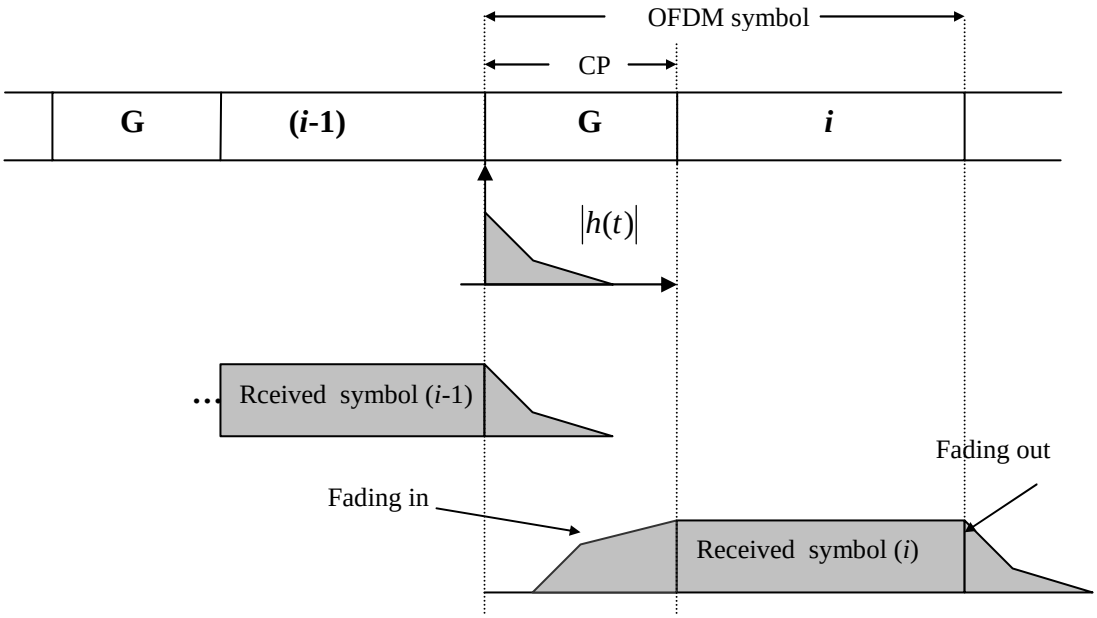


Fig. 2.11 The counteract effect of cyclic prefix against the ISI and ICI

2.7 Channel Model

The channel is the electromagnetic media between the transmitter and the receiver. The most common channel model is the Gaussian channel, which is generally called the additive white Gaussian noise (AWGN) channel. When signal is transmitted through the channel, it is corrupted by the statistically independent Gaussian noise. This channel model assumes that the only disturber is the thermal noise at the front end of the

receiver. Typically thermal noise has a flat power spectral density over the signal bandwidth.

The AWGN channel is simple and usually it is considered as the staring point to develop the basic system performance results. In wireless communication systems, the external noise and interference are often more significant than the thermal noise. Under certain conditions, the channel can not be classified as an AWGN channel but a multi-path fading channel. Multi-path fading is a common phenomenon in wireless communication environments, especially in the urban and sub-urban areas. When a signal is transmitted over a radio channel, it reflects, diffracts or scatters off the buildings, trees or other objects. The signal may have different propagation paths when it arrives at the receiver as shown in Fig. 2.12. Each path may introduce a different phase, amplitude attenuation, delay and Doppler shift to the signal. Since the transmission environment is always changing, therefore the phase, attenuation, delay and Doppler shift of the signal are random variables. At the receiver, when several versions of the transmitted signal are mixed together, at some points, they may add up constructively, and at the other points they may add up destructively. As the result the receiver may get a far diverse signal from the transmitted one. In this case, the channel is called a multi-path fading channel.

The impulse response of a multi-path channel generally exhibits a delay spread. The actual multi-path intensity profile for a certain channel needs to be estimated to have the characteristics of the channel. For a single transmitted impulse, the time between the first and the last received components is called the maximum excess delay T_m. Beyond T_m the power will fall below certain threshold level and can be discarded without

causing any significant error. If the maximum excess delay T_m lasts longer than the signal symbol time interval T_s, it will generate the ISI distortion and the channel is said to be frequency selective fading. Otherwise if $T_m < T_s$, all the received multi-path components arrive within a symbol time interval. In this case there is no channel induced ISI distortion and the channel is said to be frequency nonselective or flat fading.

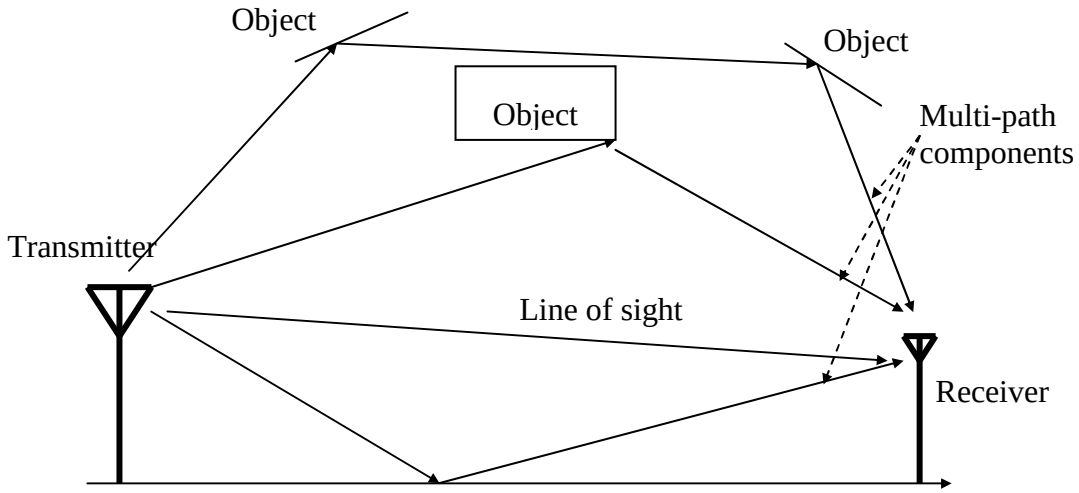


Fig. 2.12 The multi-path communication environment.

The multi-path phenomenon can also be specified in frequency domain. Another useful parameter of the multi-path fading channel is the reciprocal of the time spread, called the coherent bandwidth B_c. This quantity is a statistic measure of a range of frequency over which a signal's frequency components have a strong potential for amplitude correlation. If frequency components within this bandwidth receive approximately the same attenuation and group delay, the channel is said to be frequency nonselective. If the frequency components within the bandwidth often experience dramatically different attenuation and phase shift, the channel is frequency selective.

The coherence bandwidth B_c and maximum excess delay T_m are reciprocally related and can be calculated approximately with the knowledge of the other. The characteristic of a flat fading multi-path channel is shown in Fig 2.13. It can be seen that the coherence bandwidth is wider than the signal bandwidth, therefore all the signal frequency components are affected the same. In this situation the channel effect can be compensated easily.

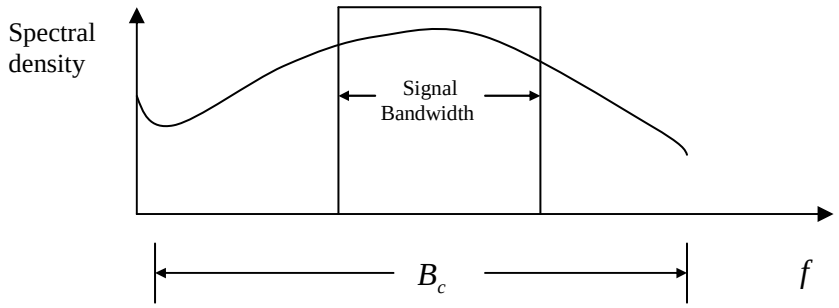


Fig. 2.13 Flat fading multi-path channel

Many diversity techniques were introduced to combat the multi-path fading effect. In an OFDM system, the broadband is divided into many narrow sub-carriers. The data transmission rate of each sub-carrier is lowered down by a factor of *N*. When the number of sub-carriers is sufficiently large, the channel transfer function is nearly flat in the interval of an individual sub-carrier. This changes a frequency selective channel into many frequency nonselective channels as shown in Fig. 2.14. It can be observed that the magnitude frequency response is not flat for all the sub-carriers, but for the i^{th} sub-carrier the frequency band is very narrow and its magnitude frequency response is considered to be flat during that interval. For this reason, the OFDM system is robust against the multi-path fading effect.

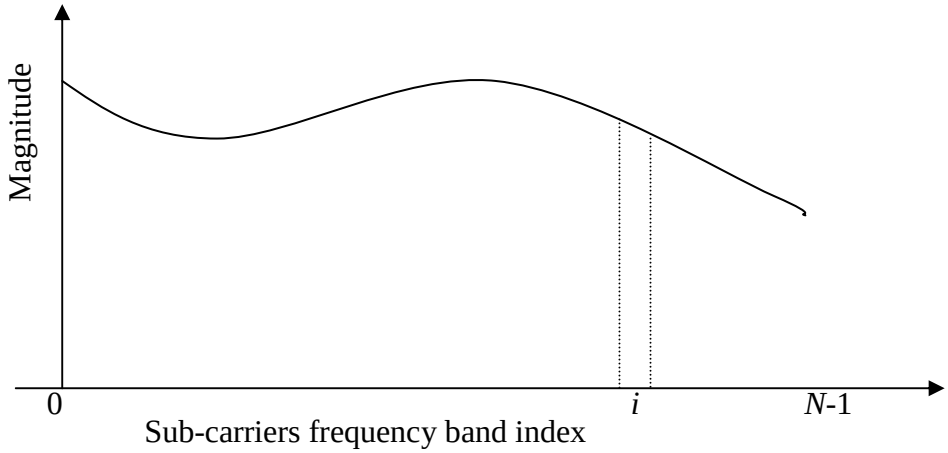


Fig. 2.14 Magnitude frequency response for the i^{th} sub-carrier in the OFDM system

As the signal is assumed to be band-limited, the time delay line model can describe this multi-path phenomenon with time varying coefficients and a fixed tap spacing, which can be shown graphically in Fig. 2.15. If assumed that there are infinite scatters, then the channel impulse response can be considered to be complex Gaussian process.

If there are multiple reflective paths and if there is no single dominant path, then the process is zero mean and the envelope of such a received signal is statistically described by a Rayleigh probability density function and the channel is said to be Rayleigh fading. If there is a single dominant path, then the process is nonzero mean and the fading envelope is described by a Rician probability density function. In this case the channel is considered to be Rician fading [2]. In practice, there are many channel models for wireless communication environments. The actual channel model needs to be estimated. In our case, the wireless channel is assumed to be the Rayleigh slow fading channel as this channel model is simple and very common. The multi-path power profile is assumed to be exponential decaying.

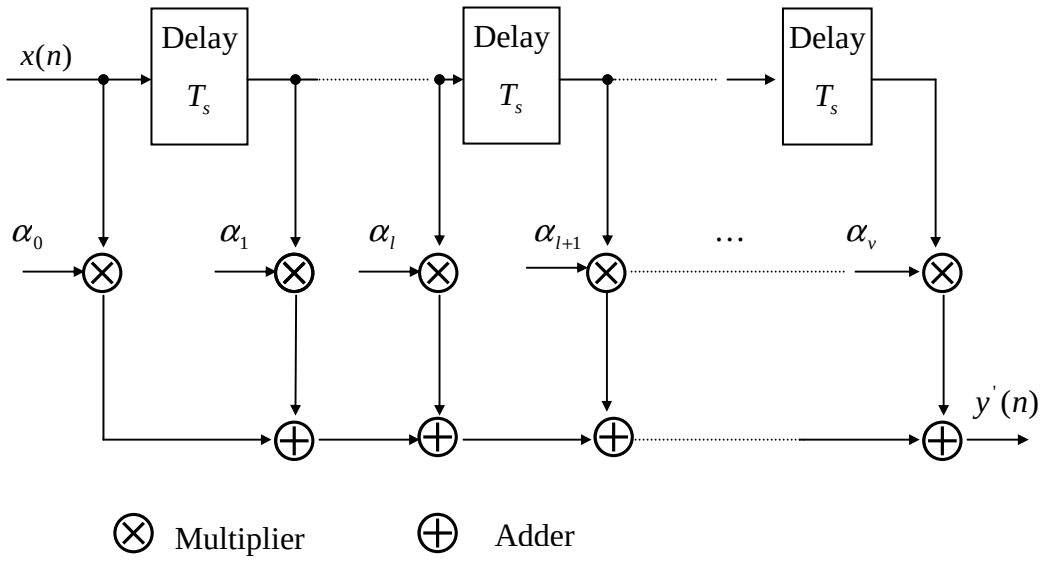


Fig. 2.15 A multi-path fading channel model

In Fig. 2.15, the maximum channel delay spread lasts *v* samples, the coefficients of $\alpha_0, \alpha_1, \cdots \alpha_v$ are the attenuations of the different paths. These coefficients are Rayleigh distributed random variables. The delay T_s denotes the unit sample delay. The output of the multi-path channel can be expressed as follows [2]:

$$y'(n) = \sum_{i=0}^{v} \alpha_i x(n-i) \tag{2.24}$$

Taking the AWGN noise $z(n)$ into account, the received signal can be written as:

$$y'(n) = \sum_{i=0}^{v} \alpha_i x(n-i) + z(n) \tag{2.25}$$

It can be seen from the above equation that the multi-path fading affects *v* samples. This will introduce ISI/ICI if no anti-measure is employed at the receiver.

2.8 Summary

In this chapter, an OFDM system model is introduced. It first discusses the format of the data frame, the punctured convolution code, the Viterbi decoder, the interleaving and deinterleaving. Then the principle of OFDM modulation and demodulation is presented. To counteract the multi-path fading effect, a guard interval using cyclic prefix is included. The length of the cyclic prefix is required to be equal to or longer than the maximum channel delay spread for the system to be free from ISI/ICI. This, however, will reduce the transmission efficiency when the maximum channel delay spread is very long. To solve this problem, a TEQ is generally introduced to shorten the effective channel impulse response to be within the range of the cyclic prefix. The principle of TEQ is discussed in next chapter.

Chapter 3 Time Domain Equalization (TEQ): Discussion and Analysis

This chapter first discusses some popular TEQ algorithms for OFDM system to provide the background information on the TEQ design. Then a reduced computational complexity TEQ algorithm for the high rate IEEE 802.11a OFDM system is introduced and mathematically analyzed. The complexity analysis shows a high computational reduction of the proposed algorithm.

3.1 System Model

Based on the OFDM system model given in Fig. 2.1, a data flow with TEQ is shown in Fig 3.1. The base band signal $x(n)$, generated by an *N*-point IFFT can be expressed as:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi kn}{N}},\ n = 0,1,\ldots,N-1 \tag{3.1}$$

where the sequence of $X(k)$ is the modulated data of the sub-carriers. After adding the cyclic prefix, the modulated signal passes through the multi-path fading channel. It is assumed that the channel is time invariant, at least during one OFDM symbol. Using the tap delay line channel model, the channel coefficient vector $\vec{h}$ is described by a vector:

$$\vec{h} = [\, h_0, h_1, \cdots, h_m \,]^* \tag{3.2}$$

Error!

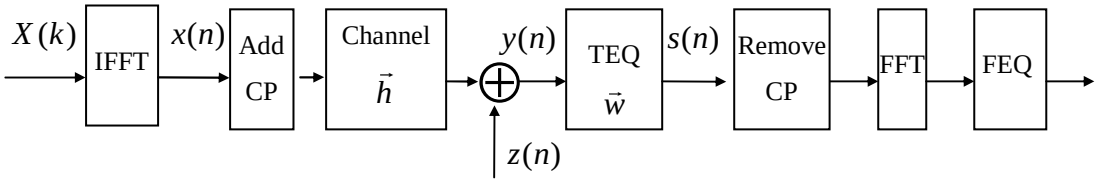


Fig. 3.1 Data flow in an OFDM system with TEQ

Here * indicates a conjugate transpose, and h_i is the attenuation of the i[th] path. The parameter *m* is the maximum channel delay spread value.

At the receiver, the received signal $\vec{y}$ can be written as:

$$\vec{y} = \vec{x} \otimes \vec{h} + \vec{z} \tag{3.3}$$

where $\otimes$ denotes the convolution operation and $\vec{z}$ is the additive white Gaussian noise (AWGN). After the insertion of TEQ, whose filter coefficients is denoted as $\vec{w}$, the received signal can be expressed as:

$$\vec{s} = \vec{x} \otimes \vec{h}_{eff} + \vec{z}_w \tag{3.4}$$

in which the effective channel impulse response $\vec{h}_{eff}$ is defined as:

$$\vec{h}_{eff} = \vec{h} \otimes \vec{w} \tag{3.5}$$

and the filtered modified noise

$$\vec{z}_w = \vec{z} \otimes \vec{w} \tag{3.6}$$

If the effective channel impulse response is limited to be in the range of the cyclic prefix, the received signal will be free from ISI and ICI without the necessary to extend the length of guard interval.

This received signal can also be expressed in frequency domain by the following equation:

$$S(f) = X(f)H_{eff}(f) + Z_w(f) \tag{3.7}$$

where $S(f), X(f), H_{eff}(f)$ and $Z_w(f)$ are the FFT of $\vec{s}$, $\vec{x}$, $\vec{h}_{eff}$ and $\vec{z}_w$, respectively. It can be observed from equation (3.7) that a simple zero forcing equalization (ZFE) technique can be implemented to compensate the effective channel impulse response. It is realized by multiplying the received signal with the vector $\frac{1}{H_{eff}(f)}$ in frequency domain. The disadvantage of this method is that when the gains of some of the sub-carriers are very low, the inversion of the gain would be very high and the noise in these sub-carriers is magnified. The signal to noise ratio (SNR) in these sub-carriers is decreased and it will introduce more bit errors, making the system performance degraded. At the receiver, after discarding the cyclic prefix, FFT operation, QAM demodulation, de-interleaving and Viterbi decoding, the approximated signal is obtained.

3.2 TEQ Algorithms

The equalization technique was originally proposed by Hirosaki [5]. Since then many equalization algorithms for OFDM systems, including TEQ, have been developed. The main approaches of the TEQ design can be formulated as follows. Given a physical multi-path fading channel, described by a FIR filter with coefficients vector $\vec{h}$, the maximum delay spread lasts *m* samples, another FIR filter with $\vec{w} = [w_0, w_1, \cdots, w_p]^*$ is used to cascade with the original channel. The effective channel can now be modeled as a FIR filter with impulse response $\vec{h}_{eff}$, known as the target impulse response (TIR) with some delays. The equalization design begins with the TIR and tries to find the optimum

coefficients $\vec{w}_{opt}$ of the TEQ to shorten the length of effective channel impulse response to be in the range of the guard interval. The criteria used for finding the coefficients of TEQ and TIR may vary, which leads to different algorithms. The most popular algorithms used to design TEQ (the MMSE, the MSSNR and the MGSNR) are discussed in the subsequent section.

3.2.1 Minimum Mean Square Error (MMSE) Algorithm and Its Variants

In [6], the minimum mean square error (MMSE) method is discussed. The structure of the algorithm is shown in Fig. 3.2.

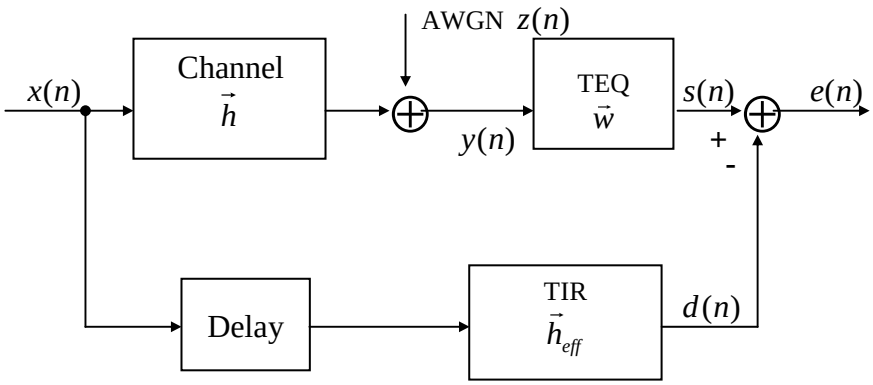


Fig. 3.2 Structure of MMSE equalizer

The MMSE algorithm is based on a channel shortening technique to decrease the system complexity of the Viterbi decoders [7]. Given the length of the TIR, it is desired to find the coefficients vector $\vec{w}$ of TEQ to minimize the mean square error (MSE) signal $\vec{e}$, given by:

$$MSE = E\{\vec{e}^{\,2}\} = \vec{h}_{eff}^{*} R_{xx} \vec{h}_{eff} - \vec{h}_{eff}^{*} R_{xy} \vec{w} - \vec{w}^{*} R_{yx} \vec{h}_{eff} + \vec{w}^{*} R_{yy} \vec{w} \tag{3.8}$$

Here $R_{xx} = E\{\vec{x}\vec{x}^{*}\}$, $R_{xy} = E\{\vec{x}\vec{y}^{*}\}$, $R_{yx} = E\{\vec{y}\vec{x}^{*}\}$, $R_{yy} = E\{\vec{y}\vec{y}^{*}\}$.

The matrices R_{xy} and R_{yx} depend on the delay. In order to minimize the mean square

error of *MSE* expressed in equation (3.8), the well-known orthogonality principle of the linear estimation theory can be exploited, to require:

$$E[\vec{e}\vec{y}^{*}] = 0 \tag{3.9}$$

which gives the solution as [7]:

$$\vec{h}_{eff}^{*} R_{xy} = \vec{w}^{*} R_{yy} \tag{3.10}$$

To avoid the trivial solutions, the MMSE method places a unit-tap constraint on TIR coefficients $\vec{h}_{eff}$ or TEQ coefficients (MMSE-UTC) $\vec{w}$ [8]. Assume f_i is the vector of zeros with unity in the i^{th} position. The problem is stated as follows [8]:

$$\min_{\vec{w},\vec{h}_{eff}} MSE \; with \; f_i^{T} \vec{h}_{eff} = 1 \tag{3.11}$$

Using equation (3.8) as a cost function, the following solution can be obtained [8]:

$$\vec{h}_{eff} = \frac{R_{x|y}^{-1} f_i}{f_i^{T} R_{x|y}^{-1} f_i} \tag{3.12}$$

where $R_{x|y} = R_{xx} - R_{xy} R_{yy}^{-1} R_{yx}$.

The optimum solution is given by examining all the possible $i \in \{0, N-1\}$ to maximize the achievable data rate in the multi-carrier system. It involves many matrix operations to find the optimum values of $\vec{h}_{eff}$ and $\vec{w}_{opt}$. If the MMSE algorithm places a unit-energy constraint on either TIR $\vec{h}_{eff}$ or the TEQ coefficients $\vec{w}$ (MMSE-UEC) [9], that is

$$\vec{w}^{*} R_{yy} \vec{w} = 1 \text{ or } \vec{h}_{eff}^{*} R_{xx} \vec{h}_{eff} = 1 \tag{3.13}$$

then it would give a smaller mean square error [9].

For the unit energy constraint on TEQ coefficients $\vec{w}$, the optimum solution can be calculated by the following equation [9]:

$$\vec{w}_{opt} = (\sqrt{R_{yy}})^{-1}\vec{g}_{opt} \tag{3.14}$$

where $\vec{g}_{opt}$ is the eigenvector corresponding to the minimum eigenvalue which satisfies the equation:

$$\vec{w}^T R_{yy}\vec{w} = \vec{w}^T \sqrt{R_{yy}}^T \sqrt{R_{yy}}\vec{w} = \vec{g}^T\vec{g} \tag{3.15}$$

with $\vec{g} = \sqrt{R_{yy}}\vec{w}$. The above calculations involve solving the minimum eigenvalue and eigenvector corresponding to the channel and noise independent matrices. Similarly the solution with the unit energy constraint on $\vec{h}_{eff}$ can also be obtained [9]. The high computational load of the algorithm makes it impractical to use. In [10] some iterative algorithms have been developed to decrease the computational complexity, but it requires many iterations to converge to a reasonable value. These algorithms are also impractical to be used in the high rate wireless communication systems.

There are some other variations of the MMSE algorithm. One is the MMSE decision feedback equalization (MMSE-DFE) [11]. This algorithm includes a feed forward filter $\vec{w}_f$ and a feed back filter $\vec{w}_b$ to adapt to the incoming signal. It involves finding the optimum values of $\vec{w}_f$ and $\vec{w}_b$ to minimize the error signal. In [12] the zero-pole MMSE-DFE is discussed. It approximates the original channel by pole-zero representation, written as in the following equation:

$$H(Z) = \frac{B(Z)}{1 + A(Z)} \tag{3.16}$$

It uses the autoregressive moving average (ARMA)-Levinson algorithm to find the coefficients of $B(Z)$ and $A(Z)$. If $1 + A(Z)$ is used as the transfer function of the equalizer, the effective transfer function becomes $H'(Z) = H(Z)(1 + A(Z)) = B(Z)$. If

the zeros of $B(Z)$ is less than or equal to the length of the guard interval, then the channel effect can be compensated free of ISI and ICI.

3.2.2 Maximum Shortening Signal to Noise Ratio (MSSNR)

A maximum shortening signal to noise ratio (MSSNR) algorithm is discussed in [13]. It is known that there are always some errors between the effective channel impulse response and the desired TIR coefficients. There is some part of the signal energy falling inside the target window, which is desired. But there is still some signal energy falling outside the target window, which contributes to the ISI. The definition of shortening SNR (SSNR) is expressed as follows:

$$SSNR = \frac{Energy_{in\_target\_window}}{Energy_{out\_target\_window}} \tag{3.17}$$

As shown in the Fig. 3.2, the target window does not need to begin at the first sample, but there can be some delays. The measured criterion in this algorithm is to find the coefficients of the TEQ, which will maximize the value of SSNR.

It is desired that the equivalent TIR would fall in the range of cyclic prefix with a delay *d* when the minimum ISI is to be obtained. A window function is defined to fit in the effective channel response $\vec{h}' = \vec{h} \otimes \vec{w}$. Let $\vec{h}_{win}$ =window ($\vec{h} \otimes \vec{w}$), then the window function can be written as the follows:

$$\vec{h}_{win} = \begin{bmatrix} h'_d \\ h'_{d+1} \\ \vdots \\ h'_{d+N_g} \end{bmatrix} \tag{3.18}$$

$$= \begin{bmatrix} h_d & h_{d-1} & \cdots & h_{d-p+1} \\ h_{d+1} & h_d & \cdots & h_{d-p+2} \\ \vdots & \vdots & \ddots & \vdots \\ h_{d+N_g} & h_{d+N_g-1} & \cdots & h_{d+N_g-p+1} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{p-1} \end{bmatrix}$$

$$= H_{win}\vec{w}$$

here , N_g is the length of cyclic prefix, *p* is the number of the TEQ coefficients and the length of original channel impulse response is *m*.

Similarly the equivalent channel impulse response outside the target window $\vec{h}_{wall}$ is defined as follows:

$$\vec{h}_{wall} = \begin{bmatrix} h'_0 \\ \vdots \\ h'_{d-1} \\ h'_{d+N_g+1} \\ \vdots \\ h'_{m+p-2} \end{bmatrix} = H_{wall}\vec{w} \tag{3.19}$$

The optimal solution becomes finding the coefficients of TEQ to maximize the SSNR. It is necessary that the coefficients of $\vec{w}$ are not all zeros, otherwise it will be meaningless. To simplify the problem, the H_{win} is normalized. Now the solution becomes to find the coefficients of TEQ to minimize $H^*_{wall} \cdot H_{wall}$ while maintaining the condition:

$$H^*_{win} \cdot H_{win} = 1 \tag{3.20}$$

The energy inside and outside the window can be expressed as

$$\vec{h}^*_{win}\vec{h}_{win} = \vec{w}^* H^*_{win} H_{win}\vec{w} = \vec{w}^* B\vec{w} \tag{3.21}$$

$$\vec{h}^*_{wall}\vec{h}_{wall} = \vec{w}^* H^*_{wall} H_{wall}\vec{w} = \vec{w}^* A\vec{w} \tag{3.22}$$

It can be observed that A and B are symmetric and positive definite. Assume B is

invertible, the Cholesky decomposition is used to decompose the matrix B [13]:

$$B = Q\Lambda Q^* = (Q\sqrt{\Lambda})(\sqrt{\Lambda}Q^*) \tag{3.23}$$

$$= (Q\sqrt{\Lambda})(\sqrt{\Lambda}Q)^* = \sqrt{B}\sqrt{B}^*$$

Here Λ is a diagonal matrix formed from the eigenvalues of B, and the columns of Q are orthonomal eigenvectors. Since B is full rank, the matrix $\sqrt{B}^{-1}$ exists.

Define $\vec{y} = \sqrt{B}^* \vec{w}$, the following equation can be obtained [13]:

$$\vec{y}^* \vec{y} = \vec{w}^* \sqrt{B}\sqrt{B}^* \vec{w} = \vec{w}^* B\vec{w} = 1 \tag{3.24}$$

Solving the above equation gives:

$$\vec{w} = (\sqrt{B}^*)^{-1} \vec{y} \tag{3.25}$$

$$\vec{w}^* A\vec{w} = \vec{y}^* (\sqrt{B})^{-1} A(\sqrt{B})^{-1} \vec{y} = \vec{y}^* C\vec{y} \tag{3.26}$$

with $C = (\sqrt{B})^{-1} A(\sqrt{B})^{-1}$. The solution to this problem is $\vec{y} = \vec{l}_{\min}$ where $\vec{l}_{\min}$ is the unit length eigenvector corresponding to the minimum eigenvalue $\lambda_{\min}$. The optimum TEQ coefficients can be obtained as follows [13]:

$$\vec{w}_{opt} = (\sqrt{B}^*)^{-1} \vec{l}_{\min} \tag{3.27}$$

So the optimum value of SSNR can be calculated accordingly.

It can be seen that the algorithm requires knowing the original channel impulse response, the length of the TIR and length of the TEQ. There are not so many constraints and pre-conditions, but requires too many matrix operations to obtain results.

3.2.3 Maximum Geometric Signal to Noise Ratio (MGSNR)

Al-Dhahir and Cioffi proposed the maximum geometric signal to noise ration (MGSNR) algorithm [14]. The criterion of the MGSNR is to maximize the bit rate. The

disadvantages of MGSNR algorithm are that it requires several assumptions to be met and also involves many matrix operations. The algorithm has a high computational complexity and is not suitable for practical use.

There are also other proposed algorithms to realize the equalization function in frequency domain, such as LMS. The problem of these algorithms is that the convergence process is very slow. They require many iterations for the multi-carrier communication system to converge to a stable solution. Therefore a new TEQ algorithm is sought for the high rate IEEE 802.11a OFDM systems.

3.3 A Reduced Complexity TEQ Algorithm

The IEEE 802.11a OFDM system does not have an initialization process as in the ADSL applications. In addition, the symbol interval is very short (4μs). For practical use in the high rate OFDM system, it is necessary for the equalization algorithm to have reasonable computational complexity and take a short time to complete. The algorithms [10-14] are therefore impractical to be implemented in a high rate WLAN. The MMSE method is discussed in section 3.2.1. Based on this MMSE criterion, an equalization algorithm for a high rate OFDM system with a relatively low complexity is proposed. In what follows, this algorithm is described. More importantly, necessary modifications to the algorithm are introduced to make the algorithm suitable for the practical implementation in the high rate OFDM system.

Given the length of the TIR, the original channel impulse response $\vec{h}$ and the length of the cyclic prefix, one want to find the optimum TEQ coefficients to minimize the error signal $e(n)$. Assume that the maximum channel delay spread is longer than the length of the cyclic prefix, thus the ISI/ICI can not be simply removed by increasing the

SNR, i.e., it is necessary to include a TEQ function block to shorten the effective channel impulse response. The TEQ coefficient vector is described by the following:

$$\vec{w} = [w_0, w_1, \cdots, w_p]^* \tag{3.28}$$

where p is the order of TEQ. The original channel impulse response is described by equation (3.2). As mentioned before, the effective channel impulse response $\vec{h}_{eff}$ is the convolution of the original channel and the TEQ FIR filter, namely $\vec{h}_{eff} = \vec{h} \otimes \vec{w}$. This filter has the length of (m+p+1). It is assumed that the length of cyclic prefix is N_g samples long. It is desired that the major components of $\vec{h}_{eff}$ are in the range of N_g samples. The components outside the range of cyclic prefix will contribute to the ISI and ICI, but it is made to be very small, thus no significant distortion is caused. Fig. 3.3 illustrates the effective channel impulse response ($\vec{h}_{eff}$) resulted from the convolution between the TEQ and the original channel impulse response.

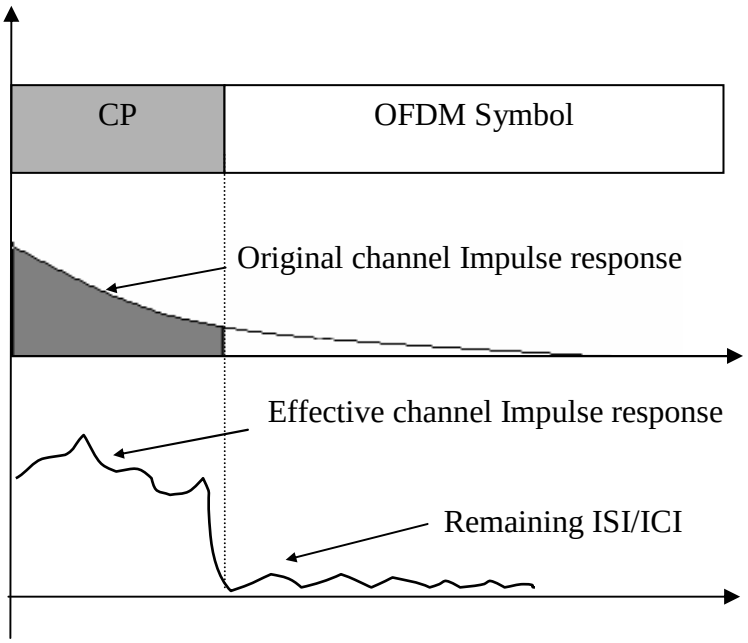


Fig. 3.3 The TEQ shortens the impulse response to the cyclic prefix

The solution for the TEQ coefficients involves the knowledge of the original channel impulse response, having the channel state information at the receiver is necessary. Numerous algorithms have been developed for channel estimation. In [17], a channel estimation algorithm with the aid of the training sequence is presented to estimate the channel for the OFDM system. In this project, as the research focuses on the design and hardware implementation of the TEQ algorithm, an ideal knowledge of the channel impulse response $\vec{h}$ is assumed.

To begin, a convolution matrix H of size $(m+p+1)\times(p+1)$ is generated as follows:

$$H = \begin{bmatrix} h_0 & h_1 & \cdots & h_m & 0 & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_m & \cdots & 0 \\ \vdots & & & & & & \vdots \\ 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_m \end{bmatrix}^* \tag{3.29}$$

Then the effective channel impulse response can be expressed by the following equation:

$$\vec{h}_{eff} = H \cdot \vec{w} \tag{3.30}$$

Ideally the effective impulse response $\vec{h}_{eff}$ should consist of the following components:

$$\vec{h}_{eff}(k) = \begin{cases} 1, & k = 0 \\ X, & 0 < k \le N_g \\ 0, & k > N_g \end{cases} \tag{3.31}$$

Here $\vec{h}_{eff}(k)$ refers to the k^{th} element of the vector $\vec{h}_{eff}$ and the symbol X means that the actual values of those components are not important. In practice, the energy of the samples outside the range of cyclic prefix should be minimized. In order to avoid the

trivial solutions, the first element of $\vec{h}_{eff}$ is set to 1. A reduced effective channel impulse response $\vec{h}_{re-eff}$ can be written in a simpler form as following:

$$\vec{h}_{re-eff}(k) = \begin{cases} 1, & k = 0 \\ 0, & 0 < k \le m + p - N_g \end{cases} \tag{3.32}$$

It then follows that a reduced convolution matrix H_{re} with size of $(m + p + 1 - N_g) \times (p + 1)$ can be generated accordingly from the matrix H by discarding the lines between 2 and $(N_g + 1)$ in H. Thus the following equation can be obtained:

$$\vec{h}_{re-eff} = H_{re} \cdot \vec{w} \tag{3.33}$$

It is desired to find the coefficient vector $\vec{w}$ of TEQ to satisfy the following equation:

$$H_{re} \cdot \vec{w} = \vec{d} + \vec{\delta} \tag{3.34}$$

where $\vec{d}$ is the $(m + p + 1 - N_g)$ ideal vector given by the following expression:

$$\vec{d} = [1, 0, \cdots, 0]^* \tag{3.35}$$

and $\vec{\delta}$ is the vector that describes the error signal:

$$\vec{\delta} = [\delta_0, \delta_1, \cdots, \delta_{m+p-N_g}]^* \tag{3.36}$$

Using these definitions, the cost function of MMSE can be written as follows:

$$\begin{aligned} E_{MSE} &= E\{\left| H_{re} \cdot \vec{w} - \vec{d} \right|^2\} = E\{|\vec{\delta}|^2\} = E[\vec{\delta}^* \cdot \vec{\delta}] \\ &= (H_{re} \cdot \vec{w} - \vec{d})^* (H_{re} \cdot \vec{w} - \vec{d}) \\ &= \vec{w}^* \cdot H_{re}^* \cdot H_{re} \cdot \vec{w} + \vec{d} \cdot \vec{d} - \vec{w}^* \cdot H_{re}^* \cdot \vec{d} - \vec{d} \cdot H_{re} \cdot \vec{w} \\ &= \vec{w}^* \cdot H_{re}^* \cdot H_{re} \cdot \vec{w} + \vec{d} \cdot \vec{d} - 2\vec{w}^* \cdot H_{re}^* \cdot \vec{d} \end{aligned} \tag{3.37}$$

The cost function is minimized on the condition of least square approach gives the following equation:

$$\frac{\partial E_{mse}}{\partial \vec{w}} = 0 \tag{3.38}$$

(or simply use the orthogonality principle). In this case the following equation exists to minimize the error signal $\vec{\delta}$:

$$\frac{\partial E_{mse}}{\partial \vec{w}} = -2H_{re}^{*} \cdot H_{re} \cdot \vec{w} + 2H_{re}^{*} \cdot \vec{d} = 0$$

The above equation yields:

$$\vec{w}_{opt} = (H_{re}^{*} \cdot H_{re})^{-1} \cdot H_{re}^{*} \cdot \vec{d} \tag{3.39}$$

In the case of AWGN environment, the coefficients of TEQ are given by [17]:

$$\vec{w}_{opt} = (H_{re}^{*} \cdot H_{re} + \gamma_{pe}^{2} \cdot I)^{-1} \cdot H_{re}^{*} \cdot \vec{d} \tag{3.40}$$

where $\gamma_{pe}^{2} = \frac{1}{SNR}$, with *SNR* is the signal to noise ratio and ***I*** is a $(p+1)\times(p+1)$ identity matrix.

Because the reduced convolution matrix H_{re} is used to calculate the optimum TEQ coefficients, the dimension of the matrix is reduced in the matrix multiplication, the computational complexity is also reduced accordingly. Observed that (3.40) involves matrix inversion, which is difficult to implement in hardware. However, this difficulty can be overcome as described below. Rewrite (3.40) as follows:

$$A = H_{re}^{*} \cdot H_{re} + \gamma_{pe}^{2} \cdot I \tag{3.41}$$

$$B = H_{re}^{*} \cdot \vec{d} \tag{3.42}$$

So the equation (3.40) can be rewritten as follows:

$$A \cdot \vec{w} = B \tag{3.43}$$

It can be observed from Equation (3.41) that $A^{*} = A$, i.e., A is symmetric. This means that it is not necessary to calculate all of its elements; only the diagonal and upper (or

lower) elements required to be calculated when performing the matrix multiplication. There are several algorithms to solve this kind of linear equation problem without solving the matrix inversion directly. Cholesky decomposition has a reduced computational complexity and is popular. But as it has a square root operation in finding the diagonal elements of the matrix, this requires long time to obtain the values. Thus the Cholesky decomposition is impractical for hardware implementation in a high data rate OFDM system. More simplification is to be sought.

Here A can be shown to be positive definite. Assume that $\{\vec{x}_k\}$ is the input stream, $\{\vec{y}_k\}$ is the response of matrix H_{re}, and the noise vector $\{\vec{z}_k\}$, then

$$\vec{y}_k = H_{re} \cdot \vec{x}_k + \vec{z}_k \tag{3.44}$$

here $\vec{y}_k = \begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+m+p-N_g} \end{bmatrix}$, $\vec{x}_k = \begin{bmatrix} x_k \\ x_{k+1} \\ \vdots \\ x_{k+p} \end{bmatrix}$ and $\vec{z}_k = \begin{bmatrix} z_k \\ z_{k+1} \\ \vdots \\ z_{k+m+p-N_g} \end{bmatrix}$.

It can be shown that

$$\begin{aligned} E[\vec{y}_k^* \vec{y}_k] &= \vec{x}_k^* \cdot H_{re}^* \cdot H_{re} \cdot \vec{x}_k + \vec{x}_k^* H_{re}^* \vec{z}_k + \vec{z}_k^* H_{re} \vec{x}_k \\ &= \vec{x}_k^* \cdot A \cdot \vec{x}_k + |\vec{z}_k|^2 + \vec{x}_k^* H_{re}^* \vec{z}_k + \vec{z}_k^* H_{re} \vec{x}_k \end{aligned} \tag{3.45}$$

For all nonzero $\vec{x}_k \in C^{(p+1)\times(p+1)}$, the above equation means the energy falling into the window with a window function defined as $h_w = \{1, \underbrace{0, \cdots, 0}_{p+1}, \overbrace{1, \cdots, 1}^{m+p+2}\}$. As stated above, the first element of the target impulse response $\vec{d}$ is set to be 1 to avoid trivial solution. Thus for any nonzero $\vec{x}_k$, $E[\vec{y}_k^* \vec{y}_k] > 0$. As noise $\{\vec{z}_k\}$ is far less than the desired signal, so $\vec{x}_k^* \cdot A \cdot \vec{x}_k > 0$ and A is positive definite and Hermitian. Furthermore, it can be derived

that the special case of LU decomposition, the LDL^T decomposition can be used to factorize A:

$$A = LDL^T \tag{3.46}$$

here L is the lower triangular square matrix, D is a diagonal matrix, $D = diag(d_1, \cdots, d_n\}$, and L^T is the symmetric transpose of L. It is known that when a matrix is ill conditioned, sometimes the LU decomposition does not exist or is numerical inaccuracy if the LU decomposition exists. Define the following quantity [18]:

$$\Omega = \frac{\left\| S \cdot T^{-1} \cdot S \right\|_2}{\left\| A \right\|_2} \tag{3.47}$$

with $T = (A + A^T)/2$, $S = (A - A^T)/2$ and $\left\| A \right\|_2$ is the norm-2 of the matrix A. It is desired to have the value of Ω not very large [18]. This condition is always satisfied whenever A is symmetric and positive definite. Therefore the decomposition of A is always numerical stable and it is safe to avoid the step to check that if it is necessary for the matrix to pivot or not in this case.

The LDL^T decomposition requires only half the computation of the LU decomposition, but the LDL^T decomposition needs one more substitution step to calculate the last result. Therefore, although LDL^T algorithm is used to factorize A, the LU decomposition's forward and backward substitutions are used to calculate the optimum TEQ coefficients. Based on the properties of matrices, there exists a lower triangular matrix L, a diagonal matrix D and an upper triangular matrix U that satisfy the following equation:

$$LDL^T = L(DL^T) = LU = A \tag{3.48}$$

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{N1} & l_{N2} & \cdots & l_{NN} \end{bmatrix} \tag{3.49}$$

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ 0 & u_{22} & \cdots & u_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{NN} \end{bmatrix} \tag{3.50}$$

$$D = diag(d_1, \cdots, d_N) \tag{3.51}$$

From the above equations, the equations to calculate the elements of *L*, *D* and *U* matrices can be written as follows:

$$\sum_{k=1}^{N} l_{ik} u_{kj} = a_{ij}, \quad i = 1, \cdots, N; j = 1, \cdots, N \tag{3.52}$$

Choose the diagonal elements:

$$l_{ii} = 1, \qquad i = 1, \cdots, N \tag{3.53}$$

For $j = 1, \cdots, N$, the equations are derived as follows:

$$\begin{cases} u_{1j} = a_{1j}, & j = 1, \cdots, N \\ u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}, & i = 2, \cdots, j \\ d_i = u_{ii}, & i = 1, \cdots, N \\ l_{ij} = \dfrac{1}{u_{jj}} (a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}), & i = j+1, \cdots, N \end{cases} \tag{3.54}$$

It can be observed that in equation (3.54) to calculate the values of l_{ij}, it includes division operations. The relationship between elements l_{ji} and u_{ij} is:

$$l_{ji} = u_{ij} / u_{ii}, \quad 1 \le j < i \le N \tag{3.55}$$

All the elements of the *L* and *U* matrices can be calculated accordingly.

After finding the L and U matrices, it is ready to solve for the optimum TEQ coefficients through substitutions. For forward substitution, it can be seen that:

$$B = H_{re}^{*} \cdot \vec{d} = [b_1, \cdots, b_{p+1}] \tag{3.56}$$

As $\vec{d} = [1, \cdots, 0]$, so it can be calculated that:

$$B = [h_0, \cdots, 0] \tag{3.57}$$

Substitute the equation (3.57) into equation (3.43), the equations to solve for $\vec{y} = [y_1, \cdots, y_{p+1}]$ can be written as:

$$\begin{aligned} y_1 &= \frac{b_1}{l_{11}} = h_0 \\ y_i &= \frac{1}{l_{ii}}(b_i - \sum_{j=1}^{i-1} l_{ij} y_j) = -\sum_{j=1}^{i-1} l_{ij} y_j, \qquad i = 2, \cdots, p+1 \end{aligned} \tag{3.58}$$

In backward substitution, the matrix of D and L^T is combined to save one step in solving the values of $\vec{w}_{opt}$ instead of calculation the intermediate values with each matrix. According to the LU decomposition algorithm, similar to the forward substitution, the equation to find the coefficients of $\vec{w}_{opt}$ can be written as:

$$\begin{aligned} w_{p+1} &= \frac{1}{u_{p+1}} y_{p+1} \\ w_i &= \frac{1}{u_{ii}}(y_i - \sum_{j=i+1}^{p+1} u_{ij} w_j), \qquad i = p, \cdots, 1 \end{aligned} \tag{3.59}$$

From the derivation above, it can be observed that the reduced convolution matrix is used, the dimension of the matrix multiplication is reduced accordingly, so it reduces the computational complexity. The matrix A is proved to be symmetric and positive define, the LDL^T decomposition is used to reduce the computation. There are some regularities between the elements of matrix A, which is exploited to further reduce the hardware

complexity. This is discussed in hardware design. The computational complexity is analyzed in the following section.

3.4 Computational Complexity Analysis

Before analyzing the complexity of the equalization algorithm employed in this thesis, the MSSNR algorithm discussed in section 3.2.2 is analyzed to provide comparative information.

3.4.1 Complexity Analysis of the MSSNR TEQ Algorithm

In section 3.2.2, the MSSNR TEQ algorithm is briefly reviewed. As mentioned after the insertion of the TEQ, the effective channel impulse response $\vec{h}_{eff}$ is represented by the equivalent impulse response composed of $\vec{h}_{win}$ and $\vec{h}_{wall}$. The part $\vec{h}_{win}$ represents the desired signal, a window of $(N_g + 1)$ consecutive samples from $\vec{h}_{eff}$. The part $\vec{h}_{wall}$ represents the signal falling out of the desired window, which contribute to the ISI. The dimensions of the convolution matrices H_{win} and H_{wall} are of $(N_g + 1) \times p$ and $(m + p - N_g - 2) \times p$. With setting the delay to a specific value, the solution for the optimum TEQ coefficient vector $\vec{w}$ is summarized as follows:

1) Calculate the matrices $A = H_{wall}^{*} H_{wall}$ and $B = H_{win}^{*} H_{win}$ first with a constraint of $\vec{w}^{*} B \vec{w} = 1$.
2) Calculate the square root matrix of *B* and matrix *C*, as shown below: $B = \sqrt{B}\sqrt{B}^{*}$, $C = (\sqrt{B})^{-1} A (\sqrt{B}^{*})^{-1}$.
3) Solve for the optimum coefficients $\vec{w}_{opt} = (\sqrt{B})^{-1} \vec{l}_{\min}$, where $\vec{l}_{\min}$ is the unit-

length eigenvector corresponding to the minimum eigenvalue $\lambda_{\min}$ of matrix *C*.

In order to find the optimum value, the above steps are iterated setting different values of the delay. Since the algorithm in this thesis does not involve iterative steps, for a fair comparison, the complexity of the MSSNR algorithm is determined without any iteration. Note that The calculation does not include additions and subtractions; and there may have different methods employed in calculation that will lead to different results. This is calculated as follows:

1) The calculation of *A* and *B* requires $p^2(N_g+1)$ and $p^2(m+p-N_g-2)$ multiplications.

2) The square root matrix of *B* can be calculated by Cholesky decomposition. The computation complexity of the decomposition is $\frac{p(p+1)(2p+1)}{12}-\frac{p(p+1)}{4}$. Substitutions require $p(p+1)$ computations. Without considering the pivoting, it requires $\frac{p(p+1)(2p+1)}{12}+\frac{3p(p+1)}{4}$ operations. The calculation of *C* requires the inversion of $(\sqrt{B})$ which also can be calculated by Cholesky decomposition. Then it requires p^4 more computations to calculate matrix *C*. The overall computation amounts to $p^4+\frac{p(p+1)(2p+1)}{6}+\frac{3p(p+1)}{2}$ for the second step.

3) As it is required to calculate the minimum eigenvalue and the corresponding minimum eigenvector. From the popular algorithms, assume the inverse power method is chosen. The computation complexity of the inverse power method is determined as follows. The first step requires $\frac{p(p+1)(2p+1)}{6}$ operations. The later

step requires p^2 operations. If it is assumed that there are 24 iterations, then combining with the calculation of $\vec{w}_{opt}$, the complexity would be $\frac{p(p+1)(2p+1)}{6} + 25p^2$ computations.

The computation complexity of MSSNR algorithm can be approximated by the following equation:

$$\boldsymbol{O}(\textit{MSSNR algorithm}) = p^2(m+p-1) + \frac{3p(p+1)}{2} + \frac{p(p+1)(2p+1)}{3} + 25p^2 + p^4.$$

3.4.2 Complexity Analysis of the Proposed TEQ Algorithm

The computational complexity of the proposed algorithm is calculated as follows. The dimension of the convolution matrix is reduced by N_g and p is the order of the TEQ. The number of multiplication is $p(m+p-N_g-2) + \frac{p(p+1)}{2}$ (exploits the regularity of the elements in the matrix A which is discussed in Chapter 4), the LDL^T decomposition and substitutions require $\frac{p(p+1)(2p+1)}{12} + \frac{3p(p+1)}{4}$ calculations. The total computation amount to:

$$\boldsymbol{O}(\textit{proposed algorithm}) = p(m+p-N_g-2) + \frac{p(p+1)(2p+1)}{12} + \frac{5p(p+1)}{4}.$$

As an example, assume that the lengths of the channel impulse response, the TEQ FIR filter and the guard interval are 15, 8 and 8 respectively. Based on the complexity analysis, the MSSNR algorithm requires 7620 computations. On the other hand, the proposed algorithm requires only 296 computations. The computation complexity is reduced by 25 times.

3.5 Summary

After the analysis of the popular TEQ algorithms, it is found that the most of them have high computational load. They are not suitable for a high rate IEEE 802.11a OFDM system. A reduced computational complexity TEQ algorithm is proposed and analyzed. Compared to other TEQ algorithms, it is very attractive for practical use in a high rate OFDM system, due to its simplicity.

Chapter 4 Simulation Results and Discussions

In this chapter, the functionality of the proposed TEQ algorithm is simulated and analyzed. Based on the IEEE 802.11a standard, a system model is set up in Simulink® and the overall system performance is tested. The effect of different orders of the TEQ FIR filters on the system performance is also investigated.

4.1 System Model

To demonstrate the validity of the proposed algorithm and the effect of the TEQ on the system performance, an OFDM system model is set up in Simulink® according to the IEEE 802.11a standard. The top-level system model is shown in Fig. 4.1. The data transmission rate is chosen to be 54Mbps. There are 48 sub-carriers to be used for data transmission. As the pilot signal is not processed in this research, the 4 pilot sub-carriers are not used. A rate-3/4 punctured convolution code is employed for the FEC. The outer and inner block interleavers are also included; the size of the block interleaving is 288. The inner and outer interleaving patterns are formulated according to the standard. The bit stream is mapped to complex signals according to the 64-QAM constellations. The 64-point IFFT and FFT are used. The AWGN is also included in simulation. The length of the cyclic prefix defined in the IEEE 802.11a is 16 samples, but in this thesis, different lengths are assigned to show the effect of the TEQ on the system performance.

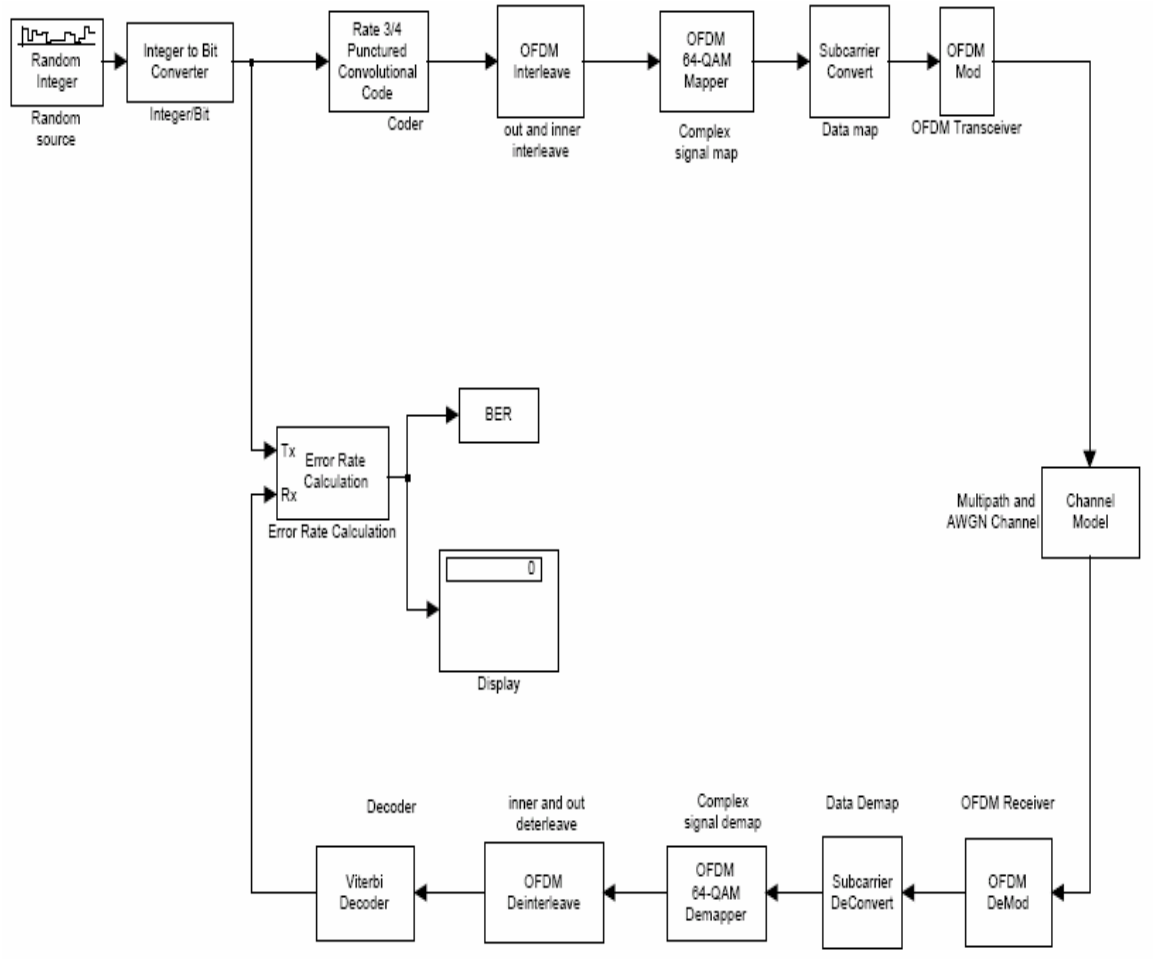


Fig. 4.1 Top-level Simulink model

4.2 Simulation Results and Discussions

As an example, the functionality of the TEQ algorithm is illustrated in Fig. 4.2. It is assumed that the original channel impulse response lasts 15 samples (shown in Fig. 4.2-(a)) and the length of the cyclic prefix is 8 samples. This implies that if a TEQ is not included, the ISI/ICI will greatly degrade the system performance. After the pre-FFT TEQ with an order of 31 (shown in Fig. 4.2-(b)) is inserted, the effective channel impulse response is shortened to 9 samples long as shown in Fig. 4.2-(c). It can be

observed that the major components of the effective channel impulse response fall in the range of the cyclic prefix. All samples beyond the cyclic prefix (after the 9th sample) contribute to the ISI/ICI, but they are very small, i.e., producing insignificant error.

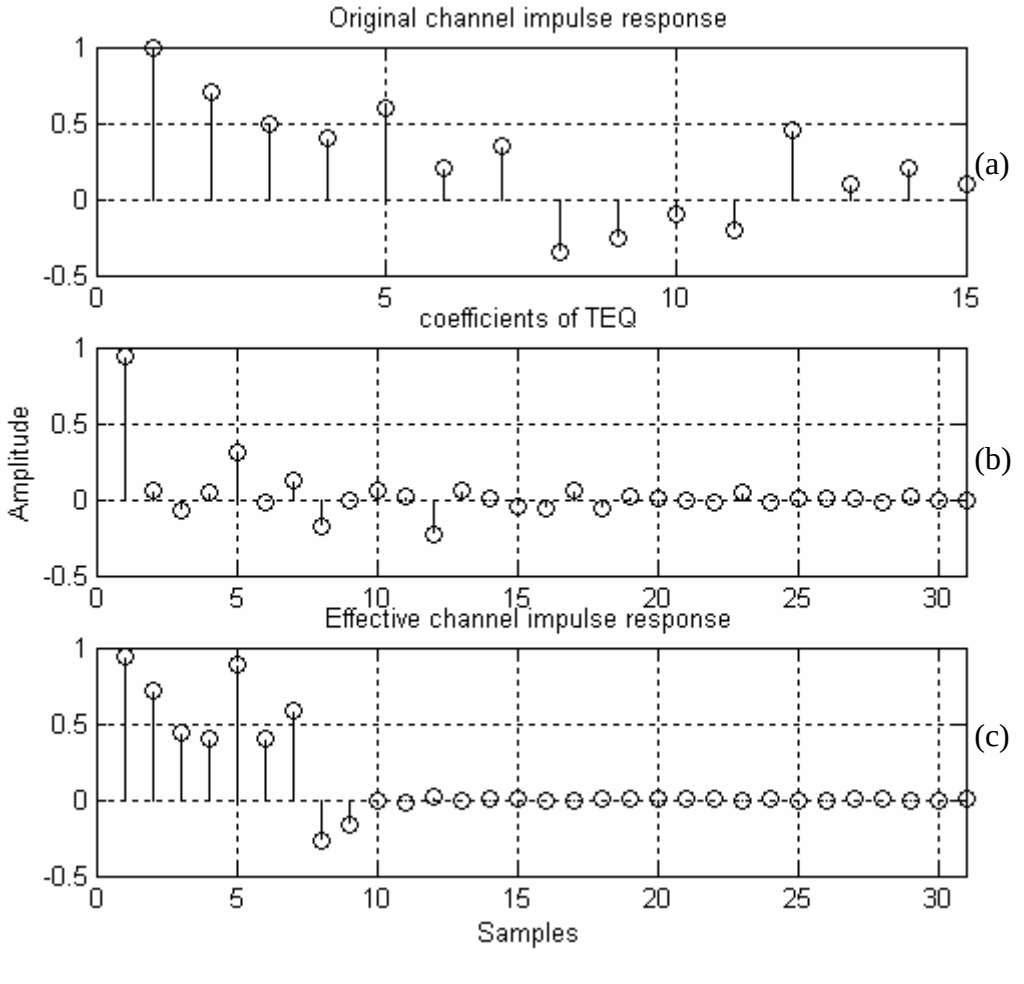


Fig. 4.2 Functionality validation of the TEQ

The effect of using different lengths of cyclic prefix on the system performance is also tested. The bit error rates are shown in Fig. 4.3. In this case no TEQ is employed and the ZFE is used for compensating the effective channel impulse response. It can be observed that when the length of the cyclic prefix is small (for example, the CP=8, 9 or 10), the ISI/ICI is severe and the bit error rate is very high. While extending the length of the cyclic prefix, more energy will fall in the cyclic prefix and the energy that

contributes to the ISI/ICI will become less. The overall system performance is improved. For the system with the length of cyclic prefix to be 14, it can be seen that increasing the signal to noise ratio from 12dB to 22dB, the system performance can be improved significantly compared with the bit error rate with the length of the cyclic prefix of 8.

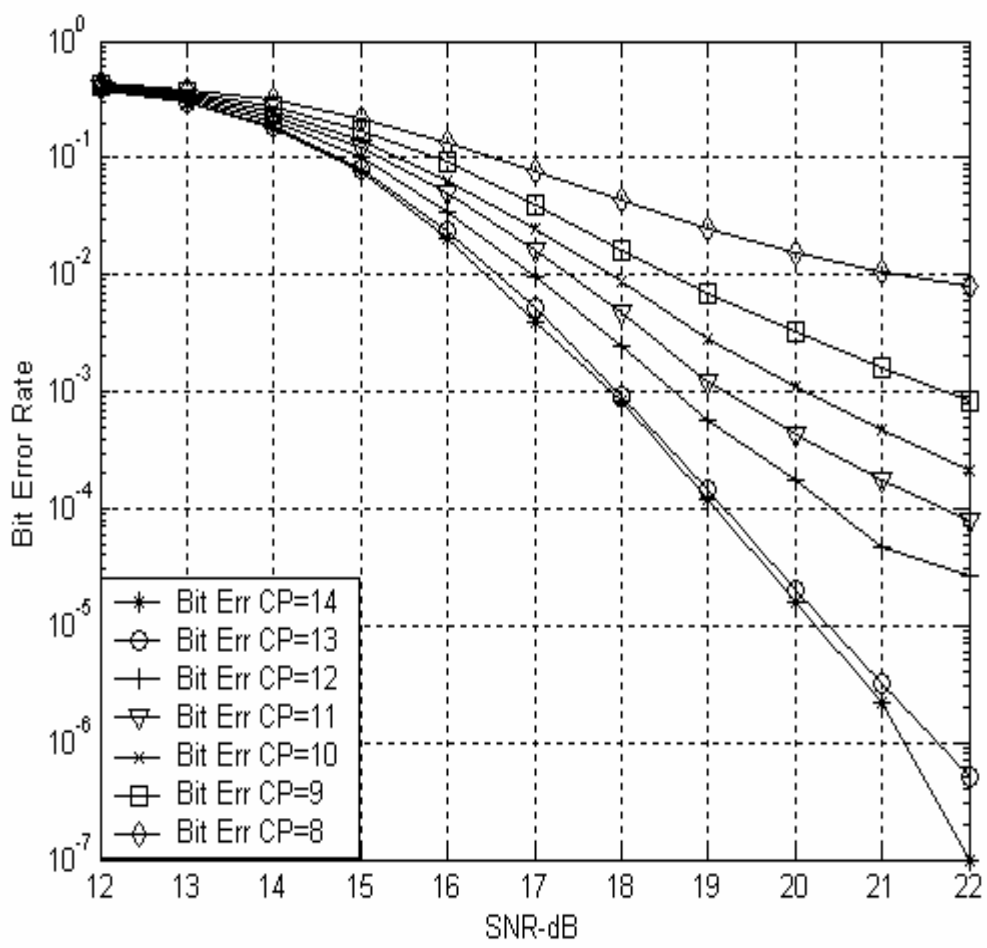


Fig. 4.3 System performances with different length of cyclic prefix

With the channel model in Fig. 4.2-(a), Fig. 4.4 illustrates the effect of TEQ on the system performance. The top line shows the bit error rate of the system without using TEQ. It can be seen that if the maximum channel delay spread exceeds the guard interval and if TEQ is not performed and without the extension of the guard interval, the system performance can not be improved by simply increasing the SNR. Increasing the

SNR from 12dB to 22dB, the bit error rate is only decreased to an order of 10^{-2}. This is because the existence of the ISI/ICI causes the error floor.

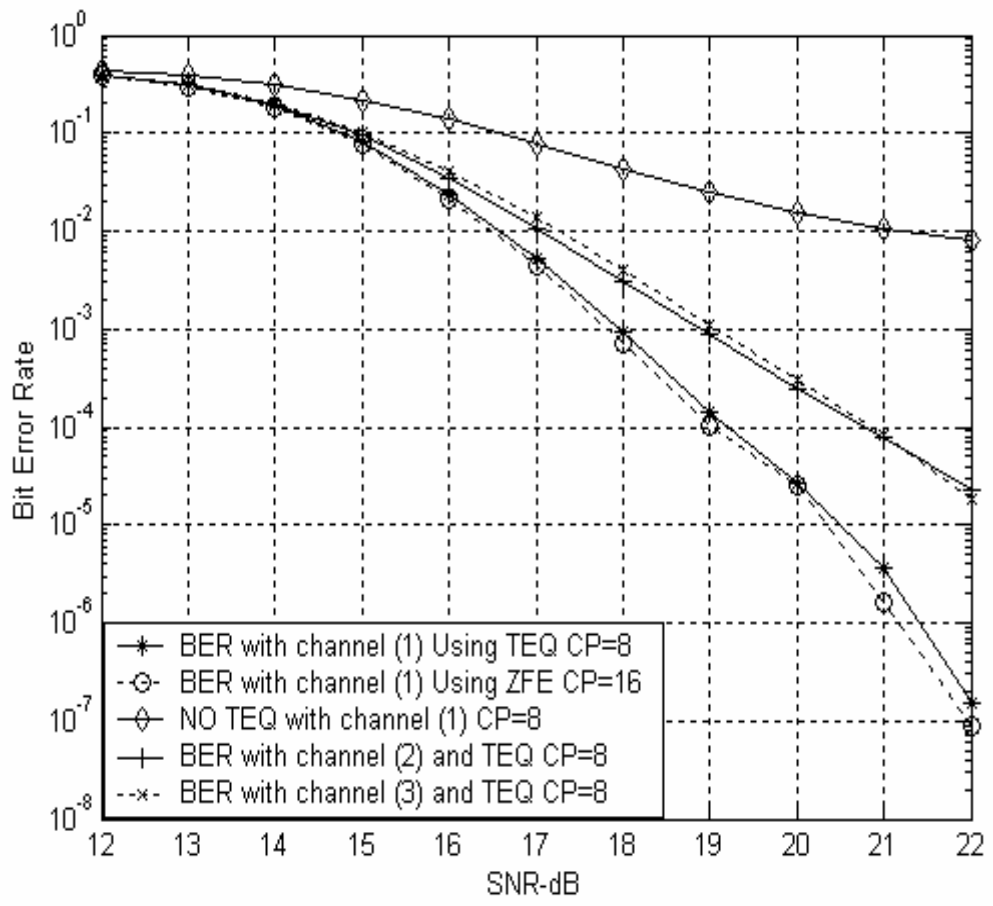


Fig. 4.4 Simulation results for system BER performance

The lowest curve is the system performance for the case that the length of cyclic prefix is extended to 16 samples and the ZFE is used for channel compensation. It can also be seen that if a pre-FFT TEQ is inserted to shorten the effective channel impulse response to be 9 samples, and with the use of ZFE, the system performance is improved significantly. This improvement makes the performance approach the system with cyclic prefix equal to 16 samples and compensated with ZFE method. Thus we can obtain higher transmission efficiency and a reasonable system performance at the same time.

The system performances with TEQ using two other channel models, $\vec{h}_2$ and $\vec{h}_3$, are also shown in Fig. 4.4. The impulse responses of these two channels are as follows:

$$\vec{h}_2 = 0.85\delta(k) + 0.402\delta(k-6) + 0.19\delta(k-14) + 0.09\delta(k-20) + 0.042\delta(k-25)$$

and

$$\vec{h}_3 = 0.881\delta(k) + 0.416\delta(k-6) + 0.197\delta(k-11) + 0.093\delta(k-15) + 0.044\delta(k-18) + 0.021\delta(k-22) + 0.01\delta(k-30) + 0.005\delta(k-37)$$

The maximum channel delay spreads are 25 and 37 respectively and the path gains follow an exponential decaying distribution. As can be expected and observed, the system performance degrades when the maximum delay spread increases. However the performance with TEQ is still acceptable at the high SNR values.

Figure 4.5 shows the simulation results with different orders of FIR filter to realize the TEQ with channel model $\vec{h}_1$ shown in Fig. 4.2-(a). It can be observed that even with a small order of FIR filter (p=3), the ISI and ICI can be significantly reduced and gain is very high compared to the case when the channel impulse response is longer than the guard interval and neither extension of the guard interval nor TEQ is employed. When the order of TEQ increases to 7, the equalizer already achieves the most performance gain. As the order increases to 17 or 31 the OFDM system reaches the highest performance gain. When the order is 45 or over, the gain due to the application of the TEQ becomes saturated. If the order of TEQ is even higher, no further performance gain can be achieved, but the performance starts to deteriorate. This is due to the fact that when the effective channel impulse response is longer than 64 samples, to realize the ZFE to compensate the effective channel, truncation is required for the 64-point FFT.

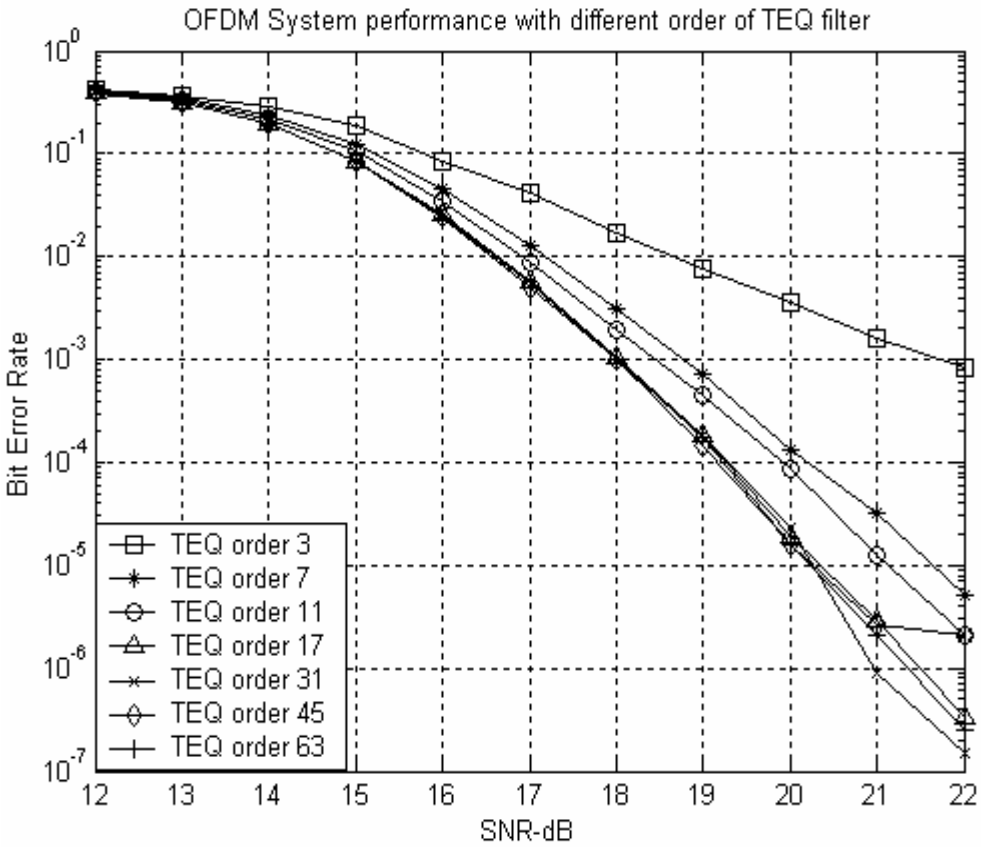


Fig. 4.5 Simulation results with different orders of the FIR to implement the TEQ

The optimum TEQ FIR order depends on the channel impulse response and the required BER level. Based on the results tested with all three channel models and considering the computational complexity to find the coefficients of TEQ and the difficulty of hardware implementation of the FIR filter, a reasonable order of the FIR filter can be selected to be between 7 and less than 31 to achieve the most performance improvement.

The above simulation results assume a perfect knowledge of the channel state information at the receiver. Such assumption is not the case in the real system. The realistic channel impulse response needs to be estimated. Using the channel estimation method in [13], the influence of the channel estimation error on the system performance

was also investigated. Using the channel model $\vec{h}_3$ and a 15-order TEQ, the test result is shown in Fig. 4.6. The system performance with an ideal knowledge of the channel impulse response is also provided for comparison. It can be seen that there is a very small performance loss due to the imperfect channel state information. This degradation is less than 1dB compared to the case of assuming ideal channel state information.

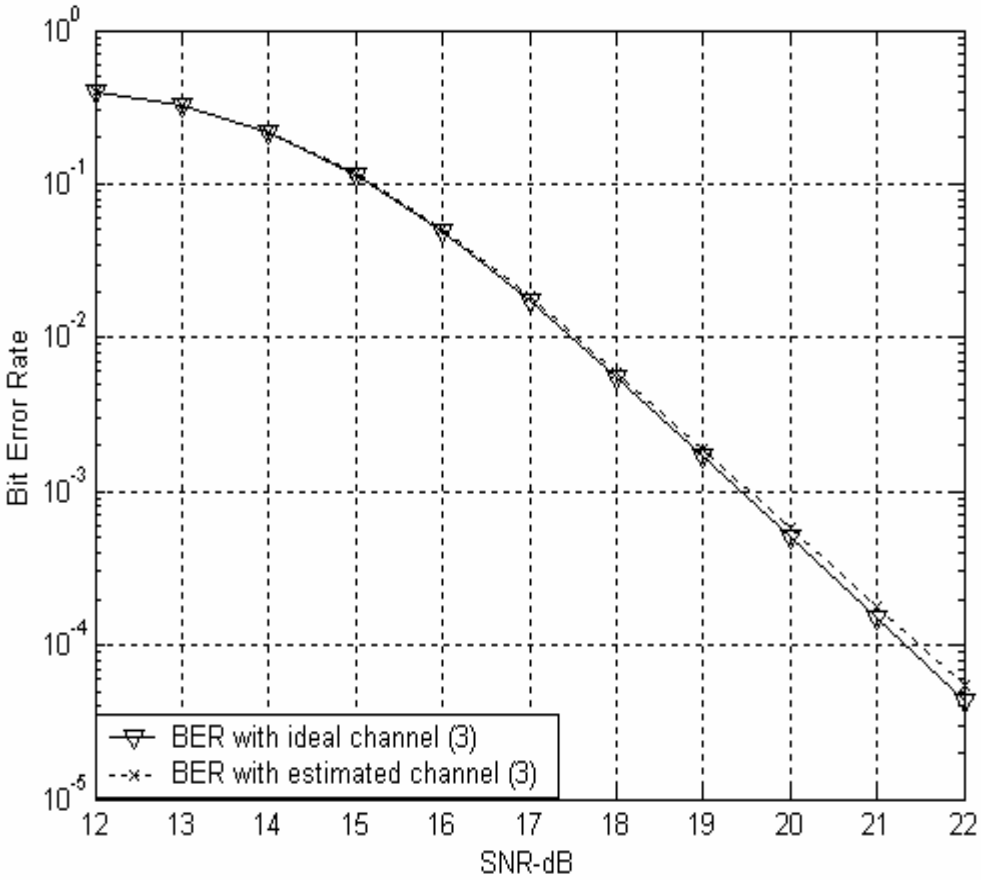


Fig. 4.6 Bit error rate with estimated and ideal channels

4.3 Summary

In this chapter, the simulation results were provided to verify the functionality of the proposed algorithm. When the maximum channel delay spread is very long, the proposed TEQ can be used to shorten the effective channel impulse response. The high

system performance and transmission efficiency can be obtained at the same time with a moderate order of TEQ filter.

Chapter 5 Hardware Design Model for Algorithm Implementation

In this chapter, following the discussion in the previous chapters, a hardware design model is presented. The detailed function blocks to solve the optimum TEQ coefficients are discussed. After the implementation of the TEQ FIR filter, a zero forcing equalization function block is designed to compensate the effective channel impulse response. To speed up the calculation, a 64-point radix-4 multi-path pipeline FFT is employed as the fundamental structure to be implemented in FPGA. The algorithm is also implemented in DSP to show the advantage in the running speed of FPGA implementation.

5.1 Introduction

Today various communication standards such as WLAN, digital television (DTV), cable modem and wideband code division multiple access (WCDMA) have been rapidly developed. Custom application specific integrated circuits (ASIC) have been implemented to reduce their cost, size and power consumption and to increase the running speed. However the time cycle of design and redesign, if there are any modifications, is comparatively long and expensive. In addition, due to the competitive

pressure from the market, ASIC-based solutions may be inadequate for adopting various evolutions of the communication standards. On the other hand, with the constant advances in very large scale integration (VLSI) technologies and architecture design, field programmable gate array (FPGA) has experienced extensive architectural innovations in the past several years. Advanced process technology has enabled the development of high-density devices, which can have a multi-millions of gates system on a programmable chip (SOPC) computing platforms, from originally serving as simple platforms for small ASIC prototyping to glue logic implementation.

New generations of FPGA have made it possible to integrate a large number of computation resources, such as embedded microprocessors, optimized digital signal processing (DSP) blocks and high capacities of logic elements on one silicon die. New FPGAs also include the embedded memory for on-chip storage, fast routing matrices which can be used to implement many complex applications. One of such devices is the Altera FPGA Stratix. This device family is based on the 1.5V, 0.13µm, all-layer-copper process CMOS technology. The device consists of up to 79,040 logic elements (LEs), 7 Mbits of embedded on-chip memory, DSP blocks, high-performance I/O capabilities and other advanced features. It can be used at the heart of these complex high-bandwidth systems to accelerate system performance and to enable new functionalities. Redesign or modification of the original system is very quick and easy if it is implemented in FPGA. Furthermore, new technology like Altera HardCopy devices provides a low-risk, cost-effective, and time-saving alternative to ASICs for high-volume production. The maximum running speed of FPGA system also increases tremendously in the past few years. They become extremely suitable to the needs of high-performance real-time signal processing. The research motivation in this project is to develop a cost-effective,

high performance and flexible TEQ for the high rate IEEE 802.11a OFDM system using the state-of-art technique. FPGA is preferred for hardware implementation and the Altera Stratix is chosen.

Channel equalization in high data rate WLAN systems is among the most arithmetically demanding tasks. According to the algorithm discussed in Chapter 3, the implementation of the algorithm in hardware involves the design of a fixed-point matrix multiplication, the LDL^T & LU decomposition, the finite impulse response (FIR) filter and the high-speed fast Fourier transform (FFT). All these functions are computational intensive and the system should operate at a data rate of 54Mbps, therefore a high efficient system design is essential.

Matrix multiplication is computationally intensive operation. The multiplication of two matrices with dimensions $(n \times m)$ and $(m \times n)$ involves $n(n \times m)$ multiplications. There are many algorithms and hardware architectures designed to improve the matrix multiplication calculation. As it is proved in Chapter 3, the matrix *A* defined in Equation (3.41) is symmetric and positive definite; thus it is not necessary to calculate all the elements in this matrix. Furthermore, certain relationships exist among these elements that make the reuse of the available values possible. The exploitation of these characteristics reduces the computational intension and makes the calculation of the matrix *A* efficiently. This will be further discussed in this chapter.

Solving a linear system equation is very common in various applications. As discussed in Chapter 3, instead of directly finding the inversion of the matrix, many algorithms have been developed to reduce computational complexity, such as Cholesky decomposition, *LU* decomposition and QR factorization, eigenvalue and eigenvector

methods. The Cholesky decomposition involves computing the square roots of the diagonal elements, which limits its use in hardware implementation. QR factorization involves iterative process in finding the result. Eigenvalue and eigenvector solution involves inverse power method and seems comparatively more computational intensive. The LDL^T decomposition and LU [18] substitutions are preferred in this hardware implementation to solve such a problem.

As already discussed, the system matrix is symmetric and positive definite, the LDL^T decomposition is exploited to factorize the matrix, which is only half computational complexity of the LU decomposition. To further increase the speed, the forward and backward substitutions of LU decomposition are used instead of the substitutions of the LDL^T decomposition to solve the linear equation.

FFT and IFFT are the fundamental applications in digital signal processing; high speed FFT and IFFT are the critical parts of the OFDM system. As it is known that the IEEE 802.11a system can run up to at a 54Mbps data transmission rate, the OFDM symbol has a very short interval (4.0μs). To design a FFT for the zero forcing equalization in FPGA, it is required to achieve a high speed real-time processing. According to the system architecture and the parameters defined in the IEEE 802.11a, the transform operation of the FFT should be accomplished in the order of microsecond.

As radix-4 is the most computational efficient for the implementation of 64-point FFT, it is preferred in the FFT hardware implementation. To speed up the calculation, the pipeline parallelism structure is employed. The pipeline radix-4 FFT has been designed and implemented. After the equalizer coefficients are obtained, the pipeline FFT transforms the vectors of the FIR coefficients and the original channel impulse

response to frequency domain for the zero forcing equalization.

5.2 Block Diagram of TEQ Hardware Implementation

Based on the above discussion, given the values of the original channel impulse response $\vec{h}$, and the order *p* of the TIR FIR filter, the optimum TEQ coefficients $\vec{w}_{opt}$ can be calculated by Equations (3.49) to (3.51). They are repeated here for convenience.

$$A\vec{w} = B \tag{5.1}$$

with $A = H_{re}^{*} \cdot H_{re} + \gamma^2 \cdot I$ and $B = H_{re}^{*} \cdot \vec{d}$.

From the discussions in Chapter 3, the block diagram of hardware implementation is presented in Fig. 5.1.

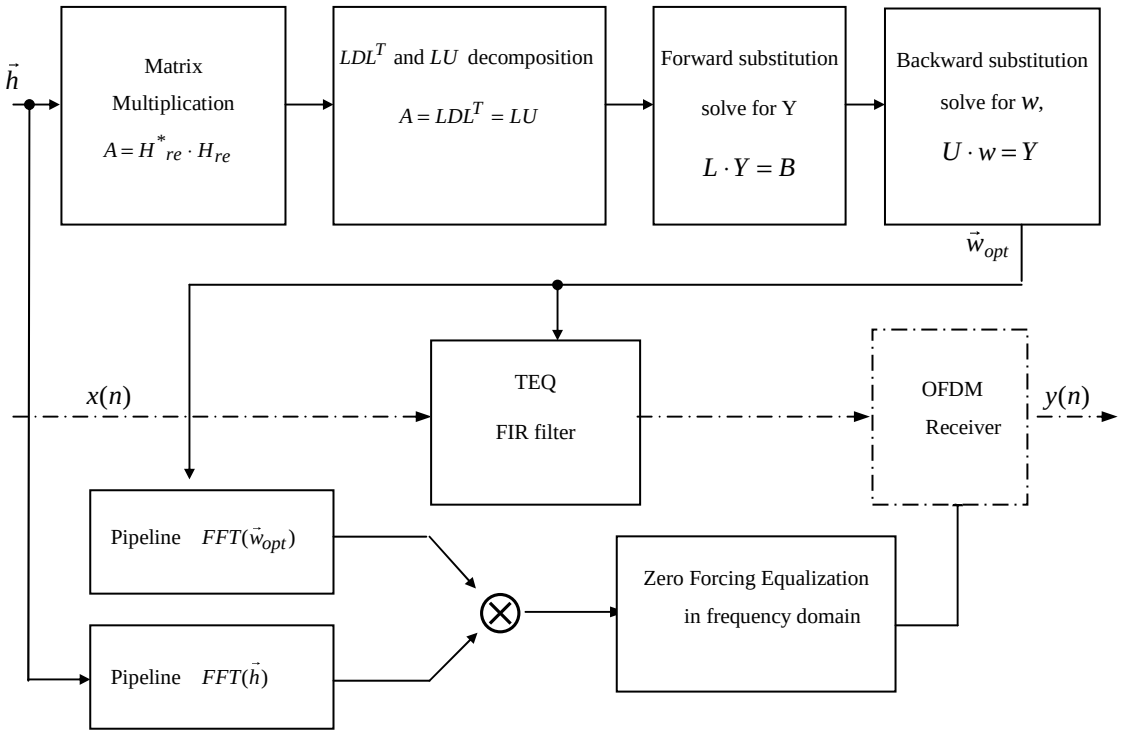


Fig. 5.1 A block diagram of the TEQ hardware design.

The hardware design flow chart can be clearly observed from the diagram. Given the

channel impulse response $\vec{h}$ and the order of the TEQ as the input parameters, the reduced convolution matrix can be obtained; a matrix multiplication function block is designed to calculate *A*. The LDL^T decomposition is designed to factorize *A*. After decomposition the forward substitution and backward substitution of *LU* decomposition are designed accordingly to find the optimum coefficients $\vec{w}_{opt}$. The TEQ FIR filter is implemented using these coefficients.

The next task is to design the zero forcing equalization to compensate the effective channel in frequency domain. First, the vector of $\vec{h}$ and $\vec{w}_{opt}$ are transferred to frequency domain by the 64-point radix-4 pipeline FFT. Then calculate the multiplication $FFT(\vec{h}) \cdot FFT(\vec{w}_{opt})$ and find the inversion of the multiplication result. Lastly at the receiver multiplies each sub-carrier with the individual element of the inversion of the multiplication. As the attenuation of each path is normalized and its values is considered to be less than 1, so the fractional number multiplication and division will be involved in the implementation.

In hardware implementation, considering the tradeoff between accuracy and operation speed, it is decided that the input channel coefficients and the output matrix are all represented by 16-bits, which includes 3-bit integer and 13-bit fraction. The arrangement can be changed under different requirements. The fix point representation is employed in the implementation and the fix point multiplier and divider are designed as the fundamental circuits.

5.2.1 Matrix Multiplication

It was already shown that the matrix *A* is symmetric and positive definite. When

calculating the elements of the matrix it is efficient to make use of these properties to further reduce the computational complexity.

The detailed process of calculating the matrix *A* in hardware is demonstrated. It is assumed that the physical channel coefficient vector is given by:

$$\vec{h} = [h_0\ h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7\ h_8\ h_9\ h_{10}\ h_{11}\ h_{12}\ h_{13}\ h_{14}]$$

The maximum delay spread is 15 items and the order of the TEQ FIR filter is p=7, the length of the cyclic prefix is 8 samples and the TEQ is used to truncate the effective channel impulse response to be less than 9 samples. The reduced convolution matrix H_{re}^{*} and H_{re} can be generated accordingly and the multiplication result of $A = H_{re}^{*} \cdot H_{re} + \gamma^2 \cdot I$ is a (p+1) x (p+1) matrix. As *A* is symmetric, one has

$$a_{ij} = a_{ji}, \qquad \text{for} \quad i \neq j \tag{5.2}$$

Thus only the diagonal elements and upper (or lower) elements of the matrix need to be calculated. The equation to calculate the diagonal elements of the matrix *A* can be expressed as follows:

$$\begin{cases} a_{11} = h_0^2 + h_9^2 + h_{10}^2 + h_{11}^2 + h_{12}^2 + h_{13}^2 + h_{14}^2 \\ a_{22} = h_8^2 + h_9^2 + h_{10}^2 + h_{11}^2 + h_{12}^2 + h_{13}^2 + h_{14}^2 \\ \vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ a_{88} = h_2^2 + h_3^2 + \cdots + h_{14}^2 \end{cases} \tag{5.3}$$

From the above equation, it is seen that the result of the last calculation can be exploited to calculate the next element. For a_{11}, it can be separated into two parts: one is h_0^2 and the other is $h_{9-14}^2 = h_9^2 + h_{10}^2 + h_{11}^2 + h_{12}^2 + h_{13}^2 + h_{14}^2$. The second part can be used to calculate the value of a_{22} as follows: $a_{22} = h_8^2 + h_{9-14}^2$, then a_{22} can be used to calculate $a_{33} = h_7^2 + a_{22}$, and so on. Same as above, the equation to calculate the elements a_{12} to

a_{78} can be written as follows:

$$\begin{cases} a_{12} = h_9 h_8 + h_{10} h_9 + h_{11} h_{10} + h_{12} h_{11} + h_{13} h_{12} + h_{14} h_{13} \\ a_{23} = h_8 h_7 + h_9 h_8 + h_{10} h_9 + h_{11} h_{10} + h_{12} h_{11} + h_{13} h_{12} + h_{14} h_{13} = h_8 h_7 + a_{12} \\ a_{34} = h_7 h_6 + h_8 h_7 + h_9 h_8 + h_{10} h_9 + h_{11} h_{10} + h_{12} h_{11} + h_{13} h_{12} + h_{14} h_{13} = h_7 h_6 + a_{23} \\ \vdots \qquad \vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ a_{78} = h_3 h_2 + a_{67} \end{cases} \tag{5.4}$$

The element a_{12} can be exploited to calculate a_{23}, and so on. It is observed that this property exists among other elements of A. In general this rule can be written as:

$$a_{(i+1)(j+1)} = \Delta + a_{ij},\ 1 < i < j <= 7 \quad i \neq j \tag{5.5}$$

$$a_{(i+1)(i+1)} = \Delta + a_{ii},\ 1 < i <= 7$$

where Δ is the increment.

It requires only one more multiplication to calculate the value of Δ and one more addition to get the value of the corresponding element in the second row, and so on. The values of the elements in other rows can be calculated easily. This property greatly reduces the computational complexity of matrix multiplication, from $O(n^3)$ to $O(n^2)$, therefore decreases the requirements of the hardware implementation. The registers required to store the intermediate results are also reduced accordingly.

5.2.2 Decomposition and Substitution Function Block

After implementation of the matrix multiplication, the LDL^T decomposition module is implemented to factorize *A*, and the forward and backward substitution modules of *LU* decomposition are used to find the optimum coefficients $\vec{w}_{opt}$ as shown in Fig. 5.1. These circuits can be designed efficiently in FPGA according to the equations (3.54), (3.58) and (3.59).

5.2.3 TEQ FIR Filter

After finding the optimum TEQ coefficients vector of $\vec{w}_{opt}$, a FIR filter can be implemented accordingly. The basic structure of a FIR filter operation can be represented by the following equation:

$$y(n) = \sum_{i=0}^{p} x(n-i)w(i) \tag{5.6}$$

Here $x(n)$ represents the input sequence, $w(n)$ represents the TEQ coefficients and $(p+1)$ is the number of taps. The FIR is constructed by the multipliers and unit-time delays as shown in Fig. 5.2. In hardware implementation, the unit time delay can be realized by a shift register.

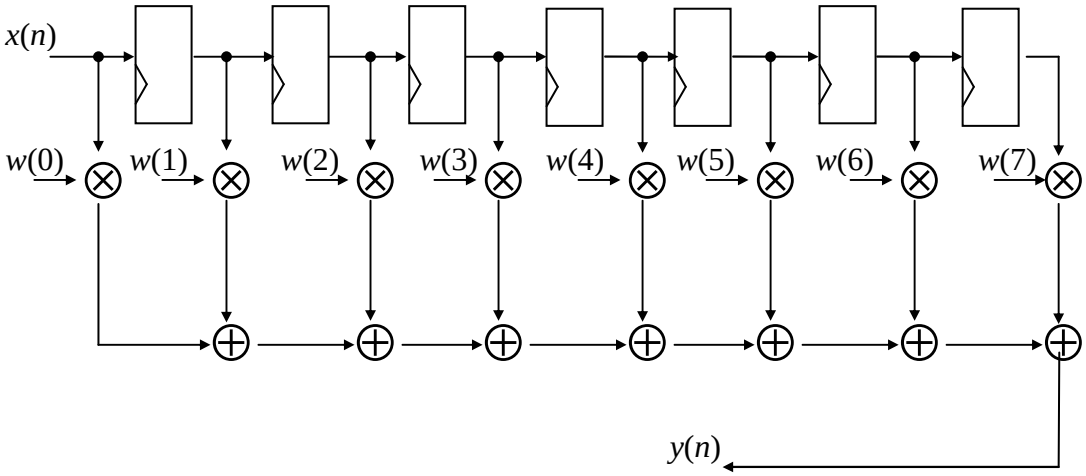


Fig. 5.2 Structure of the FIR filter

5.3 Radix-4 Pipeline FFT Implementation

In the IEEE 802.11a system, besides the use of the IFFT/FFT to realize the OFDM modulation and demodulation, another FFT function block is also used to realize the zero forcing equalization. It is the most critical feature of the transceiver of OFDM

system. The IEEE 802.11a OFDM system is different from the lower rate OFDM systems such as ADSL application by the speed limitation and no initialization at the beginning of the communication process. The FFT/IFFT operation needs to be accomplished in a very short interval. For the IEEE802.11a system, an OFDM symbol interval lasts 3.2μs and the guard interval extends 0.8μs. In order to escape the necessity of too much buffer or un-buffer mechanism, it is desired that the operation be finished in less than 4.0μs. It is imperative to design a high performance FFT to satisfy the timing requirements.

5.3.1 An Introduction of Radix-4 FFT Algorithm

As discussed in Chapter 2, since the major breakthrough of Cooley-Tukey radix-2 algorithm [19] in the 60's, a large number of FFT algorithms have been developed for the efficient computation of the DFT. Research also leads to fast Hartley transform (FHT) [20] and the split radix FFT (SRFFT) [21]. Recently there are some algorithms on the quick Fourier transform (QFT) [22] and the decimation-in-time-frequency (DITF) [23]. Efficient DFT algorithms suitable for hardware implementation are required in many real-time applications, however because of the high computational complexity, not all of the algorithms are suitable to be used in high rate systems.

To realize the FFT/IFFT, radix-2, radix-4, radix-8 or split radix can be used in the evaluation of the DFT. But for practical and theoretical interests, which radix to be used should be studied under specific conditions. In the IEEE 802.11a system, the size of the FFT is 64-point, thus the more computationally efficient radix-4 scheme is preferred, the number of complex multiplications for 64-point FFT by radix-2, radix-4 and radix-8 is calculated as the follows:

1) For the radix-2 decimation-in-time (DIT) FFT, the first stage consists of a multiplication by $W^0 = 1$, so actually no multiplication is required when multiplied by W^0. The second stage has twiddles $W^0 = 1$ and $W^{16} = j$, so again no multiplications is necessary when multiplied by the two twiddles. The third stage has one-half of the full complement of (*N/2*) complex multiplications and so on. The equation is written as

$$M_N = \frac{N}{2} \sum_{n=1}^{\log_2 N - 2} \frac{2^n - 1}{2^n} = \frac{N}{2}(\log_2 N - 2) - \frac{N}{2}[\frac{1 - 2^{-(\log_2 N - 1)}}{1 - 2^{-1}}] + 1 \tag{5.7}$$

$$M_N = \frac{N}{2}\log_2 N - \frac{3}{2}N + 2 \tag{5.8}$$

For *N*=64, the number is $M_N = 98$.

2) For radix-4 and radix-8, the number of complex multiplication calculated is 76 and 80 respectively [3]. For the 64-point FFT, the radix-4 seems to be the optimum one if the criterion rule is the number of complex multiplications. The radix-4 FFT is also faster than the split radix FFT and the discussion on radix-4 FFT algorithm is provided here.

5.3.2 Radix-4 FFT Algorithm

In radix-4 FFT [3], the number of samples *N* is the power of 4, i.e., $N = 4^p$ and *p* is a positive integer. The *N*-point FFT is decimated every 4 samples rather than every second sample as in the radix-2. The input sequence is divided into 4 *N*/4 sub-sequences. These sub-sequences are then continuously decimated until the DFT size of the sub-sequences is reduced to 4-point. The 4-point DFT is called radix-4 butterfly, or dragonfly as referred in some texts. Similar to the radix-2 FFT algorithm, the radix-4 FFT can be derived by the DIT or the DIF method.

For the *N*-point radix-4 DIT DFT algorithm, it subdivides the input sequence into

four subsequences in time domain, i.e., $x(4n)$, $x(4n+1)$, $x(4n+2)$, $x(4n+3)$, $n = 0, 1, ..., (N/4-1)$. The algorithm can be written as the following equation [3]:

$$X(p,q) = \sum_{l=0}^{3} [W_N{}^{lq} F(l,q)] W_4{}^{lp} \tag{5.9}$$

where $$F(l,q) = \sum_{m=0}^{N/4-1} x(l,m) W_{N/4}^{mq} \tag{5.10}$$

$p = 0,1,2,3$, $l = 0,1,2,3$, $q = 0,1,2,...,(N/4-1)$ and

$$x(l,m) = x(4m+1) \tag{5.11}$$

$$X(p,q) = X(\frac{N}{4} p + q) \tag{5.12}$$

The four *N/4*-point subsequences DFT $F(l,q)$ obtained from the above equation are combined according to equation (5.9) to yield the *N*-point DFT. If Equation (5.9) is expressed in matrix form, then the radix-4 FFT can be written as the follows:

$$\begin{bmatrix} X(0,q) \\ X(1,q) \\ X(2,q) \\ X(3,q) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} W_N^0 F(0,q) \\ W_N^q F(1,q) \\ W_N^{2q} F(2,q) \\ W_N^{3q} F(3,q) \end{bmatrix} \tag{5.13}$$

Since $W_N^0 = 1$, only the W_N^q, W_N^{2q} and W_N^{3q} involve complex multiplications, and the multiplication with –j is just the swapping between the real and imaginary parts of a complex number. Each dragonfly involves 3 complex number multiplications and 12 complex number additions, and it can be reduced to 8 complex number additions by a proper arrangement of the samples. The algorithm is graphically illustrated in Fig. 5.3. The decimation process continues until the DFT size of 4-point which can be calculated directly. In the radix-4 FFT algorithm, there also exists the digital reverse problem required to be solved as in the radix-2 algorithm, so an input data shuffler or an output

data shuffler needs to be included in the implementation.

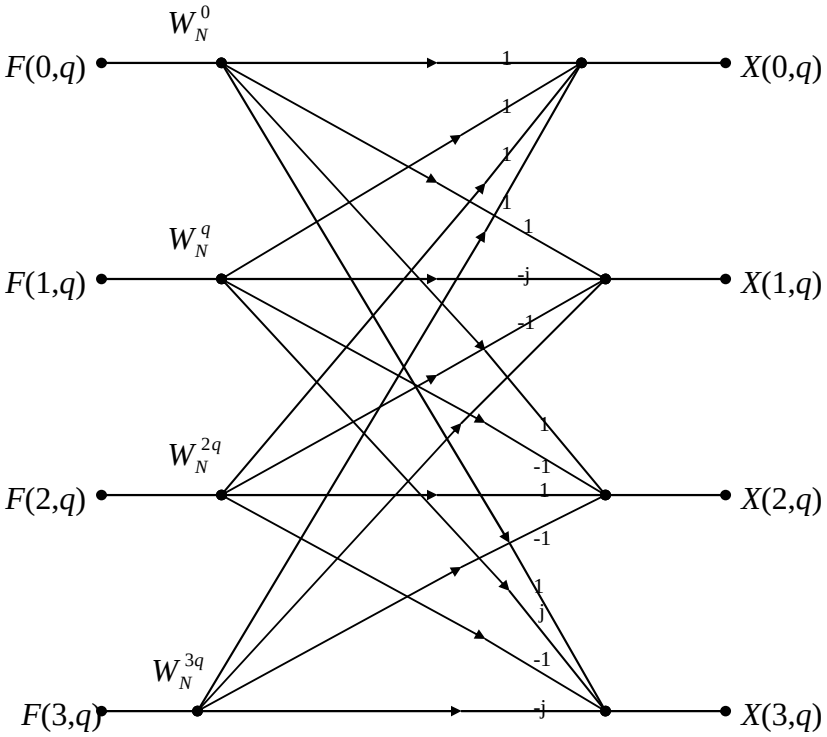


Fig. 5.3 Basic dragonfly computation in a radix 4 FFT

If decimating the input sequence in frequency domain, an *N*-point radix-4 DIF DFT algorithm can be derived as the followings [3]:

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk} \tag{5.14}$$

$$=\sum_{n=0}^{N/4-1}x(n)W_N^{nk}+\sum_{n=N/4}^{N/2-1}x(n)W_N^{nk}+\sum_{n=N/2}^{3N/4-1}x(n)W_N^{nk}+\sum_{n=3N/4}^{N-1}x(n)W_N^{nk}$$

$$=\sum_{n=0}^{N/4-1}x(n)W_N^{nk}+W_N^{Nk/4}\sum_{n=0}^{N/4-1}x(n+\frac{N}{4})W_N^{nk}+W_N^{Nk/2}\sum_{n=0}^{N/4-1}x(n+\frac{N}{2})W_N^{nk}+W_N^{3Nk/4}\sum_{n=0}^{N/4-1}x(n+\frac{3N}{4})W_N^{nk}$$

From the definition of the twiddle factors, one obtains get the following expressions:

$$W_N^{\,0}=1,\ W_N^{\,Nk/4}=(-j)^k,\ W_N^{\,Nk/2}=(-1)^k,\ W_N^{\,3Nk/4}=(j)^k,\ W_N^{\,4nk/4}=W_{N/4}^{\,nk}$$

Equation (5.14) can be rewritten as:

$$X(k)=\sum_{n=0}^{N/4-1}[x(n)+(-j)^k x(n+\frac{N}{4})+(-1)^k x(n+\frac{N}{2})+(j)^k x(n+\frac{3N}{4})]W_N^{nk} \tag{5.15}$$

Further on subdivides the *N*-point DFT sequence into 4 *N/4*-point subsequences, $X(4k)$, $X(4k+1)$, $X(4k+2)$, $X(4k+3)$, $k=0,1,...,N/4$, the radix-4 DIF FFT can be expressed as:

$$X(4k)=\sum_{n=0}^{N/4-1}[x(n)+x(n+\frac{N}{4})+x(n+\frac{N}{2})+x(n+\frac{3N}{4})]W_N^{0}W_{N/4}^{nk} \tag{5.16}$$

$$X(4k+1)=\sum_{n=0}^{N/4-1}[x(n)-jx(n+\frac{N}{4})-x(n+\frac{N}{2})+jx(n+\frac{3N}{4})]W_N^{n}W_{N/4}^{nk}$$

$$X(4k+2)=\sum_{n=0}^{N/4-1}[x(n)-x(n+\frac{N}{4})+x(n+\frac{N}{2})-x(n+\frac{3N}{4})]W_N^{2n}W_{N/4}^{nk}$$

$$X(4k+3)=\sum_{n=0}^{N/4-1}[x(n)+jx(n+\frac{N}{4})-x(n+\frac{N}{2})-jx(n+\frac{3N}{4})]W_N^{3n}W_{N/4}^{nk}$$

It can be observed that each N/4-point DFT is a linear operation of the 4 signal samples scaled by a twiddle factor. The decimation continues to the DFT size of 4-point as in the radix-4 DIT algorithm, the decimation is repeated $p=\log_4 N$ times.

5.3.3 Radix-4 Pipeline FFT

There are different architectures of the FFT/IFFT processors, such as column FFT, fully parallel FFT and pipeline FFT. Just as other computer algorithms, fully parallel processing structure can be employed to realize faster FFT evaluation, but the hardware implementations of the fully parallel FFT are too intensive and expensive. Among the parallel architectures, the pipeline FFT is an interesting special purpose structure. It is very suitable for high-speed real time applications since it can easily operate on a clock frequency close or equivalent to the sampling frequency, which will lower the system

complexity. The pipeline FFT is characterized by continuously processing the input data sequences and it is easy to realize the IFFT on the same core architecture with only minimal modification.

The design of the pipeline FFT processor had been studied since 1970's [3]. It contains an amount of parallelism equal to $\log_r N$, where *r* is the FFT radix. There will be $\log_r N$ separate hardware butterfly computations processing in parallel. Some of the popular types of pipeline FFT are:

1) Radix-2 multi-path delay commutator (R2MDC) [3] is the most classical method to implement the radix-2 pipeline FFT. In each stage, the input data sequence is divided into two parallel sub-sequences with proper delays to switch the data sequence to the desired order. It requires $((\log_2(N)-2))$ numbers of multipliers, $\log_2 N$ numbers of butterflies and $(\frac{3}{2}N-2)$ registers as the delay elements. The disadvantage is that both butterflies and multipliers are in 50% utilization.

2) Radix-2 single-path delay feedback (R2SDF) can improve the utilization of the registers more efficiently, to 100%. It requires the same number of butterflies and multipliers as R2MDC, but it requires only (*N*-1) registers.

3) Radix-4 single path delay feedback (R4SDF) is the radix-4 version of R2SDF. It improves the utilization of the multipliers to 75%, but decreases the utilization of butterflies to only 25%. It requires $((\log_4(N)-1))$ multipliers, $\log_4 N$ radix-4 butterflies and (*N*-1) registers.

4) Radix-4 multi-path delay commutator (R4MDC) [24] is a radix-4 version of R2MDC. Its utilization of multipliers and butterflies is 25%. It requires $3\log_4 N$

multipliers and $\log_4 N$ radix-4 butterflies and $(\frac{5}{2}N-4)$ registers.

5) Radix-4 single-path delay commutator (R4SDC) [25] increases the utilization of the radix-4 butterflies to 75% by modifying the radix-4 butterfly. It reduces the requirement of registers to (2*N*-2), but the butterfly and the delay commutator control mechanism become more complicated.

The architectures of these algorithms all have their own characteristics and can be used in different applications. As discussed above, the radix-4 is more computational efficient, so the radix-4 algorithm is preferred in this implementation. In the IEEE 802.11a OFDM system, at the beginning of a transmitted data packet, the training sequence is sent for the purpose of the channel estimation, frequency synchronization, etc, then the preamble header and data OFDM symbol follow. It is assumed that the channel impulse response will not change dramatically during the transmission interval of the data packet. Generally the channel estimation and the ZFE in frequency domain can be considered necessary to operate only once in the interval of packet transmission, if not considering the fine tuning operation. The utilization of the butterfly, multipliers and registers is not a major problem. And if necessary, the utilization can be improved simply by introducing some buffer scheme or overlapping. Considering the control mechanism of a radix-4 multi-path delay commutator (R4MDC) pipeline FFT is comparatively simple, therefore in this hardware implementation, the R4MDC pipeline FFT is adopted as the fundamental architecture. A block diagram of the 64-point R4MDC pipeline FFT hardware implementation is shown in Fig. 5.4. This FFT structure consists of $\log_r N = \log_4 64 = 3$ stages. Each stage includes:

1) A delay commutator, to rearrange the input data sequences between arithmetic

elements as required by the algorithm. When the sequences pass through the commutator, they are switched to the desired paths.

2) An arithmetic element (AE), which performs the arithmetic functions (complex multiplications and additions) and butterfly.

There is a control mechanism block, which generates the necessary control signals, such as counters to control the time to switch the data sequence, the time to read the twiddle factors and the time to shuffle the output signals.

In this case, the parallel output of the TEQ coefficients, $\vec{w}_{opt}$ and $\vec{h}$, are converted to serial inputs by time delays, zeros are padded for the 64-point FFT input vectors. The delay commutator is used to delay the input stream and then transform the input vector into the desired sub-sequences. In the first stage, the delay commutator distributes the one path input sequence to the four-path output sub-sequences. The first data path does not have any delay, the second path has a delay of 4^{n-i-1}, the third path has a delay of $2\times 4^{n-i-1}$ and the fourth path has a delay of $3\times 4^{n-i-1}$, with i is the stage number and $n=\log_4 N$. After some specific delays, the sub-sequences are fed to the first stage arithmetic element and butterfly (AE1). At the same time, the twiddle factors are feed into the arithmetic element in the correct order to calculate the 4-point DFT. The twiddle factors are calculated in advance, converted to binary format and saved in the read only memory (ROM), it will be read out by the control signal. There are registers required to store the intermediate results.

The other stages are almost the same as the first stage, including the delay commutator, arithmetic element, butterfly, registers and twiddle factor reading. After three stages operation and calculation, the FFT result is available, but the output is still

in digital inverse order, a shuffler is used to switch the output sequence to its normal order.

Most of the complexity is due to the control mechanism for proper delay, switching, shuffling and reading out the twiddle factor. In order to avoid the distribution of the data among different modules, the whole pipeline FFT functionalities are implemented in a single chip. Buffers are introduced to make the chip able to process the input stream continuously. Information about hardware implementation will be discussed in next chapter.

5.4 Summary

In this chapter, the detailed design of the TEQ function block is presented. The regularities and characteristics of the matrix *A* defined in Equation (3.41) are exploited to reduce the computation complexity. To realize the zero forcing equalization, a design of radix-4 pipeline FFT is also implemented in FPGA. All the hardware designs aim at improving the running speed of the high data rate OFDM system.

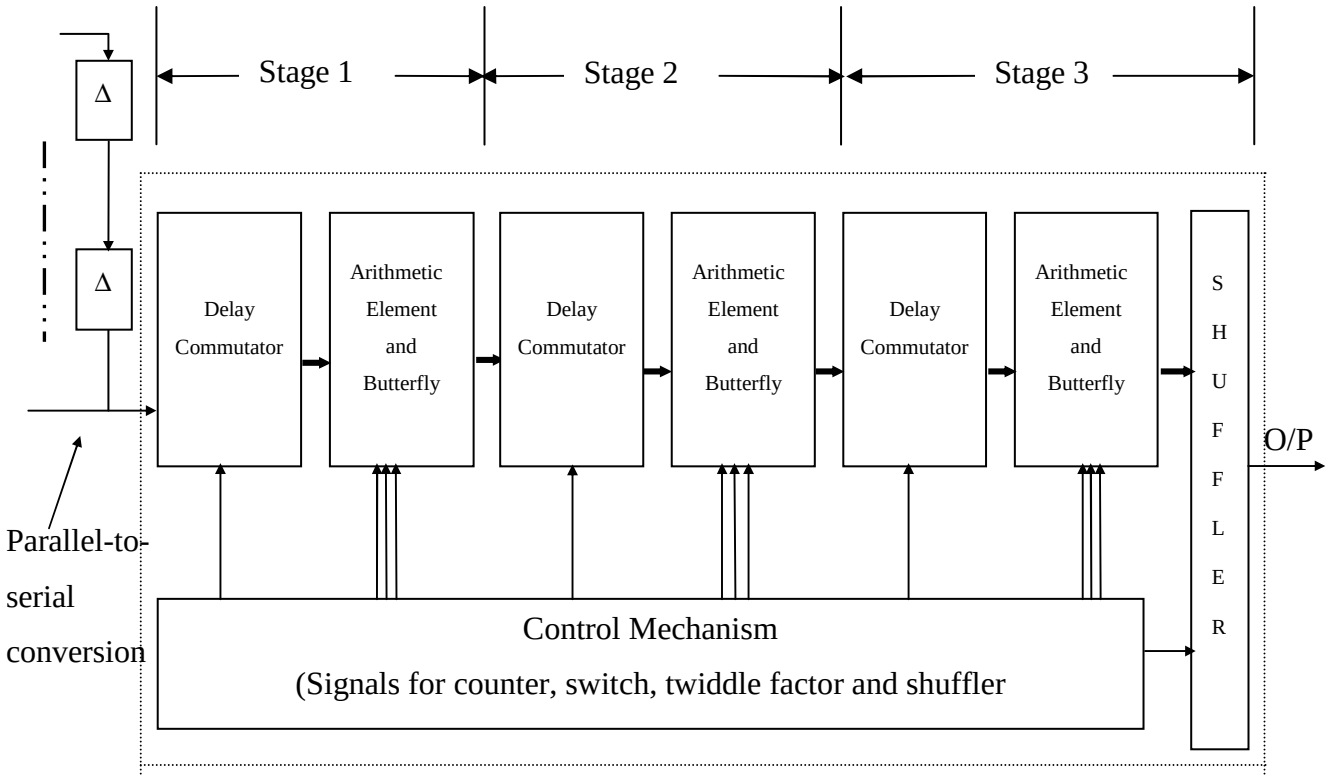


Fig. 5.4 Radix-4 64-point pipeline FFT block diagram

Chapter 6 Information on Hardware Implementation, Simulation and Discussions

In this chapter the detailed information on hardware implementation is provided. An example of hardware simulation is demonstrated to verify the functionalities of the hardware chips. The error introduced by finite binary representation and truncation is analyzed mathematically. The difference between the hardware implementation and the mathematical calculation is also provided. The system performance using the hardware outputs for the algorithm's parameters is tested and the performance loss introduced by hardware implementation is discussed. The algorithm is also implemented in DSP. The information of the DSP solution is provided and the operating speed is tested.

6.1 Information on Hardware Implementation

6.1.1 Initial Hardware Implementation Parameters

Based on the discussions in Chapter 5, the hardware is designed, debugged, synthesized and tested in the Altera's Quartus II®. Quartus II is one of the most efficient, comprehensive software for designing with the programmable logic and the mask-programmed devices. The TEQ function block and the 64-point radix-4 pipeline FFT consist of many logic elements (more than 80% of usage of the logic elements of the Altera Straix EP1S25F780C6). Considering the flexibility, operation speed and to test

the functions of TEQ block, pipeline FFT and FIR filter independently, 3 FPGA chips are implemented. The first chip is to find the TEQ coefficients, the second is to implement the FIR filter accordingly, and the third is to realize the function of the 64-point radix-4 pipeline FFT. The devices family used for the TEQ coefficients and pipeline FFT is the Altera's high density Stratix FPGAs. The device to implement the FIR filter can use some lower level FPGA chips. In this case the FIR filter implementation is compiled and simulated in the APEX20k. In this hardware implementation, it is assumed that the original channel impulse response lasts 15 samples, the order of TEQ FIR filter is 7. The input and output data widths are 16 bits (3-bit integer and 13-bit fraction). The twiddle factors are chosen to be 8 bits (2-bit integer and 6-bit fraction). Two's complement operation is employed for the negative numbers. The brief connection between modules is shown in Fig. 6.1. The data widths of input, output and interconnections are all 16 bits.

After the channel estimation, the channel impulse response is passed to the input pins of the TEQ. The outputs are the optimum TEQ coefficients, which are the input parameters to realize the FIR filter. The original channel vector $\vec{h}$ and optimum TEQ coefficient vector $\vec{w}_{opt}$ are converted from parallel to serial and passed to the 64-point radix-4 pipeline FFT to be transferred to the frequency domain. Then the multiplication of the two output sequences is calculated and the divider is used to find the inversion, the result is used to realize the ZFE to compensate the effective channel.

6.1.2 Detailed Hardware Implementation Information

The detailed information on the hardware implementation of the TEQ FIR filter, the 64-point radix-4 pipeline FFT and the TEQ coefficients solver chips is listed in Tables

6.1, 6.2 and 6.3 respectively.

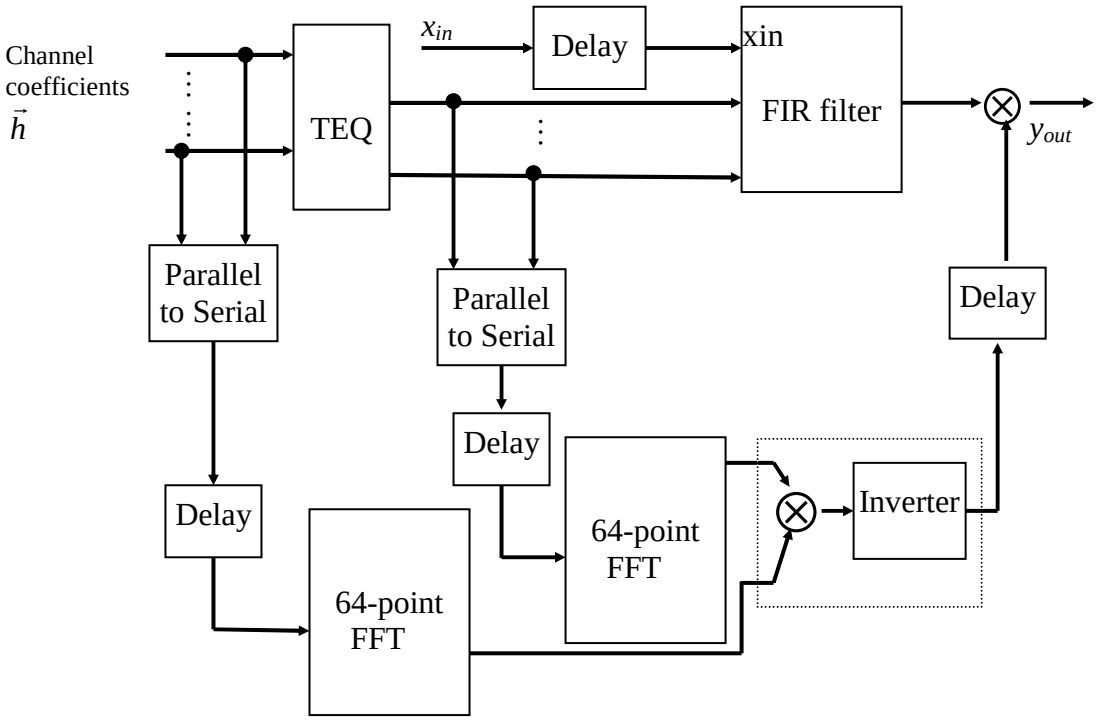


Fig. 6.1 The module connection diagram.

Table 6.1 Hardware implementation information on the TEQ FIR filter

Chip name	fir_filter
FPGA Device	Altera APEX20K EP20K100BC356-1V
Logic elements	3,209/4,160 (77%)
Data width	16-bit, 3-bit integer, 13-bit fraction
Filter Order	7
Total Pins	163
Clock frequency	20MHz

Table 6.2 Hardware implementation information on the 64-point radix-4 pipeline FFT

Chip name	pipelinefft
FPGA Device	Stratix EP1S25F780C6
Logic elements	14,780/18,460 (80%)
FFT Length	64
Radix	4
Data width	16-bit, 3-bit integer, 13-bit fraction
Twiddle factor width	8-bit, 2-bit integer, 6-bit fraction
DSP block 9-bit elements	56
Total Pins	77
Memory (shift registers)	1,792
Internal main clock	20MHz
Clock cycle and output delay	64/82
Transform time interval	3.2μs

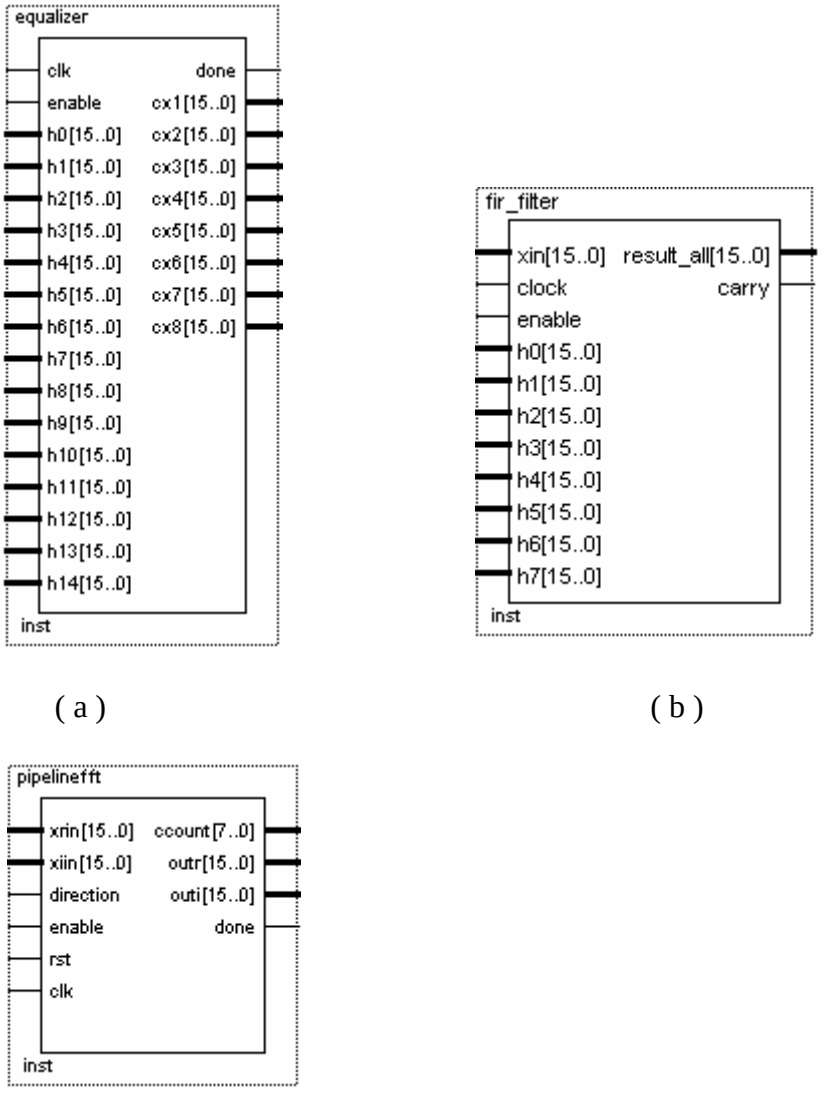
Table 6.3 Hardware implementation information on the TEQ coefficient solver

Chip name	equalizer
FPGA Device	Altera Stratix EP1S25F780C6
Logic elements	15,909/18,460 (86%)
Data width	16-bit, 3-bit integer, 13-bit fraction
TEQ Order	p=7
DSP block 9-bit elements	24
Total Pins	376
Internal main clock	16MHz
Clock cycle	55
Time interval	3.34μs

For the 64-point radix-4 pipeline FFT, it requires 65 clock cycles to finish the operation. Buffer method is employed to enable it to process the input stream continuously. For the TEQ solver, from Table 6.3 it can be seen that it requires about 3.34μs to calculate the optimum TEQ coefficients vector $\vec{w}_{opt}$. This time is less than one OFDM symbol interval (original OFDM symbol + guard interval) of the IEEE 802.11a system. The implementation of the chip to solve the TEQ coefficients involves multiplication and division, when the data width is large, this will decrease the maximum running speed of the chip. In simulation, it was found that the divider limits the highest running speed. The TEQ solver chip can run as high as 16MHz. In future work, an optimized divider shall be implemented. The TEQ FIR filter and the 64-point radix-4 pipeline FFT chips can run at a clock frequency greater than 20MHz. It requires two clocks to satisfy the

requirements of the system to transmit data up to 54Mbps in the IEEE 802.11a system.

The block symbols of these chips are shown in the Fig. 6.2.



(a) (b)

(c)

Fig. 6.2 Block symbols of the chips

Part (a) is the symbol block of the chip to find the optimum TEQ coefficients. The pins, h_0 to $h_{14,}$ are for the inputs of the original channel impulse response. If “enable” signal is high, the chip begins to work. After finishing the calculation, the “done” signal

will be high. Part (b) is for the TEQ FIR filter, the “carry” signal indicates the overflow. Part (c) is for the 64-point radix-4 pipeline FFT, “xrin” and “xiin” are the real and imaginary parts of the complex input signal, “outr” and “outi” are the real and imaginary parts of the FFT output, all are 16 bits width.

6.2 Simulation Results and Discussions.

In hardware implementation, since the numbers are represented by the finite binary bits, some errors will be introduced through the truncation or rounding. As it assumes that the fraction number is 13-bit, therefore the least significant bit (LSB) of the calculation is 2^{-13}. If the actual data value is not a multiplication of *LSB*, there are some errors in the fix point representation.

It is known that if a 16-bit×16-bit, the result is a 32-bit. Truncation is required to shorten the multiplication result to the 16-bit word. This also introduces some errors. If the number is too large, a scale is necessary to be introduced in calculation to avoid overflow. The expectation and variance of error generated by binary truncation can be calculated using the following equation.

$$\mu = \frac{2^N - 1}{2^{N+1}} LSB = \frac{2^{16} - 1}{2^{17}} 2^{-13} \approx 2^{-14} \tag{6.1}$$

$$\sigma^2 = [\frac{(2^N - 1)(2^{N+1} - 1)}{6 \cdot 2^{2N}} - \frac{(2^N - 1)^2}{4 \cdot 2^{2N}}] LSB^2 \approx \frac{1}{12} 2^{-26} \tag{6.2}$$

An example is demonstrated next. Assume the original channel coefficients are given by:

$$\vec{h} = [0.5\ 0.4\ 0.1\ 0.35\ 0.6\ 0.2\ 0.3\ -0.3\ -0.2\ -0.1\ -0.15\ 0.2\ 0.1\ 0.2\ 0.1]$$

These decimal values are first converted to binary format before entering the FPGA

chip. The corresponding binary numbers are shown in Table 6.4; the decimal values represents the converted binary numbers are also shown for comparison.

Table 6.4 Mapping of decimal values of original channel to binary bits

Value of h	Binary	Real Value
0.5	0001000000000000	0.5
0.4	0000110011001100	0.3999
0.1	0000001100110011	0.1
0.35	0000101100110011	0.349975
0.6	0001001100110011	0.599975
0.2	0000011001100110	0.19995
0.3	0000100110011001	0.2999267
-0.3	1111011001100111	-0.299967
-0.2	1111100110011010	-0.19995
-0.1	1111110011001101	-0.1
-0.15	1111101100110100	-0.1499023
0.2	0000011001100110	0.19995
0.1	0000001100110011	0.1
0.2	0000011001100110	0.19995
0.1	0000001100110011	0.1

There are offsets between these two sets of numbers. If more binary bits are used to represent the data value, the smaller would be the value of the *LSB*, the smaller the offset. But increasing the binary bits, it requires more time to complete the

multiplication and division operation, thus decrease the maximum running speed of the system. So there is a tradeoff between the highest running frequency and accuracy. The AWGN noise is also considered in hardware implementation.

The hardware solution of TEQ coefficients $\vec{w}_{opt}$ is shown in Table 6.5 in decimal and binary data formats. The mathematical calculation values are also shown for comparison.

Table 6.5 Hardware and mathematical calculations

TEQ	Binary number	Decimal values (Hardware implementation)	Mathematical calculation values
w_0	0011110010000011	1.89099121093750	1.88905660432045
w_1	1110111100001001	-0.53015136718750	-0.53678706108282
w_2	1110011000101001	-0.80749511718750	-0.79308852895277
w_3	1111011111100001	-0.25378417968750	-0.24499303067784
w_4	0001011111100011	0.74645996093750	0.74253305861786
w_5	1111101000011010	-0.18432617187500	-0.18674375786207
w_6	1111110000100101	-0.12048339843750	-0.11244022219353
w_7	1111001110100001	-0.38659667968750	-0.38361822089975

From table 6.5 it can be seen that there are some differences between the hardware output and the mathematical calculations. The offset is not constant for all elements due to the signed multiplications and divisions. It can be observed that the offset is very small.

6.3 System Performance Test

The TEQ coefficients $\vec{w}_{opt}$ vector and the original channel impulse response $\vec{h}$ are transformed to frequency domain by the 64-point pipeline FFT chip shown in Fig 6.1. The outputs $FFT(\vec{w}_{opt})$ and $FFT(\vec{h})$ are combined for the zero forcing equalization. The inversion of $FFT(\vec{h}_{eff}) = FFT(\vec{h}) \times FFT(\vec{w}_{opt})$ is calculated in hardware. The difference between the calculation of the inversion of $FFT(\vec{h}_{eff})$ in hardware and the mathematical calculation for the 64 sub-carriers is also plotted in Fig. 6.3. Part (a) is the offset of real part and part (b) is the offset of imaginary part. It can be seen that the differences are not always the same among the channels, some sub-channels have large offset, some have smaller, but all are tolerable.

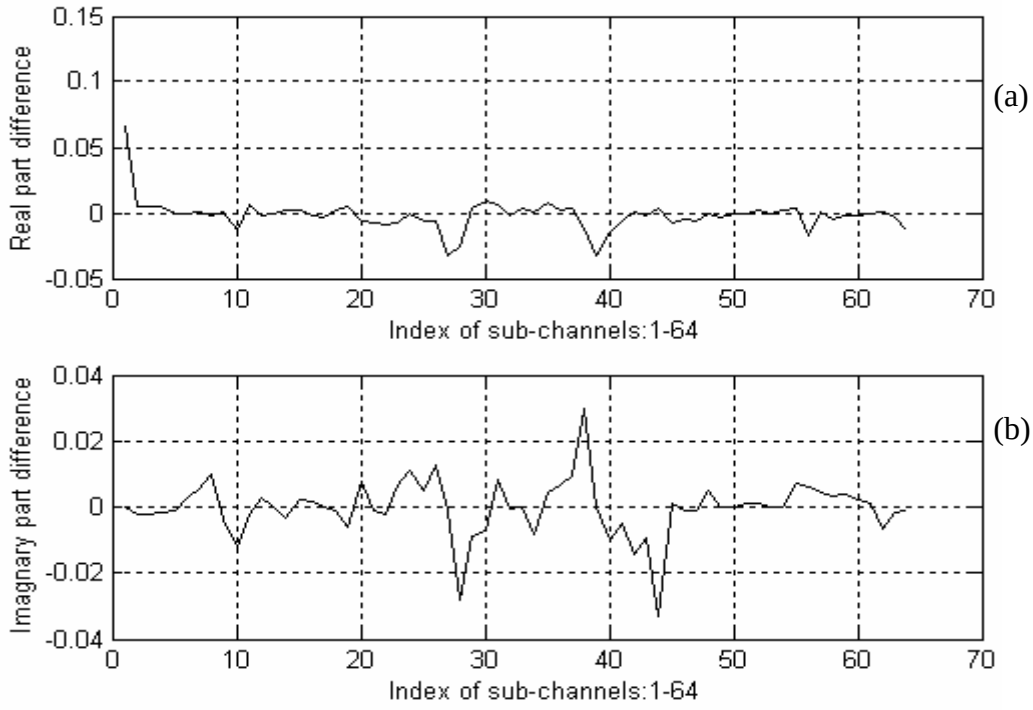


Fig. 6.3 Difference of $1/FFT(\vec{h}_{eff})$ between hardware implementation and mathematical calculation

The system performance is also tested to show the success of the hardware implementation. The vectors $\vec{w}_{opt}$, $\vec{h}$ and the inversion of $FFT(\vec{h}_{eff})$ calculated in hardware are used for the algorithm's parameters to test the system performance. The result is shown in Fig. 6.4, the bit error rate using mathematical calculation for the algorithm's parameters is also shown for comparison. Due to the finite length of binary representation and truncation, the offset introduced by the hardware implementation somewhat degrades the system performance, but the degradation is very minimal. The two curves are very close to each other. The overall system performance is very good. The high system performance verifies the proposed equalization technique and the hardware implementation.

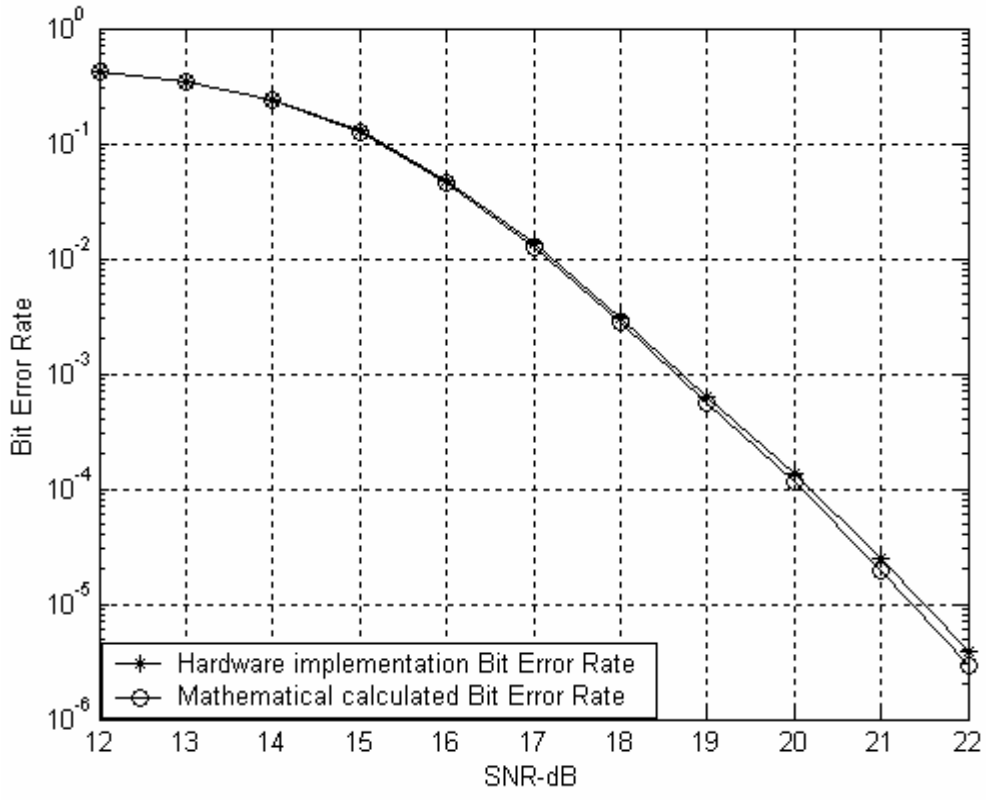


Fig. 6.4 Bit error rate obtained with hardware calculation and mathematical calculation for the algorithm's parameters

6.4 Results of DSP Implementation

The TEQ algorithm is also implemented in the TI TMS320C6711 to demonstrate its feasibility. The detailed information is shown in Table 6.6.

Table 6.6 Information on DSP implementation

Device	TI TMS320C6711
Clock cycles	31,472
Clock frequency	300MHz
Time to solve TEQ	104.91μs

With the same assumptions as in the implementation in FPGA, for the 8 tap TEQ FIR filter, it requires 104.91μs to find the coefficients by float point calculation. Compared with the FPGA implementation, the DSP solution is 31 times slower. The DSP calculation and mathematic calculation are shown in Table 6.7.

Table 6.7 DSP implementation and mathematical calculations

TEQ	Hardware calculations	Mathematical values
w_0	1.889057	1.88905660432045
w_1	-0.536787	-0.53678706108282
w_2	-0.793089	-0.79308852895277
w_3	-0.244993	-0.24499303067784
w_4	0.742533	0.74253305861786
w_5	-0.186744	-0.18674375786207
w_6	-0.11244	-0.11244022219353
w_7	-0.383618	-0.38361822089975

It can be concluded that using the DSP to implement the TEQ equalizer in float point

calculation, it has a higher accuracy, but it takes longer to find the TEQ coefficients.

In [26], the information on real-time MSSNR TEQ design on TI TMS320C620 is provided. For the 7^{th} order (8 taps) MSSNR TEQ FIR filter, it requires 603,440 clock cycles on 300MHz DSP to calculate the TEQ coefficients in fix point calculation. This is equivalent to 2011μs [26]. There is no straight comparison between these two implementations for the use of the different channel models and different data width. The MSSNR algorithm was also implemented in DSP TMS320C6711 in float point calculation with the same assumption as FPGA implementation, to show how it operate compared with the float point DSP implementation of the proposed algorithm. The characteristics of elements in matrices H_{win} and H_{wall} are exploited in calculations; Cholesky decomposition is used to calculate the square-root matrix and the power method is used to find the maximum eigenvalue and eigenvector. The DSP implementation information is shown in Table 6.8.

Table 6.8 DSP implementation of MSSNR TEQ algorithm

Device	TI TMS3206711
Algorithm (float point)	Maximum shortening SNR (MSSNR)
TEQ taps	8
Clock cycles	3,022,240
Clock frequency	300MHz
Calculation time	10,100μs

The hardware calculation of the TEQ coefficients using MSSNR TEQ algorithm is shown in Table 6.9. It can be seen that the TEQ coefficients are different from the

solutions of the algorithm proposed in this thesis shown in Table 6.7. It takes 3,022,240 clock cycles to find the optimum coefficients. On 300MHz DSP it takes about 10,100μs to find the optimum TEQ coefficients. It is about 96 times slower than the DSP implementation of the proposed TEQ algorithm.

Table 6.9 DSP implementation of MSSNR algorithm

TEQ	Hardware calculations	Mathematical calculations
w_0	-10.934371	-10.93437171147581
w_1	6.403067	6.40305765838586
w_2	4.250992	4.25099251696741
w_3	–0.530990	-0.53098569948867
w_4	–4.913623	-4.91362565421088
w_5	3.234580	3.23457405366343
w_6	0.463822	0.46382389085372
w_7	1.211181	1.21118184238945

6.5 Summary

In this chapter, the detailed information about the hardware implementation on TEQ solver, TEQ FIR filter and radix-4 pipeline FFT is provided. The hardware simulation results verify the functions of the chips. They satisfy the system requirements to operate at a data rate of 54Mbps with very high system performance. The DSP implementation results show that they would have higher accuracy but more time in solving the optimum TEQ coefficients.

Chapter 7 Conclusions and Suggestions for Future Study

7.1 Summary

Broadband communications attract much attention in the last few years. Using the IEEE 802.11a OFDM system to construct the WLAN and to achieve a high data rate of 54Mbps is appealing. In OFDM communication systems, the broadband is divided into many sub-carriers for parallel data transmission. The system has a high spectral efficiency and multi-path resistant capability tolerance. A guard interval using cyclic prefix is inserted to avoid the ISI and ICI. The technique is very simple but it compromises the transmission efficiency, especially when the maximum channel delay spread is long and the number of sub-carriers is low as in the IEEE 802.11a standard. To maintain the transmission efficiency, a pre-FFT TEQ is usually included in the system to improve the system performance.

In this thesis, some popular TEQ algorithms for OFDM system are studied and their computational complexities are analyzed. Due to their high computational complexities, they are not suitable for a practical use in the high rate IEEE 802.11a OFDM systems. Based on the MMSE algorithm, a reduced computational complexity design of a TEQ for the high rate IEEE 802.11a OFDM application to increase the transmission efficiency is then proposed. The algorithm is tested in Matlab® to verify its functionality. A system model with TEQ is setup in Simulink® based on the IEEE

802.11a specifications. The system performance is tested under different situations, including different lengths of cyclic prefix, different channel models, different orders of TEQ FIR filter and with estimated channel information. The system performance that does not include a TEQ is also tested for comparison. Finally the hardware design of the algorithm is studied. As an example, the algorithm with TEQ order 7 is implemented in FPGA. The result is compared with the mathematical calculations in Matlab®. The error introduced by the hardware implementation is also studied and analyzed. To make the algorithm feasible for a practical use, a high performance 64-point radix-4 pipeline FFT is also implemented in FPGA. The system performance with the hardware output for the algorithm's parameters is also tested. The proposed TEQ algorithm and the MSSNR TEQ algorithm are also implemented in TI TMS320C6711 with float point calculations to gain some comparative information.

7.2 Conclusions

In this thesis a reduced computational complexity TEQ algorithm is proposed. It uses a reduced dimensional convolution matrix to find the optimum coefficients of TEQ. After the derivation of the algorithm, the optimum TEQ coefficients can be solved by the equation (3.40). In solving the linear equation $A\vec{w} = B$, the matrix A is proved to be symmetric and positive definite. The LDL^T decomposition is also used to reduce the computational complexity further.

Simulation results verify the functionality of the proposed algorithm. The optimum order of the TEQ FIR filter depends on the channel impulse response. The overall system performance tests show that a high system performance gain can be obtained with a moderate order of the TEQ FIR filter. When the maximum channel delay spread

increases, the system performance degrades. However at high SNR values, the system performance is still acceptable. Based on the test results and considering the difficulty of the hardware implementation, a reasonable order of TEQ FIR filter can be selected to be between 7 and 31.

Considering the advances in VLSI technology and FPGA, the proposed algorithm with an order p=7 of the TEQ FIR filter is implemented in the Altera's Stratix FPGA device family. The characteristics of the matrices, i.e., the symmetry and the regularities between the elements of the reduced convolution matrix, are exploited to further reduce the computational load and hardware complexity. The optimum TEQ coefficients can be found in less than 4μs (which is the OFDM symbol interval in the IEEE 802.11a). Simulation output shows that the offset between the hardware implementation and the mathematical calculation is minimal. This verifies the functionality of the hardware implementation.

To compensate the effective channel impulse response, a zero forcing equalization technique is employed. To meet the high running speed of the IEEE 802.11a system, a 64-point radix-4 pipeline FFT is designed and implemented in FPGA. It is able to finish the FFT transfer in 65 clock cycles and can operate with clock frequency greater than 20MHz. Thus the TEQ FIR filter and the FFT chips satisfy the system requirement to operate at 54Mbps. Furthermore, system performance with hardware output for the algorithm's parameters shows that the performance loss due to the hardware implementation is minimal.

The proposed technique is attractive when it is necessary to shorten the long delay communication channels to increase the transmission efficiency. The simplicity of the technique makes its implementation possible in FPGA for the high rate IEEE 802.11a

systems.

7.3 Suggestions for Future Study

Suggested future work includes the design of a high speed fix point divider in FPGA to increase the TEQ running speed, the optimization of hardware implementation and the implementation of the TEQ algorithm with a higher order (for example 11 or 17) of FIR filter to further improve the system performance. Some pipeline parallel mechanism may be employed when the maximum channel delay spread is very long. In a practical OFDM system, the channel is required to be estimated first. A reconfigable TEQ chip structure based on the channel estimation and required TEQ FIR filter order makes it flexible and attractive.

In the IEEE 802.11a system, the pilot signal is used to fine tune the channel state information. This method may be introduced to improve the system performance in slowly time-varying multi-path fading channels.

Bibliography

[1] IEEE standard 802.11a-1999, adopted by the ISO/IEC and redesignated as ISO/IEC 8802-11:1999/Amd 1:2000(E). http://standards.ieee.org/getieee802/download/ 802.11a-1999.pdf.

[2] John G. Proakis, "Digital Communications", 4th Edition, McGraw–Hill Press, 2001.

[3] Lawrence R. Rabiner, Bernard Gold, "Theory and Application of Digital Signal Processing", Englewood Cliffs, N.J., Prentice-Hall, 1975.

[4] Kunio Takaya, "Digital Signal Processing", EE 880 class note, University of Saskatchewan, September 2000.

[5] B. Hirosaki, "An Analysis of Automatic Equalizers for Orthogonally Multiplexed QAM Systems", *IEEE Trans. on Commun.*, Vol. 28, pp. 73-83, January 1980.

[6] I. S. Chow and J. M. Cioffi, "Maximum Likelihood Receiver for Multicarrier Systems", *Proc. IEEE Int. Conf. Commun.*, Vol. 2, pp. 761-765, Geneva, Switzerland, May 1993.

[7] D. D. Falconer and F. R. Magee, "Adaptive Channel Memory Truncation for Maximum Likelihood Sequence Estimation", Bell Syst. Tech. J., Vol. 52, pp. 1541-1562, November 1973.

[8] J. S. Chow and J. M. Cioffi, "A Cost-effective Maximum Likelihood Receiver for Multicarrier Systems", *Proc. IEEE Int. Conf. Commun.*, Vol 2, pp. 948-952,

Illinois, USA, June 1992.

[9] J. S. Chow, J. M. Cioffi and J. A. Bingham, "Equalizer Training Algorithms for Multicarrier Modulation Systems", *Proc. IEEE Int. Conf. Commun.*, Vol 2, pp. 761-765, Geneva, Switzerland, May 1993.

[10] M. Nafie and A. Gather, "Time Domain equalizer training for ADSL", *Proc. IEEE Int. Conf. Commun*., Vol 2, pp. 1085-1089, Montreal, Canada, May 1997

[11] N. AL-Dhahir and J. Cioffi, "MMSE Decision Feedback Equalizers: Finite-Length Results", *IEEE Trans. On Information Theory*, Vol. 41, pp. 961-975, July 1995.

[12] N. AL-Dhahir, A. Sayed, J. Cioffi, "A High-Performance Cost-Effective Pole-Zero MMSE-DFE", *Proc. Allerton Conference on Comm., Control, and Computing*, pp. 1166-1175, Illinois, USA, September 1993.

[13] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse Response Shortening for Discrete Multitone Transceivers", *IEEE Trans. on Commun.*, Vol. 44, pp. 1662-1672 , December 1996.

[14] N. AL-Dhahir, J. Cioffi, "Optimum Finite-Length Equalization for Multicarrier Transceivers", *IEEE Trans. On Commun.*, Vol 44, pp. 56-63, January 1996.

[15] T. Starr, J. M. Cioffi, and P. J. Silverman, "Understanding Digital Subscriber Line Technology", Englewood Cliffs, NJ: Prentice-Hall, 1999.

[16] N. AL-Dhahir and J. M. Cioffi, "Efficiently Computed Reduced-parameter Input-aided MMSE Equalizers for ML Detection: A Unified Approach", *IEEE Trans. Inform. Theory*, Vol 42, pp. 903-915, May 1996.

[17] Heiko Schmidt, Karl-Dirk Kammeyer, "Impulse Truncation for Wireless OFDM Systems", *5th International OFDM-Workshop*, pp. 341-345, Hamburg, Germany, September 2000.

[18] Roger A. Horn, Charles R. Johnson, “Matrix Analysis”, Cambridge University Press, Cambridge, United Kingdom, 1985.

[19] J. W. Cooley and J. W. Tukey, “An Algorithm for Machine Computation of Complex Fourier Series”, Math. Comp., Vol. 19, pp. 297-301, April 1965.

[20] R. N. Bracewell, “The Hartley Transform”, Oxford Press, Oxford, England, 1985.

[21] P. Duhamel and H. Hollomann, “Split Radix FFT Algorithm”, Electronic Letters, Vol. 20, pp. 14-16, January 1984.

[22] H. Guo, G. A. Sitton, and C. S. Burrus, “The Quick Discrete Fourier Transform,” *Proc. of the ICASSP’94*, pp. 445-447, Adelaide, Australia, April 1994.

[23] A. Saidi, “Decimation-in-Time-Frequency FFT Algorithm”, *Proc. of the ICASSP’94*, pp. 453-456, Adelaide, Australia, April 1994.

[24] 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, pp. 702-709, October 1984.

[25] G. Bi and E. V. Jones. “A Pipelined FFT Processor for Word Sequential Data”, *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 37, pp. 1982-1985, December 1989.

[26] Brian L. Evans http://www.ece.utexas.edu/~bevans/projects/adsl/”Real-Time Minimum-ISI and MSSNR DMT TEQ Design on the TI TMS320C6200 fixed-point VLIW DSP”, UT Austin, February, 2001.