

國立交通大學

電子工程學系 電子研究所

博士論文

適用於高速行動之無線都會區域網路之
基頻接收機設計

**Design of Baseband Receiver for High-Mobility
Wireless Metropolitan Area Network**

研究生：陳筱筠

指導教授：周世傑 教授

中華民國九十八年九月



適用於高速行動之無線都會區域網路之
基頻接收機設計

**Design of Baseband Receiver for High-Mobility
Wireless Metropolitan Area Network**

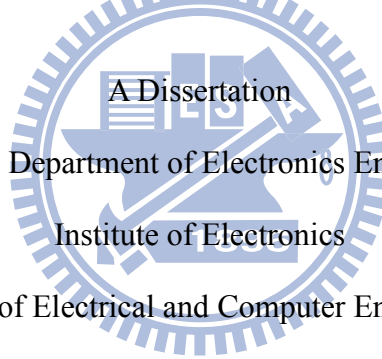
研究生：陳筱筠

Student : Hsiao-Yun Chen

指導教授：周世傑 教授

Advisor : Dr. Shyh-Jye Jou

國立交通大學
電子工程學系 電子研究所
博士論文



Submitted to Department of Electronics Engineering and
Institute of Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in

Electronics Engineering

September 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年九月



適用於高速行動之無線都會區域網路之 基頻接收機設計

研究生：陳筱筠

指導教授：周世傑 教授

國立交通大學

電子工程學系 電子研究所

摘要

近來，分集技術 (diversity techniques) 於無線通訊網路系統上受到各方矚目。使用多根妥善分隔的天線於傳送端與接收端時，多輸入多輸出 (multi-input multi-output; MIMO) 系統可以明顯地提升頻譜效率與系統傳輸能力。而空時分組碼 (space-time block code; STBC) 已被廣泛地應用於正交分頻多工 (orthogonal frequency-division multiplexing; OFDM) 系統上以得到傳送分集的增益效果，並改善無線通訊系統之系統效能。IEEE 802.16e 標準已經採納空時分組碼與正交分頻多工系統結合之系統應用 (STBC-OFDM systems)。該標準為 IEEE 802.16-2004 標準之延伸，主要是為了支援無線都會區域網路之移動性而提出的修正規格。在 STBC-OFDM 系統上，分集結合 (diversity combining)、同調偵測與解碼 (coherent detection and decoding) 需要相當精確的通道狀態資訊 (channel state information; CSI)。然而，精確的通道狀態資訊於行動無線通道上卻很難獲取。

在此論文中，我們提出應用於行動模式 IEEE 802.16e 規格之下行基頻接收機設計方法，並且此基頻接收機可應用於兩根傳送天線與一根接收天線 (2×1 天線) 之 STBC-OFDM 系統上。主要目標是為了能提供高系統效能傳輸於車速達

120 km/hr 行動環境中。首先，我們提出簡單但有效之符元邊界偵測與載波頻率漂移估計方法。接著，提出精確且擁有可接受的硬體複雜度之兩階段通道估測策略，並且此種通道估測方法在與內插式通道估測方法比較上有明顯的傳輸效能提升。此論文所提出之基頻接收機經由 2×1 天線之 STBC-OFDM 系統模擬驗證。在車速 120 與 240 km/hr 的環境中應用於四相位偏移調變 (quadrature phase shift keying; QPSK) 之下，此接收機仍可分別達位元錯誤率 (bit error rate; BER) 約 10^{-3} 與 10^{-4} ，並且未使用任何通道編碼技術。我們以 90 nm CMOS 製程實現此接收機晶片。在 16 正交振幅調變 (16 quadrature amplitude modulation; 16-QAM) 下，此晶片最快可支援下行資料傳輸達 27.32 Mbps。在通道估測硬體設計上，我們提出前置碼匹配 (preamble match)、直接式多路徑干擾消除解相關器 (straight multipath interference cancellation-based decorrelator)、最小平方估測器 (least square estimator) 及通道路徑解相關器 (path decorrelator) 等新的架構設計有效地減少硬體面積需求與功率消耗。此接收機晶片核心面積為 $2.4 \times 2.4 \text{ mm}^2$ 。在 78.4MHz 的操作頻率與 1 V 工作電壓下，其功率消耗為 68.48 mW。

此外，我們提出一組用於串列傳輸之傳輸碼，4-脈衝振幅調變對稱式傳輸碼 (4-PAM symmetric code)，其主要用於差動 4-脈衝振幅調變信號串列傳輸系統。而此傳輸碼保留 8B/10B 之傳輸優點，如直流平衡之位元串列資料與提供足夠的時序資料給予接收端進行時序回復等特性，並具有有效降低時脈回復時所產生的資料轉換邊界抖動之功能。此串列傳輸編、解碼器以 0.18 μm 製程實現，並證明若在接收端做時序回復時能夠有效的改善參考之時序資訊轉換邊界抖動達 $\pm 25\%$ 資料轉換時間。此 4-脈衝振幅調變信號串列傳輸編、解碼器之設計可分別操作於 819 MHz 與 704 MHz 並且源頭資料傳輸率可到達 13.1 Gbps 與 11.3Gbps。

Design of Baseband Receiver for High-Mobility

Wireless Metropolitan Area Network

Student: Hsiao-Yun Chen

Advisor: Dr. Shyh-Jye Jou

Department of Electronics Engineering & Institute of Electronics

National Chiao Tung University

Abstract

Recently, diversity techniques are receiving particular attention in wireless communication systems. By using well-separated antennas at the transmitter and/or receiver, the multiple-input and multiple-output (MIMO) systems can significantly increase the bandwidth efficiency and system capacity. Space-time block code (STBC) has suggested to be applied in an orthogonal frequency division multiplex (OFDM) system to utilize the transmit diversity for improving system performance in wireless communication systems. STBC-OFDM systems have been adopted in IEEE 802.16e which is an extension of IEEE 802.16-2004 for supporting mobility of wireless metropolitan area network (WMAN). However, for diversity combining, coherent detection and decoding, STBC-OFDM systems require accurate channel state information (CSI), which is particularly difficult to obtain in mobile wireless channels.

This dissertation proposes a downlink baseband receiver scheme for IEEE 802.16e in mobile mode. The proposed receiver is applied in STBC-OFDM system with two transmit antennas and one receive antenna and aims to provide high performance under the vehicle speed up to 120 km/hr. First, the simple and robust symbol boundary detection and carrier frequency offset estimation schemes are

proposed. Then, an accurate but hardware affordable two-stage channel estimation strategy is adopted to overcome the challenge of outdoor mobile channels. It has significant performance improvement as compared with the interpolation-based channel estimations. The performances of the proposed receiver have been demonstrated through the simulation of an STBC-OFDM system with two transmit antennas and one receive antenna. At vehicle speed of 120 and 240 hr/km for quadrature phase shift keying (QPSK) modulation, the proposed design can achieve the bit error rate (BER) of about 10^{-4} and 10^{-3} without using channel coding. The proposed receiver implemented in 90nm CMOS technology can support up to 27.32 Mbps (uncoded) downlink data transmission in 16 quadrature amplitude modulation (16-QAM). In the proposed channel estimator, novel architectures of a preamble match, a straight multipath interference cancellation (SMPIC)-based decorrelator, a least square (LS) estimator, and a path decorrelator are designed to reduce the area and power of the blocks. The receiver has the core area of $2.4 \times 2.4 \text{ mm}^2$ and dissipates 68.48 mW at 78.4MHz operating frequency from a supply voltage 1 V.

In addition, a novel DC-balanced low-jitter transmission code, a 4-PAM symmetric code, for a 4-PAM signaling system is presented. The 4-PAM symmetric code preserves all the useful characteristics of 8B/10B transmission code, such as DC-balanced serial data and guaranteed transitions in the symbol stream for data/clock recovery. Moreover, the proposed method decreases the jitter of the timing transition of the data in the receiver. The design results using 0.18 μm process demonstrate that the new transmission code can decrease the jitter of the transition point by $\pm 25\%$ of the transition region. The operation speeds of the encoder and decoder for the 4-PAM symmetric code are 819 MHz with 16-bit inputs (throughput 13.1 Gbps) and 704 MHz with 16-bit outputs (throughput 11.3 Gbps), respectively.

誌謝

在碩、博研究生涯的兩千五百多個日子中，最要感謝的人莫過於我的指導教授周世傑老師，無論在研究或做人處事上，老師總是給我許多的建議與鼓勵，指引我讓我能以謹慎的態度去學習、思考並解決問題，心中對老師是無限的感恩。

再者要感謝我最可愛的學弟紹維與俊男，因為有你們的加油鼓勵與幫忙，讓我有信心能去面對研究及生活上的種種挑戰，研究路上充滿喜悅。謝謝古孟霖學長在演算法與論文撰寫上給我許多協助與討論，能和學長合作真的很開心也很愉快。謝謝小胖，總是幫我解決繁瑣問題。謝謝舒蓉，我科法修課上的好伙伴。

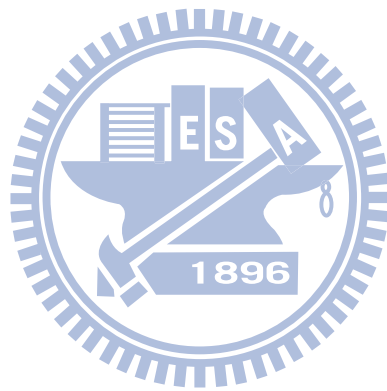
謝謝實驗室所有一起成長、奮鬥的學長、同學和學弟妹們，TOTORO、育群、庭楨、盈志、儷蓉、晉欽、誠文、阿賢、子捷、小朱、誌華、spice、阿樸、俊誼、國光、運翔、祥甦、喻喧、明銓、代暘、志宇、馥淳、為凱、雅雪、祥譽、銘謙、以樂，讓 316 實驗室充滿許多歡笑與回憶。謝謝所有照顧我的長輩與好友們，在我生活上給予各項的協助與鼓勵。謝謝大家。

感謝我的男友和我心愛的 maimai 一直陪伴在我身邊加油打氣，給我前進的力量。最後要感謝我最親愛的家人，有你們的關懷與支持，讓我得以專心致力於研究，感謝之情，不勝言表。謹將這份成就和喜悅與摯愛的家人分享。

筱筠

謹誌於 新竹

2009 年 9 月



Contents

Chapter 1 Introduction.....	1
1.1 Background.....	1
1.2 Motivation.....	6
1.3 Thesis Organization.....	7
Chapter 2 Overview of IEEE 802.16e OFDMA and MIMO Systems.....	9
2.1 Introduction.....	9
2.2 Overview of OFDMA.....	10
2.3 MIMO Systems.....	16
2.4 IEEE 802.16e OFDMA Specification.....	17
2.4.1 Frame Structure.....	17
2.4.2 Preamble Format.....	20
2.4.3 Pilot Modulation.....	21
2.4.4 Basic Specification in IEEE 802.16e OFDMA.....	22
Chapter 3 Downlink Baseband STBC-OFDM System Architecture.....	25
3.1 Introduction.....	25
3.2 System Specification and Design Targets.....	26
3.3 Transmitter Architecture.....	28
3.3.1 Channel Coding.....	30
3.3.1.1 Randomization.....	31
3.3.1.2 FEC coding.....	31
3.3.1.3 Interleaving.....	32
3.3.2 Signal Mapping.....	33
3.3.3 Subcarrier Allocation.....	33
3.3.4 STBC Encoding.....	36

3.4	<i>Receiver Architecture</i>	37
3.4.1	Synchronization	38
3.4.1.1	Symbol Timing Synchronization	39
3.4.1.2	Carrier Frequency Synchronization	42
3.4.2	Channel Estimation	48
3.4.3	STBC Decoding	49
3.4.4	Channel Decoding	49
3.4.4.1	De-interleaving	50
3.4.4.2	FEC Decoding	50
3.4.4.3	De-randomization	50
3.5	<i>System Simulation</i>	51
3.6	<i>Summary</i>	57
Chapter 4 A Robust Channel Estimator for STBC-OFDM Systems		59
4.1	<i>Introduction</i>	59
4.2	<i>Pilot-aided Channel Estimation Methods</i>	61
4.2.1	Interpolation-based Channel Estimation	61
4.2.2	DFT-based Channel Estimation	63
4.3	<i>Two-Stage Channel Estimation Method</i>	66
4.3.1	Initialization Stage	66
4.3.2	Tracking Stage	68
4.4	<i>Proposed Channel Estimator</i>	70
4.4.1	Initialization Stage: Preamble Match	71
4.4.2	Initialization Stage: SMPIC-based Decorrelator	71
4.4.3	Tracking Stage: STBC Decoder and Demapper	73
4.4.4	Tracking Stage: LS Estimator	74
4.4.5	Tracking Stage: Hessian Matrix Calculator and Path Decorrelator	76
4.5	<i>Computational Complexity</i>	78
4.6	<i>Simulation Results</i>	80
4.7	<i>Summary</i>	86
Chapter 5 Novel Programmable FIR Filter Based on Higher Radix Recoding .87		
5.1	<i>Introduction</i>	87
5.2	<i>Algorithm Reformulations</i>	88
5.2.1	Higher Radix Recoding	88

5.2.2	Secondary Radix Recoding.....	89
5.2.3	Algorithm Reformulation for FIR Filter	91
5.3	<i>Programmable FIR Filter Based on Higher Radix Booth Recoding</i>	92
5.3.1	Higher Radix Booth Multiplier (HRBM)	92
5.3.2	HRBM ² Based on Secondary Radix Recoding.....	94
5.3.3	Programmable FIR Filter Architectures.....	96
5.4	<i>Results and Comparisons</i>	97
5.5	<i>Summary</i>	98
Chapter 6 Downlink Baseband Receiver Implementation		101
6.1	<i>Introduction</i>	101
6.2	<i>Word Length Optimization</i>	105
6.3	<i>Synchronizer</i>	107
6.3.1	Symbol Boundary Detector.....	107
6.3.2	Carrier Frequency Offset Estimator	109
6.4	<i>Proposed Channel Estimator</i>	114
6.4.1	Initialization Stage: Preamble Match.....	114
6.4.2	Initialization Stage: SMPIC-based Decorrelator	116
6.4.2.1	Merge Sorting Network with Programmable and Partial Sorting Capability (MSNP ²).....	117
6.4.2.2	Triangular Decorrelator (TD).....	122
6.4.3	Tracking Stage: STBC Decoder and Demapper	124
6.4.4	Tracking Stage: LS Estimator	127
6.4.5	Tracking Stage: Hessian Matrix Calculator and Path Decorrelator	129
6.5	<i>FFT/IFFT module</i>	131
6.6	<i>Data Flow of the Proposed Receiver</i>	134
6.7	<i>Simulation Results</i>	136
6.8	<i>Design Results</i>	138
6.9	<i>Summary</i>	144
Chapter 7 Conclusions.....		147
Appendix A: A DC-balance Low-jitter Transmission Code for 4-PAM Signaling		149
A.1	<i>Introduction</i>	149

A.2 Design of the 4-PAM Symmetric Code 150

 A.2.1 The Structure of 4-PAM Symmetric Code 151

 A.2.2 Low-jitter of Data Transitions 153

 A.2.3 The Design Concept of Running Disparity 154

 A.2.4 Error Detection 156

 A.2.5 Comparison of Transmission Codes 156

A.3 Architecture and Implementation of 4-PAM Symmetric Encoder / Decoder 157

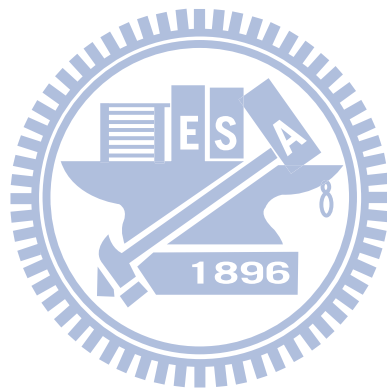
A.4 Conclusions 162

References 163



List of Tables

Table 1.1 Comparisons of wireless standards	3
Table 2.1 Scalable OFDMA parameters	14
Table 2.2 Major parameters of IEEE 802.16e OFDMA specification	23
Table 3.1 Major parameters of the proposed STBC-OFDM system	27
Table 3.2 Puncture configuration	32
Table 3.3 STBC encoding	37
Table 3.4 Required SNR (dB) for $BER < 10^{-6}$ in AWGN channel	53
Table 3.5 Required SNR (dB) for $BER < 10^{-6}$ in ITU Veh-A channel	56
Table 4.1 Computational complexity and storage requirements	79
Table 5.1 Radix-16 recoding scheme of HREnc	93
Table 5.2 Radix-32-modulo-7 recoding scheme of HRSEnc	95
Table 5.3 Design results of 10-tap programmable FIR filter	98
Table 6.1 Word lengths of several key signals in the proposed receiver	105
Table 6.2 State definitions of the MSNP ² state diagram	121
Table 6.3 Control signals used in the MSNP ² state diagram	122
Table 6.4 Read or write address for the butterfly unit in each stage	133
Table 6.5 Implementation results of the proposed receiver	138
Table 6.6 Implementation results of the proposed channel estimator	141
Table 6.7 Comparison with other design operating at 200 MHz	144
Table A.1 8B/10B, 8B5Q and 4-PAM symmetric transmission codes for 4-PAM system	157
Table A.2 Synthesis results of the 4-PAM symmetric encoder / decoder	160



List of Figures

Fig. 1.1 Wireless network area definition [4].	2
Fig. 1.2 Concept of AMC [15].	5
Fig. 2.1 (a) Conventional non-overlapping sub-bands. (b) Overlapping sub-bands.	11
Fig. 2.2 OFDM symbol with cyclic prefix.	11
Fig. 2.3 OFDMA subcarrier allocation.	12
Fig. 2.4 Two-dimensional resources of OFDMA transmission.	12
Fig. 2.5 Adjacent and distributed subcarrier allocations.	13
Fig. 2.6 An OFDM symbol.	14
Fig. 2.7 Two-dimensional basic terms in OFDMA data structure.	15
Fig. 2.8 An OFDMA frame in TDD mode [8].	18
Fig. 2.9 An OFDMA frame with multiple zones [8].	19
Fig. 2.10 Cluster structure.	21
Fig. 2.11 PRBS generator for pilot modulation.	21
Fig. 3.1 Frame format.	28
Fig. 3.2 (a) Transmitter and (b) receiver of the proposed STBC-OFDMA system with two transmit antennas and one receive antenna.	29
Fig. 3.3 Block diagrams of channel coder.	30
Fig. 3.4 PRBS generator for data randomization.	31
Fig. 3.5 Convolutional encoder of code rate 1/2.	32
Fig. 3.6 QPSK, 16-QAM and 64-QAM constellations.	34
Fig. 3.7 Subcarrier allocation of PUSC (FFT size = 1024).	35
Fig. 3.8 Receiver of the proposed STBC-OFDM system.	37
Fig. 3.9 Synchronization architecture of the proposed STBC-OFDM system.	39
Fig. 3.10 Common ISI free region of two transmit antennas.	41
Fig. 3.11 Matching results when CFO is 0.49 subcarrier spacing.	44
Fig. 3.12 CFO region partition for the proposed ping-pong algorithm.	45
Fig. 3.13 Error probability of the ICFO detection versus FCFO at v_e of 60 km/hr with E_b/N_0 of 16 dB.	45
Fig. 3.14 Performance for different quantization method.	47

Fig. 3.15 RMS errors of CFO under different vehicle speed.	47
Fig. 3.16 Uncoded BER performance for QPSK modulations at v_e of 3, 60 and 120 km/hr.	52
Fig. 3.17 Uncoded BER performance for 16-QAM modulations at v_e of 3, 60 and 120 km/hr.	52
Fig. 3.18 Uncoded BER performance for 64-QAM modulations at v_e of 3, 60 and 120 km/hr.	53
Fig. 3.19 Coded BER performance for QPSK modulation.	54
Fig. 3.20 Coded BER performance for 16-QAM modulation.	55
Fig. 3.21 Coded BER performance for 64-QAM modulation.	56
Fig. 4.1 Block diagram of a classic DF DFT-based channel estimation method.	64
Fig. 4.2 Flowchart of MPIC-based decorrelation method.	67
Fig. 4.3 Overall block diagram of the proposed channel estimator.	70
Fig. 4.4 Flowchart of the proposed SMPIC-based decorrelation.	72
Fig. 4.5 Pilots transmitted in the cluster structures over different time slots.	76
Fig. 4.6 BER performance for QPSK at v_e of 120 km/hr.	81
Fig. 4.7 BER performance for 16-QAM at v_e of 120 km/hr.	81
Fig. 4.8 BER performance for QPSK at v_e of 240 km/hr.	82
Fig. 4.9 BER performance for 16-QAM at v_e of 240 km/hr.	83
Fig. 4.10 BER performances versus the vehicle speed.	84
Fig. 4.11 MSE of channel estimations using the proposed scheme, predictive algorithms and 2-D interpolation-based methods at v_e of 120 km/hr.	85
Fig. 4.12 MSE of channel estimations using the proposed scheme and 2-D interpolation-based methods with different interpolations in frequency axis at v_e of 120 km/hr.	86
Fig. 5.1 Parallel architecture of 19×19 -bit radix-16 ($\beta=24$) HRBM.	92
Fig. 5.2 Odd multiples pre-computation structure and pre-compute $5 \times$ architecture.	93
Fig. 5.3 Parallel architecture of 19×19 -bit radix-32-modulo-7 ($\beta=25$; $\lambda=7$) HRBM ² .	94
Fig. 5.4 Programmable FIR filter using HRBM.	96
Fig. 5.5 Programmable FIR filter using HRBM ² .	97
Fig. 5.6 Design results comparison of different multiplier-based.	99
Fig. 6.1 Architectures of (a) the proposed downlink baseband receiver and (b) the proposed two-stage channel estimator with the STBC decoder and demapper.	102
Fig. 6.2 Implementation flow of the proposed receiver.	104
Fig. 6.3 Output SNR versus different word lengths for the received sample input.	106

Fig. 6.4 Output SNR versus different word lengths for the channel response output.	106
Fig. 6.5 Match filter design.	109
Fig. 6.6 Storage units for storing previous correlation results.	110
Fig. 6.7 Cross correlation circuit for the FCFO estimation.	110
Fig. 6.8 CORDIC rotation.	112
Fig. 6.9 CORDIC design.	114
Fig. 6.10 Basic design unit of the preamble match.	115
Fig. 6.11 Output SNR versus the value of β .	116
Fig. 6.12 Block diagram of the MSNP ² .	117
Fig. 6.13 Batcher classic sorting network with I/O size of eight.	118
Fig. 6.14 The memory bank arrangement of merge sorting network.	118
Fig. 6.15 32-item merge sorting sequence.	119
Fig. 6.16 128-item merge sorting sequence.	120
Fig. 6.17 State diagram of the proposed MSNP ² .	121
Fig. 6.18 Design of the decorrelated unit.	123
Fig. 6.19 Process steps of the TD operation.	124
Fig. 6.20 State diagram of the proposed TD.	125
Fig. 6.21 Design of the divider-free STBC decoder.	126
Fig. 6.22 Block diagram of the two-stage dichotomy demapper.	127
Fig. 6.23 Design of the LS estimator.	128
Fig. 6.24 Design of the coordinate precalculator.	128
Fig. 6.25 Block diagram of the path decorrelator.	130
Fig. 6.26 Radix-8 1024-point parallel memory-based FFT architecture.	132
Fig. 6.27 Radix-8 processing element.	132
Fig. 6.28 Addressing scheme of the write-in data.	133
Fig. 6.29 FFT processor arranged with five memory banks.	135
Fig. 6.30 Flow chart of the proposed receiver.	135
Fig. 6.31 BER performances at v_e of 120 km/hr.	136
Fig. 6.32 BER performances versus the vehicle speed.	137
Fig. 6.33 Chip layout of the proposed STBC-OFDM downlink baseband receiver IC.	139
Fig. 6.34 Area proportion of the proposed STBC-OFDM downlink baseband receiver.	140
Fig. 6.35 Area proportion of the proposed two-stage channel estimator.	142
Fig. 6.36 Hardware reduction of the proposed channel estimator.	143
Fig. A.1 Three transition types in the differential 4-PAM symbol stream.	150

Fig. A.2 Comparisons of type 1 and type 2 transitions in the differential 4-PAM signals.	152
Fig. A.3 Structure of 4-PAM symmetric encoders and decoders for 4-PAM signaling system.	152
Fig. A.4 (a) Type 1 transition in the 4-PAM symmetric code and (b) frame synchronization.	153
Fig. A.5 Unit cell of running disparity.	154
Fig. A.6 Disparity versus time plot.	155
Fig. A.7 Block diagram of the 4-PAM symmetric encoder.	158
Fig. A.8 Block diagram of the 4-PAM symmetric decoder.	159
Fig. A.9 Measurement results [90] of (a) all types of transitions and (b) only maximum-swing type1 transitions of the differential 4-PAM signal.	161
Fig. A.10 Frequency responses of the 4-PAM symmetric code and the 8B/10B code.	162



Chapter 1

Introduction

1.1 Background

Data networking and telecommunication are fundamental changed by the wireless communication development, which is making integrated networks a reality. The users can enjoy the fully distributed computing and communications without the wire, any time, anywhere. There are two different types of wireless services: 1) the fixed wireless provides a set of services similar to that of the traditional fixed-line services but using wireless as the medium of transmission; 2) the mobile wireless offers the additional functionality of portability and mobility. IEEE (Institute of Electrical and Electronics Engineers) have developed different methods and standards of wireless communication based on various commercially driven requirements. As shown in Fig. 1.1, based on their specific application and transmission range, the standards of wireless communication can be roughly classified into four categories: Wireless Personal Area Network (WPAN), Wireless Local Area Network (WLAN), Wireless Metropolitan Area Network (WMAN) and Wireless Wide Area Network (WWAN). These categories are summarized in Table 1.1 and briefly introduced below.

- WPAN is a computer network which communicates devices centered on an individual personal workspace. The devices include cell phones and personal digital assistants (PDAs), laptops, PCs, printers, digital cameras and video game. The range of WPAN is typically within about 10 m. The

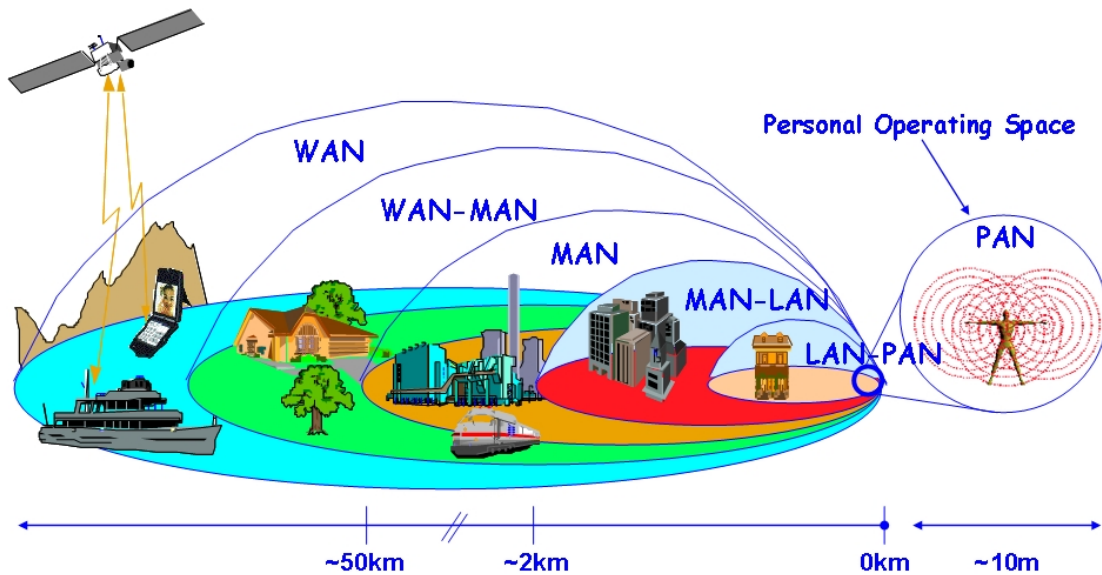


Fig. 1.1 Wireless network area definition [4].

standards of WPAN include IEEE 802.15.1 Bluetooth [1], IEEE 802.15.3 Ultra Wideband (UWB) [2], and IEEE 802.15.4 low-rate WPAN (LR-WPAN or ZigBee) [3].

- WLAN also uses radio communication in unlicensed band to realize the same functionality that a wired LAN has. It is designed to provide in-building broadband coverage. The range of WLAN is typically within about 100 m. IEEE 802.11 family defines a set standard (Wi-Fi) of WLAN which currently includes six over-the-air modulation techniques, and the most popular techniques are those defined by the a, b, g and n amendments to the original standard [5], [6]. However, the usage of IEEE 802.11 standard is restricted by the small transmitting range and the lack of mobility.
- WMAN is optimized for a larger geographical area than a WLAN, ranging from several blocks of buildings to entire cities, and initially proposed for last mile connectivity. WMAN is defined by IEEE 802.16 Working Group on Broadband Wireless Access (BWA) Standards and commercially known as Worldwide Interoperability for Microwave Access (WiMAX) which defines broadband Internet access from fixed or mobile devices via antennas.

TABLE 1.1
COMPARISONS OF WIRELESS STANDARDS

	802.15	802.11	802.16	802.20
Range	1-10 m	100 m	4-6 miles	9 miles
Application	Personal workspace	Indoor environment	Outdoor fixed or mobile environment	Outdoor high mobile environment
Spectrum	Unlicensed	Unlicensed	Licensed Unlicensed	Licensed
Frequency Band	Adapt to different spec.	2.4 GHz, 5 GHz	10-66 GHz 2-11 GHz	< 3.5 GHz
Data Rates	1-3 Mbps for Bluetooth 11-55 Mbps for UBW 20-250 Kbps for LR-WPAN	Up to 54 Mbps in 20 MHz bandwidth for IEEE 802.11-2007 Up to 600 Mbps using MIMO in 40 MHz bandwidth for IEEE 802.11n	Up to 75 Mbps in 20 MHz bandwidth for IEEE 802.16-2004 and IEEE 802.16e-2005	Downlink > 1 Mbps Uplink > 300 kbps
Channel	NLOS	NLOS	LOS (10-66 GHz) NLOS (2-11 GHz)	NLOS

The average cell ranges for WMAN networks is about 4-6 miles in non-line of sight (NLOS) environment, and service ranges up to 10 miles are in line of sight (LOS) applications [7], [8].

- WWAN is a computer network covering a more broad geographical area as contrasted with WMAN and optimized for full mobility up to vehicular speeds of 250 km/hr. WWAN is specified by IEEE 802.20 standard approved on June 2008 [9]. The air interface operates in bands below 3.5 GHz and with a peak data rate of over 1 Mbps.

Under the fast development of portable Internet services, the wireless communication technologies providing high data rate and credible performance have gradually become the mainstream in order to satisfy the requirements of various multimedia transmissions.

IEEE 802.16 working group established in 1999 standardizes the air interface and related functions associated with broadband WMAN. The first version of IEEE 802.16 standard (802.16.1) was an air interface for 10 to 66 GHz and completed in December 2001 [10]. It only supports single carrier (SC) modulation in the physical (PHY) layer with only a LOS capability. IEEE 802.16a standard ratified in January 2003 was an amendment to IEEE 802.16 [11]. It provides the capacities of a point-to-multipoint connection and NLOS transmission in the 2-11 GHz band. The PHY layer was therefore extended to include orthogonal frequency division multiplex (OFDM) and orthogonal frequency division multiple access (OFDMA). In September 2003, IEEE 802.16d was a revision project aiming to align the standard with aspects of the European Telecommunications Standards Institute (ETSI) HiperMAN standard [12]. This project concluded in 2004 with the release of IEEE 802.16-2004 [7] which superseded the earlier IEEE 802.16 documents. An industry consortium, WiMAX adopted IEEE 802.16-2004 as the first solution of fixed applications.

IEEE 802.16e-2005 (IEEE 802.16e) concluded in 2005 [8], which is often referred to as mobile WiMAX, is an extension of IEEE 802.16-2004 for providing high data rate transmission and mobility of WMAN. It enables mobile speed up to 120 km/hr, but also be backward compatible to support the fixed mode in IEEE 802.16-2004. Operation in mobile mode is limited to the licensed bands between 2-6 GHz; on the other hand, operation in fixed mode is limited to 2-11 GHz. It is based on an OFDMA technique to support multiple access scheme and multiple-input multiple-output (MIMO) systems over multipath fading channels. IEEE 802.16e is followed in this dissertation. Several distinct features of IEEE 802.16e standard are described as follows [13], [14]:

- Throughput: The technology at theoretical maximums could support about 75 Mbps in a 20 MHz channel using 64 quadrature amplitude modulation (QAM) with code rate 3/4. Moreover, MIMO technology can be exploited to further improve the transmission rate under good transmitting conditions.

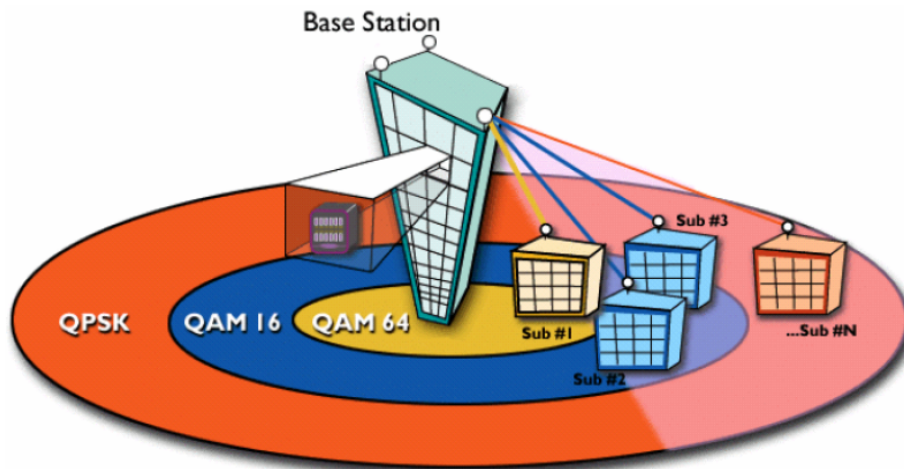


Fig. 1.2 Concept of AMC [15].

- Adaptive modulation and coding (AMC): IEEE 802.16e supports the dynamic adaptive modulation scheme which can adaptively configure the modulation and forward error correction coding schemes according to the channel conditions. As shown in Fig. 1.2, when the channel is good, the base station (BS) transmits data in 64-QAM with high code rate to achieve high transmission rate. In contrast, when the channel is poor, the base station chooses the small constellation and lower code rate to transmit. AMC significantly increases the transmission capacity and the coverage range.
- Quality of service (QoS): QoS is to resource reservation control mechanisms over the air interface on a frame-by-frame basis. It is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. The sub-channelization and medium access protocol (MAP) scheme let the scheduling in medium access control (MAC) layer become more flexible.
- Mobility: The mechanism of handover is optimized to be no longer than 50 ms. It can ensure the real-time operation without service degradation. This is especially important for the real-time applications such as Voice over Internet Protocol (VoIP). The system also supports the power

management with sleep and idle modes to extend the battery life of handheld subscriber devices.

- Security: The security specification is the most advanced in current wireless access systems. It offers Extensible Authentication Protocol (EAP) based authentication, Advanced Encryption Standard-Counter with CBC-MAC mode (AES-CCM) based authentication encryption, and Cipher-based Message Authentication Code (CMAC) and Hashed Message Authentication Code (HMAC) based control message protection scheme.

1.2 Motivation

IEEE 802.16e systems effectively provide wireless transmission of data using a variety of transmission modes from point-to-multipoint links to portable and fully mobile internet access. The research and development of IEEE 802.16e systems have gained more and more interest and became a worldwide topic. However, in mobile wireless communication, the channels often vary rapidly, which results in a large Doppler spread, particularly when the mobile station (MS) moves at a vehicular speed. A fundamental phenomenon that makes credible wireless transmission expensive and difficult is time-varying multipath fading. In order to improve the transmission quality in fast and selective fading channels, transmit diversity is an effective technology for reducing fading effect in mobile wireless communication, especially when receive diversity is expensive or impractical to acquire.

In recent years, space-time block code (STBC) [16], [17] has been shown to give high code efficiency and good performance. It is suggested to be applied in an OFDM system since OFDM systems with multiple antennas can provide better communication performance by utilizing transmit diversity [16]-[20]. Transmit diversity has been studied extensively as a method of combating impairments in wireless fading channels. However, the multiple antenna technologies bring some new challenges in attempts to realize STBC-OFDM systems. The most important challenge is the channel estimation [21]. Accurate channel state information (CSI) is

required for diversity combining, coherent detection, and decoding. Nevertheless, it is usually difficult to obtain in particular for time-varying multipath fading channels. Although STBC can still achieve the desired transmit diversity with small errors in CSI, a significant degradation is observed when the channel estimation error increases. Sending more preamble symbols or inserting more pilots in an OFDM symbol during the transmission could improve the channel estimation accuracy, but the bandwidth efficiency is also degraded. Hence, a robust channel estimation scheme is needed for good operation when the CSI is not exactly known. Moreover, the STBC-OFDM system performance is also susceptible to the synchronization errors which have been known to rotate the phase and destroy the orthogonality of an OFDM symbol. The mechanisms of synchronization are necessary to overcome these problems. As a result, the success of implementing an STBC-OFDM system in high mobility decides on two crucial challenges, synchronization and channel estimation.

In this dissertation, we focus on the design and development of a downlink baseband receiver applied in STBC-OFDM systems for IEEE 802.16e in mobile mode. Two main tasks, high quality synchronization and robust channel estimation, are performed. The proposed STBC-OFDM system is expected to present a solid foundation for WMAN system design in fixed and mobile wireless communication.

1.3 Thesis Organization

The rest of this dissertation is organized as follows. Chapter 2 introduces the concepts of OFDM, OFDMA and MIMO techniques and also briefly overviews the IEEE 802.16e OFDMA specification. Chapter 3 presents the architecture design and the performance simulation of the proposed STBC-OFDM system with two transmit antennas and one receive antenna. The proposed system aims to provide high performance for WMAN communication in fixed and mobile environments. It provides the simple and robust symbol boundary detection and carrier frequency offset estimation schemes and an accurate but hardware affordable two-stage channel estimation strategy to overcome the challenge of outdoor fading channels. In Chapter 4, the robust two-stage channel estimator for STBC-OFDM systems is proposed and analyzed. An efficient architecture of the proposed channel estimator is provided for

low-complexity hardware implementation while keeping the high performance. Chapter 5 discusses a novel programmable FIR filter based on higher radix recoding for low-power and high-performance applications. In order to verify the feasibility, the hardware implementation and the experimental results of the proposed receiver are presented in Chapter 6. Finally, Chapter 7 gives the conclusions of the proposed receiver. In addition, Appendix A presents a study of a DC-balanced low-jitter transmission code for 4-PAM signaling.



Chapter 2

Overview of IEEE 802.16e

OFDMA and MIMO Systems

2.1 Introduction

IEEE 802.16-2004 [7] specifies the air interface for fixed BWA systems and has been updated and extended to IEEE 802.16e-2005 [8] for mobile BWA systems with the concept of scalable OFDMA which is an effective technique for high bit rate applications over multipath channels. There are several specifications of IEEE 802.16e-2005 PHY layer for different applications and frequency range. For example, SC operates from 10 GHz to 66 GHz in LOS; besides, SCa, OFDM and OFDMA operate below 11 GHz in NLOS.

In this dissertation, we focus on the IEEE 802.16e OFDMA specification that supports the multi-antenna technology. We will briefly introduce the concepts of OFDM and OFDMA in this chapter. MIMO technologies will also be introduced. Finally, we give an overview of the IEEE802.16e OFDMA standard.

2.2 Overview of OFDMA

OFDM modulation offers an attractive technique for high-speed data transmission in mobile communication since it can effectively combat frequency-selective multipath fading using relatively simple frequency-domain equalization. OFDM has been widely adopted in several standards, e.g. digital video broadcasting for terrestrial (DVB-T) [22], high-speed WLAN and WMAN such as IEEE 802.11b/g (Wi-Fi) and IEEE 802.16 (WiMAX). Moreover, the high computational complexity of OFDM implementation has become possible due to the success development of digital signal processing (DSP) and very large scale integrated circuit (VLSI).

The concept of OFDM is coming from frequency division multiplexing (FDM). In FDM, the entire signal bandwidth is divided into several non-overlapping sub-bands as shown in Fig. 2.1 (a). The conventional parallel data transmission modulates each independent data on different sub-bands, and then these sub-bands are frequency-multiplexed. In order to prevent from the adjacent channel interference, the spectrum spacing allocated between sub-bands leads to inefficient utilization of the signal bandwidth. This problem is overcome by employing overlapping sub-bands as shown in Fig. 2.1 (b), and the signal bandwidth can be effectively utilized. Furthermore, OFDM technology is invented to divide the entire signal bandwidth is into mutual orthogonal overlapping subcarriers, and the orthogonality guarantees subcarrier transmission without interference from each other. As a result, the OFDM technology can achieve high spectral efficiency. Each transmitted signal in each narrow band subcarrier experiences flat channel fading; thus, the channel equalization can be performed by a simple one-tap frequency-domain equalizer. However, the channel delay spread causes the inter-symbol interference (ISI) which destroys the orthogonality of subcarriers in OFDM. In order to avoid ISI effect, a guard interval with the length longer than the maximum channel delay spread is inserted to the front of an OFDM symbol. Although the guard interval can be used to resolve ISI problem, the loss of orthogonality among subcarriers causes inter-channel interference (ICI). A cyclic prefix (CP), which is equal to a part of the OFDM symbol tail, is widely adopted in current standards to maintain the subcarrier orthogonality and avoid ICI effect as shown in Fig. 2.2.

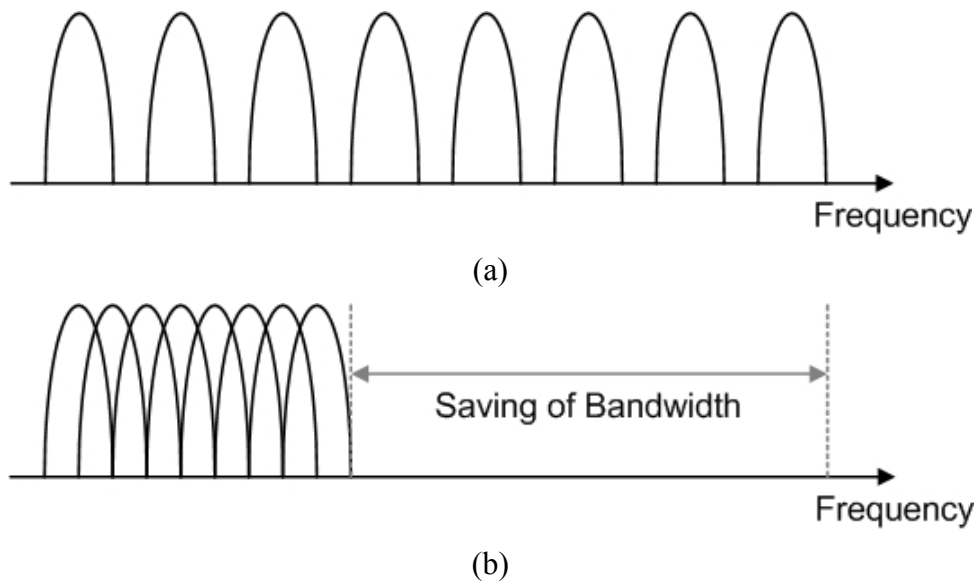


Fig. 2.1 (a) Conventional non-overlapping sub-bands. (b) Overlapping sub-bands.

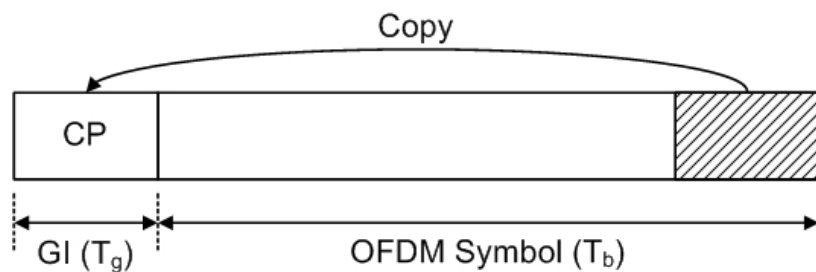


Fig. 2.2 OFDM symbol with cyclic prefix.

OFDMA technology is a multi-user version of OFDM modulation, and it is a major multiple access method considered for future wireless systems. IEEE 802.16e-2005 PHY layer provides OFDMA specification for multi-user communication. Multiple access is achieved in OFDMA by assigning mutually exclusive subsets of subcarriers to individual users, which allows simultaneous low data rate transmission from several users. The all available subcarriers of uplink and downlink in OFDMA are divided into several subsets of subcarriers termed as sub-channels as shown in Fig. 2.3.

OFDMA can be seen as an alternative to combining OFDM with time division

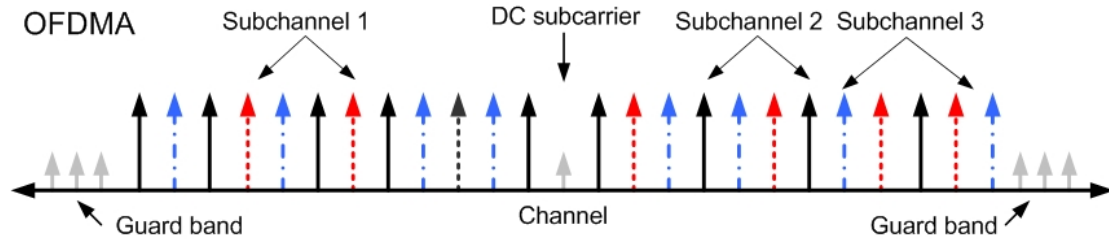


Fig. 2.3 OFDMA subcarrier allocation.

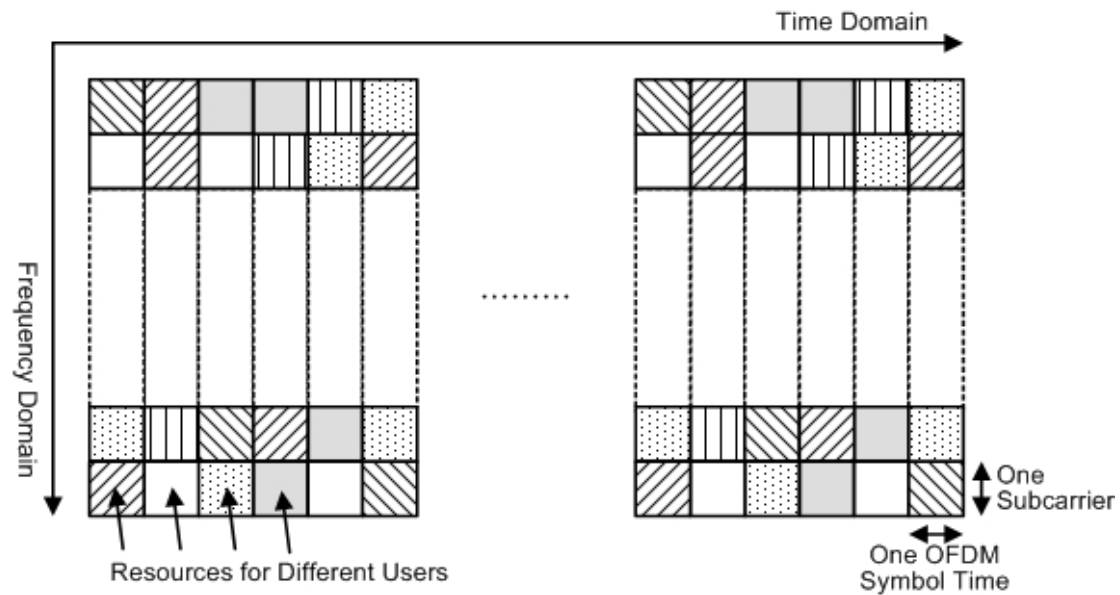


Fig. 2.4 Two-dimensional resources of OFDMA transmission.

multiple access (TDMA) or time-domain statistical multiplexing, i.e. packet mode communication. As shown in Fig. 2.4, the resources of OFDMA transmission are partitioned in the time-frequency space, and time slots are assigned along the OFDM symbol index as well as OFDM subcarrier index. OFDMA is considered as highly suitable for broadband wireless networks due to the advantages including scalability, MIMO easy applying, and the multi-user diversity ability to take advantage of channel frequency selectivity. OFDMA has another advantage of that low-data-rate users can send continuously with low transmission power. Constant delay and shorter delay can be achieved. In IEEE 802.16e OFDMA specification, there are two types of data allocation for sub-channelization, contiguous and diversity. As shown in Fig. 2.5, the

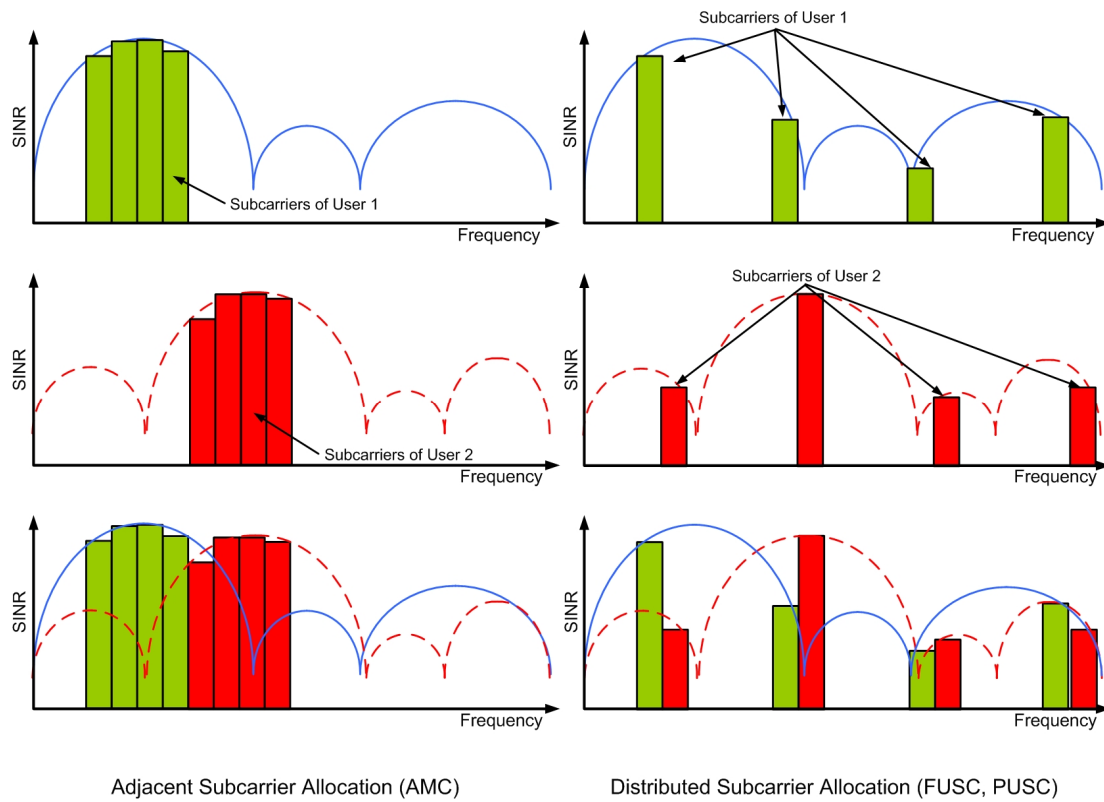


Fig. 2.5 Adjacent and distributed subcarrier allocations.

contiguous permutation collects contiguous subcarriers to form a sub-channel. On the contrary, the diversity permutation pseudo-randomly spread out the subcarriers of sub-channel over the entire bandwidth and brings the benefit of frequency diversity and robustness against the frequency select fading channel.

A concept of scalable OFDMA (S-OFDMA) [23] is also introduced to the IEEE 802.16e OFDMA specification, which supports for 128, 512, 1K, and 2K fast Fourier transform (FFT) sizes to address a variable bandwidth sizes from 1.25MHz to 20MHz for NLOS operations as shown in Table 2.1.

Several basic term definitions should be established to help us to follow IEEE 802.16e OFDMA specification.

- *Subcarrier*: An OFDM symbol is made up of subcarriers as shown in Fig. 2.6. There are three subcarrier types: data subcarriers for data transmission, pilot subcarriers for channel estimation purposes and null subcarriers for no transmission, guard bands, non-active subcarriers and

TABLE 2.1
SCALABLE OFDMA PARAMETERS

Parameters	Values			
System channel bandwidth (MHZ)	1.25	5	10	20
FFT Size	128	512	1024	2048
Sampling Factor	28/25			
Sampling Frequency	1.4	5.6	11.2	22.4
CP Ratio	1/32, 1/16, 1/8, and 1/4			
Modulation Mode	QPSK, 16QAM, and 64QAM			
Subcarrier Frequency Spacing	10.94 kHz			
Frame Duration	5 ms			

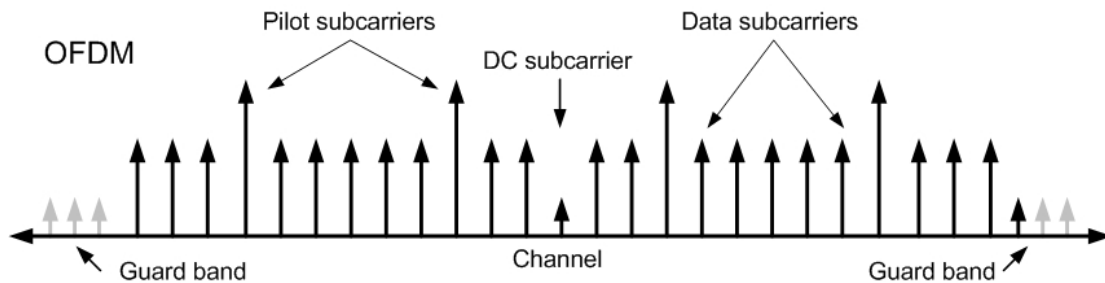


Fig. 2.6 An OFDM symbol.

the DC-subcarrier. Subcarrier spacing is 10.9375 KHz.

- Sub-channel*: It is a set of subcarriers. IEEE 802.16 OFDMA systems define two modes of sub-channel building method. In the distributed subcarrier permutation mode, subcarriers of a sub-channel are not contiguous, and their distribution is determined by the permutation types of Partial Usage of Sub-channels (PUSC) and Full Usage of Sub-channels (FUSC). In adjacent subcarrier mode, sub-channels are constituted of bins and determined the distribution by the permutation type of AMC.

IEEE 802.16e OFDMA specification supports multiple schemes for dividing the frequency-domain and time-domain resources between users, which is called sub-channelization. The relationship between the basic terms of the two-dimensional units is shown in Fig. 2.7 and brief introduced as follows.

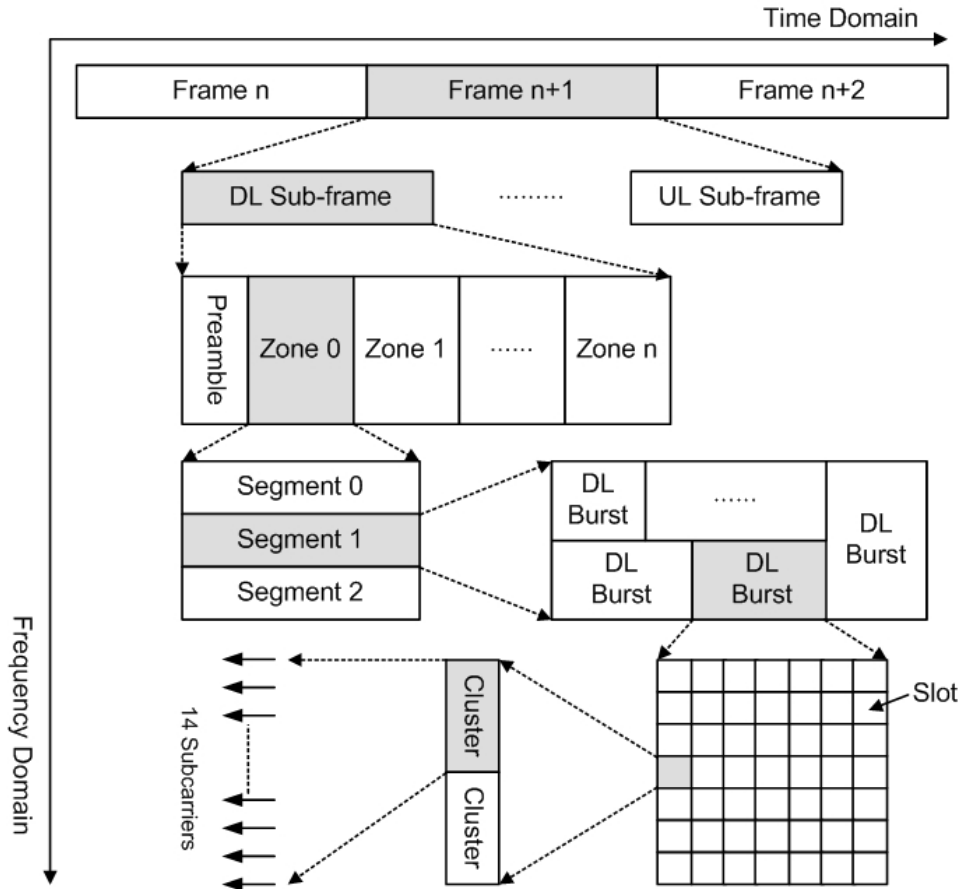


Fig. 2.7 Two-dimensional basic terms in the OFDMA data structure.

- *Frame*: It is an essential packet format of transmitted data sequence. A frame may contain both an uplink sub-frame and a downlink sub-frame.
- *Sub-frame*: It is a component to make up a frame and identified as an uplink sub-frame and a downlink sub-frame.
- *Zone*: A zone is the region of contiguous OFDMA symbols which are applied with the same permutation scheme (i.e., PUSC, FUSC or AMC). It is allowed to have different zones in a sub-frame.
- *Segment*: The set of available sub-channels form a segment which is applied with the same MAC definition. There can be three segments in a zone.
- *Burst*: It is a region which includes the contiguous sub-channel and OFDMA symbol to transmit the broadcast or unique data for

corresponding users.

- *Slot*: It is the minimum possible data allocation unit and spans both the time domain (OFDM symbol) and frequency domain (sub-channel). It contains 48 data subcarriers for all sub-channelization schemes, but their arrangement is different in different schemes [24].
- *Cluster*: It contains 14 adjacent subcarriers over 2 contiguous symbols with 4 pilot subcarriers in PUSC permutation scheme.

2.3 MIMO Systems

MIMO technology is the use of multiple antennas at both the transmitter and receiver to improve communication performance. IEEE 802.16e standard incorporates MIMO-OFDMA specification. MIMO is a current theme in wireless communication research since it can offer significant increases in data throughput, coverage range, and system reliability without additional bandwidth or transmit power. Nevertheless, these advantages usually conflict with one another, for example, increasing the data throughput will restrict the reliability improvement. Depending on application requirements, the MIMO technology can be divided into three main schemes, beamforming [25], spatial multiplexing (SM) [26] and space-time coding (STC) [27] as introduced below.

Beamforming technology achieves the spatial selectivity by using adaptive or fixed receive/transmit directional patterns for directional signal transmission or reception. The same signal is emitted from each transmit antennas with appropriate weighting and phasing such that the signal power is maximized at the receiver input. Beamforming can bring the advantages of improving the signal gain from constructive combining and reducing the multipath fading effect. Conventional beamforming employs a fixed directional pattern to filter the signals. In contrast, adaptive beamforming uses an adaptive directional pattern to filter the signals with properties of the signals actually received, and it can effectively reject the unwanted noise from other directions. This process may be carried out in the time or frequency domains. Note that beamforming requires knowledge of CSI at the transmitter.

SM technology is used to increase the transmission rate by using multiple antennas at both transmitter and receiver. It transmits a high rate data stream by dividing the stream into multiple lower rate sub-streams and transmitting these sub-streams from different transmit antennas in the same frequency channel. If these signals arrive at the receiver antennas with sufficiently different spatial signatures, the receiver can separate these streams. Spatial multiplexing can efficiently increase channel capacity at higher signal to noise ratio (SNR).

STC technology is to use multiple transmit antennas to improve the reliability of data transmission in wireless communication systems. It relies on transmitting multiple and redundant copies of the transmitted data stream to the receiver in the hope of the survival physical path between transmission and reception in a good enough channel state to allow reliable decoding. Space time codes can be separated into two main types, STBC and space-time trellis code (STTC). STBC is similar to block codes and provides only diversity gain but has much less complexity of implementation than STTC. STTC distributes a trellis code over multiple antennas and multiple time slots and provides both coding gain and diversity gain.

More details on these schemes can be found in [27]. Herein, we focus in Alamouti's STBC in two transmit-antenna systems which will be introduced in Chapter 3.

2.4 IEEE 802.16e OFDMA Specification

IEEE 802.16e includes multiple PHY specifications such as, SC, SCa, OFDM, and OFDMA, for providing different channel conditions and applications. Herein, the OFDMA specification that supports the multi-antenna technology is followed in this dissertation.

2.4.1 Frame Structure

In the licensed bands, IEEE 802.16 systems can support time-division duplexing (TDD) and frequency-division duplexing (FDD). The other license-exempt bands, the duplexing method shall be TDD. The frame can be composed of several zones that are

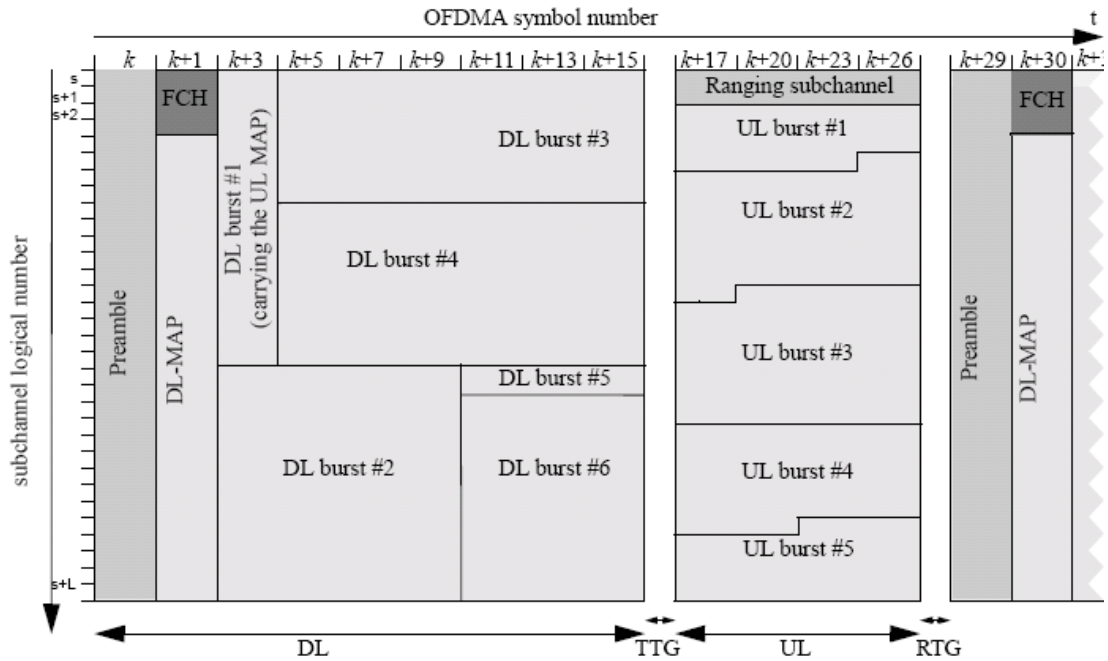


Fig. 2.8 An OFDMA frame in TDD mode [8].

divided according to subcarrier allocation methods or MIMO modes. In an FDD frame structure, the downlink (DL) and uplink (UL) sub-frames are allocated in a different frequency band without the transmit transition gap (TTG) and receive transition gap (RTG). In a TDD frame structure, the DL and UL sub-frames are allocated respectively on the same frequency band but at different time. Fig. 2.8 shows an example of an IEEE 802.16e OFDMA frame structure in TDD mode, which is a model frequently referred.

The frame structure consists of the following elements. The first symbol of a transmitted frame is a preamble symbol which is the known pattern in the receiver for the use of cell search, synchronization and channel estimation. Following the preamble symbol, the frame control header (FCH) with fixed size is transmitted for resource allocation such as the subcarriers used, length of DL-MAP and DL_Frame_Prefix. The quadrature phase shift keying (QPSK) modulation with code rate 1/2 and four repetitions is used for FCH transmission to ensure the robustness and reliable performance. DL_MAP and UL_MAP following FCH message for resource allocation of the various users in DL and UL data bursts. TTG is used to give BS and subscriber station (SS) enough time to change from downlink mode to uplink mode.

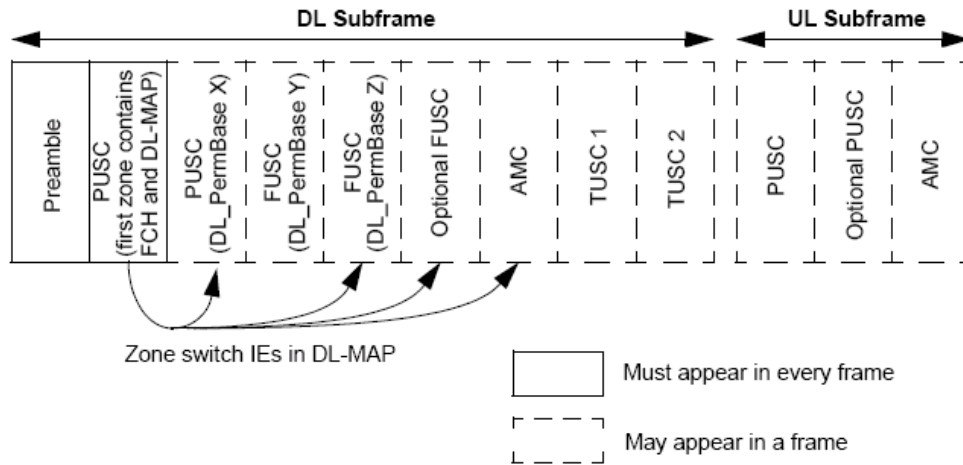


Fig. 2.9 An OFDMA frame with multiple zones [8].

For the same reason, RTG is inserted at the end of each frame. Besides, the ranging sub-channel specified in the UL_MAP message is used for contention-based bandwidth request in UL transmission.

For supporting various physical channel conditions, IEEE 802.16 OFDMA systems define two modes of sub-channel building method: the distributed subcarrier permutation mode, including PUSC and FUSC types, and the adjacent subcarrier permutation mode, including AMC type. In this dissertation, PUSC is mainly supported and will be described for the subcarrier allocation scheme in Section 3.3.3. The ratio of these modes can be flexible in the IEEE 802.16 standard. However, one burst of data transmission consists of several slots, and one slot is the minimum possible data allocation unit. In addition, the definition of this slot depends on the OFDMA symbol structure, which varies for DL and UL, for FUSC and PUSC, and for distributed subcarrier permutations and adjacent subcarrier permutation. Fig. 2.9 shows an OFDMA frame with multiple zones. Due to no information about the permutation scheme, the first zone in the DL sub-frame is essentially PUSC to ensure the FCH and DL_MAP can be received successfully. Depending on the requirements, the following zones can be applied in PUSC, FUSC, AMC, or Tile Usage of Sub-channels (TUSC). The information of zone transition is indicated in the DL_MAP and UL_MAP. The maximum number of DL zones in one DL sub-frame is eight. The maximum number of bursts to be decoded in one DL sub-frame is 64.

2.4.2 Preamble Format

The preamble symbol consists of a specific pattern known to the receiver and occupies the duration of an OFDM symbol time. It is used for frame detection, synchronization and initial channel estimation. IEEE 802.16e standard provides three types of carrier sets for different segments in the preamble symbol which can be expressed as

$$PreambleCarrierSet_s = s + 3 \cdot k \quad (2.1)$$

where $s=0,1,2$ is the index of the carrier set, and k denotes a running subcarrier index. These subcarriers in the preamble symbol are modulated by binary phase shift keying (BPSK) with a specific Pseudo-Noise (PN) code. The PN series modulating the pilots in the preamble can be found in [7]. Each segment in a zone uses one type of preamble carrier sets. For different FFT size, there are total 114 PN series to be chosen by the ID cell parameter and the segment index. The guard band subcarriers are contained both on the left and right side of the spectrum. The DC subcarrier is always be zeroed even if the type of carry set is 0. The power of the preamble subcarriers is boosted by a factor, $2\sqrt{2}$, to increase the reliability of preamble. The pilot subcarriers p_k in the preamble symbol are modulated as

$$\text{Re}\{p_k\} = 4\sqrt{2} \cdot \left(\frac{1}{2} - w_k\right), \quad \text{Im}\{p_k\} = 0 \quad (2.2)$$

where w_k denotes PN series, and $\text{Re}\{.\}$ and $\text{Im}\{.\}$ stand for the real part and the imaginary part of $\{.\}$.

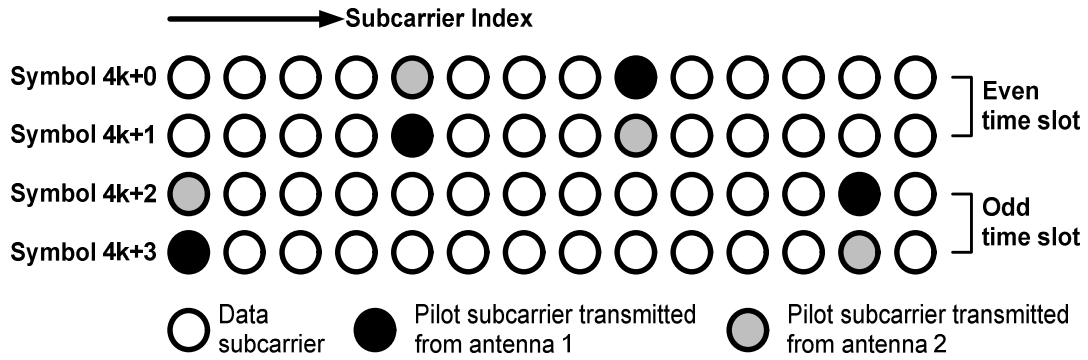


Fig. 2.10 Cluster structure.

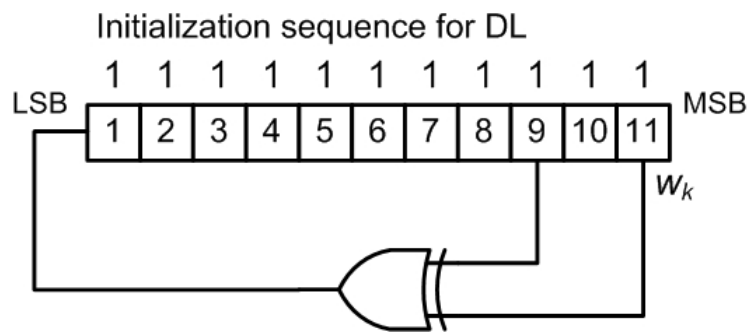


Fig. 2.11 PRBS generator for pilot modulation.



2.4.3 Pilot Modulation

The OFDM symbol structure is constructed using pilots, data, and null subcarriers. The symbol is first divided into basic clusters and null carriers are allocated. In DL PUSC mode, pilots and data carriers are allocated within each cluster as shown in Fig. 2.10. For the proposed system with two transmit antennas, when the pilot subcarrier is transmitted from one antenna, the other antenna will not transmit a pilot on the same subcarrier to avoid the inter-antenna interference. The pilot location schemes periodically change every four OFDMA symbols.

The pseudo-random binary sequence (PRBS) generator depicted in Fig. 2.11 is used to produce a sequence, w_k . Each pilot is boosted 2.5 dB over the average non-boosted power of each data subcarriers. The value of the pilot modulation, on subcarrier k , shall be derived from w_k . The pilot subcarriers p_k are modulated as

$$\operatorname{Re}\{p_k\} = \frac{8}{3} \left(\frac{1}{2} - w_k \right), \quad \operatorname{Im}\{p_k\} = 0. \quad (2.3)$$

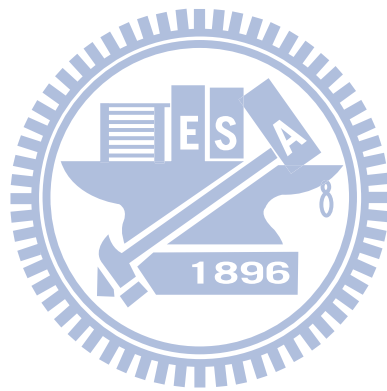
2.4.4 Basic Specification in IEEE 802.16e OFDMA

IEEE 802.16e OFDMA specification defines different transmission types according to different purposes and applications. The modulation schemes of QPSK, 16-QAM, and 64-QAM are supported for data subcarrier. The transmission types are constructed by different modulation schemes with different code rates such as QPSK 1/2, QPSK 3/4, 16-QAM 1/2, 16-QAM 3/4, 64-QAM 1/2, 64-QAM 2/3, and 64-QAM 3/4. The major parameters can be derived and described in Table 2.2.



TABLE 2.2
MAJOR PARAMETERS OF IEEE 802.16E OFDMA SPECIFICATION

Parameters		Deriving formulas
FFT Size (N)		2048, 1024, 512, and 128
Channel Bandwidth (BW)		1.25-20MHz
Sampling Factor (n)		$n=28/25$ if BW is a multiple of 1.25, 1.5, 2, and 2.75 MHz $n=8/7$ for the other cases
Ratio of CP to Useful Symbol Time (G)		1/32, 1/16, 1/8, and 1/4
Sampling Frequency (F_S)		$\text{floor}(n \cdot BW/8000) \times 8000$
Sampling Time		T_b / N
Subcarrier Spacing (Δf)		F_S / N
Useful Symbol Time (T_b)		$1/\Delta f$
Guard Time (T_g)		$G \cdot T_b$
OFDMA Symbol Duration (T_S)		$T_b + T_g$
Frame Duration (T_F)		
Number of OFDMA Symbols		$\text{floor}(T_F / T_S)$
DL PUSC	Number of Null Subcarriers (N_{Null})	
	Number of Clusters (N_C)	$(N - N_{Null})/14$
	Number of Sub-channels (N_{SC})	$N_C/2$
	Number of Pilot Subcarriers (N_{Pilot})	$N_C \times 2$
	Number of Data Subcarriers (N_{Data})	$N_C \times 12$



Chapter 3

Downlink Baseband STBC-OFDM

System Architecture

3.1 Introduction

In this chapter, the proposed downlink baseband STBC-OFDM system architecture will be described. This architecture can provide high transmission rate in IEEE 802.16e downlink communication as an alternative solution for WMAN in fixed and mobile wireless communication.

Recently, STBC-OFDM systems have received a lot of attention [28], [29] and are also adopted in IEEE 802.16e systems. Although STBC-OFDM systems with multiple antennas can provide diversity gains to improve transmission efficiency and quality of mobile wireless systems, accurate CSI is required for diversity combining, coherent detection, and decoding. Moreover, the system performance is also susceptible to the synchronization error. Therefore, synchronization and channel estimation are two crucial challenges for realizing a successful STBC-OFDM system.

Hence, a downlink baseband receiver scheme for STBC-OFDM systems is proposed and can be applied in IEEE 802.16e specification. In the proposed receiver, two main tasks, synchronization and channel estimation, are implemented. The synchronization includes symbol timing detection and carrier frequency recovery. A novel match filter is proposed to precisely detect symbol boundary, and a ping-pong algorithm is presented to effectively improve the performance of carrier frequency

synchronization [30]. Moreover, a refined two-stage channel estimation method with an initialization stage and a tracking stage is adopted to accurately estimate CSI over doubly selective fading channels [31], [32]. The initialization stage uses discrete Fourier transform (DFT)-based channel estimation method [33], [34] with the multipath interference cancellation (MPIC)-based decorrelation scheme to identify significant channel paths of the primary channel impulse responses. Then, the tracking stage uses decision-feedback (DF) DFT-based channel estimation with Newton's method [35], [36] to track the path variations of these paths.

The rest of this chapter is organized as follows. In Section 3.2, we describe the system specification of the proposed STBC-OFDM system with two transmit antennas and one receive antenna. In Section 3.3, we delineate the transmitter architecture. We then describe the proposed receiver in Section 3.4. The system simulation results are presented in Section 3.5. Finally, the conclusions of this chapter are drawn in Section 3.6.

3.2 System Specification and Design Targets

IEEE 802.16e includes multiple PHY specifications such as, SC, SCa, OFDM, and OFDMA, for providing different channel conditions and applications. The OFDMA specification that supports multi-antenna technology is adopted in the proposed STBC-OFDM system. In downlink transmission, the distributed subcarrier allocation of PUSC is supported, and the major parameters and the design targets of the proposed STBC-OFDM system are summarized in Table 3.1.

The system occupies a bandwidth of 10 MHz and operates in 2.5 GHz frequency band. The sampling frequency is 11.2 MHz. FFT size (N) is set to 1024. Each OFDM symbol is composed of 1024 subcarriers among which 720 and 120 subcarriers are used to transmit data symbols and pilots, and the remaining 184 subcarriers are used as either a DC subcarrier or virtual subcarriers. In IEEE 802.16e, the modulation schemes of QPSK, 16-QAM, and 64-QAM are supported for data subcarriers, while BPSK is adopted for pilot subcarriers and preamble symbol. In the hardware design, the data subcarrier modulation schemes target to support QPSK and 16-QAM. The length of CP is 128 sampling periods, i.e., 1/8 of the useful symbol time (T_b). Fig. 3.1 depicts the frame format which starts with one preamble symbol and is followed by

TABLE 3.1
MAJOR PARAMETERS AND DESIGN TARGETS OF THE PROPOSED STBC-OFDM SYSTEM

Parameters		Values
RF Frequency		2.5 GHz
System Channel Bandwidth (BW)		10 MHz
Sampling Frequency (F_s)		11.2 MHz
FFT Size (N)		1024
Subcarrier Spacing (Δf)		10.9 kHz
Useful Symbol Time (T_b)		91.4 μ s
Guard Time (T_g)		11.4 μ s
OFDM Symbol Duration (T)		102.9 μ s
Number of OFDM Symbols		40
PUSC	Number of Null Subcarriers (N_{NULL})	184
	Number of Pilot Subcarriers (N_{Pilot})	120
	Number of Data Subcarriers (N_{Data})	720
Design Targets		
CFO		± 7 ppm
Vehicle Speed		Up to 120 km/hr
Modulation Scheme		QSPK, 16-QAM
Uncoded Data Rate		27.32 Mbps

40 consecutive OFDM data symbols, and a time slot is equivalent to the duration of two OFDM symbols. Based on the design targets, the proposed STBC-OFDM system is defined to support the frequency offset to ± 7 ppm variation in transmitter and receiver which is equivalent to maximum CFO of 35 KHz in 2.5 GHz carrier frequency. Moreover, the proposed STBC-OFDM system is optimized to enable vehicle speed up to 120 km/hr. The maximum Doppler frequency f_d is about 0.025 (normalized to subcarrier spacing). The coherence time T_c calculating by a typical way [37] is $0.423/f_d=1.5$ ms which is about 14.8 times of an OFDM symbol period

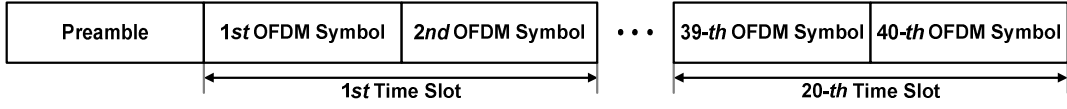


Fig. 3.1 Frame format.

and is small than a frame period. Therefore, the channel can be treated as quasi-static in several symbol times but may vary quickly during one frame transmission. As the vehicle speed further increases, ICI effect caused by Doppler spread will become more significant and cannot be ignored. When f_d is larger than 0.08 of the subcarrier spacing, the signal-to-ICI plus noise ratio is less than 20 dB [38]. Among various ICI cancellation methods, an important requirement is to accurately estimate channel variation for calculating ICI channel matrix [39]-[41]. The proposed system does not include the ICI cancellation block. However, a robust two-stage channel estimation method is adopted to precisely track channel gain variations. It is conducive to the future development of ICI cancellation to apply in higher mobile environment.

The proposed STBC-OFDM system with two transmit antennas and one receive antenna is shown in Fig. 3.2. The detail descriptions of the transmitter and receiver will be presented in the following sections.

3.3 Transmitter Architecture

As shown in Fig. 3.2 (a), in the transmitter, the transmitted data are first randomized to decrease the peak to average power ratio (PAPR) in OFDM transmission. The data are then channel encoded to resist channel effects and interleaved to avoid burst errors. Fig. 3.3 depicts the block diagrams of channel coder. After channel coding, transmitted data pass through signal mapper and then go through serial-to-parallel (S/P) to form two transmitted OFDM symbols

$$\mathbf{X}_F = \{X_F[k]: 0 \leq k \leq N-I\} \quad (3.1)$$

$$\mathbf{X}_S = \{X_S[k]: 0 \leq k \leq N-I\} \quad (3.2)$$

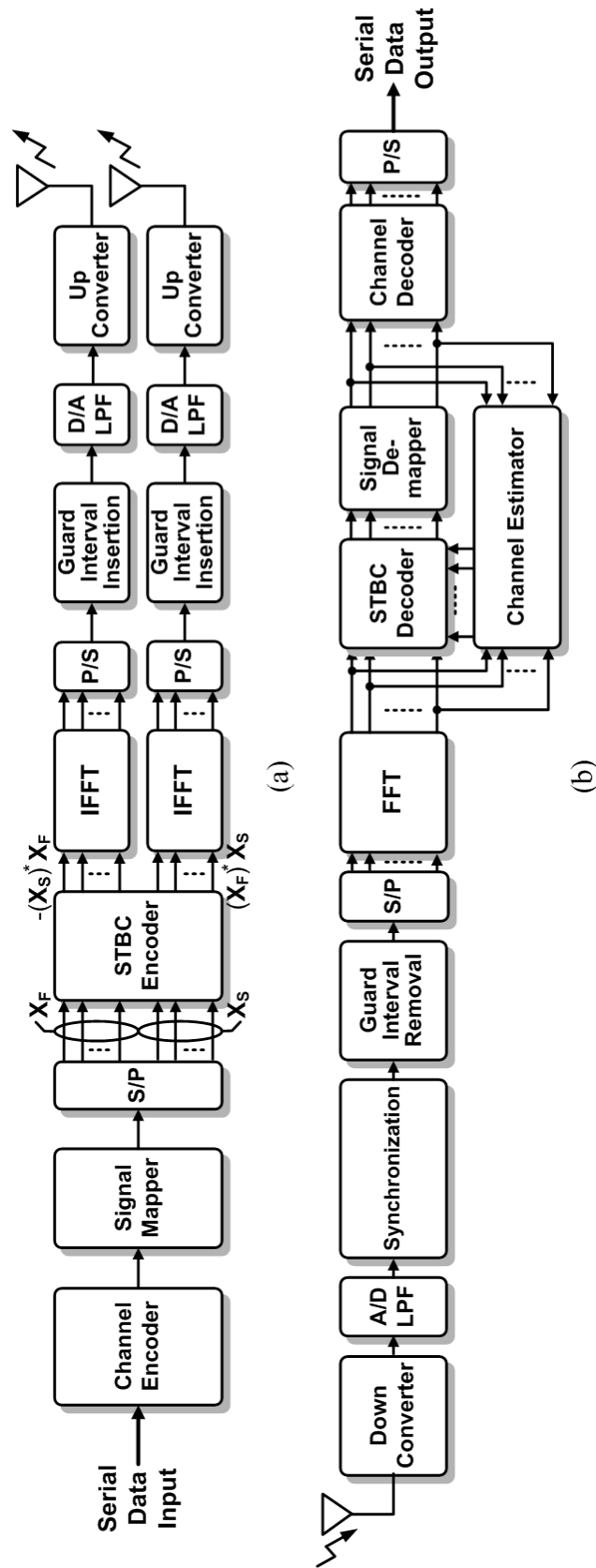


Fig. 3.2 (a) Transmitter and (b) receiver of the proposed STBC-OFDMA system with two transmit antennas and one receive antenna.

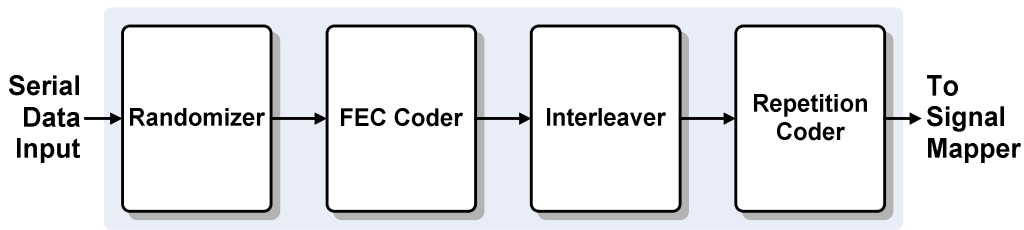


Fig. 3.3 Block diagrams of channel coder.

within a time slot. Alamouti's STBC encoding method is adopted to encode these two transmitted OFDM symbols [16]. An N -point inverse fast Fourier transform (IFFT) unit is used in each arm to transform the frequency domain OFDM symbols into time domain. After the parallel-to-serial (P/S) converters, the CP with time duration T_g is then inserted as a guard interval to combat ISI. A complete OFDM symbol with symbol duration T_b+T_g is converted into an analog signal with a digital-to-analog (D/A) converter. Finally, the analog signals are filtered by a low-pass filter (LPF), up converted to RF band, and transmitted in air.

3.3.1 Channel Coding

Channel coding helps the transmitted data to combat channel effects and avoid burst errors. As shown in Fig. 3.3, the procedure of channel coding in IEEE 802.16e includes randomization (Section 3.3.1.1), forward error correction (FEC) coding (Section 3.3.1.2), interleaving (Section 3.3.1.3), and repetition coding. Repetition coding can provide performance improvement for protecting the important control message transmission, e.g. FCH or DL-MAP. After FEC coding and interleaving, the data bits are segmented into slots, and a slot applied by repetition coding will be repeated R times to form R contiguous slots following the normal slot ordering that is used for data mapping, where $R = 2, 4, \text{ or } 6$ is the repetition code factor. This repetition scheme applies only to QPSK modulation. The other functional blocks will be discussed in the following subsections.

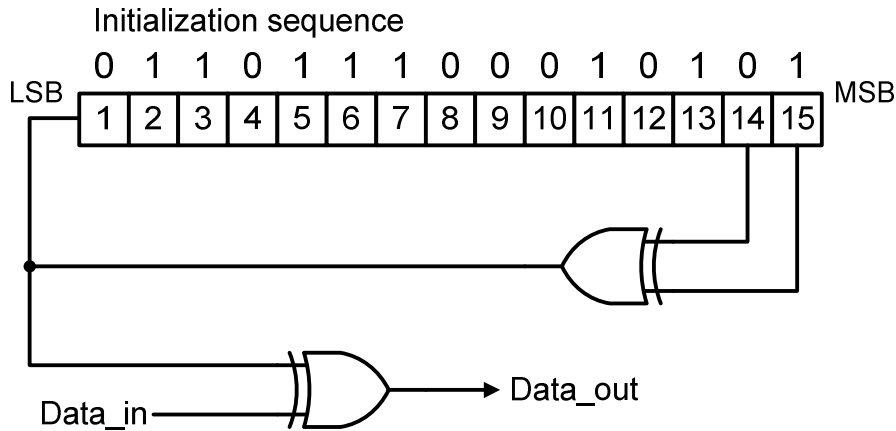


Fig. 3.4 PRBS generator for data randomization.

3.3.1.1 Randomization

Data randomization shall be performed on each data burst of the downlink and uplink, except the FCH and preamble. A randomizer is used to scramble the transmitted data and consists of a PRBS generator with the polynomial, $1 + x^{14} + x^{15}$, as shown in Fig. 3.4. The randomizer is initialized with initialization sequence “011 0111 0001 0101”. The bit stream sequentially enters the randomizer, started from MSB, to generate information bits.

3.3.1.2 FEC coding

There are several FEC coding methods are provided in IEEE 802.16e OFDMA specification such as convolutional codes (CC), block turbo codes (BTC), convolutional turbo codes (CTC) and low density parity check codes (LDPC). The mandatory coding scheme is the tail-biting convolutional coding, and the others are optional modes. Fig. 3.5 shows the CC encoder which has a code rate of 1/2 and a constraint length $k = 7$ and uses the following generator polynomials to derive its two coded bits X and Y .

$$G_I = (171)_{\text{OCT}} \quad \text{for } X \quad (3.3)$$

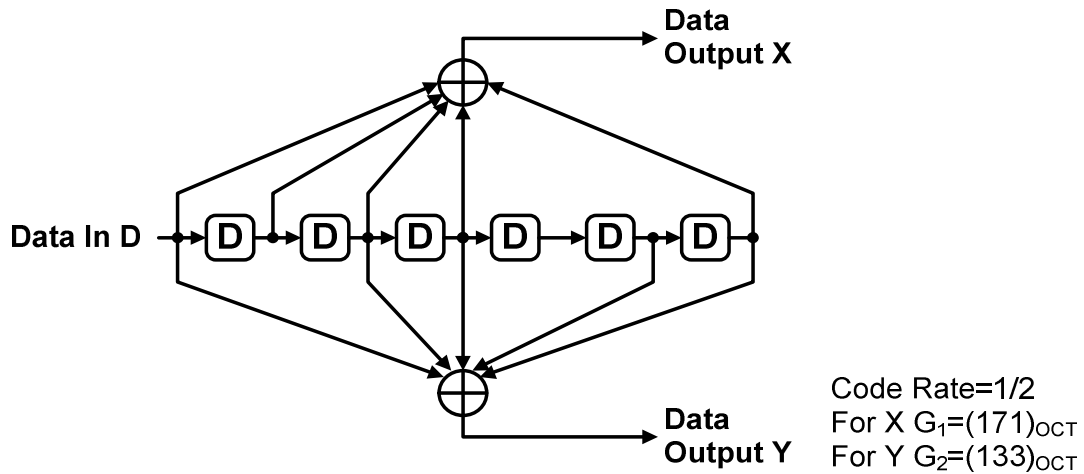


Fig. 3.5 Convolutional encoder of code rate 1/2.

TABLE 3.2
PUNCTURE CONFIGURATION

Code Rates	1/2	2/3	3/4
X	1	10	101
Y	1	11	110
Output	X_1Y_1	$X_1Y_1Y_2$	$X_1Y_1Y_2X_3$

$$G_2=(133)_{\text{OCT}} \text{ for } Y \quad (3.4)$$

Table 3.2 defines the puncturing patterns and serialization order that shall be used to realize different code rates. In the table, “1” means a transmitted bit and “0” denotes a removed bit.

3.3.1.3 Interleaving

Interleaving is used to combat the burst error and improve the bit error rate of FEC coding. All coded data bits shall be interleaved by a block interleaver with a block size corresponding to the number of coded bits per the encoded block size N_{cbps} .

A two-step permutation is adopted. The first step maps adjacent coded bits onto nonadjacent subcarriers as defined by (3.5). The second step alternately maps adjacent coded bits onto less or more significant bits of the constellation as defined by (3.6). The value N_{cpc} is the number of coded bits per subcarrier, i.e., 2, 4, or 6 for QPSK, 16-QAM or 64-QAM, respectively. The value of s is $N_{cpc}/2$. Within a block of N_{cbps} transmitted bits, k is the index of the coded bit before the first permutation, m_k is the index of that coded bit after the first and before the second permutation, and j_k is the index after the second permutation. Beside, $d = 16$ is the modulo used for the permutation. The first permutation is defined as

$$m_k = \left(N_{cbps} / d \right) \cdot k_{\text{mod}(d)} + \text{floor}(k/d) \quad (3.5)$$

and the second permutation is defined as

$$j_k = s \cdot \text{floor}(m_k/s) + \left(m_k + N_{cbps} - \text{floor}(d \cdot m_k / N_{cbps}) \right)_{\text{mod}(s)} \quad (3.6)$$

for $k = 0, 1, \dots, N_{cbps}-1$.

3.3.2 Signal Mapping

After interleaving, transmitted data bits serially enter to the signal mapper and then are converted into complex data symbols. IEEE 802.16e supports three constellation schemes, QPSK, 16-QAM, and 64-QAM as shown in Fig. 3.6. The constellation is Gray mapped to reduce the error bits of incorrect detected symbols. Moreover, the constellations shall be normalized by multiplying each constellation point with the normalized factor c to achieve equal average power.

3.3.3 Subcarrier Allocation

In IEEE 802.16e OFDMA systems, interference averaging is realized by the

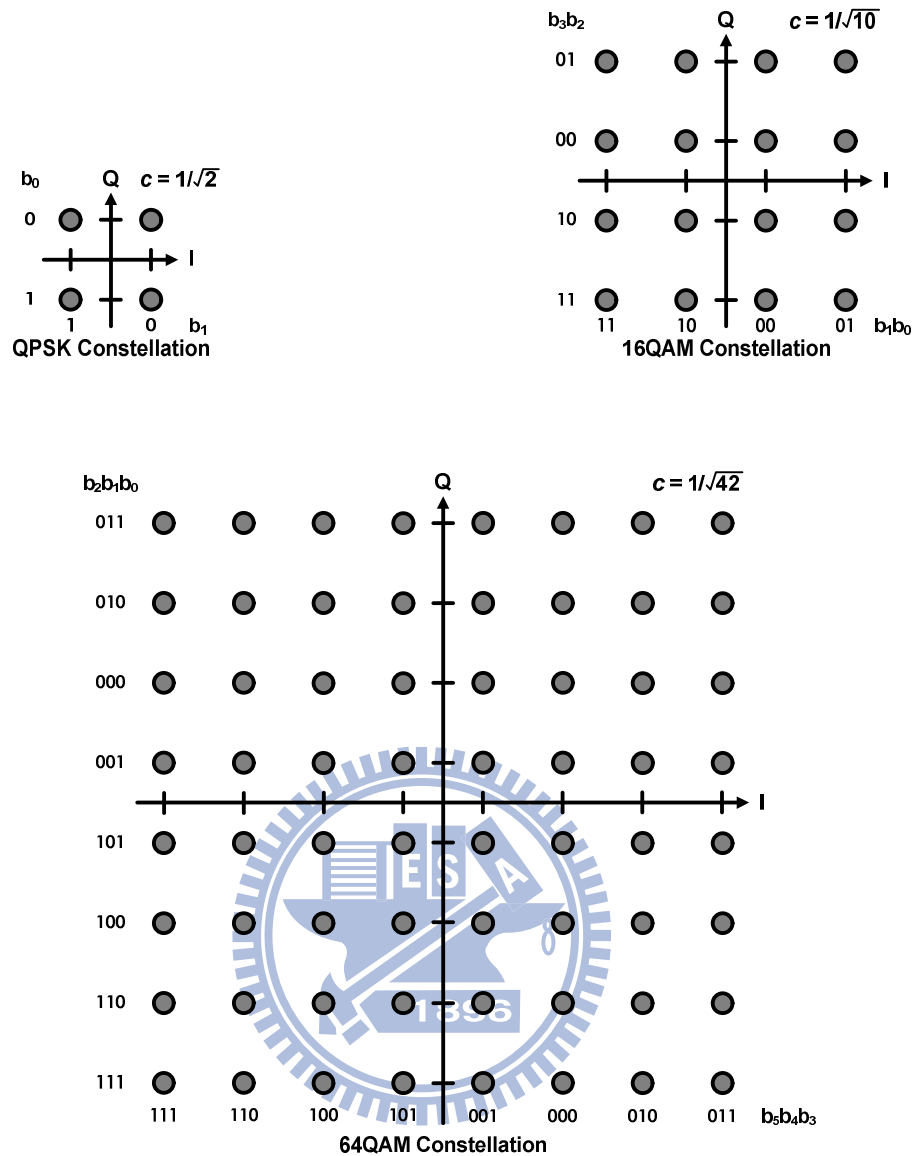


Fig. 3.6 QPSK, 16QAM and 64QAM constellations.

distributed subcarrier permutation which is designed to assign subcarriers pseudo randomly across the entire transmission spectrum, and users all have cross-spectrum subcarrier assignment. In-band interference and frequency selective fading affect all users evenly. This permutation method is adopted by PUSC and FUSC.

In PUSC, some of the sub-channels are allocated to one transmitter to form a segment, and remaining sub-channels are attributed to different segments. Each segment is assigned to different sector. PUSC is used both in UL and DL. FUSC is

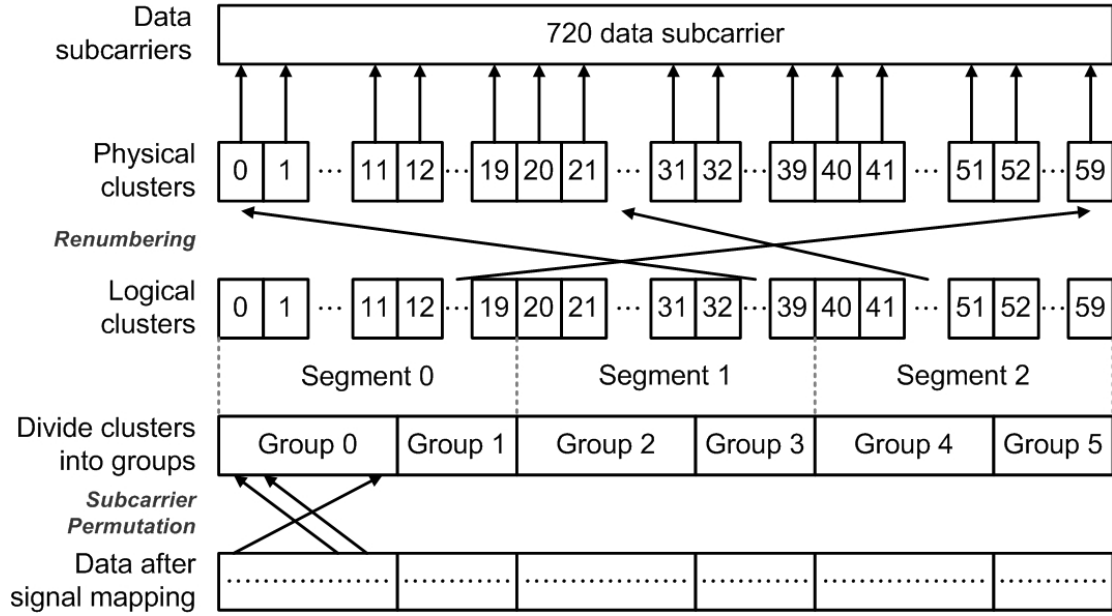


Fig. 3.7 Subcarrier allocation of PUSC (FFT size=1024).

that all sub-channels are allocated to one transmitter, and it is used only in DL. This thesis adopts PUSC, and the subcarrier allocation is described as follows. As shown in Fig. 3.7, the subcarrier allocation to sub-channels is performed using the following procedure.

1. The subcarriers are divided into the number of clusters ($N_{clusters}$) of physical clusters, and each physical cluster contains 14 adjacent subcarriers which are 12 data subcarriers and two pilot subcarriers. The number of clusters, $N_{clusters}$, varies with FFT sizes. For example, there are total 60 physical clusters with 1024-point FFT.
2. Using the equation given below, the physical clusters are renumbered to form logical clusters, and this process is also called outer permutation.

$$\begin{aligned}
 \text{LogicalCluster} = \\
 \text{RenumberingSequence}(PC + 13 \cdot DL_PermBase) \bmod N_{clusters}
 \end{aligned}
 \tag{3.7}$$

where PC denotes physical clusters. The renumbering sequence is given in [8].

The two adjacent logical clusters are not contiguous in frequency band.

3. The logical clusters are partitioned into six major groups depending on the FFT size. Group 0 includes clusters 0-11, group 1 includes clusters 12-19, group 2 includes clusters 20-31, group 3 includes clusters 32-39, group 4 includes clusters 40-51, and group 5 includes clusters 52-59. These groups may be allocated to segments, if a segment is being used, then at least one group shall be allocated to it (by default group 0 is allocated to sector 0, group 2 is allocated to sector 1, and group 4 to is allocated sector 2).
4. The sub-channel extracts one subcarrier from each of these groups. The exact subcarrier allocation is according to the permutation rule as

$$subcarrier(k,s) = N_{subchannel} \cdot n_k + \{p_s[n_k \bmod N_{subchannels}] + DL_PermBase\} \bmod N_{subchannels} \quad (3.8)$$

where $subcarrier(k,s)$ is the subcarrier index k in sub-channel s , n_k is equal to $(k+13 \cdot s) \bmod N_{subcarriers}$, $N_{subchannel}$ is the number of sub-channels, $p_s[j]$ is the series obtained by rotating basic permutation sequence cyclically to the left s times, and $DL_PermBase$ is an integer ranging from 0 to 31.

3.3.4 STBC Encoding

In IEEE 802.16e OFDMA specification, STBC proposed by Alamouti in 1998 can be used in DL to provide transmit diversity. In the proposed STBC-OFDM system, there are two transmit antennas on the transmitter and one reception antenna on the receiver as shown in Fig. 3.2. This scheme requires multiple-input-single-output channel estimation, and decoding is very similar to maximum ratio combining. Within a time slot (the duration of two OFDM symbols), two transmitted OFDM symbols, \mathbf{X}_F and \mathbf{X}_S , are encoded and transmitted from different antenna as given in Table 3.3, where the superscript * stands for complex conjugate.

TABLE 3.3
STBC ENCODING

Transmit Antenna	Within a Time Slot	
	1 st Symbol Time	2 nd Symbol Time
Antenna 1	\mathbf{X}_F	$-\mathbf{X}_S^*$
Antenna 2	\mathbf{X}_S	\mathbf{X}_F^*

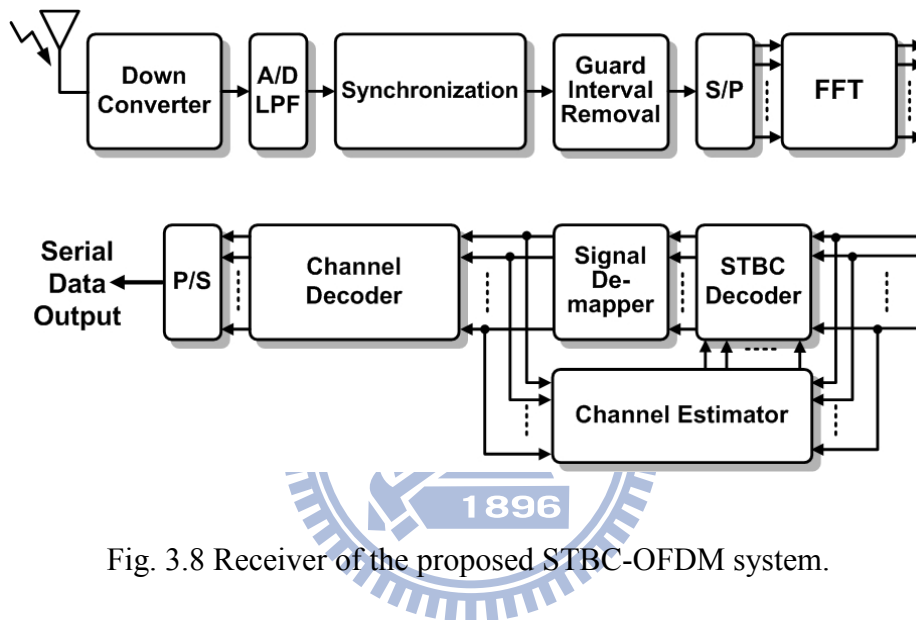


Fig. 3.8 Receiver of the proposed STBC-OFDM system.

3.4 Receiver Architecture

Fig. 3.8 shows the baseband receiver architecture for the proposed downlink STBC-OFDM communication system. The proposed baseband receiver consists mainly of synchronization, channel estimation, STBC decoding, and signal detection. The channels are assumed to be quasi-static within any two successive OFDM symbol durations. Hence, without loss of generality, the received signal processing is focused on each time slot, and the time index of symbol transmission is omitted hereafter except otherwise stated. After a RF signal is received from an antenna, it is down converted to the equivalent baseband, low-pass filtered, and digitized by an

analog-to-digital (A/D) converter. The digitized received signal is then synchronized with the symbol timing and carrier frequency to achieve the initialization in the receiver. Afterwards, when the synchronization is completed, the channel estimation estimates the channel responses, and the received signal is STBC decoded and de-mapped to retrieve the binary bits. Finally, data bits are passed to the channel decoder. The detail of the proposed receiver architecture will be described in the following subsections.

3.4.1 Synchronization

Synchronization is a critical problem in OFDM systems. The precise carrier and sampling clock frequencies are required to help the receiver to retrieve data at correct timing. However, in the transmitter and receiver, two independent crystal oscillators are used to generate the carrier frequency and sampling clock, and the oscillator mismatch results in the offsets of carrier frequency and sampling clock. Besides, the symbol timing must be derived from the received signals. In mobile environment, the Doppler effect causes the frequency shift on the received signals. Various synchronization errors in OFDM signals are defined as symbol boundary offset, sample clock offset (SCO), carrier frequency offset (CFO), and carrier phase noise. All these unavoidable problems affect the validity of synchronization detection; therefore, the carrier and timing recovery must be used in the receiver to solve these problems.

Based on the operations of the IEEE 802.16e system, there are two synchronization stages. First, the system acquisition stage is used to acquire the preamble symbol of the transmission when a mobile device first enters an 802.16 network. It includes the coarse symbol boundary detection, the coarse CFO estimation, and the preamble search. Then, in the transmission stage, the receiver starts to execute data reception. Our proposed system is focused on the transmission stage. The synchronization architecture includes symbol timing synchronization, sample clock and carrier frequency synchronization as shown in Fig. 3.9. The proposed synchronization scheme performs symbol boundary detection and CFO estimation as presented in the following subsections.

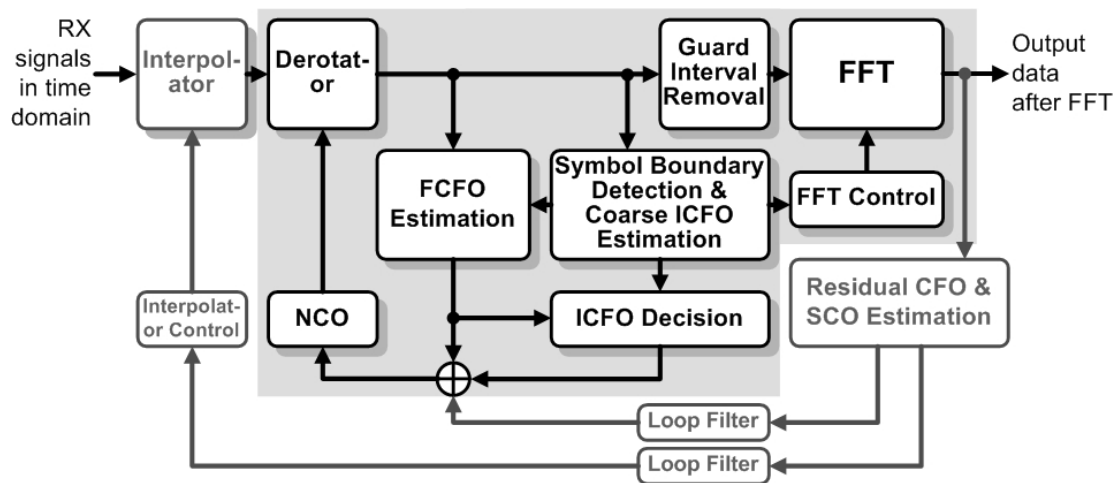


Fig. 3.9 Synchronization architecture.

3.4.1.1 Symbol Timing Synchronization

Symbol timing synchronization refers to the task of finding the symbol boundary of the individual transmitted OFDM symbols, and this information will be used to define the FFT window. If the detected OFDM symbol boundary is not correct, the performance of the receiver will be degraded.

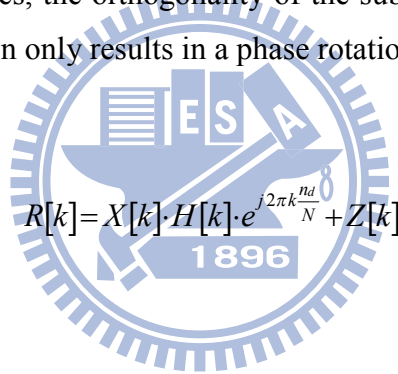
Effects of Symbol Timing Offset

In wireless multipath fading channels, the transmitted signals pass through different paths to arrive at the receiver, so the receiver receives the transmitted signals which are overlapped with different delayed versions of the signals. Assume that the maximum channel excess delay is small than the length of the guard interval and a symbol timing offset n_d (normalized to one sampling clock period) is existed in the symbol boundary detection. When the detected symbol boundary locates in the channel impulse response region or is later than the actual boundary, the data in the FFT window contains the components from the later or previous OFDM symbols, and the ISI effect is introduced. Also, the received signal is attenuated and has a phase rotation [42].

$$R[k]=X[k]\cdot H[k]\cdot\left(\frac{N-n_d}{N}\right)\cdot e^{-j2\pi k\frac{n_d}{N}}+Z[k]+ISI \quad (3.9)$$

where k is the subcarriers index, N is the number of subcarriers in an OFDM symbol, $R[k]$ is the received OFDM symbol, $X[k]$ is the transmitted OFDM symbol, $H[k]$ is the channel frequency response, and $Z[k]$ is the Gaussian noise. The term of $\exp(j\cdot 2\pi\cdot k\cdot n_d/N)$ causes a phase rotation to the subcarrier constellations.

Based on the requirements for symbol timing offset, an ISI free region of symbol timing detection is determined by the difference in length between the CP and the channel impulse response region. The detected symbol boundary must be located in an ISI free region, which is not affected by the previous symbol due to channel dispersion [43], [44]; besides, the orthogonality of the subcarriers is preserved, and a time offset within this region only results in a phase rotation.



$$R[k]=X[k]\cdot H[k]\cdot e^{j2\pi k\frac{n_d}{N}}+Z[k] \quad (3.10)$$

Symbol Boundary Detection Scheme

Since the proposed STBC-OFDM system has two transmit antennas, the signals transmitted from different antennas may arrive at the receiver with different delays due to the multipath effect as shown in Fig. 3.10. Therefore, the detected boundary must locate in the common ISI free region to prevent the respective ISI effects from the other symbols.

Because the time-domain preamble symbol is not exactly periodic, the delay correlation method cannot be used to precisely detect the symbol boundary. Since the preamble symbol is a known sequence after the system acquisition, a match filter corresponding to the time-domain preamble sequence is applied to match the received sample sequence and find the peak of matching results to obtain an accurate symbol boundary. The complexity of the match filter is depended on the coefficient length;

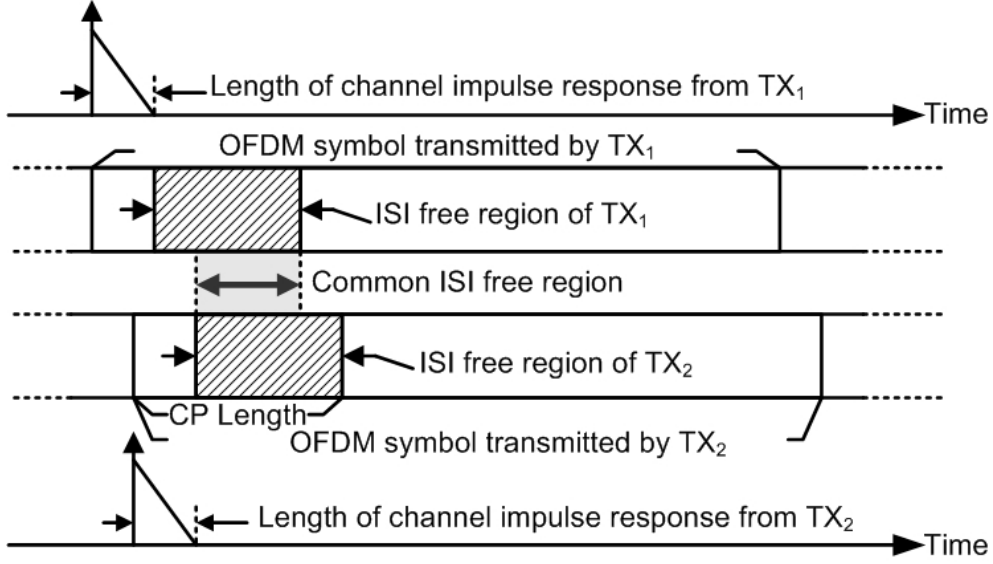


Fig. 3.10 Common ISI free region of two transmit antennas.

hence, a suitable length (L) is used to take both performance and complexity into consideration. Matching the received sample sequence $r[n]$ and a known coefficient sequence $p^{(i)}[m]$ which is the L -sample time-domain preamble sequence transmitted from i -th antenna, the symbol boundary can be found as

$$\text{SymbolBoundary}_n = \max_n \left(\sum_{m=0}^{L-1} r[n+m] \cdot (p^{(i)}[m])^* \right) \quad (3.11)$$

where n is the sample index, and m is the coefficient index. When the sample index n corresponds to the peak of the matching results, then the symbol boundary is found.

However, the mismatch of oscillator frequency in receiver and transmitter causes CFO effects to the received signals and destroys the characteristic of the matching results. In order to improve the validity of the matching results, we propose a modified match filter. Since the filter coefficient sequence is the known preamble sequence, it can be pre-shifted with the possible values of integer CFO (ICFO). An output peak will appear in the detected boundary index when matching with the preamble sequence compensated with the corresponding ICFO value; hence, this match filter has an additional advantage that the coarse ICFO can be detected

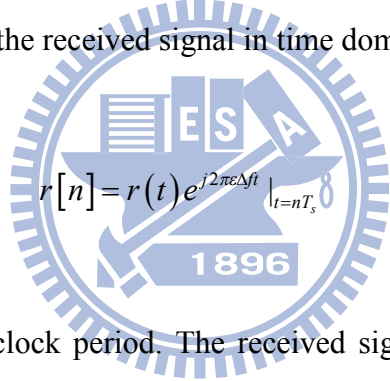
simultaneously.

3.4.1.2 Carry Frequency Synchronization

Since the transmitter and receiver generate the carrier frequency independently, the oscillator mismatch results in CFO. OFDM systems are very sensitive to synchronization particularly in CFO which destroys the orthogonal characteristic of subcarriers; hence, CFO must be precisely estimated and compensated to maintain the properties of OFDM symbols.

Effects of Carry Frequency Offset

The CFO (ε), normalized to one subcarrier spacing Δf , is divided into two values: a value of ICFO (ε_I) and a value of fractional CFO (FCFO) (ε_F), and $\varepsilon = \varepsilon_I + \varepsilon_F$ with $-0.5 \leq \varepsilon_F < 0.5$. With CFO, the received signal in time domain $r[n]$ is given by

$$r[n] = r(t) e^{j2\pi\varepsilon\Delta f t} \Big|_{t=nT_s} \quad (3.12)$$


where T_S is the sampling clock period. The received signal in frequency domain is given by

$$R[k] = X[k - \varepsilon_I] H[k - \varepsilon_I] \frac{\sin(\pi\varepsilon_F)}{N \sin(\pi\varepsilon_F/N)} e^{j2\pi \frac{i(N+N_g)+N_g(\varepsilon_I+\varepsilon_F)}{N}} e^{j\pi \frac{N-1}{N} \varepsilon_F} + I[k] + Z[k] \quad (3.13)$$

where i is the symbol index, N_g is the length of CP and $I[k]$ is the ICI term. FCFO leads to additional attenuation, phase rotation and ICI effect, while ICFO causes the subcarriers index shifting with an integer value. Moreover, the phase shift is identical at every subcarrier and also a function of the symbol index. The values of ICFO and FCFO are separately estimated by using different schemes as discussed in the following subsections.

Fractional Carrier Frequency Offset Estimation

After symbol boundary has been successfully obtained, the information of the CP is known. According to [45], the repeating characteristic of the CP can be used to estimate FCFO by correlating the CP with the corresponding received sample sequence. The correlation result is given by

$$\Lambda = \sum_{m=0}^{N_g-1} r^*[d+m]r[d+m+N] \quad (3.14)$$

where $d = 0 \dots N_g-1$ is the sample index of CP and N_g is the length of CP. Based on [46], the maximum likelihood estimation of FCFO can be derived as

$$\hat{\theta} = \tan^{-1} \left(\frac{\text{Im}\{\Lambda\}}{\text{Re}\{\Lambda\}} \right) = 2\pi \cdot \varepsilon_F \cdot \Delta f \cdot N \cdot T_s \quad (3.15)$$

$$\varepsilon_F = \frac{1}{2\pi} \cdot \hat{\theta} \quad (3.16)$$

where $\hat{\theta}$ is the rotation phase.

Integer Carrier Frequency Offset Estimation

In order to improve the accuracy of the proposed symbol boundary detection, the coefficient sequence in the proposed match filter is pre-shifted with the possible values of ICFO. Thus, the proposed symbol boundary detection has an additional advantage that the coarse ICFO can be detected simultaneously.

However, when the value of CFO is in the middle of two integer values, the accuracy of ICFO detection by using the match filter method substantially decreases. Since the ICFO effects caused by these two integer values are almost the same, there are two undistinguishable peaks in the matching results. For example, Fig. 3.11 shows the simulation with the CFO value of 0.49 subcarrier spacing, and the correct ICFO

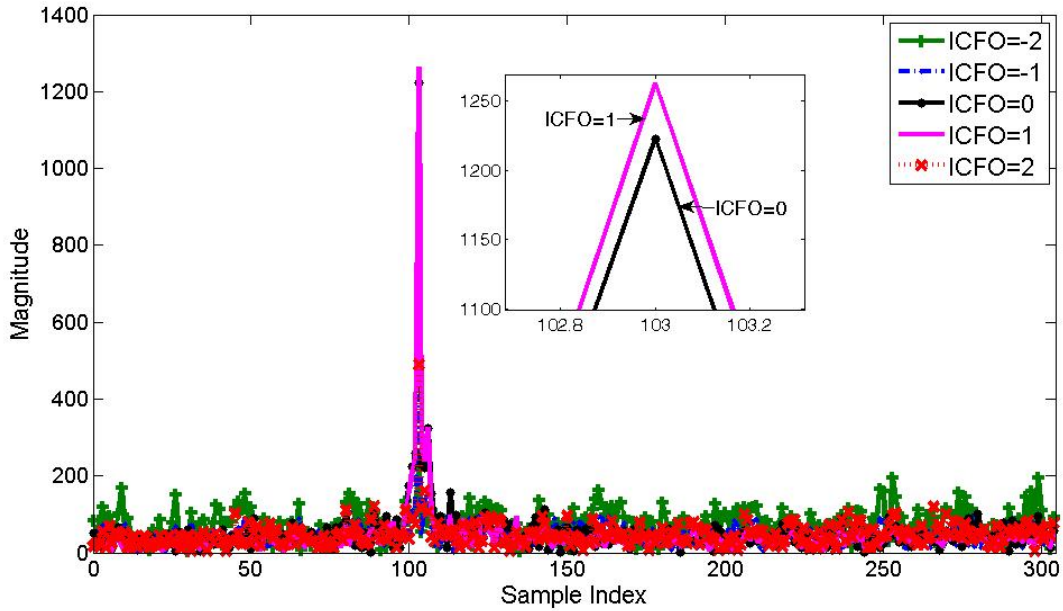


Fig. 3.11 Matching results when CFO is 0.49 subcarrier spacing.

value is 0. However, if we only consider the maximum peak of the matching results, the ICFO would be detected to be 1. The undistinguishable peaks caused by the interference of another antenna and noise may easily result in the wrong ICFO detection.

Therefore, a ping-pong algorithm is proposed to improve the performance of ICFO detection [30]. The ping-pong algorithm partitions each CFO region into the strong region and the weak region depending on the distance to each integer value, as shown in Fig. 3.12. Through the accurately estimated FCFO, this FCFO value can be used to correct the ICFO detection. When the estimated FCFO locates in the strong region, the matching results have strong reliability to determine ICFO by detecting the peak value. When the estimated FCFO locates in the weak region, there are two possible peaks in the matching results; thus, the ICFO value will be adjusted by the value of FCFO and these two peaks.

Observing the example in Fig. 3.11, there are two peaks appearing in the ICFO values of 0 and 1, and the largest matching result is in the ICFO values of 1. Assume that the FCFO value is correctly estimated to be 0.49 which is located in the weak region. If the ICFO value is determined to be 1 which is directly depending on the output peak, the estimated CFO will be 1.49. This result is unreasonable because

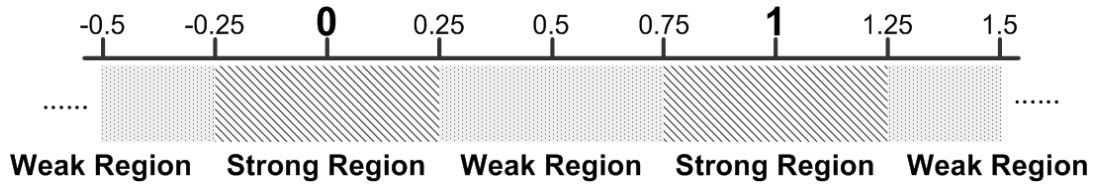


Fig. 3.12 CFO region partition for the proposed ping-pong algorithm.

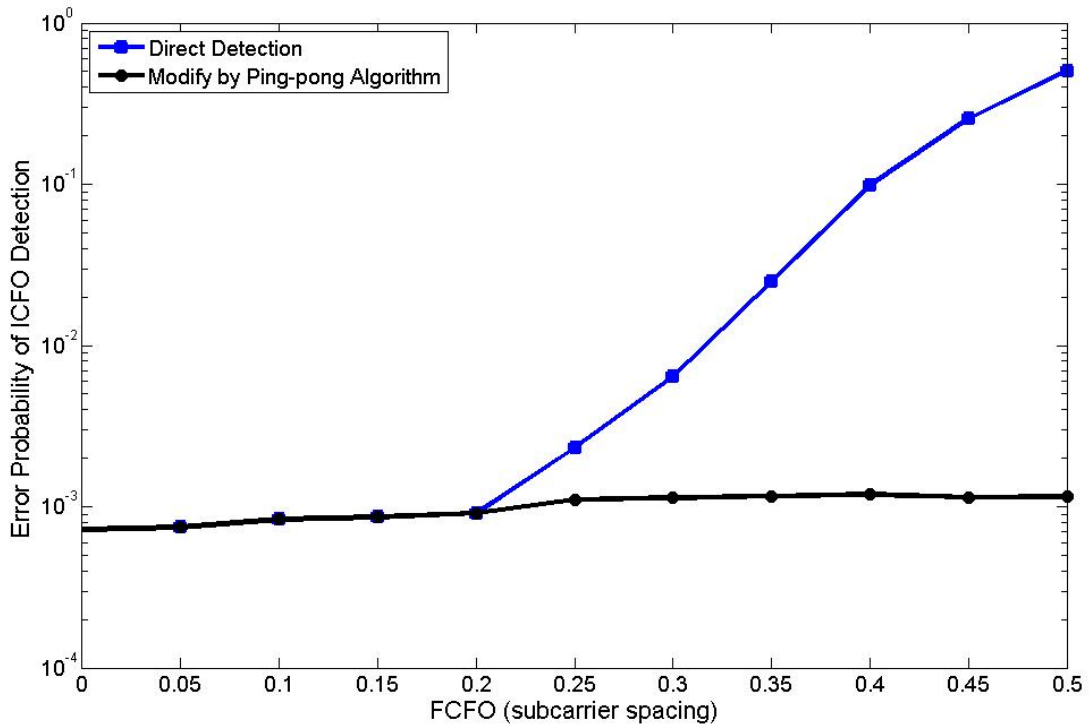


Fig. 3.13 Error probability of the ICFO detection versus FCFO at v_e of 60 km/hr with E_b/N_0 of 16 dB.

another peak is 0 but not 2. By using this ping-pong algorithm, the ICFO value will be correctly adjusted to be 0.

Fig. 3.13 shows the error probability of the ICFO detection versus different value of FCFO at the vehicle speed v_e of 60 km/hr with E_b/N_0 of 16 dB, where E_b/N_0 is defined as a ratio of received bit energy to the power spectral density of noise. The error probability of ICFO by using the direct detection of the matching results rapidly increases to about 0.5 when FCFO approaches 0.5 (the weak region), whereas it is normally about 10^{-3} . However, it is significant to see that the proposed ping-pong

scheme is effective to maintain ICFO error probability at about 10^{-3} in the weak region.

As described above, the operations of the match filter can find out the precise symbol boundary and the ICFO value. Moreover, the coefficients of the match filter, which are an L -sample preamble sequence, are quantized into $\{-1, 1\}$ values in both real and imaginary parts [47]. It can effectively reduce the complexity by using additions instead of multiplications in (3.11). Similarly, the received sample sequence can also be quantized into $\{-1, 0, 1\}$ that is represented by only two bits in the hardware implementation because it has the advantage of reducing the numbers of storing registers and computing adders. The quantization loss can be compensated by increasing the coefficient length of match filter which costs only a little hardware overhead. According to the simulation results under SNR of 9.4 dB at v_e of 120 km/hr, the correlation length of 300 is chosen, where the performance of symbol miss error probability approaching the lowest boundary. At the same vehicle speed, Fig. 3.14 shows the quantization of both coefficients and received sample sequence has almost the same performance of the case with quantization of only coefficients.

Fig. 3.15 shows the root mean square (RMS) errors of CFO which is estimated by using the proposed synchronization scheme under different vehicle speed. In the simulation, the RMS carrier frequency offset error is about 2.9×10^{-3} , 3.5×10^{-3} , 4.4×10^{-3} , and 5.7×10^{-3} times of the subcarrier spacing at v_e of 0, 30, 60, and 120 km/hr with E_b/N_0 of 16 dB. It is shown in [47] that the residual CFO must be kept below 1% of the subcarrier spacing to ensure SNR degradation smaller than 0.1 dB when using 64-QAM modulation. When using QPSK modulation, the residual CFO can be tolerated up to 5% of the subcarrier spacing.

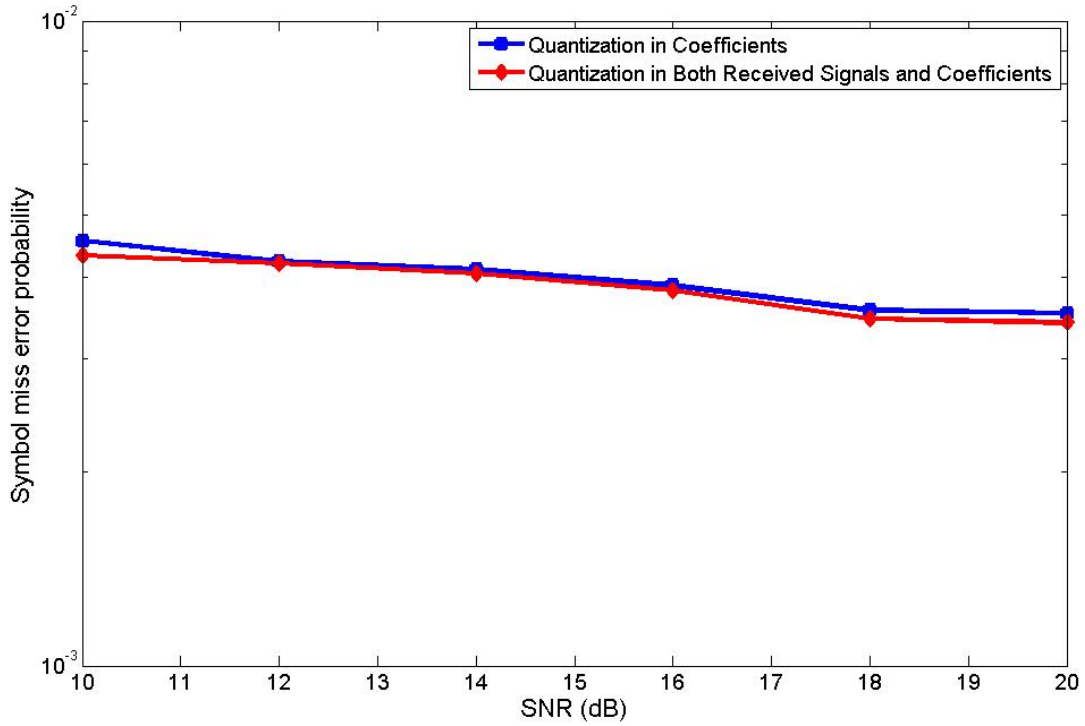


Fig. 3.14 Performance for different quantization method.

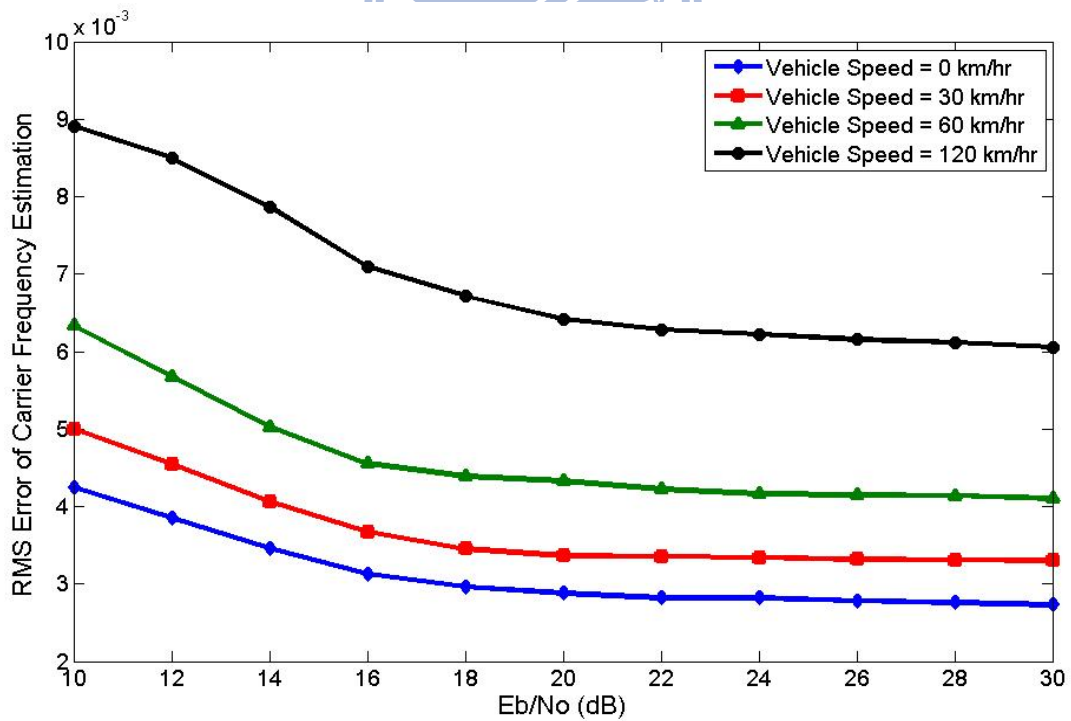


Fig. 3.15 RMS errors of CFO under different vehicle speed.

3.4.2 Channel Estimation

STBC-OFDM systems provide transmit diversity to improve communication quality of mobile wireless systems. However, the multiple antennas bring some new challenges. One of the biggest challenges is the channel estimation. Channel estimation must provide precise CSI for diversity combining, coherent detection and decoding, which is difficult to obtain especially in outdoor mobile channels. Different signals are transmitted from different transmit antennas simultaneously, and the received signal at the receive antenna is the sum of signals transmitted from all the transmit antennas; therefore, an accurate channel estimation for each pair of transmit-receive antennas is much more sophisticated than in a single antenna system.

Coherent OFDM detection requires channel estimation and tracking. The known symbols (pilots) are provided for reliable channel estimation in IEEE 802.16e systems. Two major categories of pilot-aided channel estimation methods are (a) interpolation-based channel estimation method and (b) DFT-based channel estimation method. Interpolation-based method estimates channel frequency responses by interpolating the received pilot subcarriers. However, based on the limited pilot subcarriers, accurate channel estimation of interpolation-based method is a challenge in outdoor mobile channels. Because the coherent bandwidth becomes small, the interpolation-based method becomes more difficult to estimate channel variations. DFT-based method focuses on transform domain to characterize time-domain channel impulse response (CIR) by using the DFT processing and effectively improves the performance by suppressing time-domain noise. Many DFT-based methods derived from maximum likelihood (ML) scheme have been studied for OFDM systems with preambles [33], [34]. In order to further improve the estimated performance, the decision-feedback methods employ decided data subcarriers as pilot subcarriers to track channel variations for saving transmission bandwidth and providing sufficient tracking information, which is called the DF DFT-based channel estimation method [35].

In this dissertation, we adopt a robust two-stage channel estimation method, including an initialization stage and a tracking stage, to successfully realize an STBC-OFDM system [31], [32]. The multipath fading channel is characterized by CIR consisting of a few significant paths. The delays of these paths usually vary

slowly in time, but the path gains may vary relatively fast. Therefore, within the received preamble symbol, the initialization stage uses DFT-based channel estimation method [33], [34] with the MPIC-based decorrelation scheme to identify significant channel paths of the primary channel impulse responses. Then, in the following received data symbols, the tracking stage uses DF DFT-based channel estimation with Newton's method [35], [36] to track the path variations of these paths. The detail descriptions and simulation results of the proposed two-stage channel estimation will give in Chapter 4.

3.4.3 STBC Decoding

The STBC decoder is used to decode the received signals. Two received symbols within a time slot, $\hat{X}_F[k]$ and $\hat{X}_S[k]$, should be decided. The STBC decoder is based on the latest estimated channel frequency responses to decode $\hat{X}_F[k]$ and $\hat{X}_S[k]$. The equations can be expressed as

$$\hat{X}_F[k] = \frac{(M^{(1)}[k])^* R[1, k] + M^{(2)}[k] (R[2, k])^*}{(|M^{(1)}[k]|^2 + |M^{(2)}[k]|^2)} \quad (3.17)$$

$$\hat{X}_S[k] = \frac{(M^{(2)}[k])^* R[1, k] - M^{(1)}[k] (R[2, k])^*}{(|M^{(1)}[k]|^2 + |M^{(2)}[k]|^2)} \quad (3.18)$$

where $M^{(i)}[k]$ is the latest estimated channel frequency response transmitted from the i -th antennas, $R[t, k]$ is the t -th received symbol within a time slot.

3.4.4 Channel Decoding

Channel decoding is the inverse operation of channel coding as shown in Fig. 3.3. The procedure of channel decoding is de-interleaving, FEC decoding and de-randomization. The functional blocks will be discussed in the following subsections.

3.4.4.1 De-interleaving

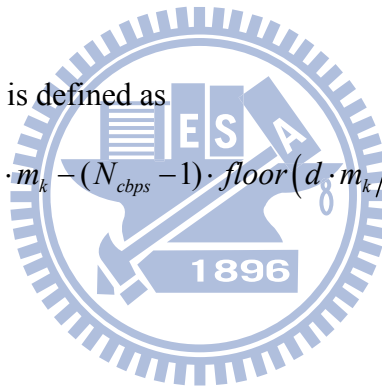
The de-interleaving is the inverse operation of the interleaving in the transmitter. A block size is corresponding to the number of coded bits per the encoded block size N_{cbps} . The value N_{cpc} is the number of coded bits per subcarrier, i.e., 2, 4, or 6 for QPSK, 16-QAM or 64-QAM, respectively. The value s is $N_{cpc}/2$. Within a block of N_{cbps} transmitted bits, j_k is the index of the received bit before the first permutation of de-interleaving, m_k is the index after the first and before the second permutation, and k is the final index after the second permutation. Beside, $d = 16$ is the modulo used for the permutation. The first permutation is defined as

$$m_k = s \cdot \text{floor}(j_k/s) + \left(j_k + \text{floor}(d \cdot j_k / N_{cbps}) \right)_{\text{mod}(s)} \quad (3.19)$$

and the second permutation is defined as

$$k = d \cdot m_k - (N_{cbps} - 1) \cdot \text{floor}(d \cdot m_k / N_{cbps}) \quad (3.20)$$

for $k = 0, 1, \dots, N_{cbps}-1$.



3.4.4.2 FEC Decoding

The Viterbi algorithm is used to decode the convolutional code. Since the puncturing is used to achieve the different code rates of 1/2, 2/3 and 3/4, a dummy bit (a zero bit) should be inserted in these removed bits. In the transmitter, the convolutional coding adopts the tail-biting method; hence, after each data block is Viterbi decoded, the block state must be recorded to be the initial state of the next data block.

3.4.4.3 De-randomization

The randomizer employs the PRBS generator to generate the pattern to do the

XOR operations with the transmitted data. The de-randomizer is based on the same PRBS generator to generate the pattern to do the XOR operations with the received data.

3.5 System Simulation

In this section, the performances of the proposed receiver are demonstrated through the simulation of the proposed STBC-OFDM system with two transmit antennas and one receive antenna. The parameters are setting based on the IEEE OFDMA specification and summarized in Table 3.1. The system occupies a bandwidth of 10 MHz. The carrier central frequency is set to 2.5 GHz with CFO= ± 7 ppm variations in the transmitter and receiver. The preamble symbols transmitted from the first and second antennas use different subcarrier sets, and the subcarrier values are set as specified in IEEE 802.16e. The multipath fading channel adopts the International Telecommunication Union (ITU) Veh-A [48] channel model with relative path power profiles of 0, -1, -9, -10, -15, and -20 (dB), and the path excess delays are uniformly distributed from 0 to 50 sample periods. Moreover, Jakes model is used to generate the Rayleigh fading in the vehicle environment [49].

Fig. 3.16 shows the uncoded bit error rate (BER) of the proposed receiver scheme for QPSK modulations at the vehicle speed v_e of 3, 60 and 120 km/hr. At v_e of 3, 60 and 120 km/hr, the uncoded BER of the proposed scheme can achieve about $6.2 \times 10^{-7}/2.2 \times 10^{-5}/1.1 \times 10^{-4}$ in E_b/N_0 of 30 dB, respectively. As we can see, at 3 km/hr which is equivalent to the maximum Doppler frequency f_D of 5.94 Hz, since the Doppler effect is small, the uncoded BER curves can smoothly decrease when the SNR increases. However, when the mobile speed become large, e.g. 60 km/hr with f_D of 138.89 Hz and 120 km/hr with f_D of 277.78 Hz, the uncoded BER will be influenced by the Doppler effects and reduced slowly. Fig. 3.17 and Fig. 3.18 also show the uncoded BER of the proposed receiver scheme for 16-QAM and 64-QAM modulations at v_e of 3, 60 and 120 km/hr. For 16-QAM modulation, the uncoded BER can achieve about $1.1 \times 10^{-4}/7.5 \times 10^{-4}/1.2 \times 10^{-3}$ at v_e of 3, 60 and 120 km/hr in E_b/N_0 of 30 dB, respectively. For 64-QAM modulation, the uncoded BER can achieve about $2.0 \times 10^{-3}/6.0 \times 10^{-3}/1.2 \times 10^{-2}$ at v_e of 3, 60 and 120 km/hr in E_b/N_0 of 30 dB, respectively.

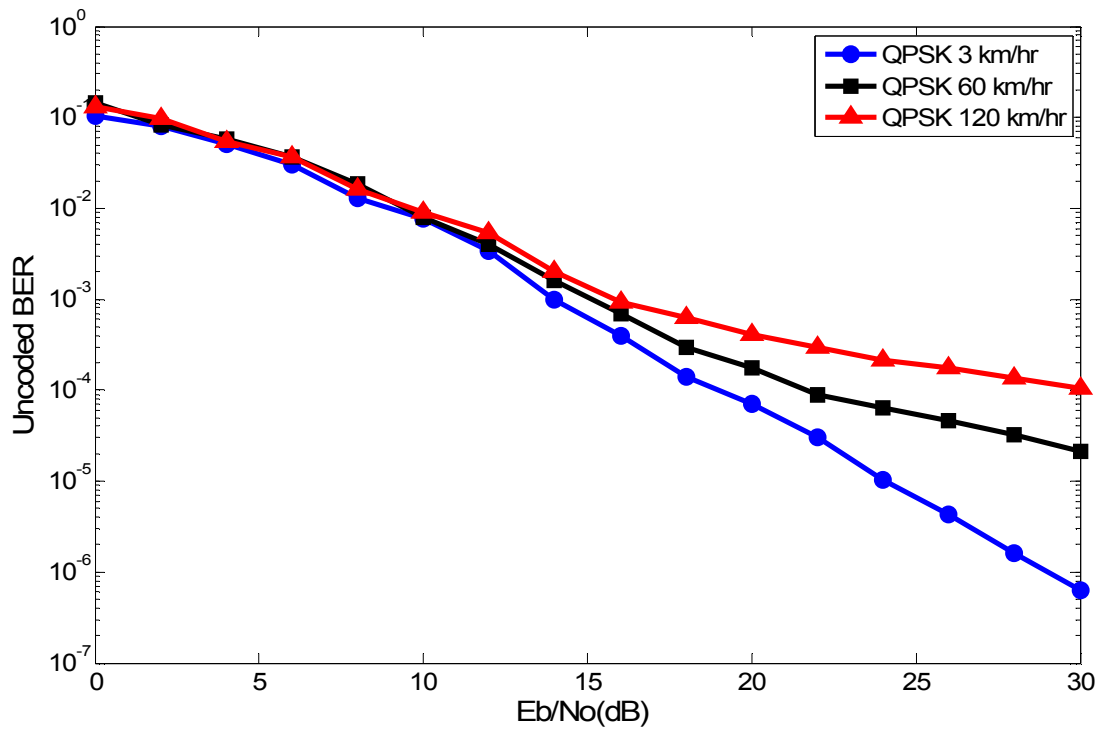


Fig. 3.16 Uncoded BER performance for QPSK modulations at v_e of 3, 60 and 120 km/hr.

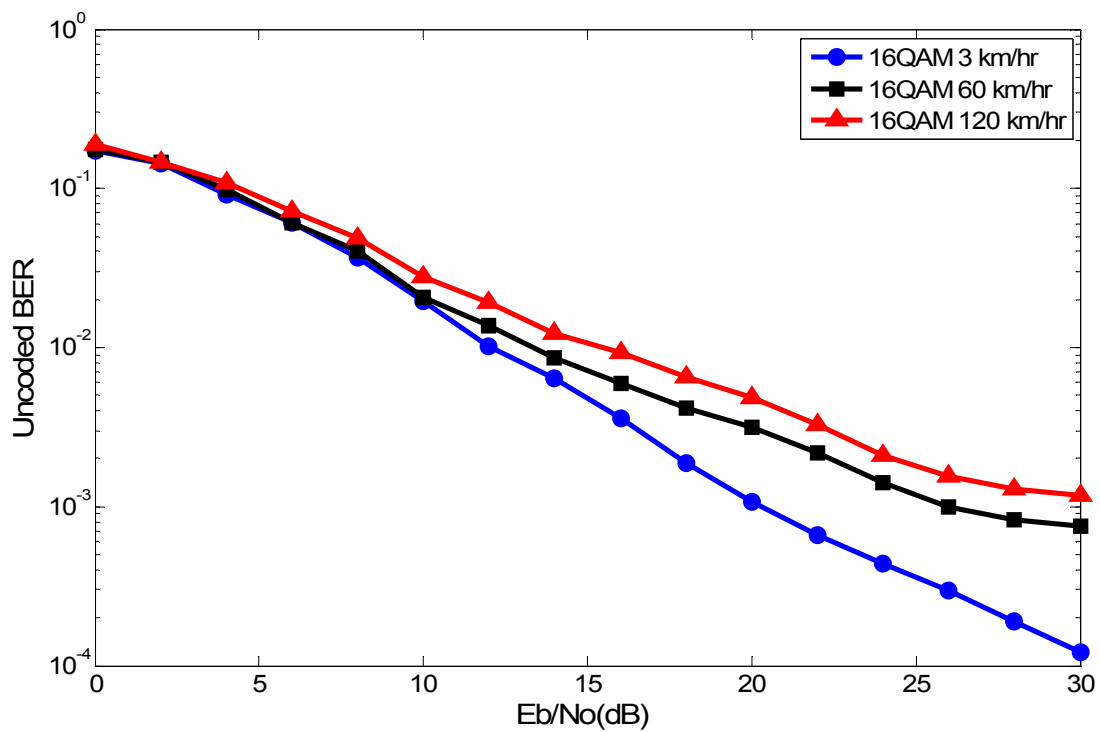


Fig. 3.17 Uncoded BER performance for 16QAM modulations at v_e of 3, 60 and 120 km/hr.

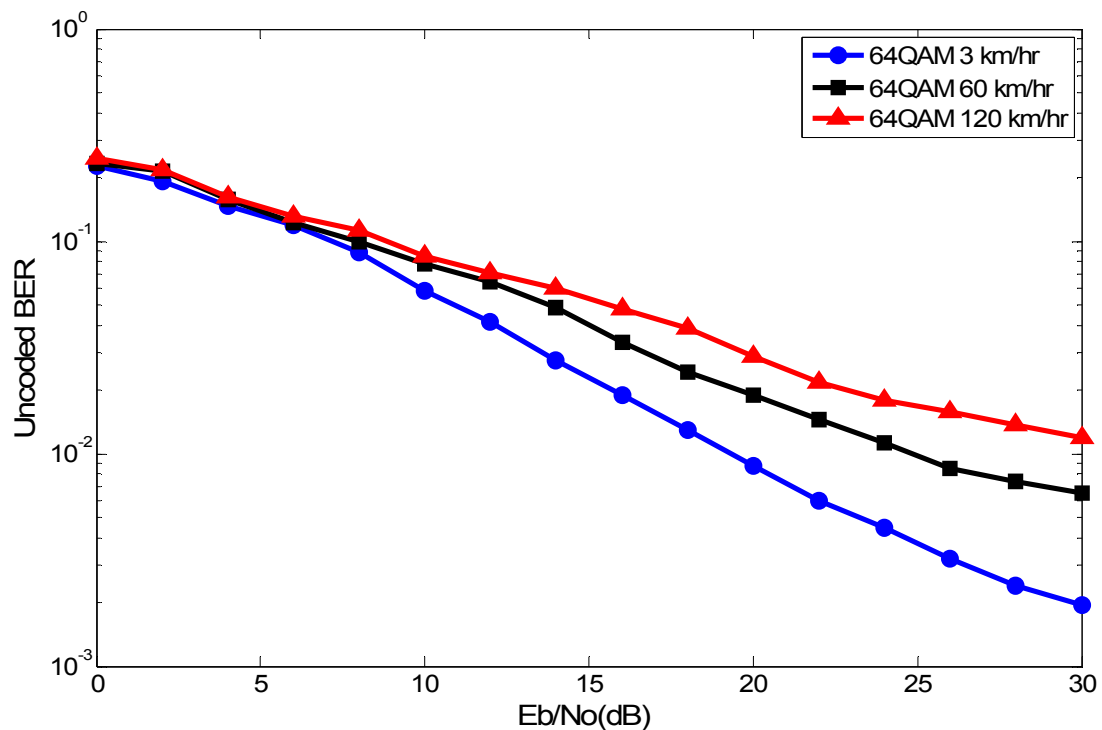


Fig. 3.18 Uncoded BER performance for 64QAM modulations at v_e of 3, 60 and 120 km/hr.

TABLE 3.4
REQUIRED SNR (dB) FOR BER 10^{-6} IN AWGN CHANNEL

Modulation	QPSK		16-QAM		64-QAM		
	1/2	3/4	1/2	3/4	1/2	2/3	3/4
Spec. of 802.16e	5.0	8.0	10.5	14.0	15.0	18.0	20.0
Proposed Receiver	3.9	5.1	8.2	11.1	13.7	15.9	17.2

In an AWGN environment, the performance requirement of IEEE 802.16e and that of our proposed system are listed in Table 3.4. The channel coding uses the convolution code specified by IEEE 802.16e as given in Section 3.3.1.2. The specification defines that BER shall be less than 10^{-6} at SNR of 5.0 and 8.0 dB for QPSK modulation in the code rate of 1/2 and 3/4, respectively; however, under the same situation and conditions, the proposed receiver only requires the SNR of 3.9 and 5.1 dB, respectively. For 16-QAM modulation, the proposed receiver requires the SNR of 8.2 and 11.1 dB in the code rate of 1/2 and 3/4, respectively. For 64-QAM

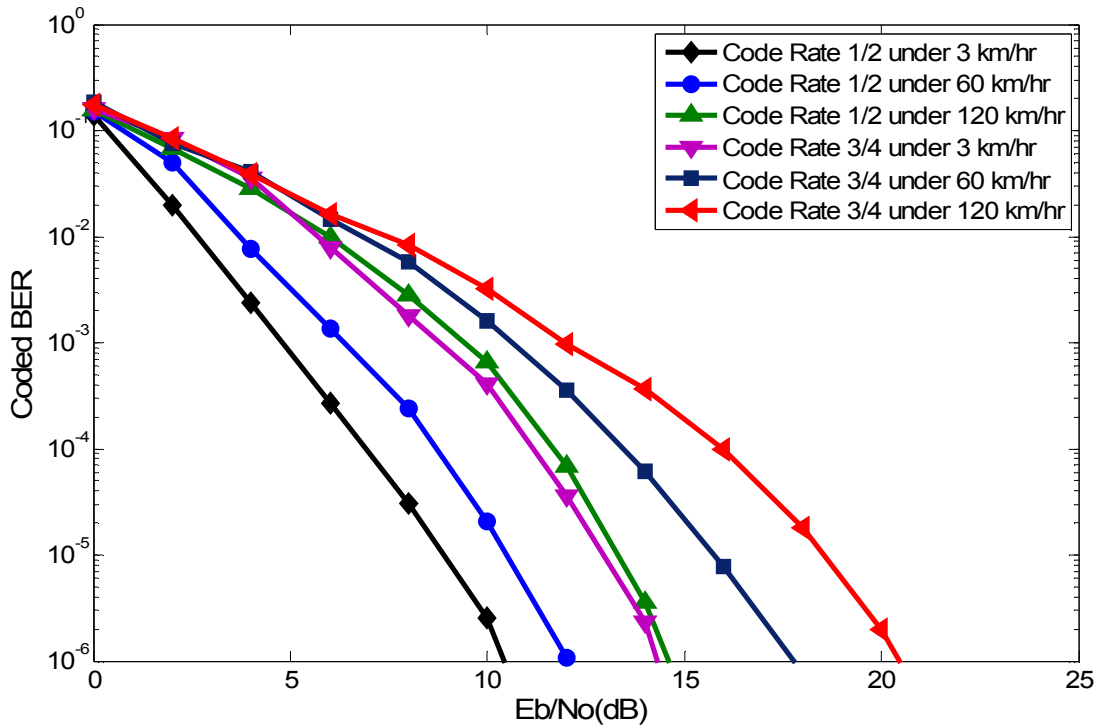


Fig. 3.19 Coded BER performance for QPSK modulation.

modulation, the proposed receiver requires the SNR of 13.7, 15.9 and 17.2 dB in the code rate of 1/2, 2/3 and 3/4, respectively. It clearly shows that the performance of the proposed receiver meets the requirements of WMAN in an AWGN environment.

Finally, for QPKS modulation, Fig. 3.19 shows the coded BER of the proposed receiver combining with the convolution code specified by IEEE 802.16e. IEEE 802.16e specification dose not provide the performance requirement in mobile transmission; however, in code rate 1/2, the proposed receiver at v_e of 3, 60 and 120 km/hr can achieve the coded BER less than 10^{-6} with E_b/N_0 of 10.6, 12.4 and 14.7 dB. In code rate 3/4, the proposed receiver under 3, 60 and 120 km/hr can achieve the coded BER less than 10^{-6} with E_b/N_0 of 14.3, 17.8 and 20.5 dB. For 16-QAM modulation, Fig. 3.20 shows the coded BER of the proposed receiver combining with the convolution code. In code rate 1/2, the proposed receiver at v_e of 3, 60 and 120 km/hr can achieve the coded BER less than 10^{-6} with E_b/N_0 of 13.1, 16.1 and 22.1 dB. In code rate 3/4, the proposed receiver under 3 km/hr can achieve the coded BER less than 10^{-6} with E_b/N_0 of 18.2 dB. Even at v_e of 60 and 120 km/hr, the proposed receiver in code rate 3/4 can achieve the coded BER to about 10^{-5} and 10^{-4} . For 64-QAM modulation, Fig. 3.21 shows the coded BER of the proposed receiver combining with

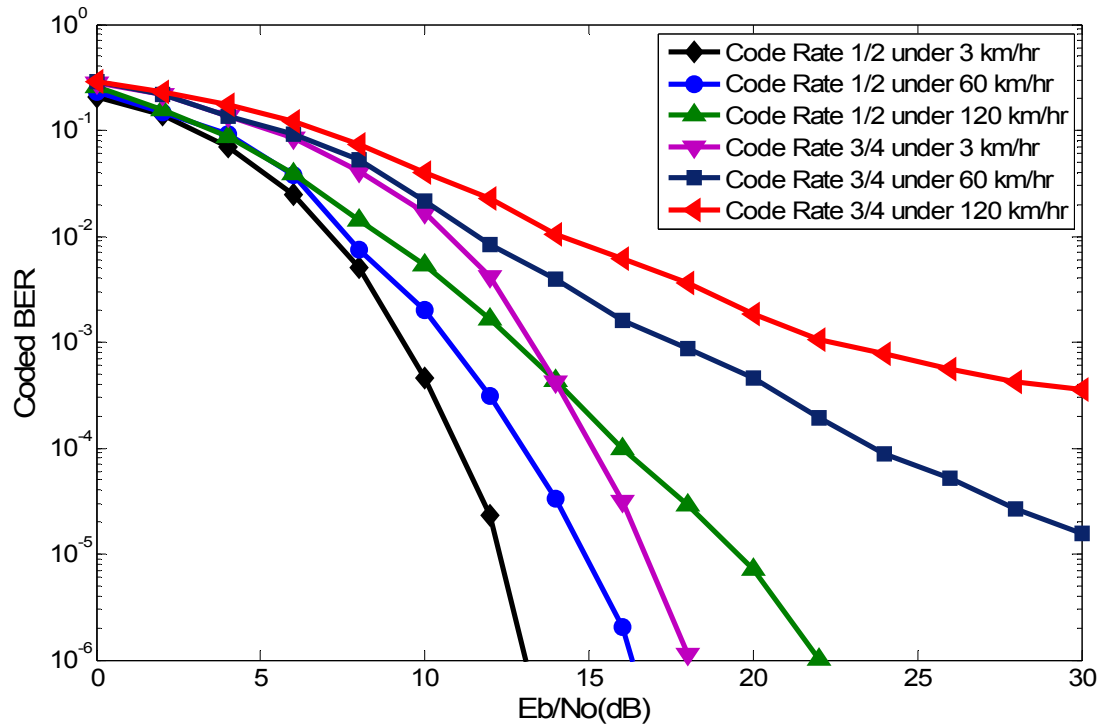


Fig. 3.20 Coded BER performance for 16QAM modulation.

the convolution code. In code rate 1/2, the proposed receiver under 3 km/hr can achieve the coded BER less than 10^{-6} with E_b/N_0 of 23.0 dB. At v_e of 60 and 120 km/hr, the proposed receiver in code rate 1/2 can achieve the coded BER to both about 10^{-5} . In code rate 3/4, the proposed receiver at v_e of 3 and 60 km/hr the proposed receiver can achieve the coded BER to both about 10^{-3} , and at v_e of 120 km/hr that can achieve the coded BER to about 10^{-2} . The SNR requirement of our proposed system achieving the coded BER less than 10^{-6} is listed in Table 3.5. Since the encoded block length of the interleaver specified by IEEE 802.16e is only 576 bits, which is much shorter than the block length using 8K bits specified by DVB-H [50], when the system adopts the large constellation such as 64-QAM, the coded BER is improved the very limited as compared with the uncoded BER. This is because that the continuous errors make the significant degradation of the Viterbi decoding efficiency. However, IEEE 802.16e has adopted the advantage channel coding schemes such as BTC, CTC and LDPC as the optional modes, and these channel coding may further improve the coded BER performances.

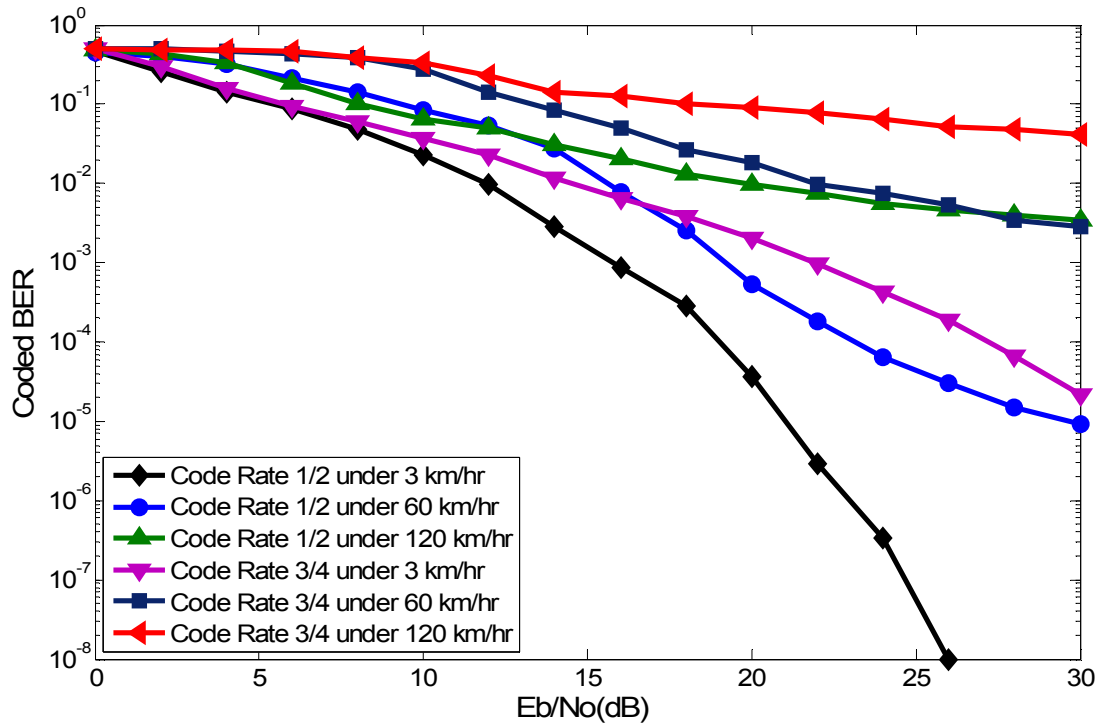


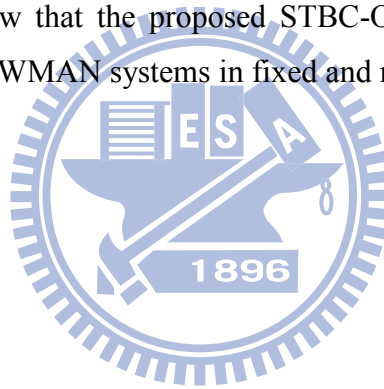
Fig. 3.21 Coded BER performance for 64-QAM modulation.

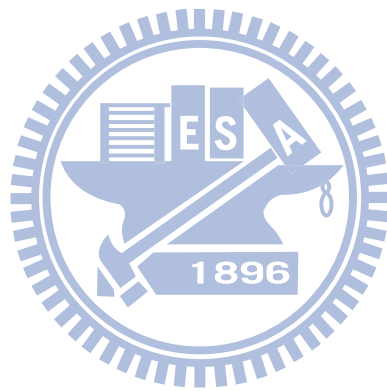
TABLE 3.5
REQUIRED SNR (dB) FOR BER 10^{-6} IN ITU VEH-A CHANNEL

Modulation	Code Rate	ITU Veh-A Channel with the Vehicle Speed		
		3 km/hr	60 km/hr	120km/hr
QPSK	1/2	13.6	15.4	17.7
	3/4	17.3	20.8	23.5
16-QAM	1/2	19.1	22.1	28.1
	3/4	24.2	-	-
64-QAM	1/2	30.8	-	-
	3/4	-	-	-

3.6 Summary

In this chapter, the proposed STBC-OFDM system is presented, and the system specification and design targets are also discussed. A simple symbol boundary detection scheme, a carrier frequency recovery loop modified by the ping-pong algorithm and an accurate two-stage DFT-based channel estimation method are included in the proposed receiver. From the system simulation results, we have shown that the CFO detection modified by the ping-pong algorithm can effectively maintain the ICFO error probability of about 10^{-3} in the weak region. The detail descriptions and simulation results of the proposed two-stage channel estimation will be given in Chapter 4. The performance of the proposed receiver meets the requirements of WMAN in an AWGN environment. Moreover, at v_e of 120 km/hr, the coded BER of the proposed receiver for both QPSK and 16QAM modulations can be less than 10^{-6} . The simulation results show that the proposed STBC-OFDM system can meet our design targets and used for WMAN systems in fixed and mobile environments.





Chapter 4

A Robust Channel Estimator for STBC-OFDM Systems

4.1 Introduction

With multiple transmit antennas, STBC can provide transmit diversity gain to improve system performance in wireless communications. However, for STBC decoding, STBC-OFDM systems require accurate CSI, which is particularly difficult to obtain in mobile wireless channels. Therefore, high quality channel estimation with acceptable hardware complexity is a crucial challenge for realizing a successful STBC-OFDM system.

Blind channel estimation method has an advantage of bandwidth-saving since it estimate channel response merely depending on the received signals. However, it must record numerous data symbols to obtain reliable channel estimation, involves high computational complexity, and only applies to slowly time-varying channels. On the contrary, pilot-aided channel estimation method using known pilots can provide more precise estimation for applications in mobile wireless communication.

Various pilot-aided channel estimation methods have been proposed for OFDM systems. Among these methods, DFT-based channel estimation methods using either minimum mean square error (MMSE) criterion or maximum likelihood (ML) criterion have been studied for OFDM systems with preamble symbols [33]-[35], [51], [52]. Since no information on channel statistics or operating SNR is required in the ML

scheme, the ML scheme is simpler to implement than the MMSE scheme [33]-[35], [51]. Furthermore, when the number of pilots is sufficient, the two schemes have comparable performances [34]. For this reason, the DF DFT-based channel estimation method is adopted to use the decided data as pilots to track channel variations for providing sufficient tracking information. Recently, Ku and Huang [19], [31] presented a DF DFT-based method derived from ML criterion and Newton's method. Moreover, they concluded that a refined two-stage channel estimation method [31] is more robust than the classical DF DFT-based method to apply in fast time-varying channels. Thus, the two-stage channel estimation method with an initialization stage and a tracking stage is adopted in this paper. Nevertheless, the two-stage channel estimation method has high computational complexity and is difficult to realize in hardware directly; hence, a novel architecture and an implementation method shall be proposed to reduce the implementation complexity.

In this chapter, a robust channel estimator for STBC-OFDM systems is proposed and applied in IEEE 802.16e baseband receiver. As compared with interpolation-based channel estimation methods which are commonly adopted in the pilot-aided channel estimator designs [53], [54], [65], and [66], our proposed channel estimator has significant performance improvements, especially when it is applied in outdoor mobile channels. This chapter is organized as follows. Section 4.2 introduces pilot-aided channel estimation methods. Section 4.3 briefly reviews the two-stage channel estimation method. Section 4.4 presents the proposed channel estimator. Then, the computational complexity of different channel estimation methods are given in Section 4.5. The simulation results are provided in Section 4.6. Finally, the conclusions of this chapter are drawn in Section 4.7.

Notation: By convention, boldface letters are used for sets, vectors, and matrices. The superscript $(\cdot)^*$ stands for complex conjugate. The notation $sign(\cdot)$ takes the sign of (\cdot) . The notations $Re(\cdot)$ and $Im(\cdot)$ stand for the real part and the imaginary part of (\cdot) . The notation $\{\dots\}$ denotes the contain elements of a set or a vector.

4.2 Pilot-aided Channel Estimation Methods

Pilot assisted channel estimation provides promising performance since it inserts known pilots periodically both in the time and frequency axes to track the time variations and frequency selectivity of the channel. Two types of pilot patterns, block type and comb type are widely adopted in various standards or literatures [55]-[59], while other two-dimension pilot patterns are discussed in [60], [61]. Comb-type pilot patterns and the time- and frequency-domain scattered pilot patterns usually perform better than the block-type pilot patterns in fast fading channels. Since block-type pilot patterns use all the subcarriers of one OFDM symbol as pilots, the insertion interval of the OFDM pilot symbols must be much less than the coherent time. Therefore, block-type pilot patterns are only suitable for slow fading channels. In order to apply in wireless mobile environments, we focus on comb-type pilot-aided channel estimation methods. Various comb-type pilot-aided channel estimation methods have been proposed for OFDM systems. Among these methods, there are two major categories of (a) interpolation-based channel estimation method and (b) DFT-based channel estimation method. These two kinds of comb-type pilot-aided channel estimation methods are briefly introduced in the following subsections.

4.2.1 Interpolation-based Channel Estimation

The channel complex gains at pilot subcarriers can be easily obtained from the received signals and known pilots, and the interpolation-based channel estimation methods are then applied to estimate the channel frequency responses at data subcarriers. Interpolation-based channel estimation methods cooperating with different pilot patterns have been extensively studied in the literatures [53], [54], [65], [66]. For the comb-type pilot-aided OFDM system, the N_{Pilot} pilot subcarriers $X_P[m]$, $m=0, \dots, N_{Pilot}-1$, are uniformly inserted into an OFDM symbol $X[k]$, $k=0, \dots, N-1$, where N is the total subcarrier number. The interval of pilot subcarriers is $L=N/N_{Pilot}$, where L is an integer. In the following, we will introduce several conventional polynomial interpolation-based channel estimation methods.

At the data subcarrier k , $mL \leq k < (m+1)L$, the estimated channel response $M[k]$ using a piecewise linear interpolation method is given by [53]

$$\begin{aligned}
M[k] &= M[mL+l] \\
&= \left(1 - \frac{l}{L}\right)M_p[m] + \frac{l}{L}M_p[m+1], 0 \leq l < L \\
&= (1 - \mu)M_p[m] + \mu M_p[m+1], \mu = l/L
\end{aligned} \tag{4.1}$$

where $M_p[m]$, $m=0, \dots, N_{Pilot}-1$, is the estimated channel response at pilot subcarriers.

The estimated channel responses using higher-order interpolation methods may have more accurate fitting than that using the linear interpolation method. However, the computational complexity also grows as the order increases. A piecewise second-order interpolation method adopted in [62] is expressed as

$$\begin{aligned}
M[k] &= M[mL+l] \\
&= C_1 M_p[m-1] + C_0 M_p[m] + C_{-1} M_p[m+1]
\end{aligned} \tag{4.2}$$

where

$$\begin{aligned}
C_1 &= \mu(\mu+1)/2 \\
C_0 &= -(\mu-1)(\mu+1) \\
C_{-1} &= \mu(\mu-1)/2
\end{aligned} \tag{4.3}$$

The coefficients of the cubic interpolation method adopted in [65] is given by

$$\begin{aligned}
M[k] &= M[mL+l] \\
&= C_1 M_p[m-1] + C_0 M_p[m] \\
&\quad + C_{-1} M_p[m+1] + C_{-2} M_p[m+2]
\end{aligned} \tag{4.4}$$

where

$$\begin{aligned}
C_1 &= -\frac{1}{6}\mu^3 + \frac{1}{2}\mu^2 - \frac{1}{6}\mu \\
C_0 &= +\frac{1}{2}\mu^3 - \mu^2 - \frac{1}{2}\mu + 1 \\
C_{-1} &= -\frac{1}{2}\mu^3 + \frac{1}{2}\mu^2 + \mu \\
C_{-2} &= -\frac{1}{6}\mu^3 - \frac{1}{6}\mu
\end{aligned} \tag{4.5}$$

Since OFDM signal is a two-dimensional function of time and frequency, pilots can be scattered in both time and/or frequency axes of an OFDM frame. Therefore, the channel frequency responses at data subcarriers can be performed by two-dimensional (2-D) interpolation-based channel estimation methods which are widely used to further improve the estimation performance. The 2-D interpolation-based channel estimation can be performed by cascading two one-dimensional interpolation-based channel estimations. Nevertheless, the conventional 2-D interpolation-based methods need huge amount of storage for implementation of the non-causal property in the time-domain interpolation. Lin and Lee [54] have effectively presented 3 kinds of the predictive algorithms to save the storage requirement in time-domain interpolation.

The interpolation-based channel estimation methods are often used in the condition that the frequency-domain channel response is oversampled by the pilot subcarriers. The number of pilot subcarriers must be larger than the normalized maximum excess delay [66]. If the normalized maximum excess delay is close to the number of pilot subcarriers, then the performance of interpolated channel response is seriously degraded. Hence, based on the limited pilot subcarriers, using the interpolation-based methods to estimate channel responses is a challenge in outdoor mobile channels. Because the coherent bandwidth becomes small, the interpolation-based method becomes more difficult to recover channel variations.

4.2.2 DFT-based Channel Estimation

The DFT-based channel estimation method obtains the time-domain CIR by inverse DFT (IDFT) transforming the frequency-domain channel response at pilot subcarriers and effectively improves the performance by suppressing time-domain noise and aliasing effect. Many DFT-based methods derived from ML scheme have been studied for OFDM systems with preambles [33]-[35], [51], [52]. In order to further improve the estimated performance, the DF DFT-based channel estimation methods employ decided data subcarriers as pilot subcarriers to track channel variations for saving transmission bandwidth and providing sufficient tracking information [33], [35], [51].

A classical DF DFT-based channel estimation method can be decomposed to

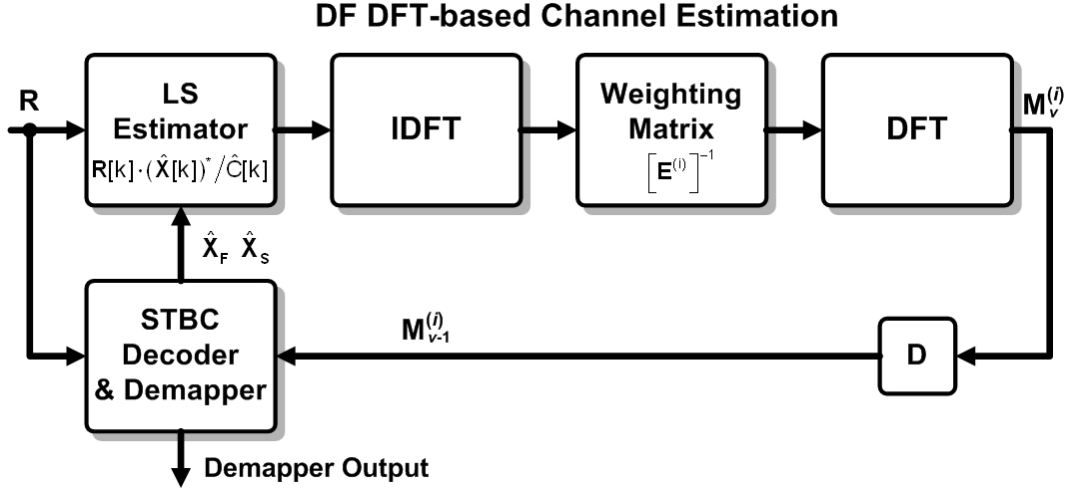


Fig. 4.1 Block diagram of a classic DF DFT-based channel estimation method.

four parts which include an least square (LS) estimator, an IDFT matrix, a weighting matrix, and a DFT matrix as shown in Fig. 4.1 [33]-[35], [52]. The LS estimator uses the decision data symbols and pilots to calculate the LS estimates which are the contaminated channel frequency response. After the IDFT processing, the LS estimates are transformed to time domain, and a weighting matrix is used to reduce noise and aliasing effect in time domain. Some of the weighting methods are to directly cut off the path gains of CIR below a threshold or keep only significant paths of the CIR. Other methods are to give the weighting of each path of CIR, and the weighting matrix can be derived from the ML or MMSE criterion [33], [52]. Finally, the weighted CIR is transformed back to frequency domain by DFT processing.

Based on the architectures of the transmitter and receiver in the proposed the STBC-OFDM system, the channel frequency response between the first / second transmit antenna and the receive antenna is denoted as $H^{(1)}[k] / H^{(2)}[k]$, respectively. Within a time slot (two OFDM symbols), after the received signals have passed through the guard interval removal and the N -point FFT, the two successive received OFDM symbols, $R[1,k]$ and $R[2,k]$, are given by

$$R[1,k] = H^{(1)}[k]X_F[k] + H^{(2)}[k]X_S[k] + Z[1,k] \quad (4.6)$$

$$R[2,k] = -H^{(1)}[k](X_S[k])^* + H^{(2)}[k](X_F[k])^* + Z[2,k] \quad (4.7)$$

for $k \in \mathbf{Q} \cup \mathbf{J}$, where \mathbf{Q} and \mathbf{J} denote the sets of data and pilot subcarrier indices, respectively, $X_F[k]$ and $X_S[k]$ are two transmitted OFDM symbols within a time slot, and $Z[1,k]$ and $Z[2,k]$ are the uncorrelated additive white Gaussian noise (AWGN) with zero-mean and variance $(\sigma_Z)^2$.

The DF DFT-based channel estimation method using the ML criterion can be summarized as follows [33]-[36]. The LS channel estimate $\delta_\nu[k]$ for $k \in \Theta$, can be obtained as

$$\delta_\nu[k] = \{\Delta_\nu^{(i)}[k] : i=1,2\} = \frac{1}{\hat{C}_\nu[k]} (\hat{\mathbf{X}}_\nu[k])^* \mathbf{R}[k] \quad (4.8)$$

where ν is the iteration number from 1 to V , i is to correspond to the i -th transmit antenna, and Θ is a subset of \mathbf{Q} used to track channel variations. Moreover, at the k -th subcarrier, the $\hat{\mathbf{X}}_\nu[k] = \{\hat{X}_\nu^{(i)}[t,k] : t=0,1\}$ is the re-encoded STBC matrix with decision symbols $\{\hat{X}_F[k], \hat{X}_S[k]\}$ as its elements obtained by applying the previously estimated channel frequency response to decode the received signal. $\mathbf{R}[k] = \{R[t,k] : t=0,1\}$ is the received signal vector, and $\hat{C}_\nu[k] = \sum_{t=1}^2 |\hat{X}_\nu^{(i)}[t,k]|^2$ is the energy normalized factor. Then, the LS estimate is improved by using a truncated DFT matrix $\mathbf{F}_{DFT}^{(i)}$ as follows:

$$\mathbf{q}_\nu^{(i)} = (\mathbf{F}_{DFT}^{(i)})^H \Delta_\nu^{(i)} = \mathbf{F}_{IDFT}^{(i)} \Delta_\nu^{(i)} \quad (4.9)$$

$$\mathbf{E}^{(i)} = (\mathbf{F}_{DFT}^{(i)})^H \mathbf{F}_{DFT}^{(i)} = \mathbf{F}_{IDFT}^{(i)} \mathbf{F}_{DFT}^{(i)} \quad (4.10)$$

$$\mathbf{M}_\nu^{(i)} = \mathbf{F}_{DFT}^{(i)} [\mathbf{E}^{(i)}]^{-1} \mathbf{q}_\nu^{(i)} \quad (4.11)$$

where $\Delta_\nu^{(i)} = \{\Delta_\nu^{(i)}[k] : k \in \Theta\}$, and $\mathbf{M}_\nu^{(i)} = \{M_\nu^{(i)}[k] : k \in \Theta\}$ is the estimated channel

frequency response vector. The truncated DFT matrix $\mathbf{F}_{DFT}^{(i)}$ is given by

$$\mathbf{F}_{DFT}^{(i)} = \begin{bmatrix} e^{\frac{-j2\pi\Theta_0\hat{\tau}_0}{K}} & \dots & e^{\frac{-j2\pi\Theta_0\hat{\tau}_{k^{(i)}-1}}{K}} \\ \vdots & \ddots & \vdots \\ e^{\frac{-j2\pi\Theta_{|\Theta|-1}\hat{\tau}_0}{K}} & \dots & e^{\frac{-j2\pi\Theta_{|\Theta|-1}\hat{\tau}_{k^{(i)}-1}}{K}} \end{bmatrix}. \quad (4.12)$$

Besides, the CSI estimated in the previous time slot is taken as the initial CSI value for the current time slot.

4.3 Two-Stage Channel Estimation Method

In the DFT-based channel estimation methods, most mobile wireless channels are characterized by CIR consisting of a few dominant paths. These path delays usually change slowly in time, but the path gains may vary relatively fast. In this section, the refined two-stage channel estimation method [31] will be briefly reviewed. An initialization stage uses a MPIC-based decorrelation method to identify the significant paths of CIR in the beginning of each frame. However, the CIR estimated by the preamble can not be directly applied in the following data bursts since the receiver is mobile. Thus, a tracking stage is then used to track the path gains with known CIR positions. The details are described in following subsections.

4.3.1 Initialization Stage

The preamble, which is the known sequence, is the first symbol of each downlink transmission frame. The significant paths can be estimated by matching the preamble; however, the preamble does not have ideal auto-correlation due to the use of either guard band or non-equally spaced pilots in most wireless standards. The MPIC-based decorrelation is used to estimate CIR path-by-path and cancel out the known multipath interference. The flowchart of the MPIC-based decorrelation is shown in Fig. 4.2. The channel estimation for each transceiver antenna pair can be

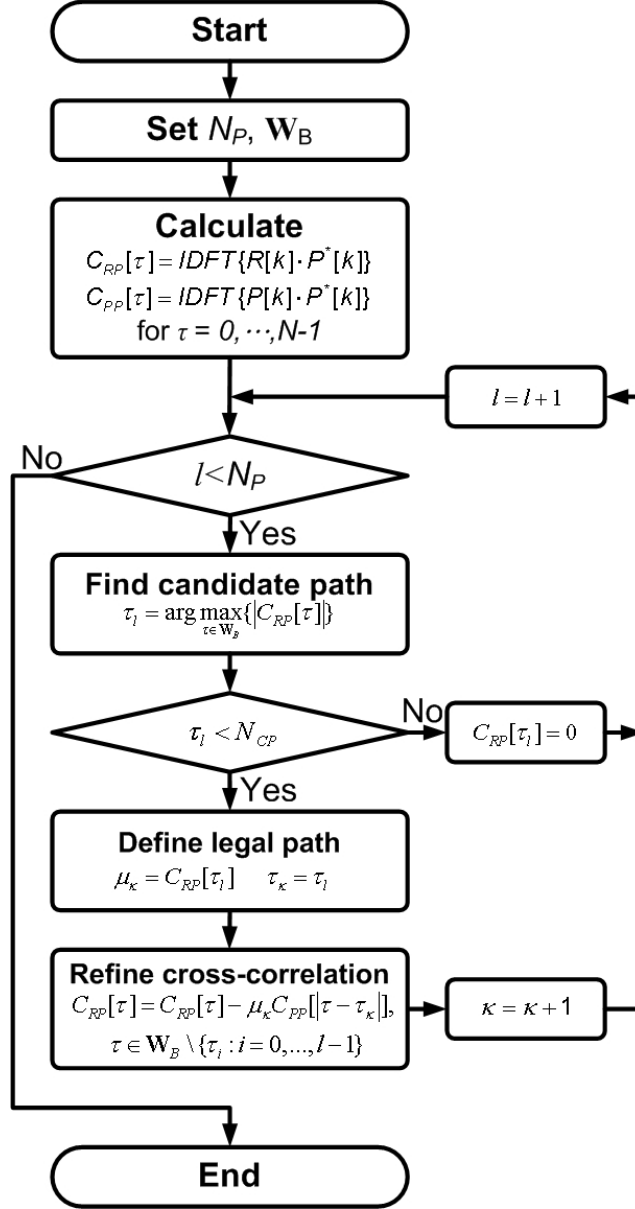


Fig. 4.2 Flowchart of MPIC-based decorrelation method.

independently performed because the preambles transmitted from different antennas do not interfere with each other. First, two parameters N_p and \mathbf{W}_B are defined as a presumptive path number of a channel and an observation window set, respectively. Second, the cyclic cross-correlation $C_{RP}[\tau]$ between the received and transmitted preambles as well as the normalized cyclic auto-correlation $C_{PP}[\tau]$ of the transmitted preamble are calculated. The indices l and κ which stand for a path counting variable and the number of the legal paths found by the MPIC-based decorrelation are initialized to zero. Third, the process is started by picking only one path whose time

delay τ_l yields the largest value in $C_{RP}[\tau_l]$, for $\tau_l \in \mathbf{W}_B$. If the path delay τ_l is larger than the length of CP, this path is treated as an illegal path and discarded by setting $C_{RP}[\tau_l]=0$. Otherwise, this path is recorded as the κ -th legal path with a time delay $\tau_\kappa=\tau_l$ and a complex path gain $\mu_\kappa=C_{RP}[\tau_l]$. Then, the interference associated with this legal path is canceled from $C_{RP}[\tau]$ to obtain a refined cross-correlation function:

$$C_{RP}[\tau] = C_{RP}[\tau] - \mu_\kappa C_{PP}[\tau - \tau_\kappa], \tau \in \mathbf{W}_B \setminus \{\tau_0, \dots, \tau_{l-1}\}. \quad (4.13)$$

Meanwhile, κ is increased by one. The value of l is also increased by one at the end of each iteration, and the iterative process is continued until l reaches the presumed value of N_P .

4.3.2 Tracking Stage

After the initialization stage, we can obtain the information of the path numbers $\kappa^{(i)}$, the multipath delays $\tau_l^{(i)}$, the multipath complex gains $\mu_l^{(i)}$, for $l \in \{0, \dots, \kappa^{(i)}-1\}$, and the corresponding channel frequency responses. Under the assumption that the multipath delays do not change over the duration of a frame, the DF DFT-based channel estimation method can be equivalently expressed in Newton's method as [36]:

$$\boldsymbol{\delta}_v[k] = \mathbf{M}_{v-1}[k] - \frac{1}{\hat{C}_v[k]} (\hat{\mathbf{X}}_v[k])^* \mathbf{R}[k] \quad (4.14)$$

$$\mathbf{q}_v^{(i)} = \mathbf{F}_{IDFT}^{(i)} \boldsymbol{\Delta}_v^{(i)} \quad (4.15)$$

$$\mathbf{M}_v^{(i)} = \mathbf{M}_{v-1}^{(i)} - \mathbf{F}_{DFT}^{(i)} [\mathbf{E}^{(i)}]^{-1} \mathbf{q}_v^{(i)}. \quad (4.16)$$

According to [36], the vector $\boldsymbol{\delta}_v[k] = \{\Delta_v^{(i)}[k]: i=1,2\}$ calculates the difference between the previous estimated channel frequency response vector

$\mathbf{M}_{v-1}[k] = \{M_{v-1}^{(i)}[k]: i=1,2\}$ and the LS estimation vector in (4.14), where v is the iteration index. The matrix $\hat{\mathbf{X}}_v[k]$ is the re-encoded STBC matrix with decided symbols, $\hat{X}_F[k]$ and $\hat{X}_S[k]$, as its entries. The decided symbols are obtained by applying the previous estimated channel frequency responses to decode the received signal vector $\mathbf{R}[k] = \{R[t,k]: t=1,2\}$, where t is the symbol index within a time slot. The value $\hat{C}_v[k] = |\hat{X}_F[k]|^2 + |\hat{X}_S[k]|^2$ is the energy normalization factor. The IDFT matrix $\mathbf{F}_{IDFT}^{(i)}$ multiplied by the vector $\Delta_v^{(i)} = \{\Delta_v^{(i)}[k]: k \in \Theta\}$ in (4.15) is to form the gradient vector $\mathbf{q}_v^{(i)}$ in Newton's method as shown in (4.16). In addition, the weighting matrix $[\mathbf{E}^{(i)}]^{-1}$ in (4.16) is in fact the inverse of the Hessian matrix in Newton's method [36]. The (l,l') -th entry of $\mathbf{E}^{(i)}$ is given by

$$\left(\mathbf{E}^{(i)}\right)_{l,l'} = \sum_{k \in \Theta} \cos\left(\frac{2\pi k (\tau_l^{(i)} - \tau_{l'}^{(i)})}{N}\right) + j \sum_{k \in \Theta} \sin\left(\frac{2\pi k (\tau_l^{(i)} - \tau_{l'}^{(i)})}{N}\right). \quad (4.17)$$

In the previous studies [33], [35], the pilots as well as the decided data symbols are simultaneously adopted to perform channel estimation at each tracking iteration. From the viewpoint of optimization, since the pilots inserted in each OFDM symbol are much more reliable than the decided data symbols, they should play a dominant role in providing a global search direction at the first tracking iteration [31]. Thus, the first iteration of the channel tracking is modified as

$$\mathbf{M}_1^{(i)} = \mathbf{M}_0^{(i)} - \gamma \cdot \mathbf{F}_{DFT}^{(i)} \mathbf{q}_1^{(i)} \quad (4.18)$$

where the gradient vector $\mathbf{q}_1^{(i)}$ is calculated according to (4.14)-(4.15) by using the pilot subcarrier set \mathbf{J} instead of the set Θ , and the value γ is an experimental constant of step size to have the best performance.

It is demonstrated in [31] that the two-stage channel estimation method has better performance than the classical DF DFT-based method, the STBC-based MMSE

method, and the Kalman filtering method for estimating channels in mobile environments, and its computational complexity is quite the same with these methods. However, the high complexity problem still needs to be solved for hardware implementation. Hence, we propose a modified two-stage channel estimation method and its architecture for implementation.

4.4 Proposed Channel Estimator

The overall block diagram of the proposed channel estimator is shown in Fig. 4.3. The initialization stage is decomposed to a preamble match, an IFFT, a straight MPIC (SMPIC)-based decorrelator, and an FFT. The tracking stage is decomposed to an STBC decoder, a demapper, an LS estimator, an IFFT, a path decorrelator, a Hessian matrix calculator and an FFT. Moreover, the IFFT and FFT are shared between the initialization stage and the tracking stage. These key blocks are described in the following subsections.

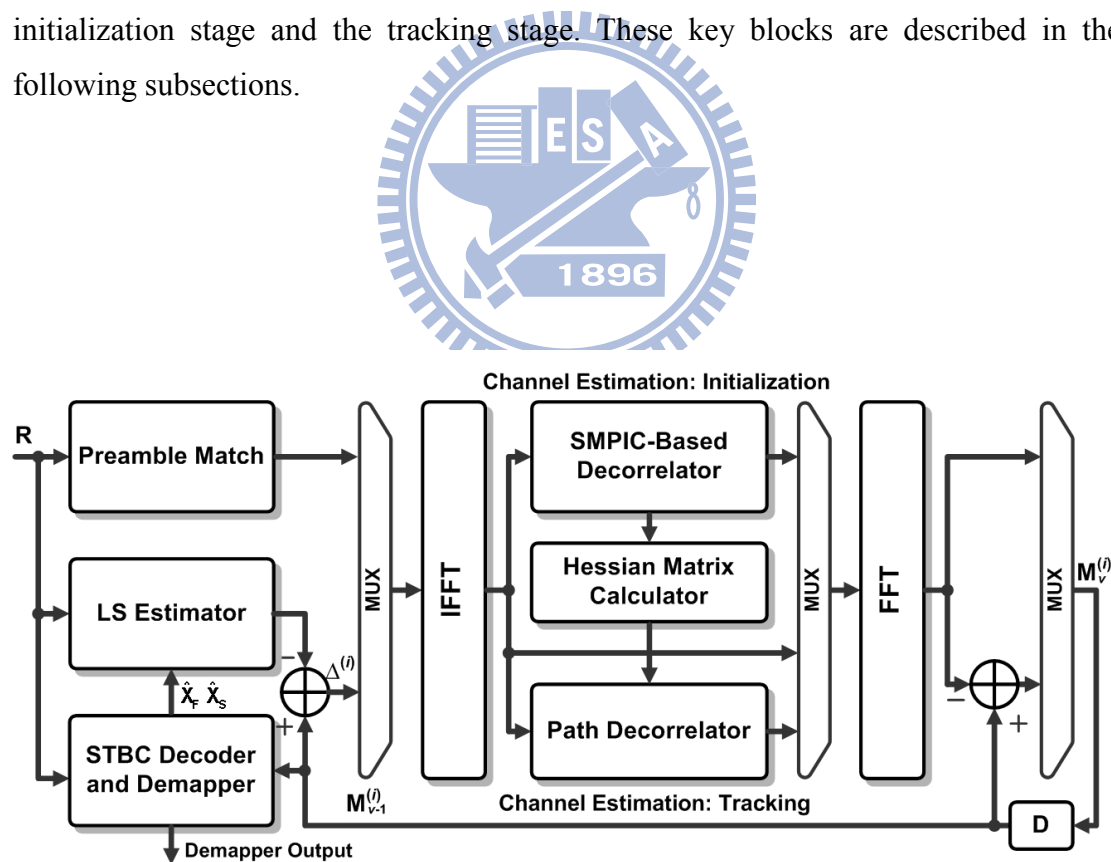


Fig. 4.3 Overall block diagram of the proposed channel estimator.

4.4.1 Initialization Stage: Preamble Match

In the initialization stage, the preamble match is first used to estimate the preliminary channel frequency responses $M^{(i)}[k]$ for $i=1,2$ by matching the received signal with the preamble symbol $P^{(i)}[k]$ transmitted from the i -th antenna. Since the preambles transmitted from different antennas do not interfere with each other, $M^{(i)}[k]$ can be independently performed by

$$M^{(i)}[k] = \frac{R[k] \cdot (P^{(i)}[k])^*}{|P^{(i)}[k]|^2} = H^{(i)}[k] + \frac{Z[k] \cdot (P^{(i)}[k])^*}{|P^{(i)}[k]|^2} \quad (4.19)$$

where $R[k]$ in the initialization stage is the first received OFDM symbol of a frame, and $Z[k]$ is AWGN. Consequently, the preliminary channel frequency responses $M^{(i)}[k]$ pass through IFFT to obtain the channel impulse responses.

4.4.2 Initialization Stage: SMPIC-based Decorrelator

After IFFT operation, the CSIs are obtained in time domain $m^{(i)}[\tau]$. The MPIC-based decorrelation method estimates CIR path-by-path. It picks the maximum path of $m^{(i)}[\tau]$ for $\tau \in \mathbf{W}_B$ and cancels the maximum path interference to other paths. If the set \mathbf{W}_B has N_B paths, the process must iterate N_P times for finding the maximum path of these N_B paths and canceling the maximum path interference to other (N_B-1) paths. This method requires too many execution cycles and is unsuitable to directly implement in the proposed channel estimator.

In order to reduce the execution cycles, we propose an SMPIC-based decorrelation method to identify N_P significant paths in a straightforward method, and the flowchart is shown in Fig. 4.4. First, the proposed scheme sorts the N_B paths to find the first N_{TP} paths with large $m^{(i)}[\tau]$. Second, the decorrelation is carried out from

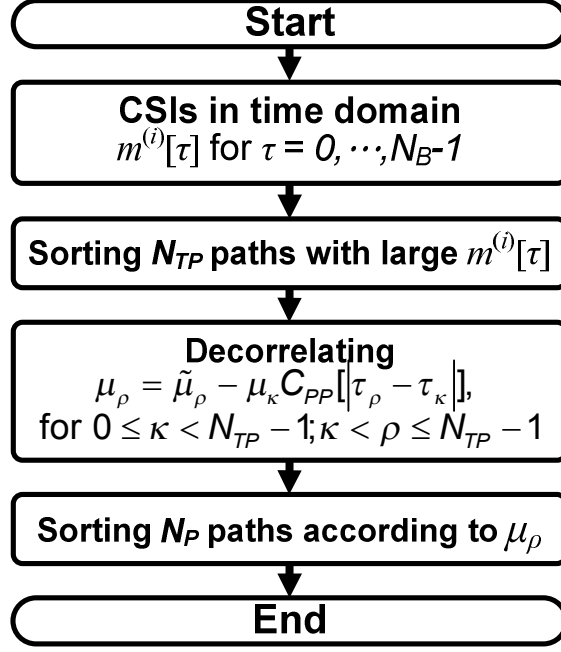


Fig. 4.4 Flowchart of the proposed SMPIC-based decorrelation.

the largest to the smallest one of these N_{TP} sorted paths to cancel the path interference. The decorrelation process starts at the first legal path which is the maximum sorted path. If κ denotes the iteration number, for $0 \leq \kappa < N_{TP}-1$, the process of the κ -th iteration is to cancel the interferences associated with the κ -th legal path gain. The process can be expressed as

$$\mu_\rho = \tilde{\mu}_\rho - \mu_\kappa C_{PP}[|\tau_\rho - \tau_\kappa|], \quad 0 < \rho \leq N_{TP} \quad (4.20)$$

where μ_κ is the κ -th legal path gain, $\tilde{\mu}_\rho$ is the ρ -th sorted path gain, μ_ρ is the ρ -th decorrelated sorted path gain, τ_κ is the κ -th legal path delay, τ_ρ is the ρ -th sorted path delay, and $C_{PP}[\tau]$ is the normalized cyclic auto-correlation of the preamble. Finally, the decorrelated N_{TP} paths are sorted again to pick up the first N_P paths. For using a sorting network of fixed I/O size N_P to sort an arbitrarily larger data set, the number of N_{TP} is defined to be $\beta \cdot N_P$, and β is an integer which is searched to optimize the computational complexity and guarantee the acceptable performance. Here, the

output SNR at the STBC decoder is used as a gauge of the system performance to determine the value of β and defined as

$$\text{output SNR} = \frac{\sum_{\ell=0}^{N_S-1} \text{SNR}_\ell \cdot n_\ell}{\sum_{\ell=0}^{N_S-1} n_\ell} \quad (4.21)$$

$$\text{SNR}_\ell = 10 \log_{10} \left(\frac{\phi_\ell^2}{2\sigma_\ell^2} \right) \quad (4.22)$$

$$\phi_\ell = \frac{1}{n_\ell} \sum_{j=0}^{n_\ell-1} |\hat{X}_j| \quad (4.23)$$

$$\sigma_\ell^2 = \frac{1}{n_\ell - 1} \sum_{j=0}^{n_\ell-1} \left(|\hat{X}_j|^2 - |\phi_\ell|^2 \right) \quad (4.24)$$

where N_S is the number of symbols in a constellation, n_ℓ is the number of data belonging to the ℓ -th symbol after being sliced, and \hat{X}_j is the desired data after STBC decoding.

If a sorting algorithm such as merge sorting is used, the computational complexity of the original MPIC-based decorrelation method requires $O(N_P N_B \log_2 N_B)$ comparisons, $O(N_P N_B)$ complex multiplications and $O(N_P N_B)$ complex subtractions because it must repeat N_P times of sorting and decorrelation of N_B paths. However, the complexity of the SMPIC-based decorrelation method only requires $O(N_B \log_2 N_B)$ comparisons, $O(N_{TP}^2)$ complex multiplications and $O(N_{TP}^2)$ complex subtractions. Thus, the requirement of execution cycles can be effectively reduced by about $O(N_P)$ times.

After the SMPIC-based decorrelator, the significant paths have been identified and are then transformed to channel frequency responses by FFT for using as the reference in the tracking stage.

4.4.3 Tracking Stage: STBC Decoder and Demapper

In the tracking stage, from (4.14), the LS estimator is used to calculate the LS

estimations followed by calculating the vector $\delta_v[k] = \{\Delta_v^{(i)}[k] : i=1,2\}$ that can be expressed as

$$\Delta_v^{(1)}[k] = M_{v-1}^{(1)}[k] - \frac{1}{\hat{C}_v[k]} \left((\hat{X}_F[k])^* R[1,k] - \hat{X}_S[k] R[2,k] \right) \quad (4.25)$$

$$\Delta_v^{(2)}[k] = M_{v-1}^{(2)}[k] - \frac{1}{\hat{C}_v[k]} \left((\hat{X}_S[k])^* R[1,k] + \hat{X}_F[k] R[2,k] \right). \quad (4.26)$$

Before the LS estimation calculation, the decided symbols $\hat{X}_F[k]$ and $\hat{X}_S[k]$ must be determined first. Based on the latest estimated channel frequency responses, the STBC decoder and the symbol demapper are used to decode these two received symbols and can be formulated as

$$\hat{X}_F[k] = \text{De} \left\{ \frac{(M_{v-1}^{(1)}[k])^* R[1,k] + M_{v-1}^{(2)}[k] (R[2,k])^*}{(|M_{v-1}^{(1)}[k]|^2 + |M_{v-1}^{(2)}[k]|^2)} \right\} \quad (4.27)$$

$$\hat{X}_S[k] = \text{De} \left\{ \frac{(M_{v-1}^{(2)}[k])^* R[1,k] - M_{v-1}^{(1)}[k] (R[2,k])^*}{(|M_{v-1}^{(1)}[k]|^2 + |M_{v-1}^{(2)}[k]|^2)} \right\} \quad (4.28)$$

where $\text{De}\{\cdot\}$ is the demapper process.

4.4.4 Tracking Stage: LS Estimator

After the decided symbols have been determined, the LS estimations, $\varepsilon_v^{(i)}[k]$ for $i=1,2$, are calculated by the LS estimator. Both $\{I_F, Q_F\}$ and $\{I_S, Q_S\}$ denote the I and Q coordinate values of $\hat{X}_F[k]$ and $\hat{X}_S[k]$, respectively. Both $\{I_{R1}, Q_{R1}\}$ and $\{I_{R2}, Q_{R2}\}$ denote the real part and the imaginary part of $R[1,k]$ and $R[2,k]$, respectively. The value of constellation normalization and the value of $1/\hat{C}_v[k]$ both have a

limited constant set; thus, these normalizations can be merged to one multiplication of λ , and the value of λ has also a limited constant set. Therefore, the possible multiplication values of λ can be pre-computed and stored in advance, and on-line computation can be saved to reduce implementation cost. The LS estimations can be expressed as

$$\begin{aligned}\varepsilon_v^{(1)}[k] &= \left(\left(\hat{X}_F[k] \right)^* R[1,k] - \hat{X}_S[k] R[2,k] \right) / \hat{C}_v[k] \\ &= \lambda \cdot \left[(I_{R1} \cdot I_F + Q_{R1} \cdot Q_F - I_{R2} \cdot I_S + Q_{R2} \cdot Q_S) \right. \\ &\quad \left. + j(Q_{R1} \cdot I_F - I_{R1} \cdot Q_F - Q_{R2} \cdot I_S - I_{R2} \cdot Q_S) \right]\end{aligned}\quad (4.29)$$

$$\begin{aligned}\varepsilon_v^{(2)}[k] &= \left(\left(\hat{X}_S[k] \right)^* R[1,k] + \hat{X}_F[k] R[2,k] \right) / \hat{C}_v[k] \\ &= \lambda \cdot \left[(I_{R1} \cdot I_S + Q_{R1} \cdot Q_S + I_{R2} \cdot I_F - Q_{R2} \cdot Q_F) \right. \\ &\quad \left. + j(Q_{R1} \cdot I_S - I_{R1} \cdot Q_S + Q_{R2} \cdot I_F + I_{R2} \cdot Q_F) \right].\end{aligned}\quad (4.30)$$

Fig. 4.5 shows the pilots which are transmitted in the cluster structures over different time slot. In IEEE 802.16e, each cluster contains 14 subcarriers, and there are 60 clusters in an OFDM symbol with 1024 subcarriers. Each cluster has two pilot subcarriers, and the pilots are modulated by BPSK. If a pilot is transmitted on one pilot subcarrier from one antenna, the other antenna will not transmit a pilot on the same subcarrier to avoid the inter-antenna interference. The dimension of \mathbf{J} is N_J . According to this allocation, if the pilot subcarrier index is $k \in \mathbf{J} = \{J_0, J_1, \dots, J_{N_J-1}\}$, the LS estimations at the first iteration can be expressed as follows:

$$\varepsilon_1^{(1)}[k] = \begin{cases} \pm\sigma \cdot R[2,k], & \text{when } k = J_{2k'} \\ \pm\sigma \cdot R[1,k], & \text{when } k = J_{2k'+1} \end{cases}\quad (4.31)$$

$$\varepsilon_1^{(2)}[k] = \begin{cases} \pm\sigma \cdot R[1,k], & \text{when } k = J_{2k'} \\ \pm\sigma \cdot R[2,k], & \text{when } k = J_{2k'+1} \end{cases}\quad (4.32)$$

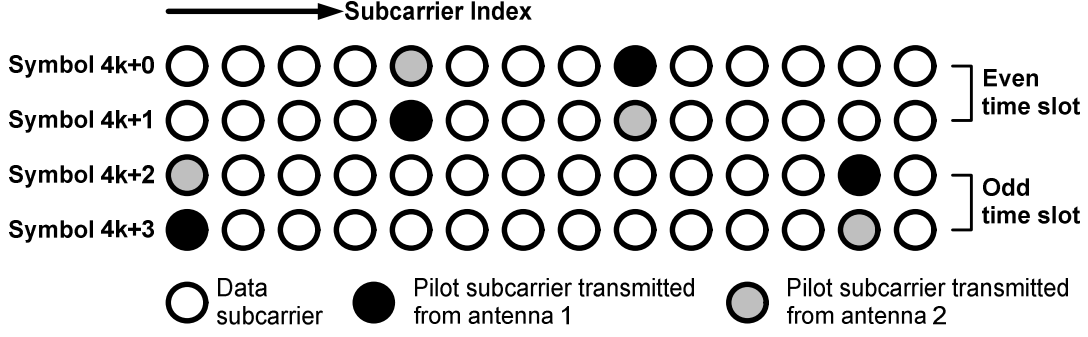


Fig. 4.5 Pilots transmitted in the cluster structures over different time slots.

where the index k' is in the range $0 \leq k' < N_f/2$, and σ is a constant value to represent the absolute pilot value normalized by the pilot power.

After the LS estimations calculation, $\delta_v[k]$ is acquired by subtracting the LS estimations from the latest estimated channel frequency responses.

4.4.5 Tracking Stage: Hessian Matrix Calculator and Path Decorrelator

We then pass $\Lambda_v^{(i)}$ through IFFT to obtain $\mathbf{q}_v^{(i)}$ in time domain. Only those gradient entries of $\mathbf{q}_v^{(i)}$ that have the same path delays as the significant paths identified in the initialization stage are considered. Therefore, white noise will be filtered out, and the decision error propagation effect can also be alleviated. Since data and pilot subcarriers are not equally-spaced, the aliasing between the paths occurs. The path decorrelator works to decorrelate the inter-path interferences. Before the path decorrelation, the inverse of the Hessian matrix, $[\mathbf{E}^{(i)}]^{-1}$, should be calculated first. Although $[\mathbf{E}^{(i)}]^{-1}$ is only calculated once within a frame operation, the matrix inverse computation needs very high complexity of $O(N_p^3)$ complex multiplications. Besides, each entry of $\mathbf{E}^{(i)}$ should take at least N_Θ cycles to calculate the cosine and sine summations by using a look-up table, where N_Θ is the dimension of Θ . If $[\mathbf{E}^{(i)}]^{-1}$ is implemented directly, it will require very large hardware module and memory. In order to reduce the requirement, the matrix inverse is avoided by considering the

strongly diagonal property [31]. If $\kappa^{(i)}$ is the significant path number, $\mathbf{E}^{(i)}$ is decomposed to $N_{\Theta}(\mathbf{I}_{\kappa^{(i)}} + \mathbf{O}_{off})$, where $\mathbf{I}_{\kappa^{(i)}}$ is a $\kappa^{(i)} \times \kappa^{(i)}$ identity matrix, and \mathbf{O}_{off} is a zero-diagonal matrix. If N_{Θ} is large enough, an approximate weighting matrix of $[\mathbf{E}^{(i)}]^{-1}$ takes the form as

$$\begin{aligned} [\mathbf{E}^{(i)}]^{-1} &= \frac{1}{N_{\Theta}}(\mathbf{I}_{\kappa^{(i)}} + \mathbf{O}_{off})^{-1} \\ &\approx \frac{1}{N_{\Theta}}(\mathbf{I}_{\kappa^{(i)}} - \mathbf{O}_{off}) = \frac{1}{N_{\Theta}^2}(2N_{\Theta}\mathbf{I}_{\kappa^{(i)}} - \mathbf{E}^{(i)}). \end{aligned} \quad (4.33)$$

Furthermore, the (l, l') -th entry value can be represented as

$$([\mathbf{E}^{(i)}]^{-1})_{l, l'} = \begin{cases} \frac{1}{N_{\Theta}}, & \text{when } l = l' \\ \frac{1}{N_{\Theta}^2}[-(\mathbf{E}^{(i)})_{l, l'}], & \text{when } l \neq l' \end{cases}. \quad (4.34)$$

Therefore, when the Hessian matrix $\mathbf{E}^{(i)}$ has been calculated, the matrix inverse $[\mathbf{E}^{(i)}]^{-1}$ only requires at most $O(N_p^2)$ real multiplications for the normalization of $1/N_{\Theta}^2$. However, the computation of $\mathbf{E}^{(i)}$ still requires at least $O(2N_p^2 \cdot N_{\Theta})$ times of the cosine and sine summations by using a look-up table and needs a large number of the execution cycles, additions and memory usage. If $\theta_{l, l'}^{(i)}$ is used to denote $(\tau_l^{(i)} - \tau_{l'}^{(i)})$, when $l \neq l'$, the value of $[-(\mathbf{E}^{(i)})_{l, l'}] / N_{\Theta}^2$, denoted as $E(\theta_{l, l'}^{(i)})$, can be further expressed to

$$E(\theta_{l, l'}^{(i)}) = -\frac{1}{N_{\Theta}^2} \cdot \left(\sum_{k \in \Theta} \cos\left(\frac{2\pi k \theta_{l, l'}^{(i)}}{N}\right) + j \sum_{k \in \Theta} \sin\left(\frac{2\pi k \theta_{l, l'}^{(i)}}{N}\right) \right). \quad (4.35)$$

Because the significant path delays are smaller than N_B , the value of $\theta_{l, l'}^{(i)}$ is in the

range $1 < |\theta_{l,p}^{(i)}| < N_B$. For low complexity implementation, all possible values of $E(\theta_{l,p}^{(i)})$ which are the coefficients of the path decorrelation can be calculated and stored in advance, and on-line computations of $\mathbf{E}^{(i)}$ can be saved.

After the path decorrelator, the decorrelated gradients pass through FFT to acquire the gradients in frequency domain. Finally, the new estimated channel frequency responses are updated by subtracting these gradients from the latest estimated channel frequency responses.

4.5 Computational Complexity

In this section, the computational complexity and storage requirement of the 2-D interpolation-based channel estimation methods, the DFT-based channel estimation methods and the proposed two-stage channel estimation method for one pair channel estimation are presented and listed in Table 4.1. The computational complexity is considered in terms of real multiplications. The operation of radix-4 N -point FFT requires $4N \log_4 N$ real multiplications.

The 2-D interpolation-based channel estimation methods first carries out the time-domain interpolation among contiguous OFDM symbols to obtain the corresponding channel frequency responses and then performs frequency-domain interpolation by using pilot subcarriers and the interpolated subcarriers obtained from time-domain interpolation. The computational complexity of the 2-D interpolation-based methods is about $2N_{Pilot} + 4J_t \cdot N_t + 4J_f \cdot (N_{Data} - N_t)$ real multiplications, where J_t is the tap number of time-domain interpolation, N_t is the number of the interpolated subcarriers obtained from time-domain interpolation, J_f is the tap number of frequency-domain interpolation, and N_{Pilot} and N_{Data} are the number of pilot and data subcarriers in an OFDM symbol, respectively. The real multiplications of $2N_{Pilot}$ are required for LS estimation at pilot subcarriers. The number of $\lfloor J_t/2 \rfloor \cdot N_{Pilot} + \lceil J_t/2 \rceil \cdot N_{Data}$ data buffers is required due to its non-causal property. The number of $(L-1) \cdot (J_t + J_f) + (J_t + J_f)$ storage is required for interpolation coefficients and estimator buffers, where L is the interval of pilot subcarriers.

The computational complexity of the classical DFT-based channel estimation methods is about $4N_{Pilot} + 4N_{Pilot} \log_4 N_{Pilot} + 4N \log_4 N$ real multiplications, where N is the

TABLE 4.1
COMPUTATIONAL COMPLEXITY AND STORAGE REQUIREMENTS

Channel Estimation Methods		Number of Real Multiplications	Storage Requirement
2-D Interpolation-based Methods		$2N_{Pilot}+4J_i \cdot N_t+4J_f(N_{Data}-N_t)$	$\lfloor J_i/2 \rfloor \cdot N_{Pilot} + \lceil J_i/2 \rceil \cdot N + (L-1) \cdot (J_i+J_f) + (J_i+J_f)$
Classic DFT-based Methods		$4N_{Pilot}+4N_{Pilot}\log_4 N_{Pilot}+4N\log_4 N$	$(2N_{Pilot}+N)+N_{Pilot}+N$
Proposed Two-stage Method	Initialization Stage (within preamble symbol)	$2(N_{Data}+N_{Pilot})+8N\log_4 N+4N_{TP}^2$	$(N_{Data}+N_{Pilot}+2N)+N_{Pilot}+N$
	Tracking Stage (at each iteration within a time slot)	$8N_{Data}+2N_{Pilot}+8N\log_4 N+4N_P^2$	$(N_{Data}+N_{Pilot}+2N)+N_P+N$

transmitted subcarrier number of an OFDM symbol. The real multiplications of $4N_{Pilot}\log_4 N_{Pilot}+4N\log_4 N$ are required for radix-4 FFT/IFFT, and that of $4N_{Pilot}$ are required for LS estimation at pilot subcarriers and windowing the CIR in time-domain. The number of N storage is shared for FFT/IFFT operation. The number of $(2N_{Pilot}+N)+N_{Pilot}$ storage is required for data buffers and weighting coefficients. When the taps of the 2-D interpolation-based methods increases to improve the estimation accuracy, the requirements of real multiplication of the 2-D interpolation-based methods and the DFT-based methods will have quite the same complexity. For example, if J_i and J_f are defined to be 6, and N_t is equal to N_{Pilot} , according to the parameters listed in Table 3.1, the 2-D interpolation-based channel estimation method requires 17,520 real multiplications, and the DFT-based channel estimation method requires 18,458 real multiplications.

By using the proposed two-stage channel estimation method, the computational complexity of the initialization stage requires about $2(N_{Data}+N_{Pilot})+8N\log_4 N+4N_{TP}^2$ real multiplications. The number of N storage is shared for FFT/IFFT operation. The number of $(N_{Data}+N_{Pilot}+2N)+N_{Pilot}$ storage is required for data buffers and SMPIC-based decorrelation. Moreover, the initialization stage is only executed once in each frame (within the preamble symbol). At each iteration, the computational complexity of the tracking stage requires about $8N_{Data}+2N_{Pilot}+8N\log_4 N+4N_P^2$ real multiplications. The storage resources can be shared in the operations of the

initialization stage and the tracking stage. The proposed two-stage channel estimation method adopts the decision data symbols as pilots to track channel variations for providing sufficient tracking information, so it is more robust than the DFT-based method to apply in outdoor mobile channels. However, the complexity is also increased. According to the parameters listed in Table 3.1, when the values of N_P and N_{TP} are defined to be 8 and 32, the proposed method requires 44,688 real multiplications in the initialization stage and requires 47,216 real multiplications at each iteration in the tracking stage. The complexity is dominated by the operations of FFT and IFFT. However, many partial FFT algorithms have been presented [63], [64], and they can be adopted in the proposed two-stage channel estimation to further reduce the computational complexity.

4.6 Simulation Results

The performances of the proposed channel estimator are demonstrated through the simulation of an STBC-OFDM system with two transmit antennas and one receive antenna. The multipath channels adopt the ITU Veh-A [48] channel model with relative path power profiles of 0, -1, -9, -10, -15, and -20 (dB), and the path excess delays are uniformly distributed from 0 to 50 sampling periods. Jakes model is also used to generate a Rayleigh fading environment [49].

Fig. 4.6 and Fig. 4.7 show the BER performances of the proposed scheme with different tracking iterations for QPSK and 16-QAM modulations at the vehicle speed v_e of 120 km/hr. The maximum Doppler frequency f_D is 277.8 Hz and is about 0.025 normalized to subcarrier spacing Δf . The result of perfect channel estimation, denoted as perfect CSI, is included for benchmarking. As shown in Fig. 4.6, due to the first tracking iteration in the tracking stage that only uses the pilot subcarriers to track channel variations and provide a global search direction, the performance curve of the proposed scheme with one tracking iteration has a performance gap as compared with the perfect CSI curve. However, when the tracking iteration number increases, the performances of the proposed scheme become more close to the perfect CSI case. In QPSK modulation, the performance curves of the proposed scheme with two, three and four tracking iterations have about 1.4, 1.1 and 0.2 dB gaps in E_b/N_0 as compared with the perfect CSI case at $BER=10^{-3}$. BER of about 10^{-4} can be achieved in E_b/N_0 of

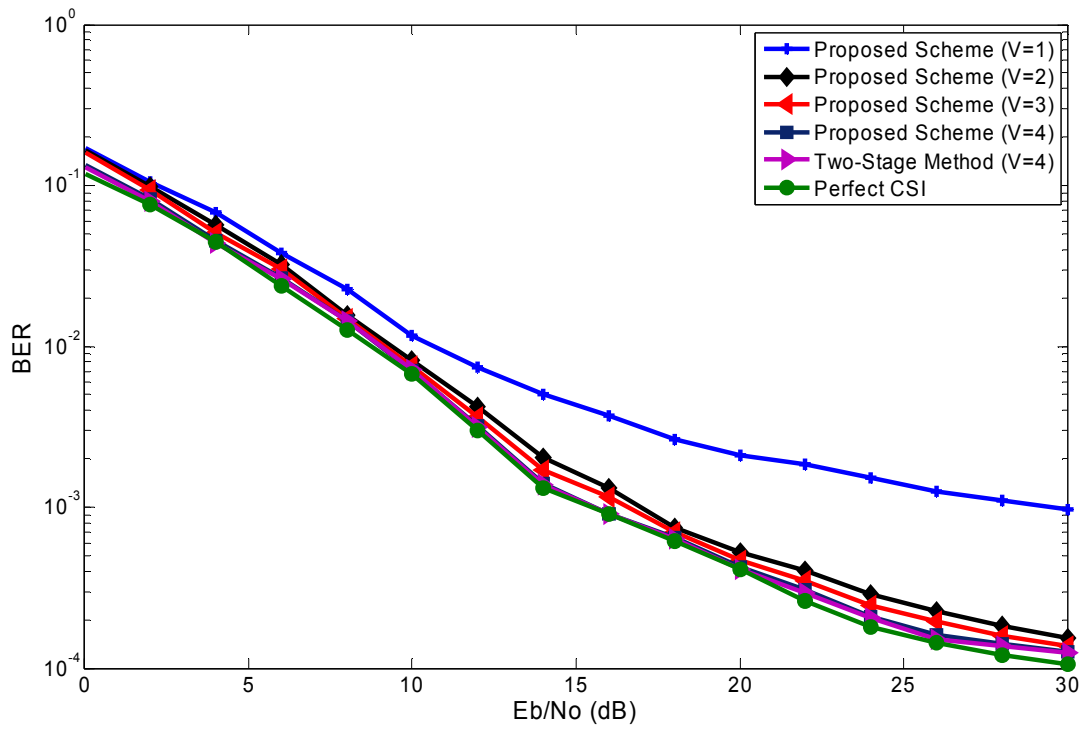


Fig. 4.6 BER performance for QPSK at v_e of 120 km/hr.

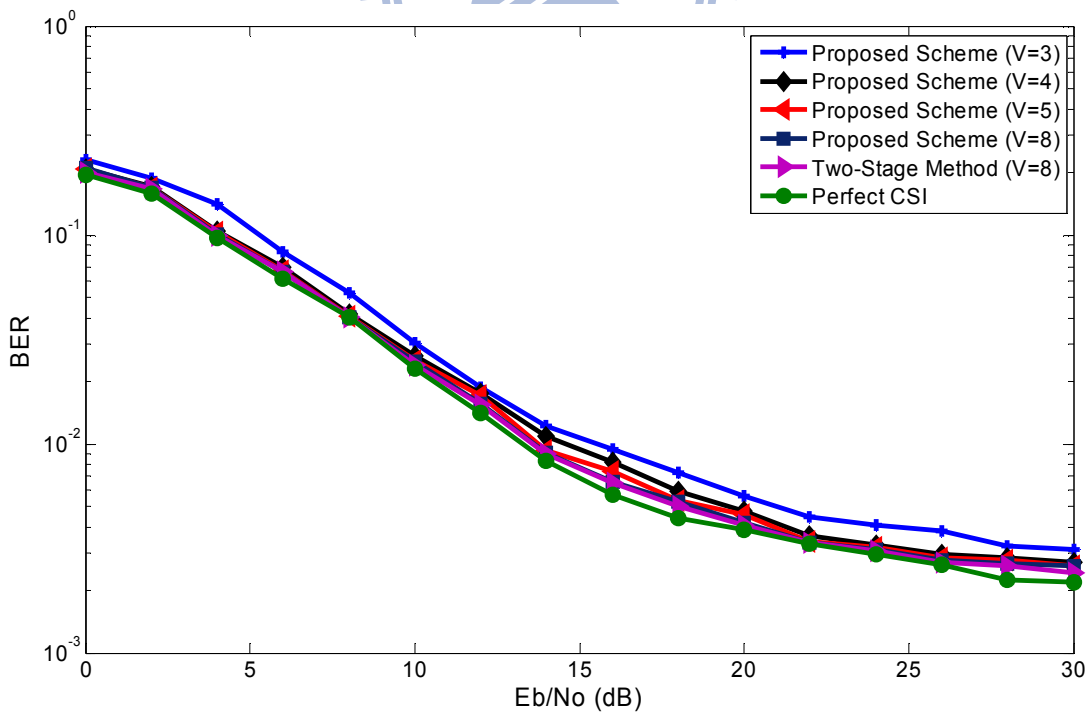


Fig. 4.7 BER performance for 16QAM at v_e of 120 km/hr.

30 dB. Moreover, the performances of the proposed scheme and the original two-stage method with four tracking iterations are almost the same. As shown in Fig. 4.7, in 16-QAM modulation, the performance curves of the proposed scheme with three, four, five, and eight tracking iterations have about 2.2, 1.2, 0.5 and 0.2 dB gaps in E_b/N_0 as compared with the perfect CSI case at $BER=10^{-2}$. BER of about 10^{-3} can be achieved in E_b/N_0 of 30 dB. Similarly, the performances of the proposed scheme and the original two-stage method with eight tracking iterations are very close.

Fig. 4.8 and Fig. 4.9 show the BER performances of the proposed scheme with different tracking iterations for QPSK and 16-QAM modulations at v_e of 240 km/hr. The maximum Doppler frequency f_D is 555.6 Hz and is about $0.051\Delta f$. As shown in Fig. 4.8, in QPSK modulation, the performance curves of the proposed scheme with two, three and four tracking iterations have about 2.1, 1.6 and 0.6 dB gaps in E_b/N_0 as compared with the perfect CSI case at $BER=10^{-2}$. BER of about 10^{-3} can be achieved in E_b/N_0 of 30 dB. As shown in Fig. 4.9, in 16-QAM modulation, the performance curves of the proposed scheme with three, four, five, and eight tracking iterations have about 2.8, 1.2, 0.5 and 0.3 dB gaps in E_b/N_0 as compared with the perfect CSI case at $BER=10^{-1.5}$. BER of about 10^{-2} can be achieved in E_b/N_0 of 30 dB.

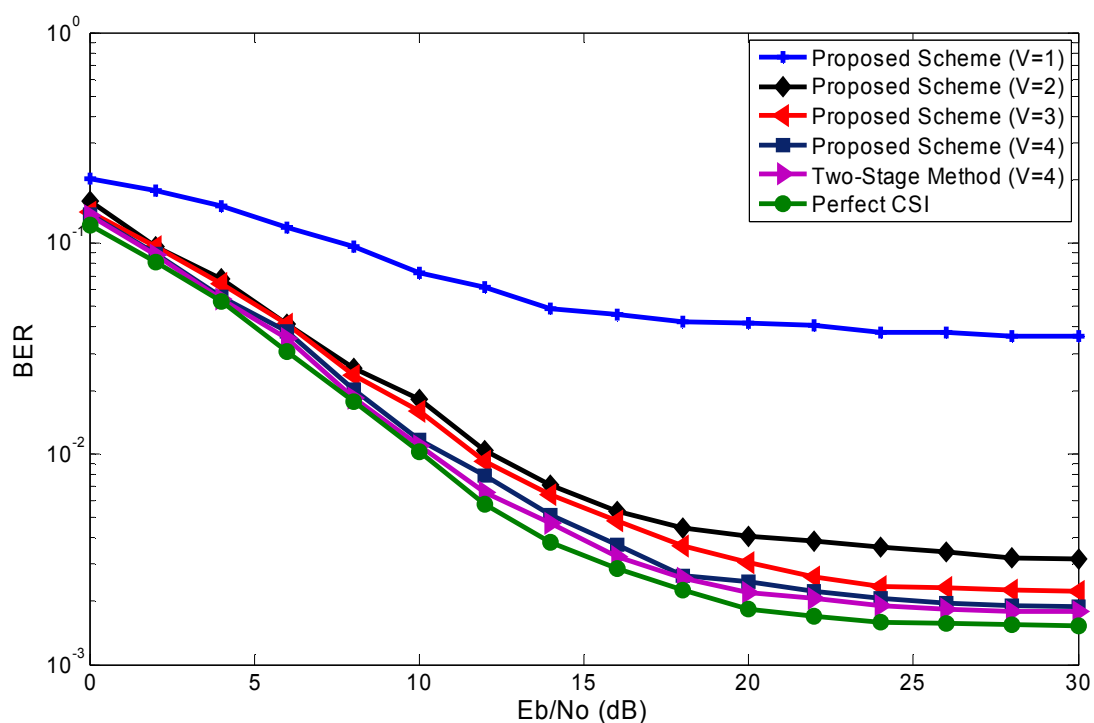


Fig. 4.8 BER performance for QPSK at v_e of 240 km/hr.

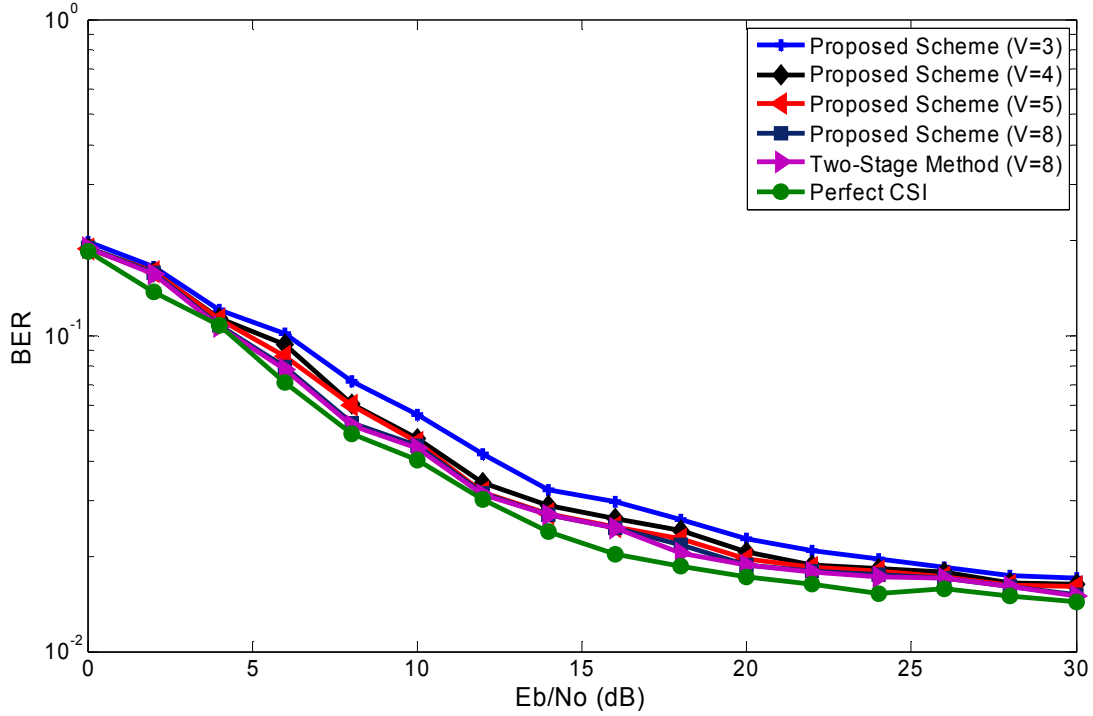


Fig. 4.9 BER performance for 16QAM at v_e of 240 km/hr.

Fig. 4.10 shows the BER performances under different v_e in E_b/N_0 of 16 dB. At v_e of 120 km/hr ($f_D=0.025\Delta f$), BER of the perfect CSI case and the proposed scheme with four tracking iterations for QPSK/16-QAM/64-QAM modulation can achieve about $8.9 \times 10^{-4}/4.6 \times 10^{-3}/3.9 \times 10^{-2}$ and $9.6 \times 10^{-4}/7.6 \times 10^{-3}/4.3 \times 10^{-2}$, respectively, without using channel coding. Even at v_e of 240 km/hr ($f_D=0.051\Delta f$), BER of the proposed scheme with four tracking iterations for QPSK/16-QAM/64-QAM can achieve about $3.1 \times 10^{-3}/2.9 \times 10^{-2}/9.7 \times 10^{-2}$.

Three kinds of interpolation-based channel estimation methods, the 1st-order predictive algorithm, the 2nd-order predictive algorithm and the two dimensional (2-D) interpolation method [53], [54], are simulated to make the performance comparisons. Considering the IEEE 802.16e OFDMA downlink specification, these methods are executed based on the cluster structures (Fig. 4.5) where a cluster consists of 14 consecutive subcarriers with alternating structures in two successive time slots. These interpolation-based methods are applied as follows: 1) for each time slot, do LS channel estimations at pilot subcarriers as described in (4.31)-(4.32), where we assume that the channel within two consecutive OFDM symbols is quasi-static; 2) according to [53] and [54], among consecutive time slots, do the time-domain

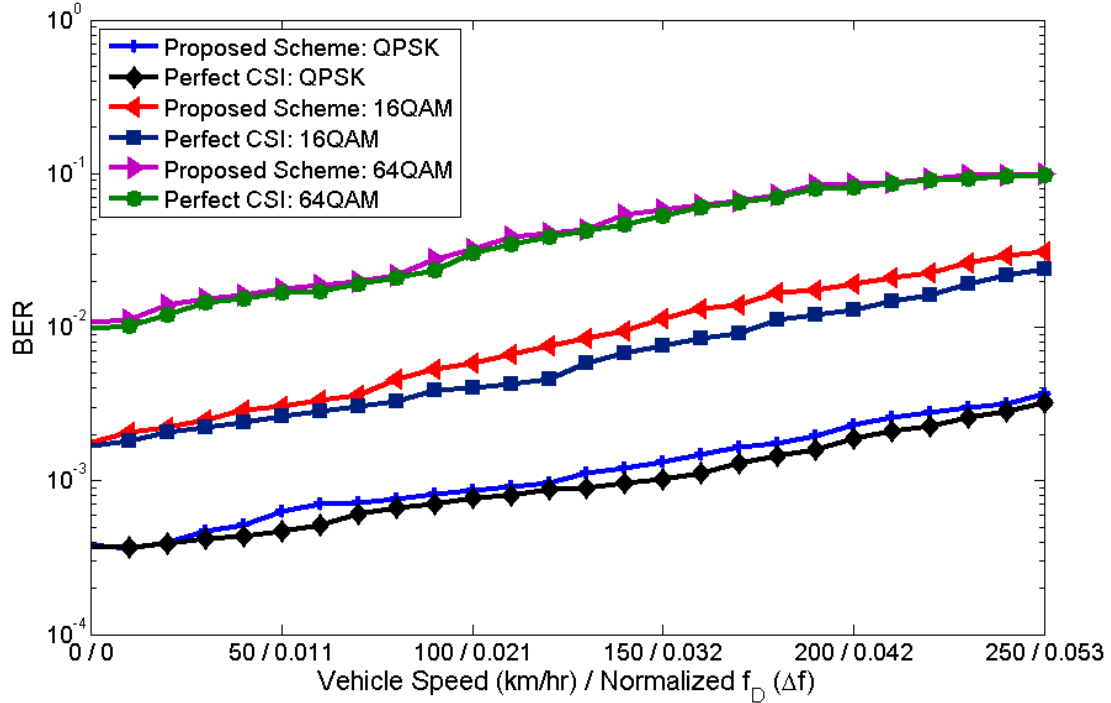


Fig. 4.10 BER performances versus the vehicle speed.

interpolation of the corresponding channel frequency response for each specific transceiver antenna pair; 3) perform linear frequency-domain interpolation by using pilot subcarriers and the interpolated subcarriers obtained from time-domain interpolation.

Fig. 4.11 shows the normalized mean square errors (MSE) of channel estimation for QPSK modulation under different methods at v_e of 120 km/hr. As shown in the figure, the performance curves of the interpolation-based methods exhibit an error floor phenomenon. Generally, there are three factors contributing to the channel estimation error of the interpolation-based methods, which are AWGN noise and model errors from both time-domain and frequency-domain interpolations. At low E_b/N_0 situation, the estimation error is mainly dominated by AWGN noise. However, the error floor phenomenon at high E_b/N_0 is due to model errors. The longest interval between the pilot subcarriers transmitted from one antenna is 12 subcarrier spacing, and even that between the pilot and interpolated subcarriers is four subcarrier spacing. Because of both the frequency selective fading caused by larger multipath delay spreads and the time selective fading caused by higher Doppler effect, the interpolation-based methods under the situation of limited pilots in the cluster

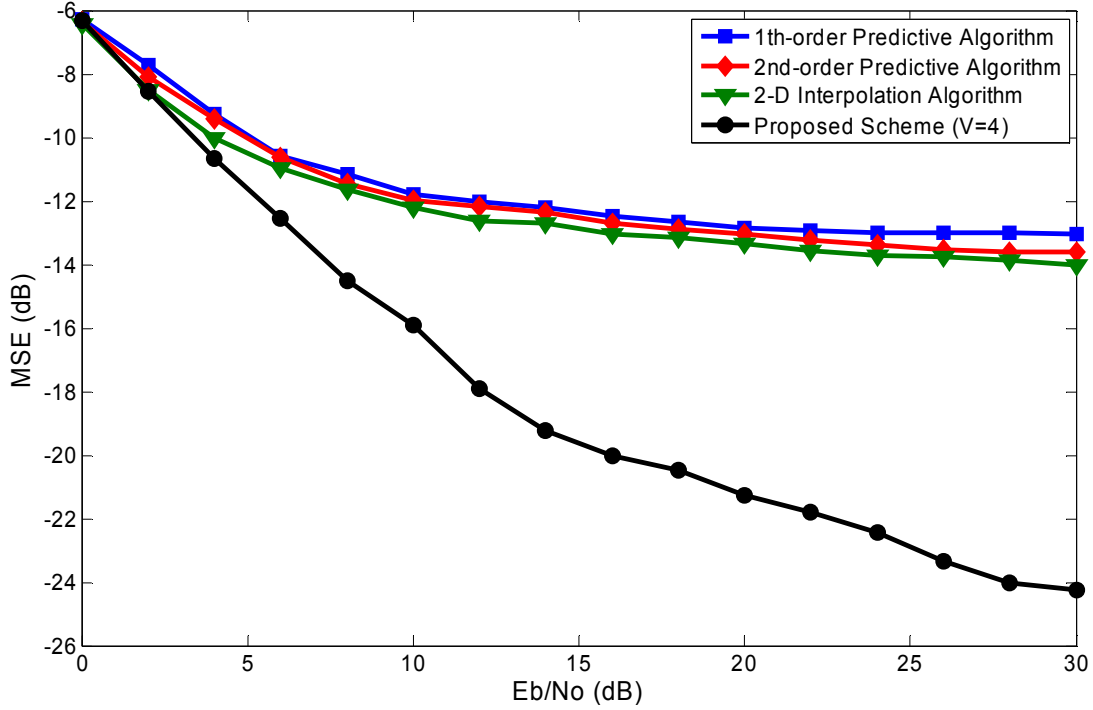


Fig. 4.11 MSE of channel estimations using the proposed scheme, predictive algorithms and 2-D interpolation-based methods at v_e of 120 km/hr.

structures cannot recover the channel frequency response well. In E_b/N_0 of 30 dB, the normalized MSEs of the proposed scheme and the 2-D interpolation method are about -24.3 dB and -13.9 dB. As discussed above, the 2-D interpolation method performs the linear interpolation in time and frequency axes. We further perform the 2-D interpolation method with the second order [65] and cubic [66] interpolations in frequency axis, respectively, and the MSE of channel estimation for QPSK modulation at v_e of 120 km/hr are shown in Fig. 4.12. Even using the second order and cubic interpolations in frequency axis, the performances of the 2-D interpolation methods are also inferior to our proposed method. The MSE of the 2-D interpolation method with the second order and cubic interpolations in frequency axis are about -15.0 dB and -15.8 dB in E_b/N_0 of 30 dB.

Although the interpolation-based methods have lower complexity for implementation, our proposed scheme has lower MSE of channel estimation and better performance especially in outdoor mobile environments.

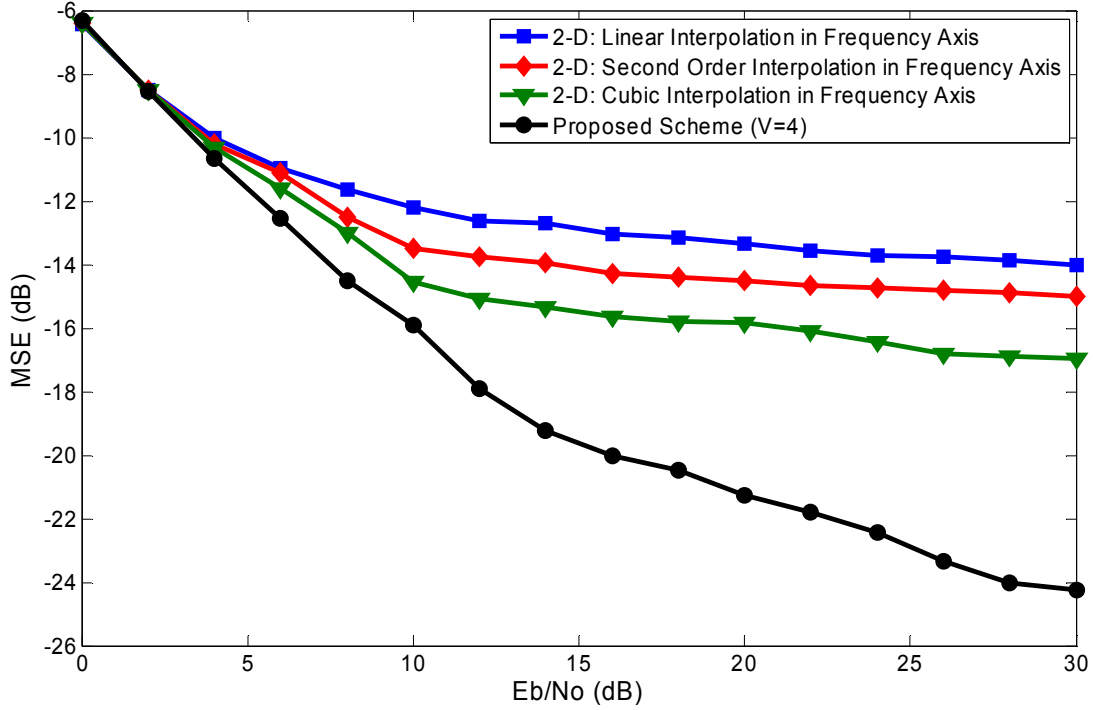
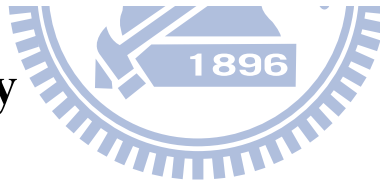


Fig. 4.12 MSE of channel estimations using the proposed scheme and 2-D interpolation-based methods with different interpolations in frequency axis at v_e of 120 km/hr.

4.7 Summary



In this chapter, a two-stage channel estimation method is proposed for STBC-OFDM systems in wireless mobile channels. The initialization stage uses a MPIC-based decorrelation method to identify the significant paths of CIR in the beginning of each frame. The tracking stage is then used to track the path gains with known CIR positions. When operating at v_e of 120 and 240 km/hr with E_b/N_0 of 16 dB for QPSK modulation, the proposed channel estimation method can achieve BER of about 10^{-4} and 10^{-3} without using channel coding. As compared with interpolation-based channel estimation methods which are commonly adopted in the pilot-aided channel estimator designs [53], [54], [65], and [66], under 120 km/hr, the proposed channel estimation method has the normalized MSE improvements of 8.5~10.4 dB in E_b/N_0 of 30 dB.

Chapter 5

Novel Programmable FIR Filter Based on Higher Radix Recoding

5.1 Introduction

In baseband communication systems, finite impulse response (FIR) filter is one of the most widely used block. In wire and wireless baseband communications, high performance programmable FIR filters are frequently used to perform adaptive pulse shaping and signal equalization on the received data. Complexity reduction of FIR filter implementations has been of particular interest because lower computational complexity leads to high performance as well as low power design.

A novel programmable FIR filter based on higher radix recoding is proposed for high performance and low power applications [67]. We employ the higher radix recoding [68]-[70] to reduce the number of partial products and reduce power consumption by pre-computation sharing in each partial product generation. However, the propagation delay of pre-computing multiples is increasing by increasing higher radices. We can extend recoding methodology by using secondary radix recoding schemes [70] to reduce the number and propagation delay of odd multiples requirement for very high radix recoding. Although the proposed receiver does not implement the programmable FIR filters, the computation sharing concept will be used in the proposed receiver for low-complexity implementation.

The remainder of this chapter is organized as follows. Section 5.2 presents the algorithm reformulations with higher radix recoding and secondary radix recoding.

Section 5.3 describes the proposed programmable FIR filters based on higher radix recoding and secondary radix recoding. Section 5.4 shows the design results and comparisons of programmable FIR filters. Finally, Section 5.5 draws the conclusions of this chapter.

5.2 Algorithm Reformulations

The reduction in partial products obtained by the well-known modified Booth multiplier recoding is offset by the need to pre-compute odd multiples of the multiplicand as inputs to each partial product generation. We propose to use higher radix recoding to decrease the number of partial products and reduce power consumption by pre-computation sharing in each partial product generation. However, increasing radices to reduce the number of partial products will increase the propagation delay of pre-computing multiples of the multiplicand. We propose to extend recoding methodology by secondary radix recoding schemes to reduce the number of odd multiples required for very high radix recoding.

5.2.1 Higher Radix Recoding

A $w \times w$ -bit multiplication $X \cdot C$ in its simplest form is implemented by the generation of w partial products. We denote X and C in binary number representation.

$$X = \sum_{i=0}^{w-1} x[i] \cdot 2^i \quad (5.1)$$

$$C = \sum_{i=0}^{w-1} c[i] \cdot 2^i \quad (5.2)$$

The product computation is based on the sum:

$$X \cdot C = \sum_{i=0}^{w-1} X \cdot c[i] \cdot 2^i \quad (5.3)$$

with the partial product, $P_i = X \cdot c[i] \cdot 2^i$. The factor 2^i is realized by a shift of the multiplicand X , and the factor $c[i] \in \{1, 0\}$ is employed by a partial product generation to select either $X \cdot 2^i$ or zero as i -th partial product in the w -term sum. The main approach to decrease the number of partial products in (5.3) is to represent the multiplier C in a higher radix. The multiplier C is recoded to a $p = \lceil (w+1)/r \rceil$ -digit minimally redundant (Booth) radix representation

$$C = \sum_{i=0}^{p-1} d_i \cdot (2^r)^i \quad (5.4)$$

in a higher radix $\beta = 2^r$, where $d_i \in \{-2^{r-1}, \dots, 2^{r-1}\}$. The representation of the product is

$$X \cdot C = \sum_{i=0}^{p-1} X \cdot d_i \cdot (2^r)^i \quad (5.5)$$

Each partial product has expanded factorization

$$X \cdot d_i \cdot (2^r)^i = X \cdot [(-1)^s \cdot \alpha_i \cdot 2^e] \cdot 2^{ir} = (-1)^s \cdot \{\alpha_i \cdot X\} \cdot 2^{e+ir} \quad (5.6)$$

where $\alpha_i \in \{0, 1, 3, \dots, 2^{r-1}-1\}$. The 2^{r-2} odd multiples of the multiplicand $\{\alpha_i \cdot X\}$ serve as primitive digits set to be conditionally selected, complemented by $(-1)^s$, and shifted by 2^{e+ir} to form the partial products to be accumulated.

5.2.2 Secondary Radix Recoding

The radices greater than eight will increase the number of required odd multiples and therefore increase the complexity and delay of the carry propagate adder to pre-compute the odd multiples. We extend higher radix (radix $\beta = 2^r$, $r \geq 5$) recoding by the secondary radix recoding to express the sum of partial products and reduce the number of required odd multiples. As show in (5.4), each Booth digit value

$-2^{r-1} \leq d_i \leq 2^{r-1}$ is recoded to n -digit value ($2 \leq n \leq 4$) in secondary radix recoding scheme. For example, $n=2$ and the secondary radix (modulus) λ , we recode the digit d_i to be

$$d_i = d_{1,i} \cdot \lambda + d_{0,i} \quad (5.7)$$

where digits $d_{1,i}$ and $d_{0,i}$ are chosen from a balanced complete residue system modulo λ forming the secondary radix digit set \mathbf{D}_λ . The residue digit sets of the form

$$\mathbf{D}_\lambda = \{0, \pm d_1, \pm d_2, \dots, \pm d_{(\lambda-1)/2}\} \quad (5.8)$$

are termed balanced complete residue systems when every integer m ($0 \leq m \leq \lambda-1$) is congruent to one member of \mathbf{D}_λ . Since complements and shifts can be employed to increase the range of digit values, all nonzero digits of \mathbf{D}_λ should be of the form $\pm \delta \cdot 2^i$, where δ is a member of $\{0, 1\}$, $\{0, 1, 3\}$ or $\{0, 1, 3, 5\}$. For $n=2$, (5.4) can be expressed as

$$C = \lambda \cdot \sum_{i=0}^{p-1} d_{1,i} \cdot (2^r)^i + \sum_{i=0}^{p-1} d_{0,i} \cdot (2^r)^i \quad (5.9)$$

where $d_{1,i}, d_{0,i} \in \mathbf{D}_\lambda$. (5.5) can be expressed as

$$X \cdot C = \lambda \cdot \sum_{i=0}^{p-1} X \cdot d_{1,i} \cdot (2^r)^i + \sum_{i=0}^{p-1} X \cdot d_{0,i} \cdot (2^r)^i. \quad (5.10)$$

The value of λ should be an odd and prime for the radix $\beta=2^r$ and must be of the order of an n -th root of β . λ is better to be implemented by 2 or 3 nonzero bits to allow multiplication by λ to be a shift and add/subtract. The n -digit secondary radix values must include all integers in the high radix Booth digit range $[-2^{r-1}, 2^{r-1}]$.

5.2.3 Algorithm Reformulation for FIR Filter

Considering an N -tap FIR filter with input sequence X_n , output sequence Y_n , and coefficients C_i , without loss of generality, we can express the FIR reformulation of high radix β recoding as

$$\begin{aligned} Y_n &= \sum_{i=0}^{N-1} C_i \cdot X_{n-i} = \sum_{i=0}^{N-1} \left[\sum_{j=0}^{p-1} X_{n-i} \cdot d_{i,j} \cdot (2^r)^j \right] \\ &= \sum_{i=0}^{N-1} \left[\sum_{j=0}^{p-1} (-1)^{s_{i,j}} \cdot \{\alpha_{i,j} \cdot X_{n-i}\} \cdot 2^{jr+e_{i,j}} \right] \end{aligned} \quad (5.11)$$

where $d_{i,j} \in \{-2^{r-1}, \dots, 2^{r-1}\}$ and $\alpha_{i,j} \in \{0, 1, 3, \dots, 2^{r-1}-1\}$. We can extend the FIR reformulation of high radix β with secondary radix recoding as

$$\begin{aligned} Y_n &= \sum_{i=0}^{N-1} C_i \cdot X_{n-i} \\ &= \sum_{i=0}^{N-1} \left[\lambda \cdot \sum_{j=0}^{p-1} X_{n-i} \cdot d_{1,i,j} \cdot (2^r)^j + \sum_{i=0}^{p-1} X \cdot d_{0,i,j} \cdot (2^r)^j \right] \end{aligned} \quad (5.12)$$

where $d_{1,i,j}, d_{0,i,j} \in \mathbf{D}_\lambda$. Each partial product can also be expanded with factorization

$$X_{n-i} \cdot d_{\eta,i,j} \cdot (2^r)^i = (-1)^{s_{\eta,i,j}} \cdot \{\delta_{\eta,i,j} \cdot X_{n-i}\} \cdot 2^{jr+e_{\eta,i,j}} \quad (5.13)$$

where $\delta_{\eta,i,j} = \{1\}, \{1,3\}$ or $\{1,3,5\}$ and $\eta=1$ or 0 .

5.3 Programmable FIR Filter Based on Higher Radix Booth Recoding

5.3.1 Higher Radix Booth Multiplier (HRBM)

Fig. 5.1 shows the parallel architecture of 19×19 -bit radix-16 ($\beta=2^4$) higher radix booth multiplier (HRBM), as shown in (5.5) and (5.6). The HRBM is composed of the odd multiples pre-computation, basic units, and adders. In the implementation of HRBM, the input X , coefficient C , and output $X \cdot C$ are represented in two's complement format.

The pre-computations of odd multiples $\{1 \times, 3 \times, 5 \times, 7 \times\}$ for radix-16 HRBM is utilized by complement and shift operations to generate the full set of partial products $\{-8 \times, -7 \times, \dots, 7 \times, 8 \times\}$ for Booth radix-16 digits. Fig. 5.2 shows the structure of the odd multiples pre-computation and shows the $5 \times$ implemented by carry propagate adder (CPA). The basic unit of HRBM is composed of higher radix encoder (HREnc), 4:1 MUX, Shifter, and AND gates. Since HREnc is directly connected to coefficients,

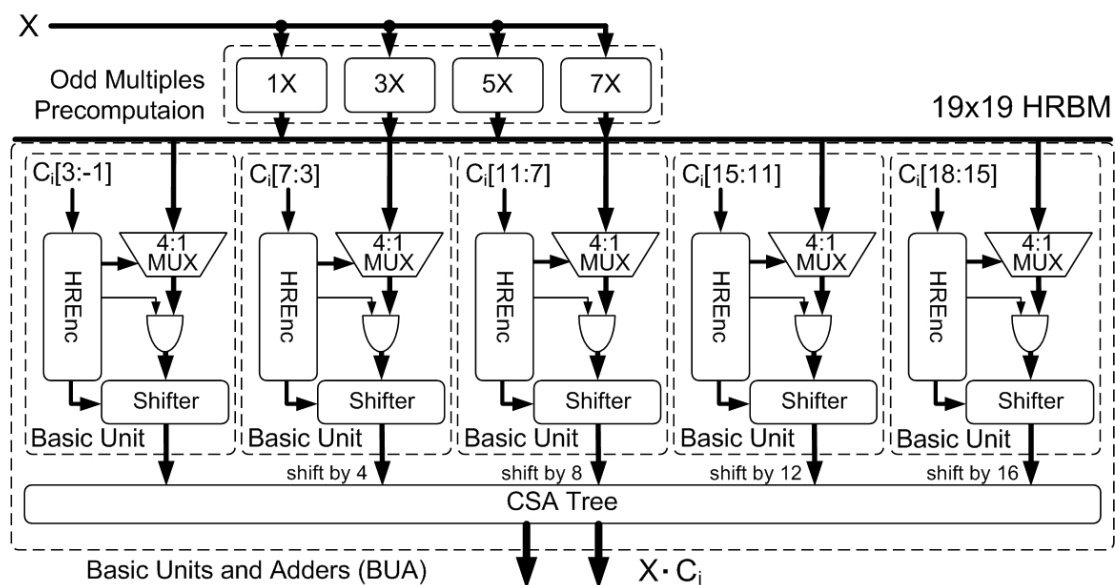


Fig. 5.1 Parallel architecture of 19×19 -bit radix-16 ($\beta=2^4$) HRBM.

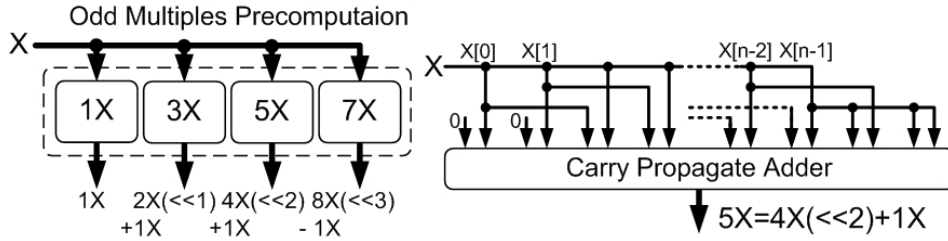


Fig. 5.2 Odd multiples pre-computation structure and pre-compute $5\times$ architecture.

TABLE 5.1
RADIX-16 RECODING SCHEME OF HREnc

Coefficient				4:1MUX	Shifter	AND	Sign		Partial Product	
$C_i[4j+3:4j-1]$				sel	sf	$zero$	sig			
00000		11111		00	00	0	0	1	0	-0
00001	00010	11110	11101	00	00	1	0	1	1	-1
00011	00100	11100	11011	00	01	1	0	1	2	-2
00101	00110	11010	11001	01	00	1	0	1	3	-3
00111	01000	11000	10111	00	10	1	0	1	4	-4
01001	01010	10110	10101	10	00	1	0	1	5	-5
01011	01100	10100	10011	01	10	1	0	1	6	-6
01101	01110	10010	10001	11	00	1	0	1	7	-7
01111		10000		00	11	1	0	1	8	-8

it does not locate on the critical path. Table 5.1 present the radix-16 recoding scheme of HREnc. 4:1 MUX is used to select the odd multiples with control signal sel . Shifter is composed of one 4:1 MUX to select the data shifting 0 to 3 bits with the control signal sf . AND gates and signal $zero$ are used to deal with zero partial product output.

Signal sig decides the partial product will be add ($sig=0$) or subtract ($sig=1$). Finally, we employ the carry save adders (CSA) tree to sum the outputs of five basic units.

5.3.2 HRBM² Based on Secondary Radix Recoding

Fig. 5.3 shows the parallel architecture of 19×19-bit radix-32-modulo-7 ($\beta=25; \lambda=7$) HRBM² based on secondary radix recoding, as shown in (5.10). The HRBM² is composed of the basic units and adders. In the implementation of HRBM², the input X, coefficient C, and output X·C are represented in two's complement format.

The 2-digit values of $\mathbf{D}_7=\{0, \pm 1, \pm 2, \pm 4\}$ cover all integers in Booth radix-32 digit range [-16,16]. The digits of \mathbf{D}_7 can be realized by a conditional complement $(-1)^s$ and a shift 2^e , $e=0, 1, \text{ or } 2$. A 19×19-bit multiplier X·C based on secondary radix recoding to recode coefficient C. (5.10) can be expressed as

$$X \cdot C = 7 \cdot \sum_{i=0}^3 X \cdot d_{1,i} \cdot (2^5)^i + \sum_{i=0}^3 X \cdot d_{0,i} \cdot (2^5)^i \quad (5.14)$$

where $d_{1,i}, d_{0,i} \in \mathbf{D}_7$. Using radix-32-modulo-7 recoding the 19×19-bit multiplier X·C has 4 partial products of each digit expression. Each digit expression can be extended as

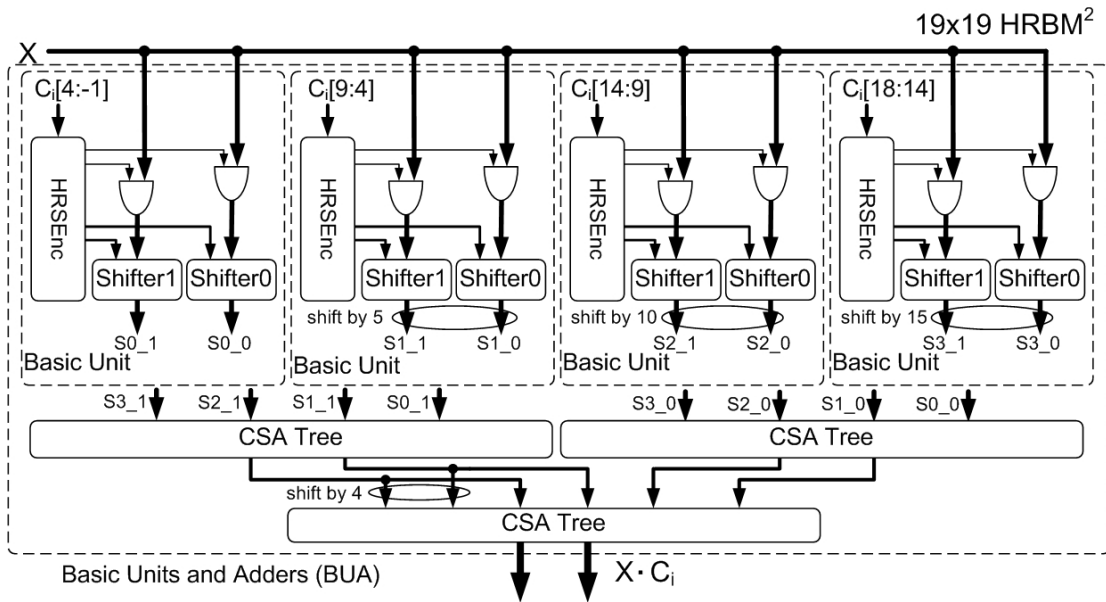


Fig. 5.3 Parallel architecture of 19×19-bit radix-32-modulo-7 ($\beta=25; \lambda=7$) HRBM².

TABLE 5.2
RADIX-32-MODULO-7 RECODING SCHEME OF HRSENC

Coefficient				Shift		AND		Sign		Partial Product			
$C_i[5j+4:5j-1]$				$sf1/sf$ 0		$z[1:0]$ J		$s[1:0]$ J		Radix-32		Modulo-7	
000000		1111111		0	00	0	0	0	1	0	-0	[0,0]	[0,0]
000001	000010	1111110	1111101	0	00	0	1	0	1	1	-1	[0,1]	[0,-1]
000011	000100	1111100	1111011	0	01	0	1	0	1	2	-2	[0,2]	[0,-2]
000101	000110	1111010	1111001	0	10	1	1	0	1	3	-3	[1,-4]	[-1,4]
000111	001000	1111000	1101111	0	10	0	1	0	1	4	-4	[0,4]	[0,-4]
001001	001010	1101110	1101101	0	01	1	1	0	1	5	-5	[1,-2]	[-1,-2]
001011	001100	1101100	1100111	0	10	1	1	0	1	6	-6	[1,-1]	[-1,1]
001101	001110	1100110	1100011	0	00	1	0	0	1	7	-7	[1,0]	[-1,0]
001111	010000	1100000	1011111	0	00	1	1	0	1	8	-8	[1,1]	[-1,-1]
010001	010010	1011110	1011101	0	01	1	1	0	1	9	-9	[1,2]	[-1,-2]
010011	010100	1011100	1010111	1	10	1	1	0	1	10	-10	[2,-4]	[-2,4]
010101	010110	1010110	1010101	1	10	1	1	0	1	11	-11	[1,4]	[-1,-4]
010111	011000	1010000	1001111	1	01	1	1	0	1	12	-12	[2,-2]	[-2,2]
011001	011010	1001110	1001101	1	00	1	1	0	1	13	-13	[2,-1]	[-2,1]
011011	011100	1001100	1000111	1	00	1	0	0	1	14	-14	[2,0]	[-2,0]
011101	011110	1000110	1000011	1	00	1	1	0	1	15	-15	[2,1]	[-2,-1]
011111		100000		1	01	1	1	0	1	16	-16	[2,2]	[-2,-2]

$$X \cdot d_{\eta,i} \cdot (2^5)^i = (-1)^{s_{\eta,i}} \cdot \{\delta_{\eta,i} \cdot X\} \cdot 2^{5i+e_{\eta,i}} \quad (5.15)$$

where $\delta_{\eta,i} = \{0, 1\}$, $e_{\eta,i} = 0, 1, \text{ or } 2$, and $\eta = 1 \text{ or } 0$.

The basic unit of HRBM² is composed of secondary radix encoder (HRSEnc), Shifter1, Shifter0, and AND gates. Since HRSEnc is directly connected to coefficients, it does not on the critical path. Table 5.2 presents the radix-32-modulo-7 recoding

scheme of HRSEnc. Shifter0 is composed of one 4:1 MUX to select the data shifting 0 to 2 bits with the control signal $sf0$. Shifter1 is composed of one 2:1 MUX to select the data shifting 0 to 1 bit with the control signal $sf1$. AND gates and signal $z[1:0]$ are used to deal with zero partial product of $\delta_{\eta,i}$. Signal $s[1:0]$ decides the partial product will be add ($s[1]$ or $s[0]=0$) or subtract ($s[1]$ or $s[0]=1$). Finally, we employ the CSA trees to sum the output of each digit expression.

5.3.3 Programmable FIR Filter Architectures

We implement the 10-tap programmable FIR filters based on HRBM and HRBM², respectively. For high performance, the transposed direct form architecture with CSA summation is used to implement the FIR filter. Fig. 5.4 shows the FIR filter using radix-16 HRBM consists of one odd multiples pre-computation and ten basic units and adders (BUAs) of HRBM. The HRBM scheme efficiently removes the redundant computations by sharing the odd multiples pre-computation in FIR filter operation, which leads to low-power design. For further improvement of high performance, Fig. 5.5 shows the FIR filter using radix-32-modulo-7 HRBM² only consists of ten BUAs of HRBM². Since the odd multiple of radix-32-modulo-7 HRBM² is only $\{1\times\}$, it efficiently eliminates the propagation delay of pre-computing multiples.

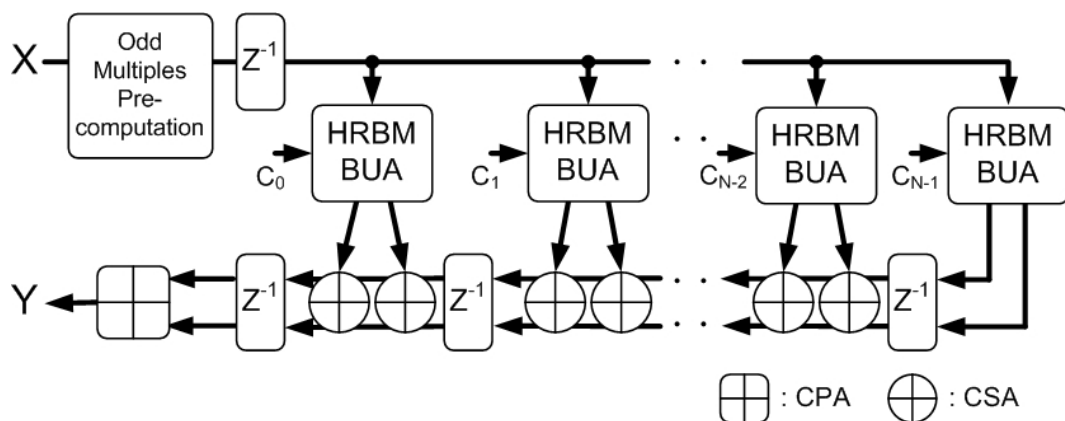


Fig. 5.4 Programmable FIR filter using HRBM.

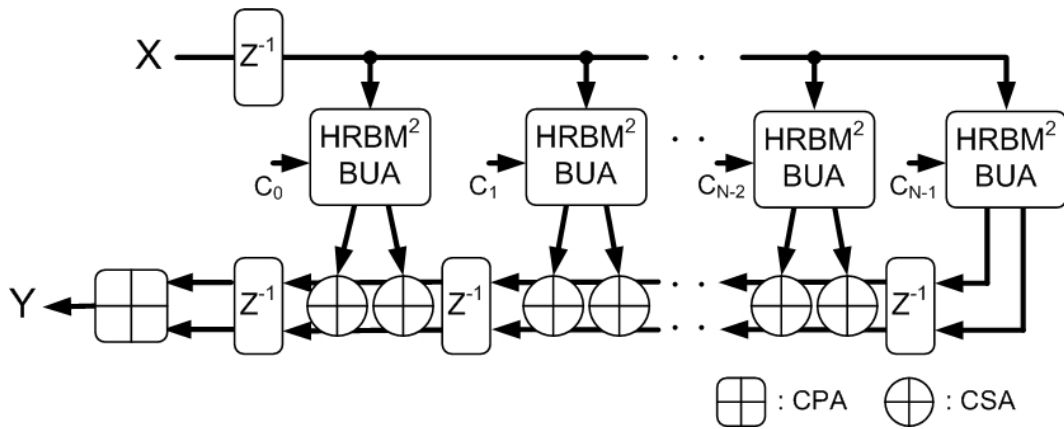


Fig. 5.5 Programmable FIR filter using HRBM².

5.4 Results and Comparisons

Two 10-tap programmable FIR filters using proposed 19×19-bit HRBM and HRBM² are synthesized by TSMC 0.18 μm CMOS technology. We also implement the 10-tap FIR filters using 19×19-bit carry save array multiplier (CSAM), Wallace tree multiplier (WTM), and computation sharing multiplier (CSHM) [71] for comparisons. WTM and CSAM are two widely used multipliers. Since the tree structure of partial products summation, WTM has better performance than CSAM. CSHM also has the concept of computation sharing for low complexity FIR filter design. We design the 10-tap FIR filter using CSHM as proposed in [71]. Since the filter architecture in [71] has one pipeline stage, we also implement the filters using different multipliers with one pipeline stage. Table 5.3 shows the clock cycle, area and power of the filters using different multipliers for low power or high speed design constraints. The power results shown in Table 5.3 are measured by the clock rate of 172 MHz (clock cycle=5.8 ns) for low power or high speed design constraints.

Since WTM- and CSAM-based architectures do not use computation sharing and perform redundant computations for all taps, the FIR filter using HRBM shows better results in terms of speed and power. CSHM-based architecture requires eight odd multiples pre-computations, {1×, 3×, ..., 15×}, that are two times of odd multiples requirement of HRBM-based architecture. The basic unit of CSHM is more

TABLE 5.3
DESIGN RESULTS OF 10-TAP PROGRAMMABLE FIR FILTER

	For Low Power Design				
	HRBM	HRBM ²	CSHM	WTM	CSAM
Clock cycle (ns)	5.8	5.8	5.8	5.8	5.8
Area (gates)	36455	45582	47261	39501	60298
Power@172 MHz (mW)	68.32	77.71	126.00	108.38	137.10
	For High Speed Design				
	HRBM	HRBM ²	CSHM	WTM	CSAM
Clock cycle (ns)	2.6	2.0	4.8	4.4	5.8
Area (gates)	37634	47193	59101	51070	60298
Power@172 MHz (mW)	69.68	78.55	160.31	142.78	137.10

complicated than the BUA of HRBM. Besides, HRBM-based architecture uses CSA summation to efficiently reduce critical path delay. Therefore, HRBM-based architecture has better speed and power results than CSHM-based architecture. Fig. 5.6 shows normalized power, area, and clock cycle of different multiplier-based architectures. The FIR filter using HRBM has 45.8%, 40.9%, and 55.2% performance improvement over FIR filter using CSHM, WTM, and CSAM. HRBM-based architecture has 45.7%, 37.0%, and 50.2% power reduction over the FIR filter based on CSHM, WTM, and CSAM. Furthermore, HRBM²-based architecture efficiently eliminates the propagation delay of pre-computing multiples. In terms of performance and power consumption, the HRBM² scheme has 54.5~65.5% and 28.3~43.3% improvement with respect to the FIR filter based on CSHM, WTM, and CSAM.

5.5 Summary

In this chapter, we present novel programmable digital FIR filters for low-power and high-performance applications. The proposed programmable FIR filters can be used to perform adaptive pulse shaping and signal equalization in wire or wireless

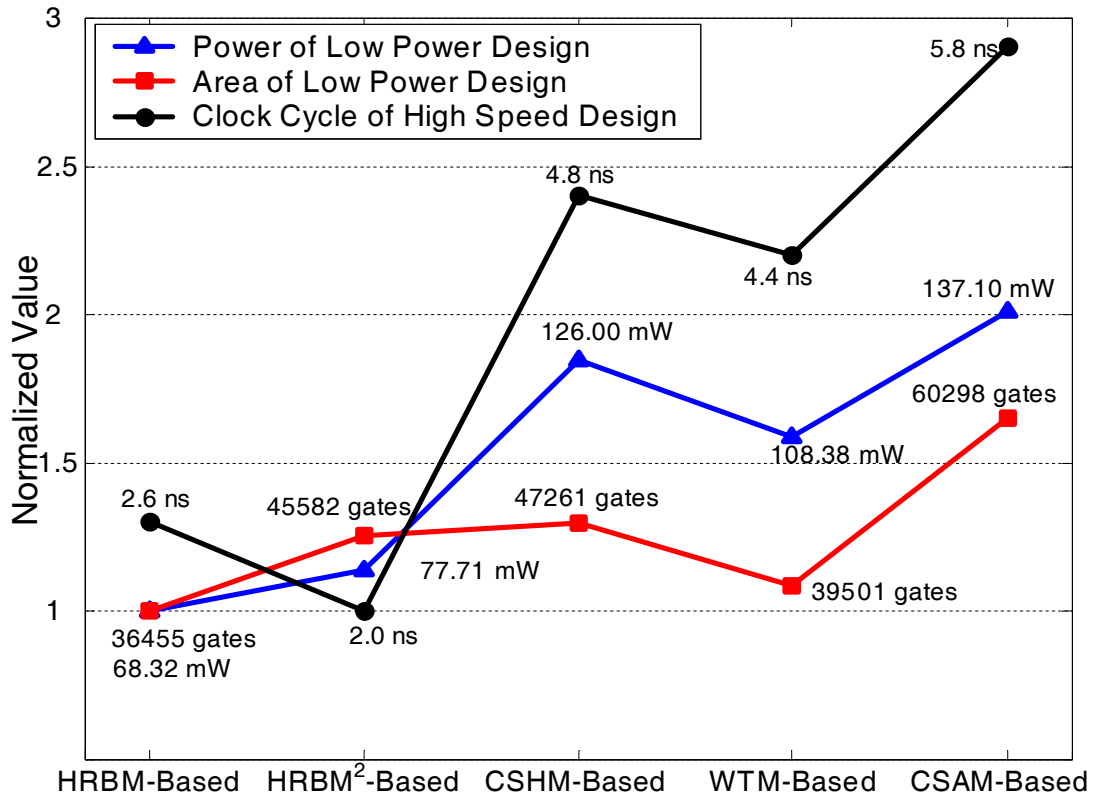
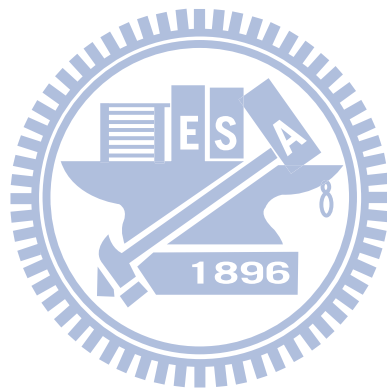


Fig. 5.6 Design results comparison of different multiplier-based.

baseband communications. In larger word length signal cases, for example, 19 bits, by using higher radix recoding to decrease partial products and pre-computation sharing in each partial product, HRBM-based architecture has better results in terms of performance and power consumption that are about 40.9~55.2% and 37.0~50.2% improvement over other designs. Extending higher radix recoding scheme with secondary radix recoding can further improve performance by reducing the propagation delay of pre-computing odd multiples. Thus, HRBM²-based architecture can improve performance about 54.5~65.5% and power consumption about 28.3~43.3% over other designs.



Chapter 6

Downlink Baseband Receiver

Implementation

6.1 Introduction

In this chapter, the hardware implementation of the proposed downlink baseband receiver for IEEE 802.16e in mobile mode will be presented. The proposed receiver applied in STBC-OFDM system with two transmit antenna and one receive antenna aims to provide high performance in wireless mobile environments. The proposed baseband receiver includes the following features:

- provision of the simple and robust schemes and architectures for the symbol boundary detector and carrier frequency offset estimator;
- adoption of a high-performance two-stage channel estimation method for providing precise CSI in wireless mobile channels;
- provision of an efficient channel estimator architecture for low-complexity hardware implementation while keeping the high performance;
- implementation of a baseband receiver applied in an STBC-OFDM system with two transmit antennas and one receive antenna;
- for IEEE 802.16e specification, the STBC-OFDM baseband receiver supporting up to 27.32 Mbps uncoded data transmission in 16-QAM

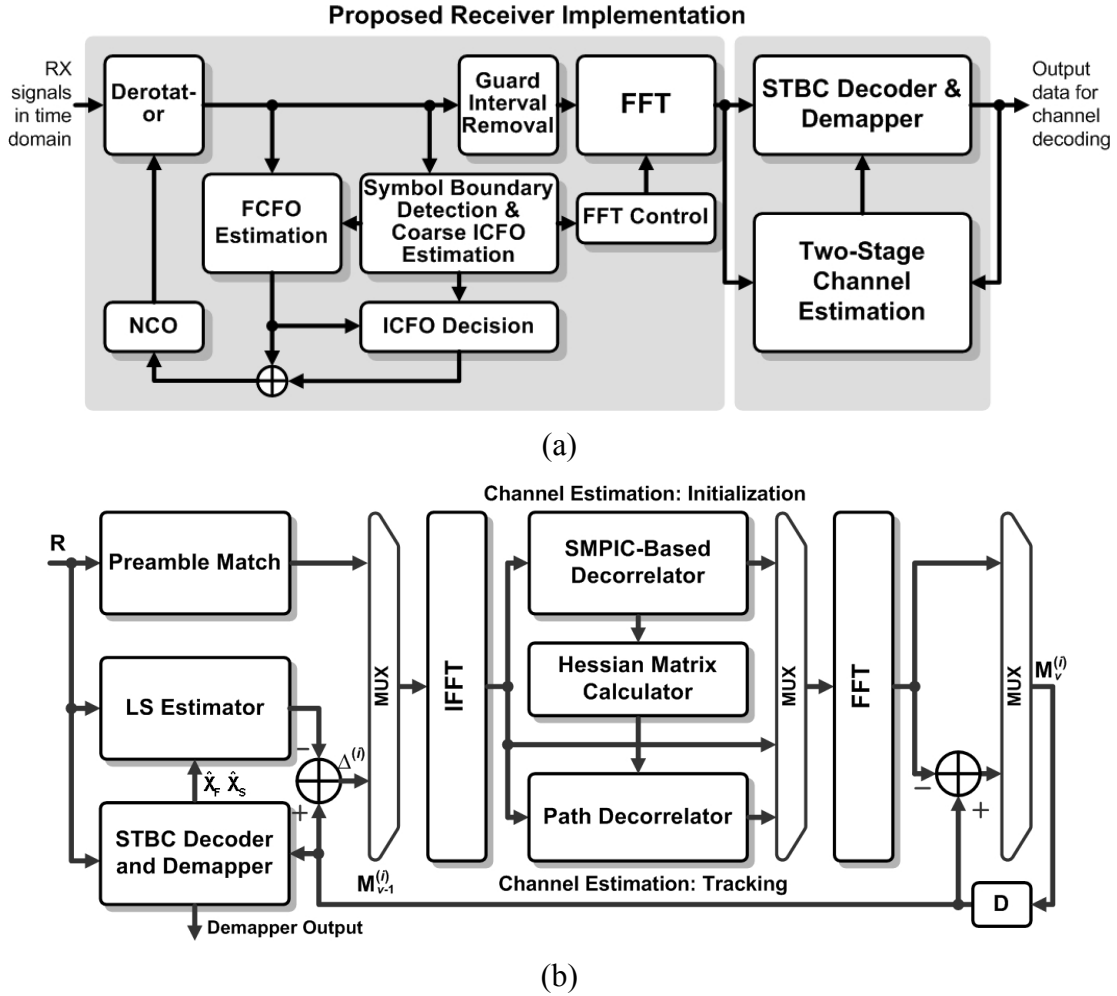


Fig. 6.1 Architectures of (a) the proposed downlink baseband receiver and (b) the proposed two-stage channel estimator with the STBC decoder and demapper.

modulation;

- operation at 11.2 MHz sampling clock and 78.4 MHz operation clock while drawing 68.48 mW from 1V supply voltage by using 90 nm CMOS process.

Fig. 6.1 (a) shows the architecture of the proposed downlink baseband receiver. The baseband receiver includes a symbol boundary detector, an ICFO/FCFO estimator, an FFT, a two-stage channel estimator, an STBC decoder, and a demapper. The architecture of the proposed two-stage channel estimator with the STBC decoder and demapper is also shown in Fig. 6.1 (b). Fig. 6.2 shows the implementation flow. The proposed STBC-OFDM system is constructed by C/C++ language for the whole

system simulation environment. Floating-point simulation is first carried out to develop the robust architectures under the proposed synchronization and channel estimation methods and to evaluate the system requirements and the target performances as described in Table 3.1. After the system architecture is defined, depending on a tradeoff between the system performance and implementation complexity, the fixed-point simulation of the whole baseband receiver is performed to decide the optimal word lengths of data paths in each function block. Fixed-point simulation is also to imitate the actual hardware behaviors. Then, based on the architecture and behaviors defined by fixed-point simulation, the proposed receiver is transformed to Verilog register-transfer-level (RTL) codes. Also, the hard macros such as SRAM modules are included to verify the hardware functional simulation. The verification of the hardware function is based on the test patterns generated by the fixed-point system program. Afterward, the gate-level design is generated by logic synthesis with applied constraints and verified again to guarantee its correct functionality. The pre-layout power consumption is estimated using the gate-level simulation. When the targets of the functionality and the system specification are achieved, the physical design is implemented from the gate-level design by an automatic placement and routing (APR) tool. The integrated circuit layout generated by the APR tool is verified by design-rule-check (DRC) and layout-versus-schematic (LVS) to ensure the layout conforming to the rules required for faultless fabrication and guarantee it really representing the circuit that you desire to fabricate. Subsequently, the post-layout gate-level net-list is also simulated to confirm its functionality, and the post-layout power consumption can be estimated. Finally, the layout file of the chip is taped out for fabrication.

This chapter is organized as follows. Section 6.2 introduces the word length optimization. Section 6.3 describes the synchronizer implementation. Section 6.4 presents the proposed channel estimator. Section 6.5 describes the design of the FFT/IFFT module. Then, the simulation results are given in Section 6.6. The design results are provided in Section 6.7. Finally, Section 6.8 draws the conclusions of this chapter.

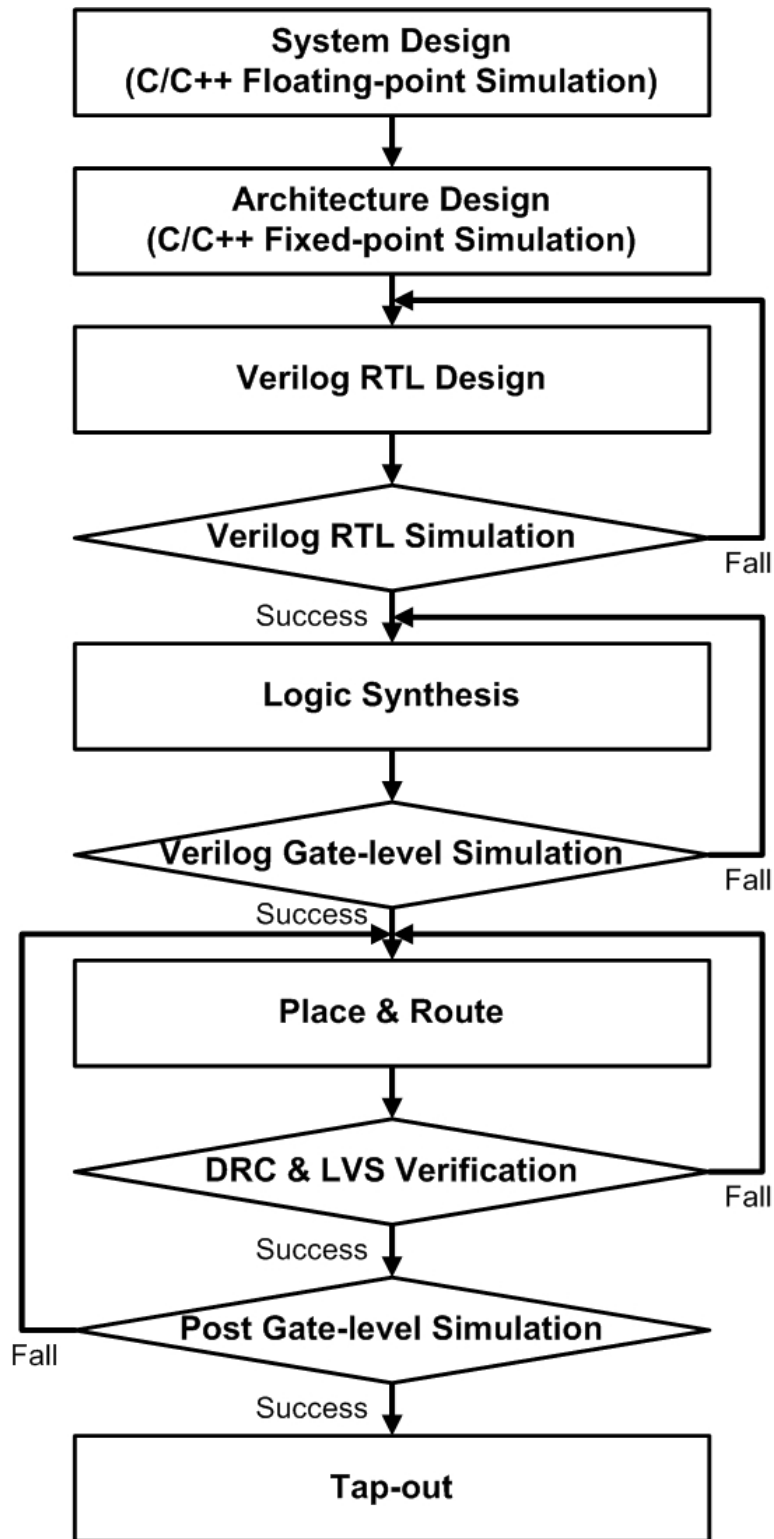


Fig. 6.2 Implementation flow of the proposed receiver.

6.2 Word Length Optimization

For hardware implementation, the finite word length has a strong effect on the system performance since it dominates the signal precisions. Although increasing word length can have a better performance, the design will have higher hardware complexity, consume more power, and have lower system operation frequency. Therefore, the signal word lengths in the proposed receiver design must be optimized. When determining the word length of each building block, two objectives are to be considered: (1) reduce the hardware cost by using the minimum word length and (2) maintain the acceptable system performance. Thus, the output SNR at the STBC decoder, as defined in (4.21)-(4.24), is used as a performance criterion to determine the appropriate word length of each building block. Fig. 6.3 shows the curves of the output SNR versus different word lengths for the received sample input. These curves are simulated at the vehicle speed of 120 km/hr with different E_b/N_0 . The word length is decided to be 10 where the curves of the output SNR get into saturation. Fig. 6.4 shows the curves of the output SNR versus the different word lengths for the channel response output, and the word length is decided to be 16 where the curves of the output SNR get into saturation. The performance loss due to the word length quantization is smaller than 0.5 dB when BER is at 10^{-4} . The word lengths of several key signals in the proposed receiver are summarized in Table 6.1.

TABLE 6.1
WORD LENGTHS OF SEVERAL KEY SIGNALS IN THE PROPOSED RECEIVER

Signal	Word Length	Signal	Word Length
Received Sample Input	10	Preamble Match Output	12
CORDIC Input	14	LS Estimator Output	14
CORDIC Output	14	STBC Decoder Output	9
FFT Output	13	IFFT/FFT Output	16
FFT Twiddle Factor	18	MPIC Decorrelator Output	14
Channel Response Output	16	Path Decorrelator Output	14

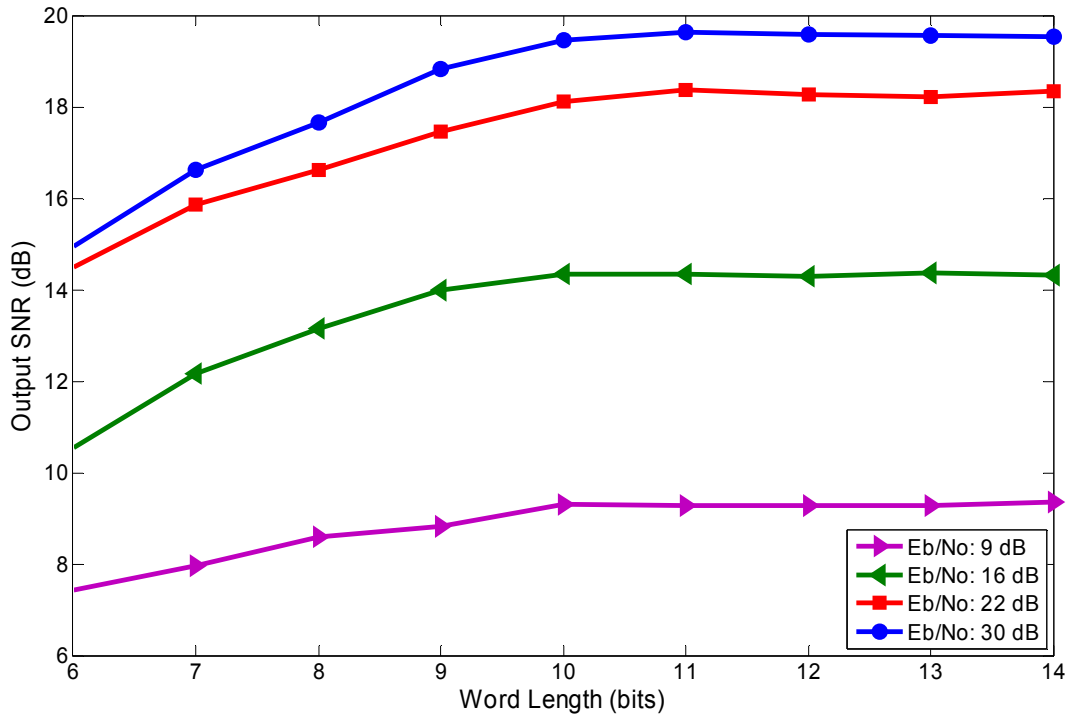


Fig. 6.3 Output SNR versus different word lengths for the received sample input.

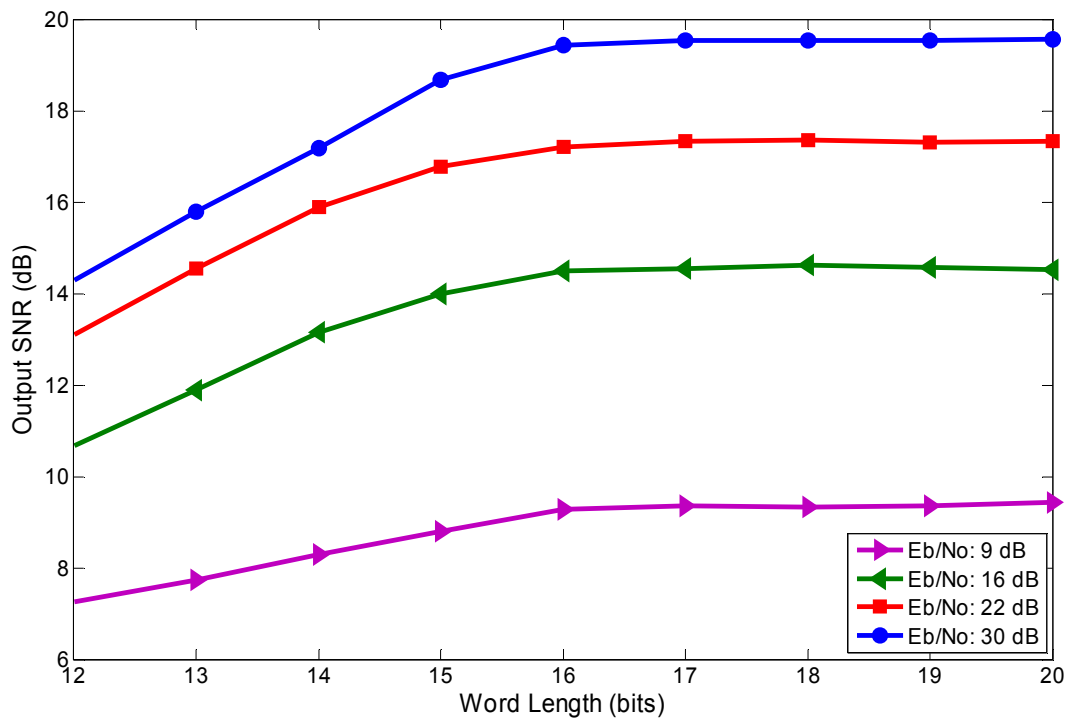


Fig. 6.4 Output SNR versus different word lengths for the channel response output.

6.3 Synchronizer

The proposed STBC-OFDM system is defined to support the frequency offset to ± 7 ppm variation in transmitter and receiver. The maximum CFO is equivalent to ± 35 KHz in 2.5 GHz carrier frequency. Therefore, there are seven possible ICFO values ranging from +3 to -3.

The modified match filter method evaluates the correlation between the received sample sequence and the known pre-compensated preamble sequence to find out the accurate symbol boundary and the corresponding ICFO value. However, there are two important problems that should be solved for hardware implementation. First, the hardware complexity is proportional to the length of correlation. Second, the match filter uses a large amount of multipliers to calculate the correlation. These problems result in the hardware being too huge and consuming a lot of power. As mentioned in Section 3.4.1, to solve the first problem, a suitable length is used to take both performance and hardware complexity into consideration. For the second problem, the coefficients of the match filter, which are the known preamble sequence, are quantized into $\{-1, 1\}$ values in both real and imaginary parts [47]. This improvement simplifies the multiplication as simple addition and subtraction and substantially replaces the multipliers into adders in the match filter. Furthermore, the received sample sequence can also be quantized into $\{-1, 0, 1\}$ and represented by two bits in the hardware implementation. By this scheme, the applied architecture only considers whether the received sample sequence and the coefficients are in phase or not; besides, it significantly reduces the requirement of registers and adders. Thus, even increasing the correlation length to compensate for the quantization loss does not pay a lot of cost. The detail descriptions are given in the following subsections.

6.3.1 Symbol Boundary Detector

Each tap of the match filter includes the complex multiplication of the coefficient and the received sample. Since both the real and imaginary part of coefficients are known and quantized to -1 or 1, each tap can be simplified to add or subtract the value of the received sample. Based on two's complement representation, the XOR gate is used to inverse the received sample for subtraction, but a lot of adders will be needed

to compensate 1 for each subtraction. Since the coefficients are known in advance, the number of subtractions is also known; hence, to compensate for each tap becomes awkward. All of the compensations can be summed and compensated in the last step for saving the hardware. The summation of the tap results and the final compensation is implemented by the CSA tree. Then the correlation result is compared with the threshold to find the maximum value. After the operation of the match filter is done with sufficient length which is decided to 300 by simulation, the symbol boundary index is found.

Modified Calculation Process

Although above architecture avoids the multiplier usage and reduces the register number, several RTL methods can be used to further reduce the hardware implementation. Due to the subtraction, the signed calculation becomes necessary; thus, more hardware effort is required for sign extension. Therefore, if the signed calculation can be avoided, the hardware and power will be saved.

In the proposed method, both real and imaginary part of the received sample sequence are quantized into $\{-1, 0, 1\}$, and the correlation length is fixed. When the numbers of 0's and 1's components stored in each tap register are known, the number of -1's components can also be obtained at the same time. Therefore, if the total number of 0's components can be ignored, the correlation only needs to sum the 1's components in each tap. One example is shown in below:

$$\begin{aligned}
 &\text{The correlation length: } 300 \\
 &\quad \text{Number of 0's: } 150 \\
 &\quad \text{Number of 1's: } 86 \\
 &\quad \text{-----} \\
 &\text{Result} = 86 + (300 - 150 - 86) * (-1) \\
 &\quad = 86 * 2 - 300 + 150
 \end{aligned}$$

Initially, because all of the registers are reset to 0, the initial value is equal to 300.

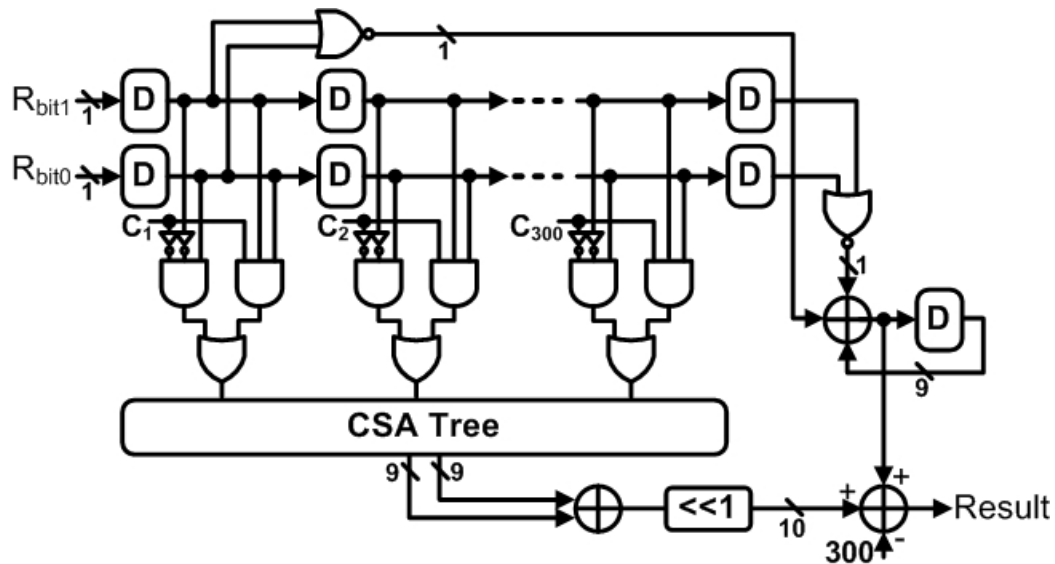


Fig. 6.5 Match filter design.

Next, a monitor circuit detects the number of zero in the register after each data updating. On the other hand, a logical circuit judges whether the result of correlation is 1. The logical circuit is designed by the truth table, and only NAND gate and NOR gates are required. The block diagram is shown in Fig. 6.5.

6.3.2 Carrier Frequency Offset Estimator

The CFO synchronization is partitioned into the ICFO and FCFO estimations. As mentioned in Section 3.4.1, in order to improve the accuracy of the proposed symbol boundary detection, the coefficient sequence in the match filter is pre-shifted with the possible values of ICFO; hence, the coarse ICFO can be detected simultaneously in the symbol boundary detection. The ICFO estimator can be implemented by the same match filter circuit of the symbol boundary detector. In order to accumulate the correlation results twice for ICFO, an additional circuit is necessary for storing the previous matching result. This operation can be realized by the combination of registers and multiplexers as shown in Fig. 6.6. A control signal is generated to decide the operation mode of this circuit. If the control signal is positive, the seven registers will respectively store the current correlation results of the seven possible ICFO values like memory. On the contrary, if the control signal is negative, the chain of the

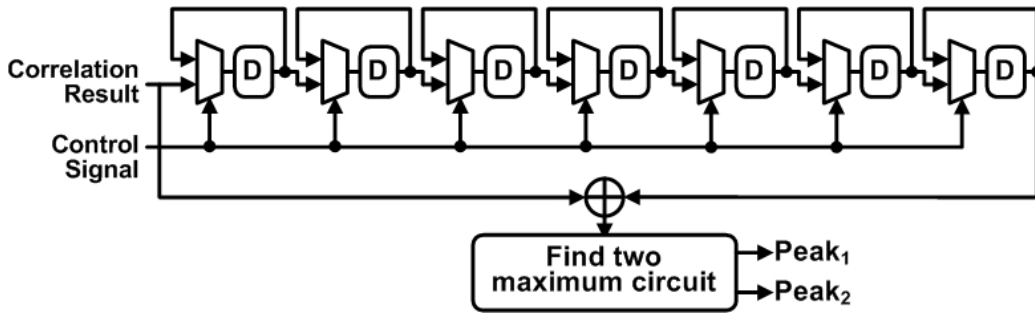


Fig. 6.6 Storage units for storing previous correlation results.

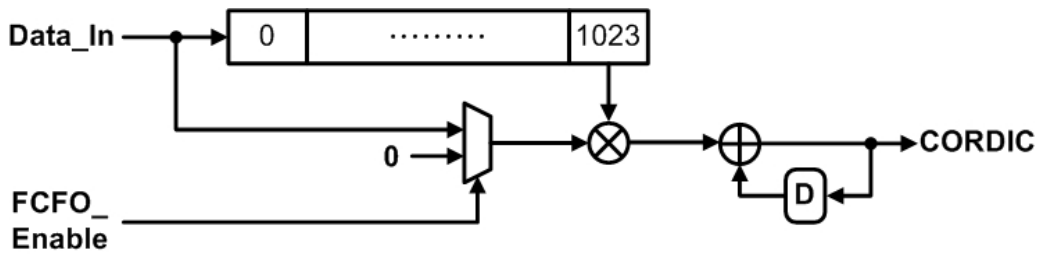


Fig. 6.7 Cross correlation circuit for the FCFO estimation.

registers works as the shift-registers. The control signal is kept positively until that the stored correlation results are needed to accumulate with the second correlation results. Finally, the accumulated results are passed to the comparator to select and save the two maximum possible ICFO values.

The repeating characteristic of guard interval can be used to estimate FCFO as expressed in (3.14) using the cross correlation algorithm. Fig. 6.7 shows the cross correlation circuit for the FCFO estimation, and the FCFO estimate is derived from the phase of this complex correlation output. In FCFO estimation, a 1024 delay-line is necessary for storing received data to correlate with. If the shift registers are used to realize the delay-line, it will cost a lot of area and power consumptions. The usage of register file or SRAM is a better choice than using shift register. A twister memory operation presented in [72] can be used to reduce the cost of memory area. The area of two single port register file modules for the size of 512 words \times 10 bits, which is 0.038 mm², reduces 77% and 28 % area than that of shift registers and one dual port SRAM

module for the size of 1024 words×10 bits, which are 0.166 mm² and 0.053 mm², respectively. In power consumption, the two single port register file modules save about 83% and 26% than the other two schemes. Therefore, two single port register modules for the size of 512 words×10 bits are adopted to realize the delay-line by using twister memory access operation. Besides, the complex multiplication can be realized from four multiplications and two additions/subtractions to three multiplications and five additions/subtractions:

$$\begin{aligned} (A + jB) \times (C - jD) &= (AC + BD) + j(BC - AD) \\ &= (AC + BD) + j[(A + B) \times (C - D) - AC + BD] \end{aligned} \quad (6.1)$$

For deriving the phase of the complex correlation output, an arctangent function can be implemented by the coordinate rotation digital computer (CORDIC) hardware. Moreover, the CORDIC circuit can also be used to provide the values of sine and cosine for de-rotating the CFO effect. The vector rotation is derived from (6.2), which a input vector $\{X_{in}, Y_{in}\}$ is rotated through an angle θ .

$$\begin{aligned} \begin{bmatrix} X_{out} \\ Y_{out} \end{bmatrix} &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X_{in} \\ Y_{in} \end{bmatrix} \\ &= \cos \theta \begin{bmatrix} 1 & -\tan \theta \\ \tan \theta & 1 \end{bmatrix} \begin{bmatrix} X_{in} \\ Y_{in} \end{bmatrix} \end{aligned} \quad (6.2)$$

The basic concept of the CORDIC computation is to decompose the desired rotation angle into the weighted sum of a set of predefined elementary rotation angles such that the rotation through each of them can be accomplished with simple shift-and-add operations. The rotation angle is represented as follows.

$$\theta = \sum_{n=0}^{N_n-1} d_n \cdot \tan^{-1}(2^{-i}) + \xi \quad (6.3)$$

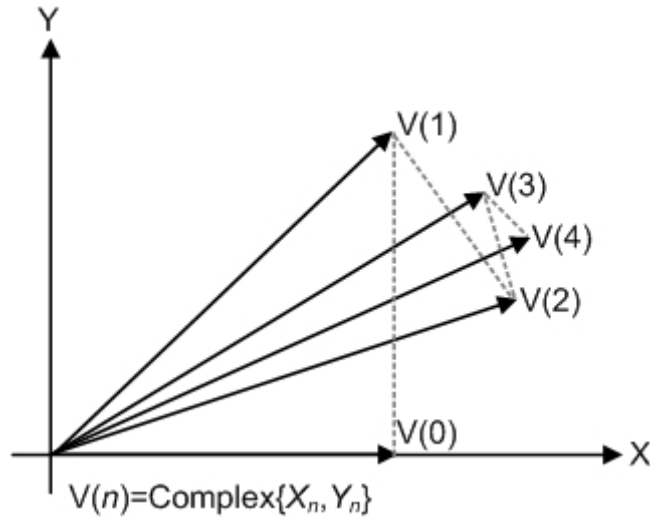


Fig. 6.8 CORDIC rotation.

where ζ is the residual angle beyond the resolution of CORDIC computation, and N_n is the iterations number. The rotation is completed by several iterations such as a windscreen wiper to rotate to the wanted values as shown in Fig. 6.8. The operation at each iteration is expressed as

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = K_n \begin{bmatrix} 1 & -d_n 2^{-n} \\ d_n 2^{-n} & 1 \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} \quad (6.4)$$

where d_n is the direction of rotation and equal to +1 or -1, and $K_n = \cos(\tan^{-1}(2^{-i}))$ is a scale constant. Removing the scale constant from the iterative equations yields a shift-add algorithm for vector rotation. The product of the K_n can be applied elsewhere in the system or treated as part of a system processing gain as given by

$$K = \prod_{n=0}^{N-1} K_n = \prod_{n=0}^{N-1} (\sqrt{1+2^{-i}}). \quad (6.5)$$

(6.4) can be represented as

$$\begin{aligned}
x_{n+1} &= x_n - y_n \cdot d_n \cdot 2^{-n} \\
y_{n+1} &= y_n + x_n \cdot d_n \cdot 2^{-n} \\
z_{n+1} &= z_n - d_n \tan^{-1}(2^{-n})
\end{aligned} \tag{6.6}$$

where z_{n+1} is the residual angle in the n -th iteration.

Two operation modes of the CORDIC algorithm, the rotation mode and the vectoring mode, are introduced as follows. The main difference of the two operations is determined by the direction of rotation. The rotation mode rotates the input vector by a desired rotation angle (z_0), and the rotation decision at each iteration is made to diminish the residual angle z_{n+1} in the angle accumulator. The decision at each iteration is therefore based on the sign of the residual angle after each step.

$$\text{rotation mode} \quad d_n = \begin{cases} -1, & \text{if } z_n < 0 \\ +1, & \text{otherwise} \end{cases} \tag{6.7}$$

On the contrary, the vectoring mode records the angle by rotating the input vector to x-axis, and the rotation decision at each iteration is to minimize the y component of the residual vector at each rotation. The sign of the residual y component is used to determine which direction to rotate next.

$$\text{vectoring mode} \quad d_n = \begin{cases} +1, & \text{if } y_n < 0 \\ -1, & \text{otherwise} \end{cases} \tag{6.8}$$

Fig. 6.9 shows the unrolled structure of CORDIC circuit which unrolls the iterative structure to N_n operating units and is adopted in this synchronizer design. The angle value in each stage is a constant and can be hardwired without using ROM to store. This structure can be pipelined to achieve the higher data throughput rate.

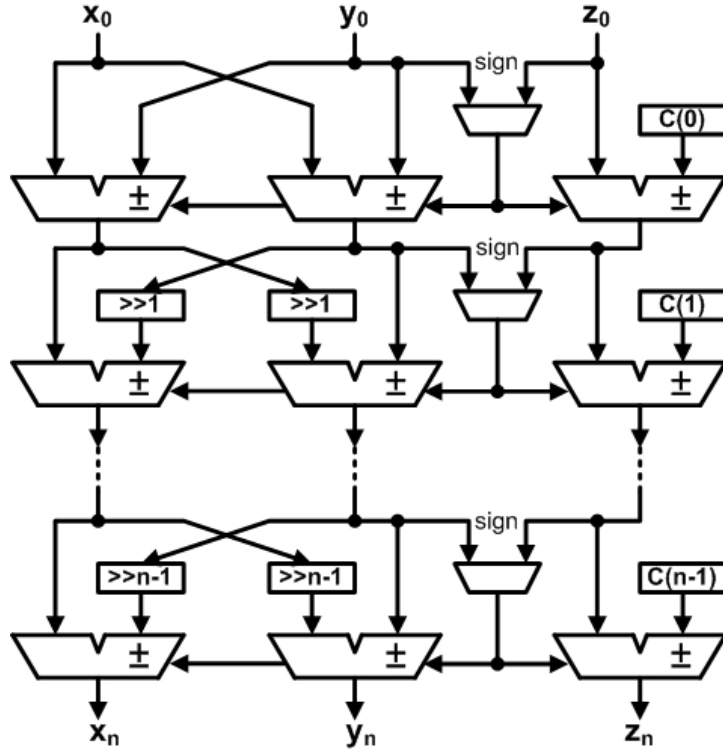


Fig. 6.9 CORDIC design.

6.4 Proposed Channel Estimator

The proposed channel estimator (Fig. 6.1 (b)) consists of the initialization stage and the tracking stage. The initialization stage is composed of a preamble match, a SMPIC-based decorrelator and an FFT/IFFT module. The tracking stage is composed of an STBC decoder, a demapper, an LS estimator, a path decorrelator, a Hessian matrix calculator and an FFT/IFFT module. Moreover, the FFT/IFFT module can be shared between the initialization stage and the tracking stage. These key block designs are described in the following subsections.

6.4.1 Initialization Stage: Preamble Match

From (4.19), the preamble match is used to estimate the preliminary channel frequency responses $M^{(i)}[k]$ for $i=1,2$ by matching the received signal with the

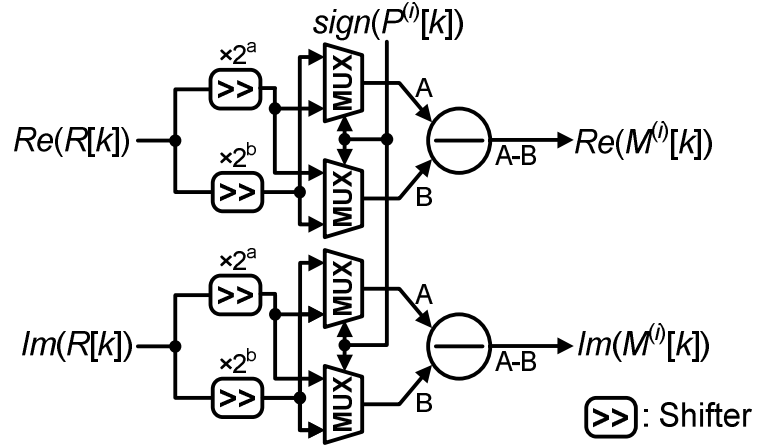


Fig. 6.10 Basic design unit of the preamble match.

preamble symbol $P^{(i)}[k]$ transmitted from the i -th antenna. The values of preamble subcarriers are known patterns; besides, the subcarriers are modulated by BPSK and boosted as a constant power to increase the reliability. Thus, the absolute values of preamble subcarriers normalized by the power can be pre-computed to a real constant α . Furthermore, the constant value α can be quantized to a canonic sign digit (CSD) code [73] with two nonzero digits (2^a-2^b) for the purpose of using only shifters and adders instead of multiplier implementation to reduce the design complexity. The $M^{(i)}[k]$ can be reformulated as

$$\begin{aligned}
 M^{(i)}[k] &= \text{sign}(P^{(i)}[k]) \cdot R[k] \cdot \frac{|P^{(i)}[k]|}{|P^{(i)}[k]|^2} \\
 &= \text{sign}(P^{(i)}[k]) \cdot R[k] \cdot \alpha
 \end{aligned} \tag{6.9}$$

The preamble match design consists of subtractors and multiplexers controlled by the sign of $P^{(i)}[k]$ for forming the operations of (6.9) as shown in Fig. 6.10. The known preamble patterns are only required to store the sign of $P^{(i)}[k]$ which can be represented by one bit data.

6.4.2 Initialization Stage: SMPIC-based Decorrelator

The proposed SMPIC-based decorrelator is used to identify N_P significant paths in a straightforward method. The proposed method first sorts the N_B paths (within an observation window) to find the first N_{TP} paths with large $m^{(i)}[\tau]$, where N_{TP} is defined to be $\beta \cdot N_P$, and β is an integer. Then, the decorrelation is carried out to cancel the path interference as expressed in (4.20). Finally, the decorrelated N_{TP} paths are sorted again to pick up the first N_P significant paths.

For this channel estimator, N_B is defined to be 128 which is the same of the CP length, and N_P is presumed to be eight. According to the output SNR evaluation in (4.21)-(4.24), Fig. 6.11 shows the curves of the output SNR in QPSK modulation versus the value of β . These curves are simulated at the vehicle speed of 120 km/hr with different E_b/N_0 . The value of β is decided to be four where the curves of the output SNR get into saturation. Hence, the value of N_{TP} is 32. As compared with the original MPIC-based decorrelation method, the performance loss due to the

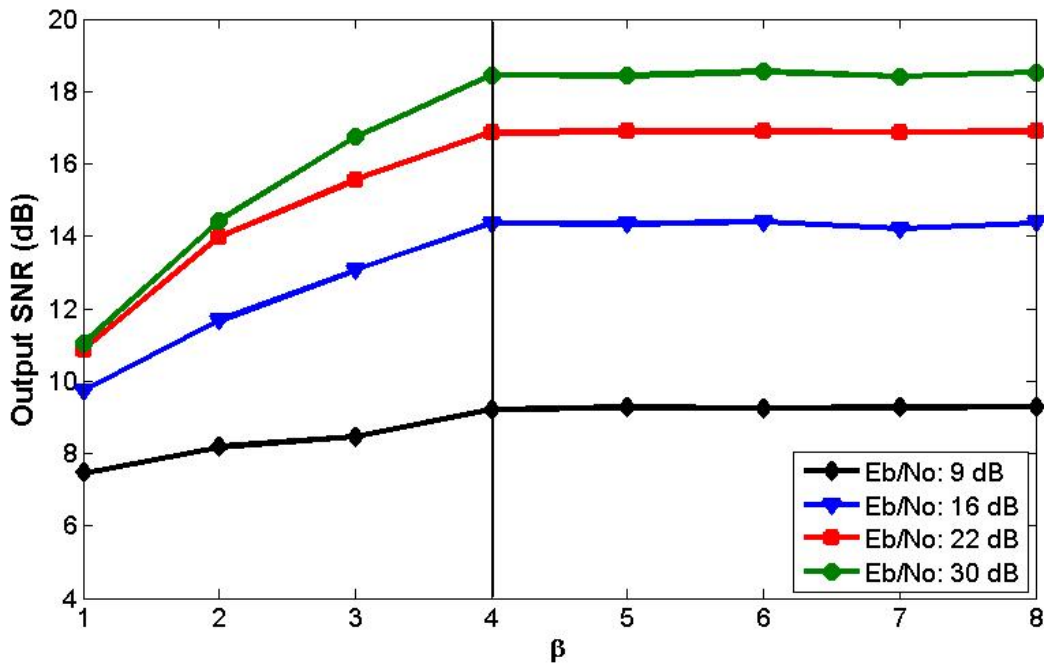


Fig. 6.11 Output SNR versus the value of β .

quantization of β is smaller than 0.5 dB when BER is at 10^{-4} .

The architecture of the SMPIC-based decorrelator requires a very efficient partial sorting network and a decorrelator. We propose a merge sorting network with programmable and partial sorting capability (MSNP²) and a triangular decorrelator (TD). The details of these two block designs are described below.

6.4.2.1 Merge Sorting Network with Programmable and Partial Sorting Capability (MSNP²)

First, the design of the proposed merge sorting network, MSNP², is discussed. In order to avoid the high complexity of parallel sorting network, a fixed I/O size sorting network and a set of memory module are used to accommodate the number of sorting elements [74]-[76]. Here, the architecture of the MSNP² with a memory bank, a sorting control unit and an 8-item sorter is shown in Fig. 6.12. The 8-item sorter is the Batcher's sorting network with I/O size of eight. The Batcher's sorting network is widely used because of its inherent parallelism and short latency [77]. Fig. 6.13 shows the 8-item sorter, and the basic unit is a 2x2 comparator which is used to perform data comparison and exchange. The memory bank which is primarily used to save the path power values is organized into eight independent memory modules denoted as $MD_1 \sim MD_8$.

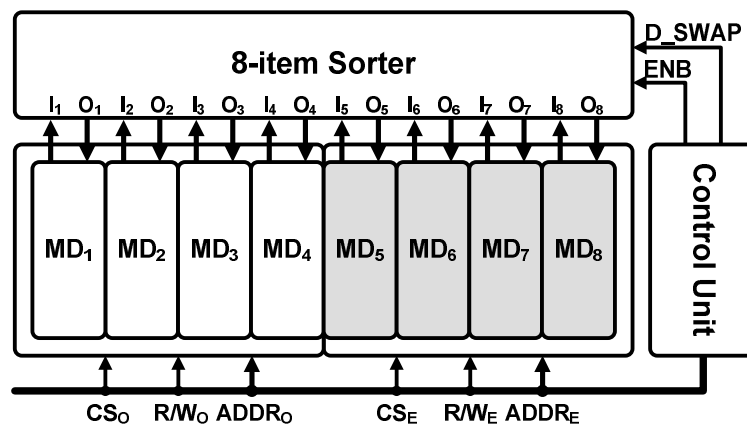


Fig. 6.12 Block diagram of the MSNP².

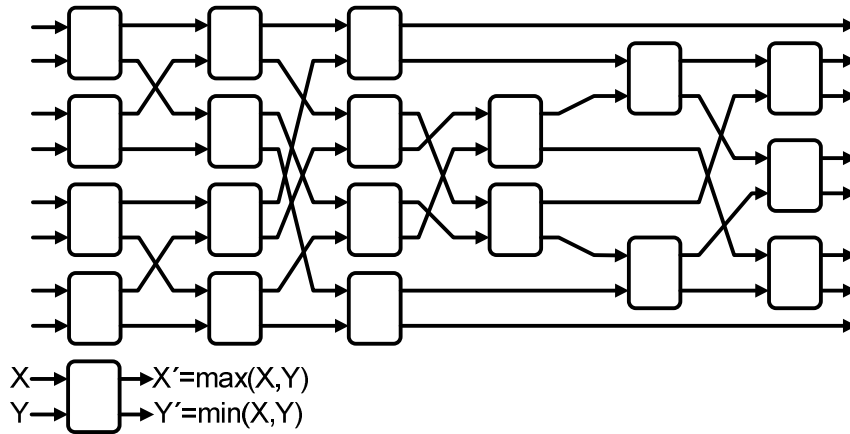


Fig. 6.13 Batcher classic sorting network with I/O size of eight.

	MD ₁	MD ₂	MD ₃	MD ₄	MD ₅	MD ₆	MD ₇	MD ₈
	000	001	010	011	100	101	110	111
0000		R ₁				R ₂		
0001		R ₃				R ₄		
0010		R ₅				R ₆		
0011		R ₇				R ₈		
0100		R ₉				R ₁₀		
0101		R ₁₁				R ₁₂		
0110		R ₁₃				R ₁₄		
0111		R ₁₅				R ₁₆		
1000		R ₁₇				R ₁₈		
1001		R ₁₉				R ₂₀		
1010		R ₂₁				R ₂₂		
1011		R ₂₃				R ₂₄		
1100		R ₂₅				R ₂₆		
1101		R ₂₇				R ₂₈		
1110		R ₂₉				R ₃₀		
1111		R ₃₁				R ₃₂		

Fig. 6.14 Memory bank arrangement of merge sorting network.

Since the maximum sorting item is 128, the sorting data (path power values) are arranged with 32 rows which the row definition is used in the sorting sequence, and each row contains four sorting data. The odd rows are loaded into $MD_1 \sim MD_4$, and the even rows are loaded into $MD_5 \sim MD_8$ as illustrated in Fig. 6.14. Based on the sorting sequence, the sorting control unit takes two rows of data to the 8-item sorter for sorting in each cycle; then, the outputs of the sorter which are divided into two

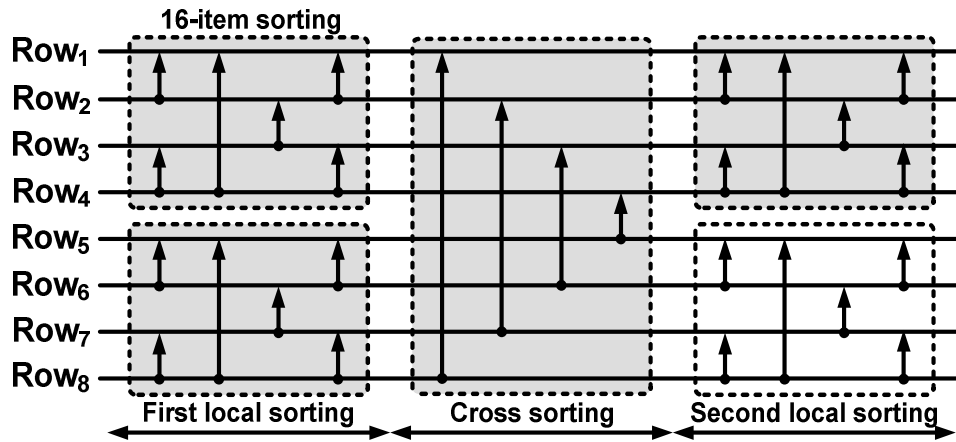


Fig. 6.15 32-item merge sorting sequence.

clusters in descending order are written back to the memory bank and replace the original two rows. The L -item merge-sorting sequence can be divided into three cycles: (1) the first local sorting cycle, (2) the cross sorting cycle, and (3) the second local sorting cycle. At the first local sorting cycle, L -item data are divided into two $L/2$ -item data clusters to do $L/2$ -item merge-sorting, respectively. Then, L -item data will be arranged in two $L/2$ -item clusters in descending order. At the cross sorting cycle, the data in the up cluster are compared and exchanged with the data in the down cluster. After cross sorting, the data in the up cluster are larger than that in the down cluster. At the second local sorting, the two clusters are sorted separately again in descending order. Finally, the sorted results are saved in the memory bank and arranged in the row order.

Fig. 6.15 shows the 32-item merge sorting sequence represented by the directed arrows in the line representation, and each arrow represents an operation of the 8-item sorter. According to a 32-item merge sorting sequence, where (x, y) represents the row x and row y sorted by 8-item sorter, the first local sorting sequences of 32-item sorting are $(1, 2)$, $(3, 4)$, $(5, 6)$, $(7, 8)$, $(1, 4)$, $(5, 8)$, $(2, 3)$, $(6, 7)$, $(1, 2)$, $(3, 4)$, $(5, 6)$, and $(7, 8)$. The crossing sorting sequences are $(1, 8)$, $(2, 7)$, $(3, 6)$, and $(4, 5)$. The second local sorting sequences are the same as the first local sorting sequences.

The merge sorting is used two times in the SMPIC-based decorrelator. The first time is to sort the 128-item data to find the first 32-item data and denoted as

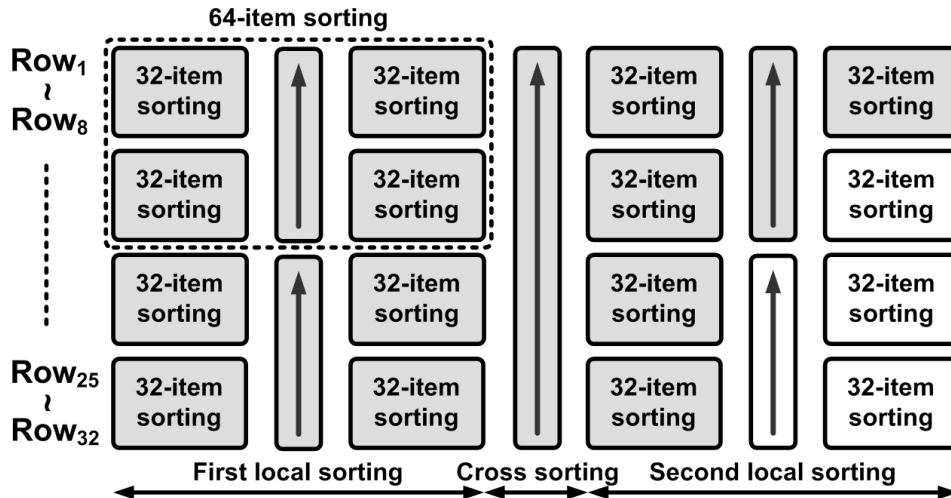


Fig. 6.16 128-item merge sorting sequence.

128-32-item sorting. The second time is to sort the 32-item data to find the first 8-item data and denoted as 32-8-item sorting. The 32-item sorting sequence is used to be a basic control sequence, and the 128-item sorting sequence can be extended by the 32-item sorting sequence and constructed as the line representation shown in Fig. 6.15. For saving execution time and power, the 128-32-item sorting only executes the grey part of Fig. 6.16, and the 32-8-item sorting executes the grey part of Fig. 6.15.

The state diagram of the proposed MSNP² is shown in Fig. 6.17. The MSNP² has the programmable and partial sorting capability to execute the 128-32-item sorting and the 32-8-item sorting. There are five state definitions as listed in Table 6.2. The basic 32-item sorting is composed of the execution of State_Set_0, State_Set_1 and State_Set_2. Moreover, the 128-32-item sorting can be performed by arranging above three states with State_Cross_0 and State_Cross_1. Several control signals are used to control the execution flow and decide the execution state in the different situations as introduced in Table 6.3.

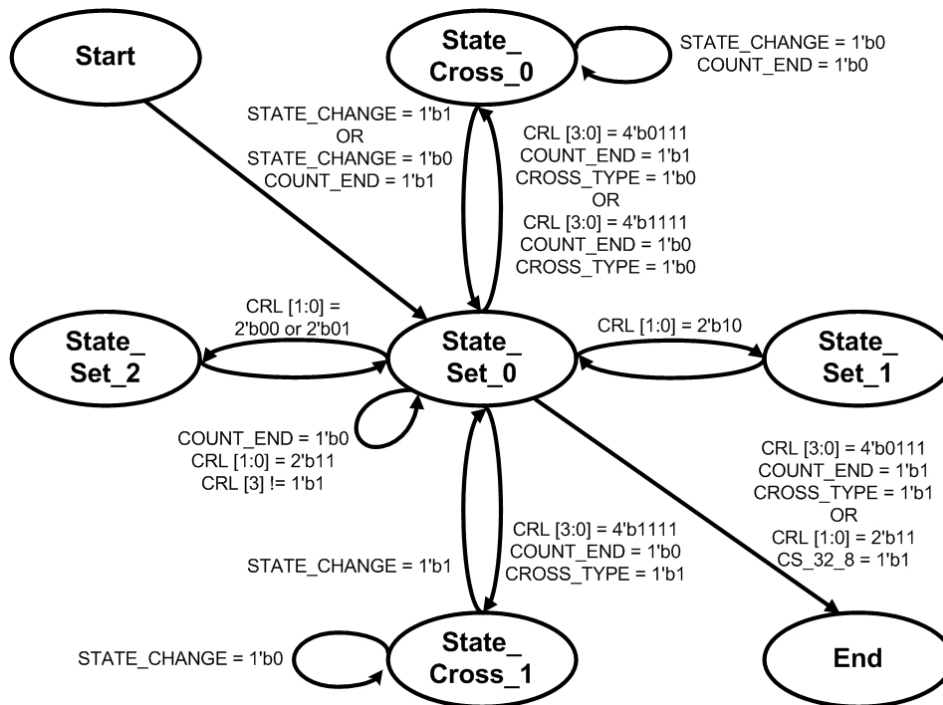


Fig. 6.17 State diagram of the proposed MSNP².

TABLE 6.2
STATE DEFINITIONS OF THE MSNP² STATE DIAGRAM

State Name	State Behavior
State_Set_0	For a basic 32-item sorting, State_Set_0 is used to execute the sorting sequence {(1, 2), (3, 4), (5, 6), (7, 8)}
State_Set_1	For a basic 32-item sorting, State_Set_1 is used to execute the sorting sequence {(1, 4), (5, 8), (2, 3), (6, 7)}
State_Set_2	For a basic 32-item sorting, State_Set_2 is used to execute the sorting sequence {(1, 8), (2, 7), (3, 6), (4, 5)}
State_Cross_0	For extending to 128-item sorting, State_Cross_0 is used to execute the sorting sequence {(1,16) (2,15) (3,14) (4,13) (5,12) (6,11) (7,10) (8,9)}
State_Cross_1	For extending to 128-item sorting, State_Cross_1 is used to execute the sorting sequence {(1 ,32) (2 ,31) (3 ,30) (4 ,29) (5 ,28) (6 ,27) (7 ,26) (8 ,25)} and {(9 ,24) (10,23) (11,22) (12,21) (13,20) (14,19) (15,18) (16,17)}

TABLE 6.3
CONTROL SIGNALS USED IN THE MSNP² STATE DIAGRAM

Control Signals	Description
CRL [3:0]	This 4-bit signal is used to count the execution times of State_Set_0.
CS_32_8	This one-bit signal is used to note that the execution should be the 128-32-item sorting (CS_32_8 = 1'b0) or the 32-8-item sorting (CS_32_8 = 1'b1).
CROSS_TYPE	This one-bit signal is used to switch the cross sorting to State_Cross_0 (CROSS_TYPE = 1'b0) or State_Cross_1 (CROSS_TYPE = 1'b1).
STATE_CHANGE	This one-bit signal is changed when the 16-row data are completed in the cross sorting (State_Cross_0 or State_Cross_1).
COUNT_END	This one-bit signal is used to sign that the second local sorting of the 128-32-item sorting is partial executed in the up cluster.

6.4.2.2 Triangular Decorrelator (TD)

The TD consists of a decorrelated control unit, a decorrelated unit, and a memory bank shared with the sorting process. The TD is executed after the first 128-32-item sorting. There are 31 iterations of the TD process, and the process starts at the first legal path which is the maximum sorted path. If κ denotes the iteration number, for $0 \leq \kappa < 31$, the process of the κ -th iteration is to cancel the interferences associated with the κ -th legal path gain. The process can be expressed as

$$\mu_\rho = \tilde{\mu}_\rho - \mu_\kappa C_{\rho\rho} [\tau_\rho - \tau_\kappa], \quad \kappa < \rho \leq 31 \quad (6.10)$$

where μ_κ is the κ -th legal path gain, $\tilde{\mu}_\rho$ is the ρ -th sorted path gain, μ_ρ is the ρ -th decorrelated sorted path gain, τ_κ is the κ -th legal path delay, τ_ρ is the ρ -th

sorted path delay, and $C_{PP}[\tau]$ is the normalized cyclic auto-correlation of the preamble. The process of the κ -th iteration executes $(31-\kappa)$ times of decorrelating calculation. At the κ -th iteration, the process does not decorrelate the interference to the first $(\kappa-1)$ legal paths since the interference value is much smaller than their path gains and does not influence the accuracy of the decision in the significant path positions. Moreover, the small path gain offset can be revised in the tracking process without loss of the performance. In this way, the TD process can effectively save about half of execution cycles and power consumption. After the κ -th iteration processing, the $(\kappa+1)$ -th legal path is acquired to execute the next iteration. Because the preamble is a known pattern and modulated by BPSK, the value of $C_{PP}[\tau]$ with different τ can be calculated and stored in ROM as constant value η_τ in advance. The iteration process can be further expressed as

$$\mu_\rho = \begin{cases} \text{Re}\{\mu_\rho\} = \text{Re}\{\tilde{\mu}_\rho\} - \text{Re}\{\mu_\kappa\} \cdot \eta_{|\tau_\rho - \tau_\kappa|} \\ \text{Im}\{\mu_\rho\} = \text{Im}\{\tilde{\mu}_\rho\} - \text{Im}\{\mu_\kappa\} \cdot \eta_{|\tau_\rho - \tau_\kappa|} \end{cases}, \quad \kappa < \rho \leq 31. \quad (6.11)$$

After 31 iterations have been completed, the 32 legal paths are obtained and then sorted again to find eight significant paths. Therefore, following the process of the 32-item decorrelation, the decorrelated control unit is designed to control the access flow of the sorted paths and the decorrelated paths. Fig. 6.18 shows the design of the

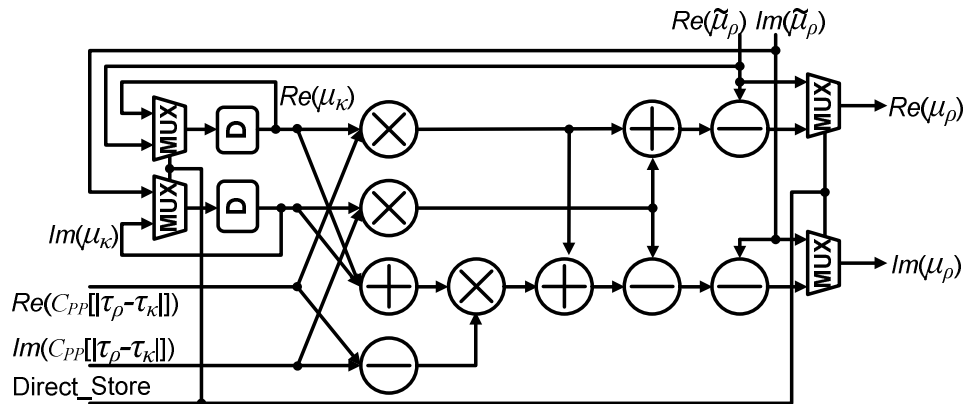


Fig. 6.18 Design of the decorrelated unit.

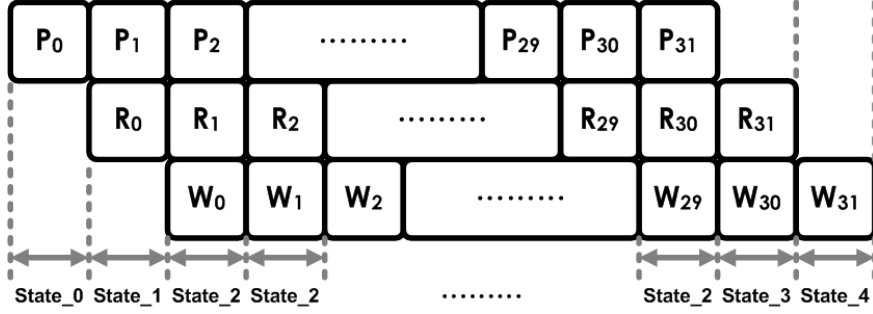


Fig. 6.19 Process steps of the TD operation.

decorrelated unit. Within the execution of the κ -th iteration, μ_κ must be used $(31-\kappa)$ times; hence, μ_κ is read from memory at the beginning and saved in the local registers to reduce memory access until the iteration is finished. As shown in Fig. 6.19, the TD operation can be decomposed to three kinds of the process steps: 1) the process of P is used to read the path delay values of τ_κ and τ_ρ and calculate the access addresses of $C_{PP}[|\tau_\rho - \tau_\kappa|]$, μ_κ and $\tilde{\mu}_\rho$; 2) the process of R is used to read the path gains of μ_κ or $\tilde{\mu}_\rho$ and the value of $C_{PP}[|\tau_\rho - \tau_\kappa|]$, and μ_κ will be directly stored in the local registers until the κ -th iteration is finished; 3) the process of W is used to write back the decorrelation results of μ_ρ . According to the process steps, the state diagram of the TD is shown in Fig. 6.20. Initial State_0~ State_4 read the path gains from the FFT/IFFT module and is only executed in the first iteration of the TD process; otherwise, State_0~ State_4 access the paths gains in the memory bank shared with the sorting process. The control signal COUNT_START is used to record the iteration κ . The control signal POS_COUNT is used to count the index ρ .

6.4.3 Tracking Stage: STBC Decoder and Demapper

Before the LS estimation calculation, the decided symbols $\hat{X}_F[k]$ and $\hat{X}_S[k]$ must be determined first. By using dividers, the decided symbols expressed in (4.27) and (4.28) can be implemented intuitively; however, the hardware design of a divider is very costly. The target of this receiver implementation is to support the modulations of BPSK, QPSK, and 16-QAM; therefore, a demapping dichotomy method with two

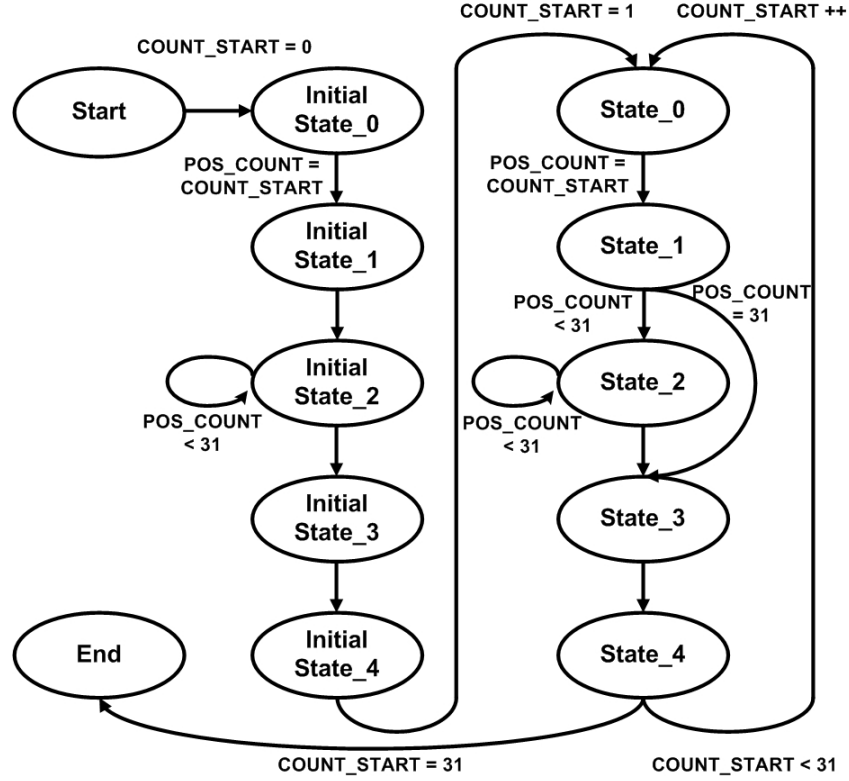


Fig. 6.20 State diagram of the proposed TD

stages [78] can be adopted to avoid the divider implementation. The division expressions of (4.27) and (4.28) are avoided and can be rewritten as

$$\begin{aligned} \zeta[k] \cdot \hat{X}_F[k] &= \tilde{X}_F[k] \\ &= (M_{v-1}^{(1)}[k])^* R[1, k] + M_{v-1}^{(2)}[k] (R[2, k])^* \end{aligned} \quad (6.12)$$

$$\begin{aligned} \zeta[k] \cdot \hat{X}_S[k] &= \tilde{X}_S[k] \\ &= (M_{v-1}^{(2)}[k])^* R[1, k] - M_{v-1}^{(1)}[k] (R[2, k])^* \end{aligned} \quad (6.13)$$

where

$$\zeta[k] = \left(|M_{v-1}^{(1)}[k]|^2 + |M_{v-1}^{(2)}[k]|^2 \right). \quad (6.14)$$

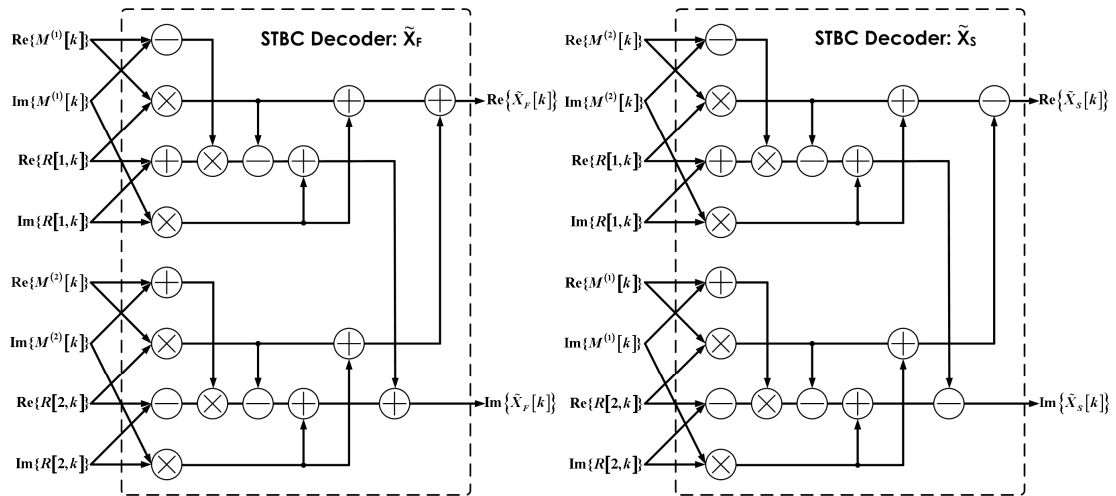


Fig. 6.21 Design of divider-free STBC decoder.

The combination of the STBC decoder and symbol demapper can be decomposed into a divider-free STBC decoder and a two-stage dichotomy demapper. Fig. 6.21 shows the design of the divider-free STBC decoder. The computation of $\tilde{X}_F[k]$ or $\tilde{X}_S[k]$ requires four complex multiplications and two complex additions/subtractions. A complex multiplication can be realized from four multiplications and two additions/subtractions to three multiplications and five additions/subtractions as expressed in (6.1). Hence, the design of the divider-free STBC decoder only requires 12 multipliers and 24 adders.

Fig. 6.22 shows the block diagram of the two-stage dichotomy demapper [78]. Since the constellation values are known, the divisions can be replaced by scaling the constellation map with the value $\zeta[k]$. However, for the 16-QAM constellation, using exhaustive method requires at least 15 comparators and 16 multipliers. Therefore, a demapping dichotomy method with two stages is joined to reduce the hardware cost. The $\tilde{X}_F[k]$ and $\tilde{X}_S[k]$ are partitioned into real part and imaginary part, and the demapping process is able to demap the real part and imaginary part separately. The first stage is based on the sign bit of the real part and imaginary part signals to determine the local quadrant, and the results of QPSK demodulation are obtained simultaneously. If the modulation mode is 16-QAM, the second stage is then used to determine the absolute value of the real part and imaginary part signals that are

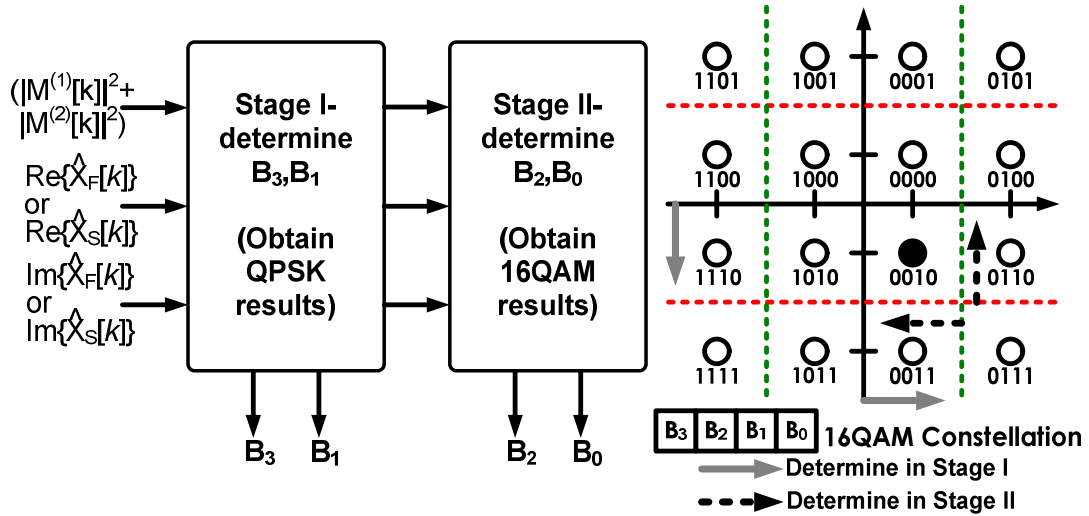


Fig. 6.22 Block diagram of the two-stage dichotomy demapper.

greater than or less than the decision boundary that has been scaled by $\zeta[k]$. After these two stages process, the signal locations in the 16-QAM constellation are decided, and the results of 16-QAM demodulation are obtained.

6.4.4 Tracking Stage: LS Estimator

As expressed in (4.29) and (4.30), $\{I_F, Q_F\}$ and $\{I_S, Q_S\}$ are used to denote the I and Q coordinate values of $\hat{X}_F[k]$ and $\hat{X}_S[k]$, respectively; besides, $\{I_{R1}, Q_{R1}\}$ and $\{I_{R2}, Q_{R2}\}$ are used to denote the real part and the imaginary part of $R[1,k]$ and $R[2,k]$, respectively. The value of λ has a limited set. Therefore, the possible multiplication values of λ can be pre-computed and stored in advance to reduce implementation cost.

Fig. 6.23 shows the design of the LS estimator which is composed of coordinate precalculators, LS units, an LS control unit and a final normalization. The coordinate precalculators are designed to generate the partial products of $R[t,k]$ multiplied by the coordinate values of $\{I_F, Q_F\}$ and $\{I_S, Q_S\}$. The coordinate precalculators support the modulations of BPSK, QPSK, and 16-QAM, and the multiples should be $\{1x, 3x\}$. Fig. 6.24 shows a coordinate precalculator implemented by carry propagate adder (CPA). An LS unit including multiplexers and adders is used to sum the corresponding partial products and generate the LS estimation results without

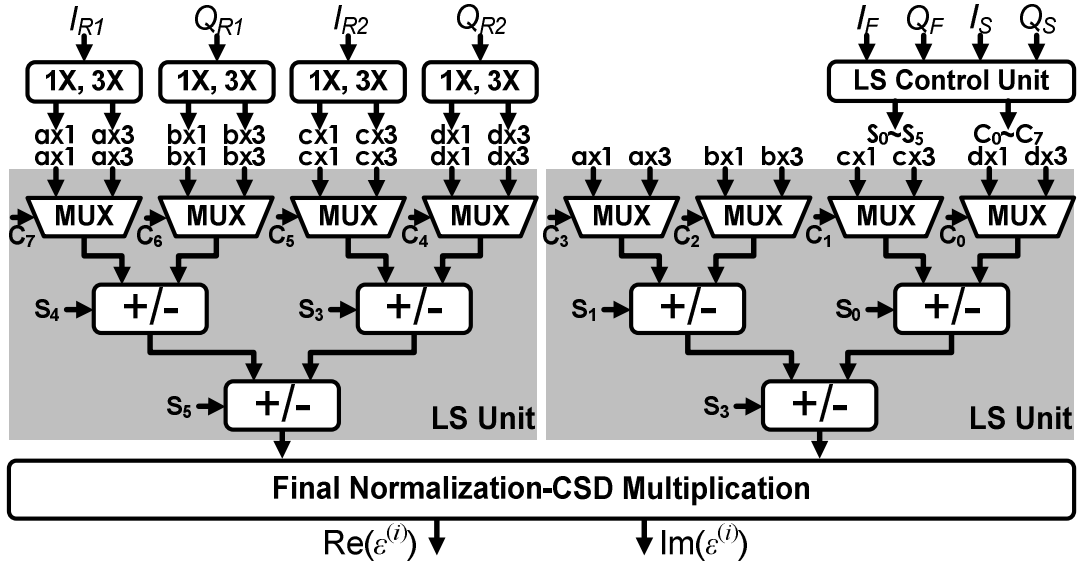


Fig. 6.23 Design of the LS estimator.

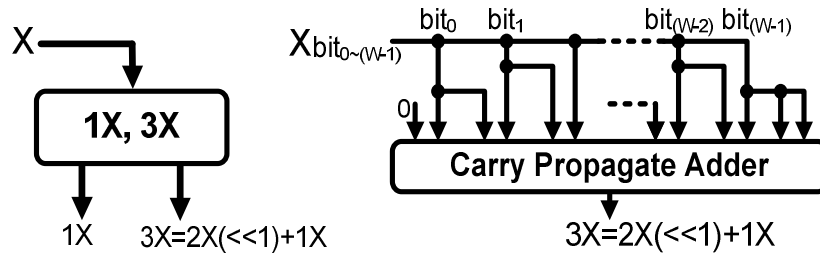


Fig. 6.24 Design of the coordinate precalculator.

normalization. The LS control unit is based on the values of I_F , Q_F , I_S , and Q_S to generate the control signals for selecting the outputs of the coordinate precalculators, controlling the adders to add or subtract and choosing the results multiplied by the corresponding value of λ . Since the value of λ has a limited constant set, all possible values can be applied by CSD coding and then searched their common subexpressions to implement CSD multiplications to avoid the usage of dividers. Finally, the result is outputted after the final normalization.

Moreover, the first iteration of the channel tracking depends on the pilots inserted in each OFDM symbol to provide a reliable global search direction. As expressed in (4.31) and (4.32), the LS estimations are performed by multiplying the corresponding $R[1,k]$ and $R[2,k]$ with a constant value σ . Therefore, the LS

estimations at the first iteration can be easily implemented by a constant CSD multiplication.

6.4.5 Tracking Stage: Hessian Matrix Calculator and Path Decorrelator

Before the path decorrelation, the inverse of the Hessian matrix, $[\mathbf{E}^{(i)}]^{-1}$, should be calculated first. As mentioned in Section 4.4.5, the computation of matrix inverse $[\mathbf{E}^{(i)}]^{-1}$, which requires about $O(N_p^3)$ complex multiplications, is successfully avoided by considering the strongly diagonal property as explained in (4.33)-(4.34). Nevertheless, we still require at least N_Θ cycles to calculate the cosine and sine summations by using a look-up table for forming each entry of $\mathbf{E}^{(i)}$, where N_Θ is the dimension of Θ . Following the expression in (4.35), since the maximum significant path delay of CIR is smaller than N_B , the value of $\theta_{l,l'}^{(i)}$ has a finite range. All possible values of $E(\theta_{l,l'}^{(i)})$ can be pre-calculated and stored in ROM to effectively avoid the on-line computations which requires at least $O(2N_p^2 \cdot N_\Theta)$ real additions and a lot of memory for saving the cosine and sine values. After the inverse of the Hessian matrix is acquired, the path decorrelation can be formulated as

$$[\mathbf{E}^{(i)}]^{-1} \mathbf{q}_v^{(i)} = \begin{bmatrix} 1/N_\Theta & \cdots & E(\theta_{0,N_p-1}^{(i)}) \\ \vdots & \ddots & \vdots \\ E(\theta_{N_p-1,0}^{(i)}) & \cdots & 1/N_\Theta \end{bmatrix} \begin{bmatrix} q_v^{(i)}[\tau_0] \\ \vdots \\ q_v^{(i)}[\tau_{N_p-1}] \end{bmatrix}. \quad (6.15)$$

The computational complexity of the path decorrelation requires $O(N_p^2)$ complex multiplications and $O(N_p^2)$ complex additions.

We further observe that the inter-path interference degrades sharply when $|\theta_{l,l'}^{(i)}|$ gradually becomes large, many values of $E(\theta_{l,l'}^{(i)})$ after the numerical quantization are very small and near to zero. Hence, all of these nonzero quantized $E(\theta_{l,l'}^{(i)})$ values can be expressed in CSD codes and searched for their common subexpressions;

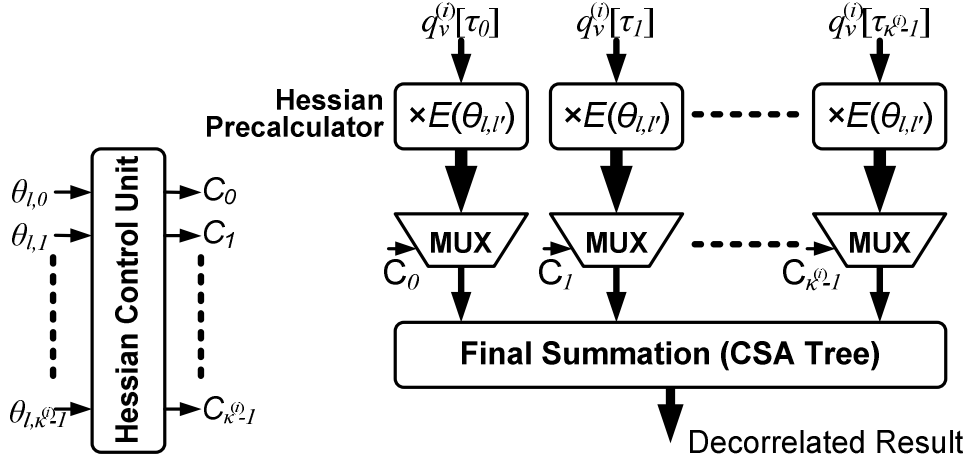


Fig. 6.25 Block diagram of the path decorrelator.

subsequently, the multiplications of $E(\theta_{l,l'})$ can be successfully implemented by CSD multiplications and free to use a large number of ROM to store all possible values of $E(\theta_{l,l'})$. Moreover, we merge the Hessian matrix calculator into the path decorrelator, and there are four components to compose this path decorrelator: Hessian precalculators, a Hessian control unit, selectors, and a final summation.

Fig. 6.25 shows the block diagram of this path decorrelator implemented in parallel form. The Hessian precalculator employs CSD multiplications to multiply the un-decorrelated $\mathbf{q}_v^{(i)}$ with the possible values of $E(\theta_{l,l'})$, and it is only executed once during the operation of the path decorrelator. Then, based on the value of $\theta_{l,l'}$, the Hessian control unit generates the control signals for selecting the corresponding results generated by the Hessian precalculators. Finally, the corresponding results are selected by the selectors and summarized by the final summation to form the result of a decorrelated $q_v^{(i)}[\tau_\kappa]$, $0 \leq \kappa \leq N_P-1$. The design just needs N_P cycles to complete all path decorrelations in parallel form; otherwise, it needs N_P^2 cycles in serial form. Based on the proposed architecture, the computational complexity of the path decorrelation is effectively reduced to at most $O(N_P \cdot (N_P + N_E))$ complex additions, where N_E denotes the nonzero digit number of all possible values of $E(\theta_{l,l'})$.

As described above, the proposed path decorrelator efficiently avoids computing

the Hessian matrix and the matrix inverse. It uses only adders and multiplexers instead of many complex multipliers and a lot of memory; besides, the redundant computations are also removed by sharing the results of the Hessian precalculators. Therefore, the proposed path decorrelator highly reduces the hardware complexity and leads to low-power application simultaneously.

6.5 FFT/IFFT module

The FFT is a key component in an OFDM system. It is used to transform the received time-domain OFDM symbols into frequency domain. In addition, the FFT and IFFT are also required by the proposed two-stage channel estimator, and the tracking stage tracks channel variations in an iterative way. In the proposed channel estimator design, the FFT and IFFT can be shared between the initialization stage and the tracking stage. Therefore, the implementation cost and latency of FFT and IFFT are two main issues to achieve the design requirement. A parallel memory-based FFT/IFFT architecture which provides multiple inputs and outputs in normal order is adopted to have a lower cost and reduce the latency requirement which is targeted to less than 1/4 of one OFDM symbol time. Fig. 6.26 illustrates the architecture of the proposed 1024-point parallel memory-based FFT/IFFT module [79]. This architecture consists of eight independent memory modules, four radix-8 processing elements (PEs), two radix-2 butterfly elements, and two commutators. The memory modules are implemented with two single-port SRAM modules, which consume less area and power than one two-port SRAM module. Two classes of PE architecture are popular in the literature: single-path delay feedback (SDF) and multi-path delay commutator (MDC) [80], [81]. Considering cost, complexity and throughput, the radix-8 PE adopts an 8-point pipelined SDF FFT architecture with a reorder buffer, a complex multiplier and the associated twiddle factor table as shown in Fig. 6.27. The complex multiplier can be implemented in three real multiplications and five real additions/subtractions as expressed in (6.1). Furthermore, depending on the symmetry in the sine/cosine functions, the look-up table stores only the sine and cosine values from 0 to $\pi/4$.

For radix-8 FFT operation, the read-out data index of the memory access is the 3-bit reversal of the write-in data index. In order to achieve the parallel inputs and

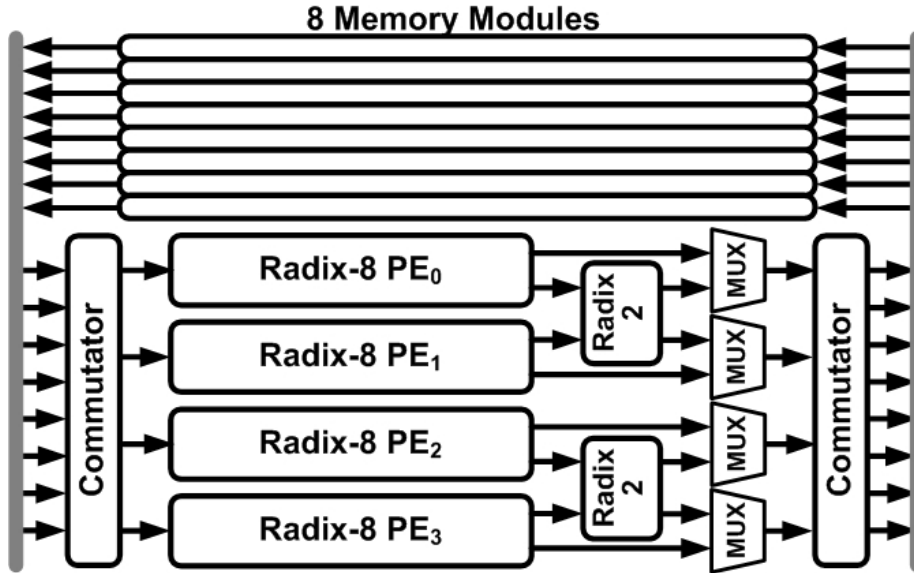


Fig. 6.26 Radix-8 1024-point parallel memory-based FFT architecture.

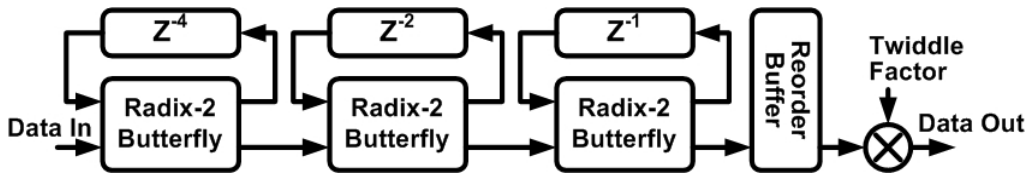


Fig. 6.27 Radix-8 processing element.

outputs in normal order, the memory access addressing for eight memory modules shall avoid the memory conflict occurred in the input or output orders. Assume that the binary index of write-in data is $\{A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0\}$, and the binary index of read-out data is $\{B_9B_8B_7B_6B_5B_4B_3B_2B_1B_0\}$. A data in the read-out index $\{B_9B_8B_7B_6B_5B_4B_3B_2B_1B_0\}$ mapping to the 3-bit reversal write-in index is $\{A_2A_1A_0A_5A_4A_3A_8A_7A_6A_9\}$. For providing the data conflict free of the parallel inputs and outputs in normal order, the parallel write-in data assigned to the eight independent memory modules are based on the addressing scheme of $\{A_2A_1A_0\} + \{A_9A_8A_7\}$, so the data located in the different memory modules can be successfully parallel outputted in normal order as illustrated in Fig. 6.28.

The commutators are used to communicate the data between the memory modules and the radix-8 PEs. According to radix-8 FFT algorithm, the 1024-point

Memory Bank	0	8	16	...	135	...	262	...	389	...	516	...	643	...	770	...	897	...
000	0	8	16	...	135	...	262	...	389	...	516	...	643	...	770	...	897	...
001	1	9	17	...	128	...	263	...	390	...	517	...	644	...	771	...	898	...
010	2	10	18	...	129	...	256	...	391	...	518	...	645	...	772	...	899	...
011	3	11	19	...	130	...	257	...	384	...	519	...	646	...	773	...	900	...
100	4	12	20	...	131	...	258	...	385	...	512	...	647	...	774	...	901	...
101	5	13	21	...	132	...	259	...	386	...	513	...	640	...	775	...	902	...
110	6	14	22	...	133	...	260	...	387	...	514	...	641	...	768	...	903	...
111	7	15	23	...	134	...	261	...	388	...	515	...	642	...	769	...	896	...

Write-In Data Index $\{A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0\}$

Memory Bank Selection $\{A_2A_1A_0\}+\{A_9A_8A_7\}$

Address $\{A_9A_8A_7A_6A_5A_4A_3\}$

Fig. 6.28 Addressing scheme of the write-in data.

TABLE 6.4
READ OR WRITE ADDRESS FOR THE BUTTERFLY UNIT IN EACH STAGE

	Read or Write Address	Address in Memory Bank
Stage1	$b_2b_1b_0b_7b_6b_5b_4b_3p_1p_0$	$\{b_2b_1b_0\}+\{b_3p_1p_0\}$
Stage2	$b_7b_6b_5b_2b_1b_0b_4b_3p_1p_0$	$\{b_7b_6b_5\}+\{b_3p_1p_0\}$
Stage3	$b_7b_6b_5b_4b_3p_1b_2b_1b_0p_0$	$\{b_7b_6b_5\}+\{b_1b_0p_0\}$

DFT operation requires four computation stages which include three stages of radix-8 computation and the last stage of radix-2 computation. Table 6.4 lists the access addressing scheme for the four radix-8 PEs in different computation stages. An 8-bit data $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$ is counted to generate the communicated address, and the two-bit data $\{p_1p_0\}$ is used to denote the ID of the four PEs, where $\{00\}$, $\{01\}$, $\{10\}$, and $\{11\}$ denote PE₀, PE₁, PE₂, and PE₃, respectively. By using this addressing scheme, only the stage 3 has the conflict read address, so a stall cycle is used when PE₂ and PE₃ read the data in stage 3.

6.6 Data Flow of the Proposed Receiver

Based on the proposed receiver, the time-domain OFDM symbols after phase compensation are forwarded through the FFT processor for OFDM demodulation. Then, the frequency-domain received OFDM symbols are executed by channel estimation and STBC decoding. Since the proposed receiver is applied in STBC-OFDM system with two transmit antenna and one receive antenna, the two received OFDM symbols within a time slot must be ready at the same time for channel estimation and STBC decoding. Thus, the FFT processor which is arranged with five memory banks, MB_R1_0, MB_R1_1, MB_R1_2, MB_R2_0, and MB_R2_1, is used to perform OFDM demodulation and buffer the received OFDM symbols as shown in Fig. 6.29. A time-domain received OFDM symbol after phase compensation is written into one of these five memory banks, and this memory bank will be used to execute the FFT process. While executing the FFT process, the time-domain received OFDM symbol is continuously written to another memory bank. After the two OFDM symbols within a time slot have been FFT transformed, the memory banks storing these two OFDM symbols are ready for the access of channel estimation and STBC decoding. The process of channel estimation and STBC decoding should be finished within a time slot. The memory banks MB_R1_0, MB_R1_1 and MB_R1_2 are used to buffer the first received OFDM symbol $R[1,k]$ within a time slot. The memory banks MB_R2_0 and MB_R2_1 are used to buffer the preamble symbol and the second received OFDM symbol $R[2,k]$ within a time slot.

The flow chart of the proposed receiver is shown in Fig. 6.30. After phase compensation, the preamble symbol is first written to MB_R2_0. The following two OFDM symbols are then written to MB_R1_0 and MB_R2_1 while the preamble symbol in MB_R2_0 are calculated by FFT processor and then ready at the time slot 0 for the channel estimation and STBC decoding access. Similarly, when the other two OFDM symbols are written to MB_R1_1 and MB_R2_0, the two OFDM symbols in MB_R1_0 and MB_R2_1 are ready at the time slot 1. Following the flow chart, the access scheme at the time slot 6 is the same as that at the time slot 0. Thus, the memory access schemes of these five memory banks are repeated every six time slots.

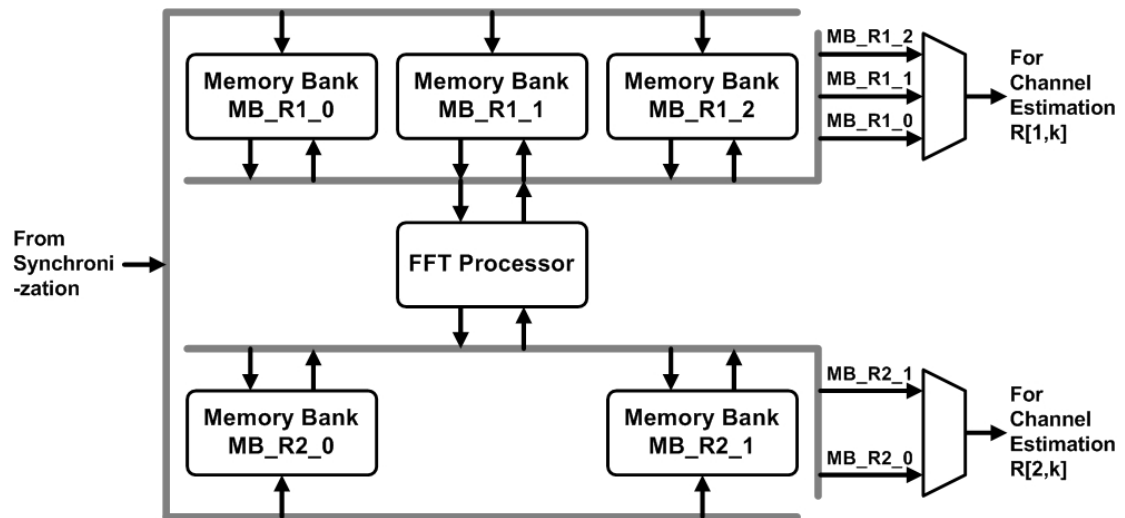


Fig. 6.29 FFT processor arranged with five memory banks.

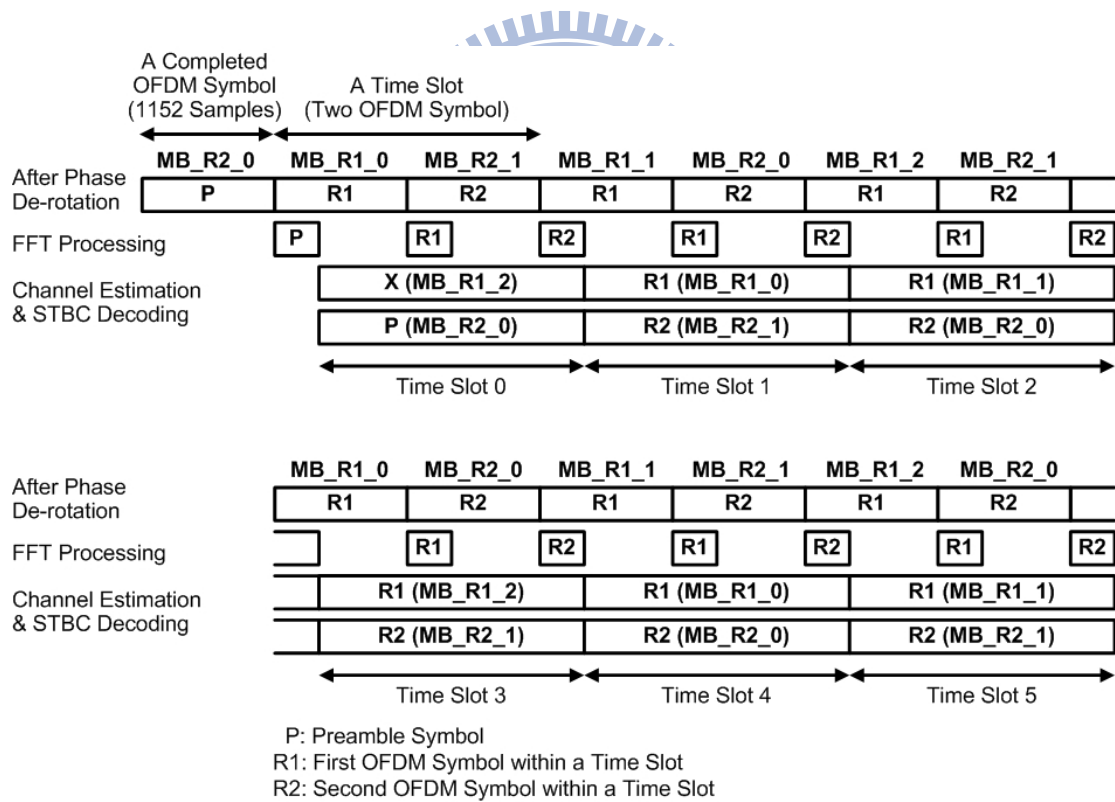


Fig. 6.30 Flow chart of the proposed receiver.

6.7 Simulation Results

The performances of the proposed receiver are demonstrated through the simulation of an STBC-OFDM system with two transmit antennas and one receive antenna. The system occupies a bandwidth of 10 MHz. The carrier central frequency is set to 2.5 GHz with CFO= ± 7 ppm variations in the transmitter and receiver. The multipath channels adopt the ITU Veh-A [48] channel model with relative path power profiles of 0, -1, -9, -10, -15, and -20 (dB), and the path excess delays are uniformly distributed from 0 to 50 sampling periods. Moreover, Jakes model is used to generate a Rayleigh fading environment [49].

Fig. 6.31 shows the BER performances of the proposed scheme and the hardware version with four tracking iterations at the vehicle speed v_e of 120 km/hr. The maximum Doppler frequency f_D is 277.8 Hz and is about 0.025 normalized to subcarrier spacing Δf . The hardware version is simulated with fixed-point word length. The result of perfect channel estimation (perfect CSI) is also included for

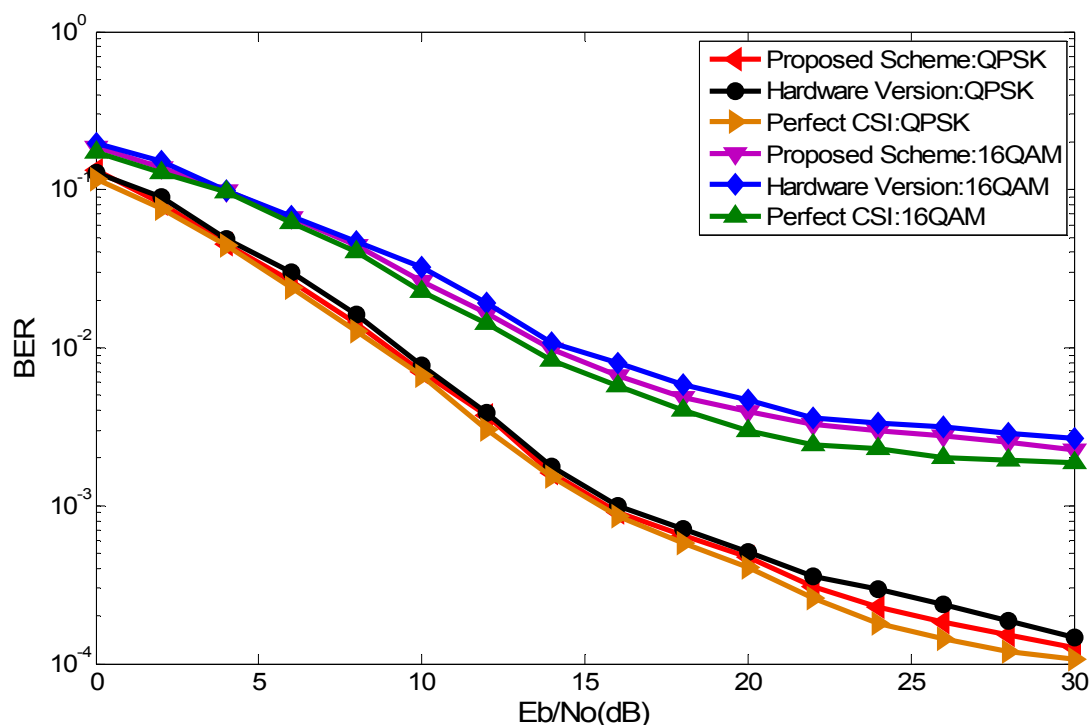


Fig. 6.31 BER performances at v_e of 120 km/hr.

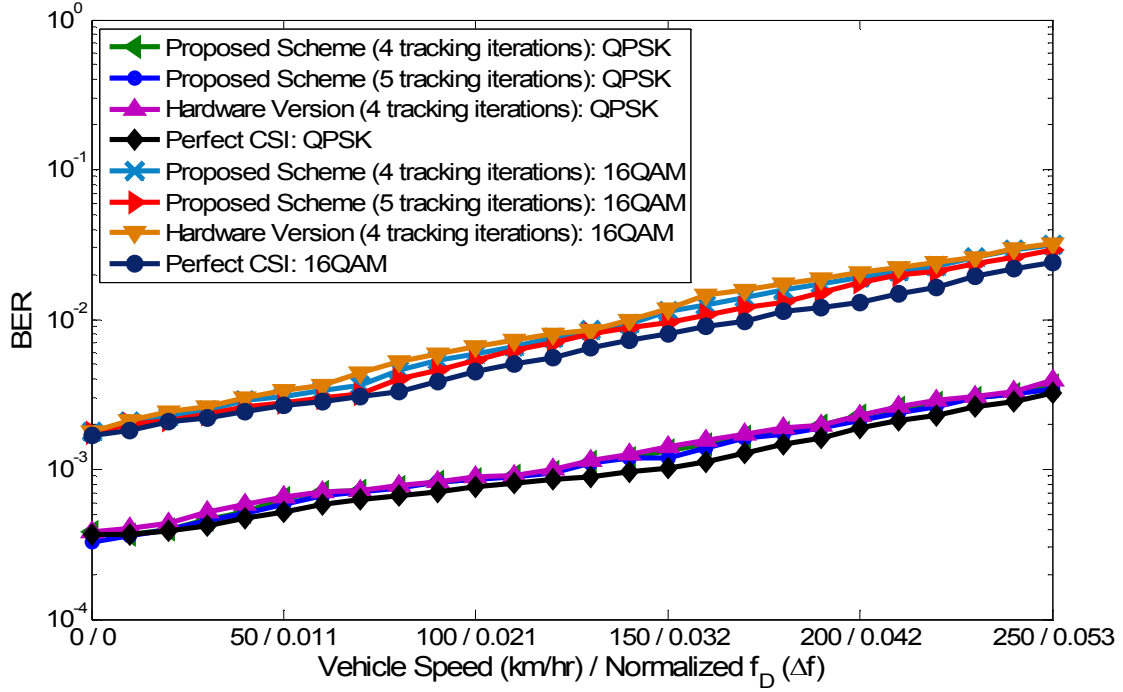


Fig. 6.32 BER performances versus the vehicle speed.

benchmarking. The performance curves of the proposed scheme and the hardware version are very close to the perfect CSI curve. In QPSK modulation, the curve of the hardware version has about 0.27 dB gap in E_b/N_0 as compared with the proposed scheme and about 0.78 dB gap in E_b/N_0 as compared with the perfect CSI case at $BER=10^{-3}$. In 16-QAM modulation, the curve of the hardware version has about 0.56 dB gap in E_b/N_0 as compared with the proposed scheme and about 1.25 dB gap in E_b/N_0 as compared with the perfect CSI case at $BER=10^{-2}$.

Finally, Fig. 6.32 shows the BER performances under different v_e at $E_b/N_0=16$ dB. At v_e of 120 km/hr ($f_D=277.8$ Hz= $0.025\Delta f$), BER of the perfect CSI case, the proposed scheme and the hardware version with four tracking iterations for QPSK/16-QAM can achieve about $8.5 \times 10^{-4}/5.5 \times 10^{-3}$, $9.6 \times 10^{-4}/7.7 \times 10^{-3}$ and $9.9 \times 10^{-4}/8.1 \times 10^{-3}$, respectively, without using channel coding. In Fig. 6.32, we further provide the BER performance of the proposed scheme with five tracking iterations. The BER performance curves for four and five tracking iterations are very close. In other words, no further improvement in BER can be achieved after four tracking

iterations with the vehicle speed up to 240 km/hr. Even at v_e of 240 km/hr ($f_D=555.6$ Hz= $0.0514f$), BER of the hardware scheme with four tracking iterations for QPSK/16-QAM can achieve about $3.3 \times 10^{-3}/3.0 \times 10^{-2}$.

6.8 Design Results

The proposed receiver is implemented in 90nm 1P9M, 1 V CMOS technology. Several memory types are available. According to the memory compiler report, the single-port SRAM solution for the received symbol buffers can save about 50% to 60% of the area than the dual-port SRAM solution. In the proposed receiver design, we relax the access time constrain and make only one read or write per memory module in the memory bank, so that we can use low cost single-port register file. For example, the area of single-port register file which is 0.023 mm^2 is significant smaller than that of dual-port SRAM which is 0.054 mm^2 for the size of 128 words \times 38 bits.

The hardware implementation result of the proposed receiver is listed in Table 6.5. The chip layout of the proposed receiver is shown in Fig. 6.33. The area is 5.55 mm^2 and equivalent to 1,385,523 gates. For this receiver, there are two clocks, 11.2

TABLE 6.5
IMPLEMENTATION RESULTS OF THE PROPOSED RECEIVER

Technology		CMOS 90nm 1P9M, 1V
Sampling Frequency		11.2 MHz
Clock Frequency		78.4 MHz
Size (mm^2)	Core	2.41×2.41
	Chip	3.21×3.21
Gate Count (gates)		1,386,523 (5.55 mm^2)
SRAM Size (bits)		412.5K
Power (mW)		68.48

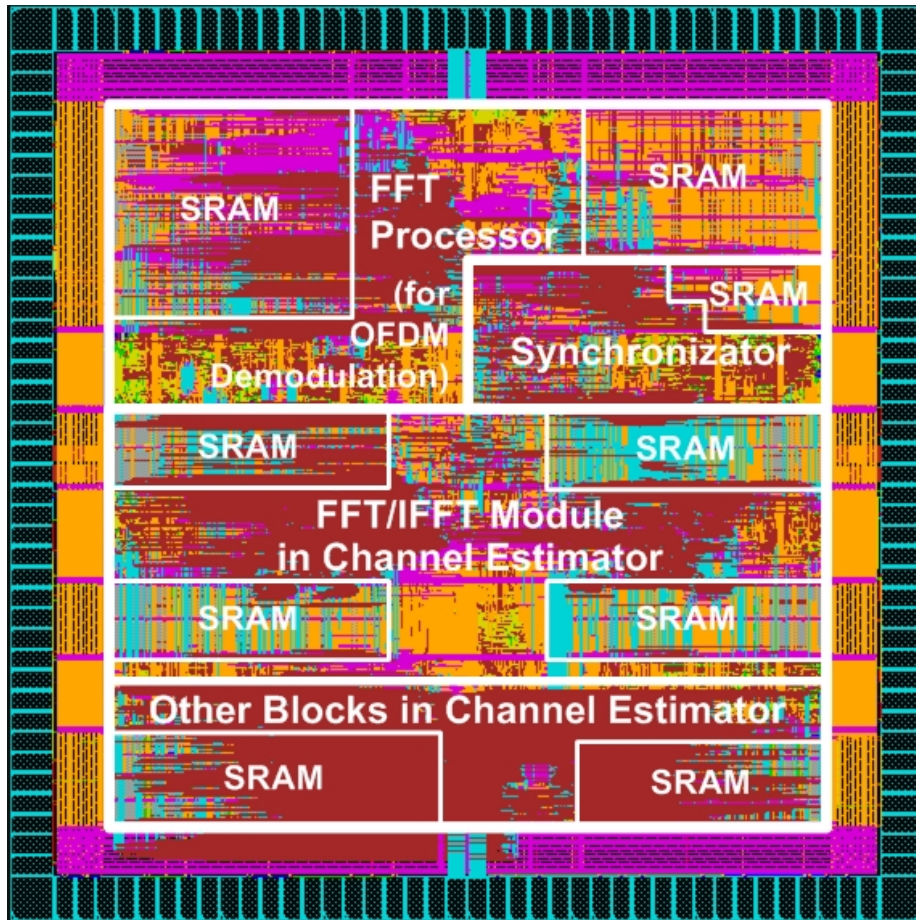


Fig. 6.33 Chip layout of the proposed STBC-OFDM downlink baseband receiver IC.

MHz and 78.4 MHz, to be used as the sampling frequency and the operation frequency, respectively. The power is evaluated to be 68.48 mW from a supply voltage 1 V. The core size of this chip is $2,411 \times 2,411 \text{ um}^2$, and the chip size is $3,211 \times 3,211 \text{ um}^2$. The chip is packaged with 208 pins. Among 208 pins, 26 pins and 66 pins are used for inputs and outputs, and 68 pins are power and ground pins. One pair of the power and ground pins is used for I/O bias pads, 6 pairs and 23 pairs are used for input and output power pads, and other 4 pairs are used for core power pads.

The area proportions of the synchronizer, the FFT processor arranged with five memory banks for OFDM demodulation and the two-stage channel estimator are about 7.26%, 30.75% and 61.99% as illustrated in Fig. 6.34. The area of the synchronizer is $402,450 \text{ um}^2$ and equivalent to 100,613 gates. It effectively uses the

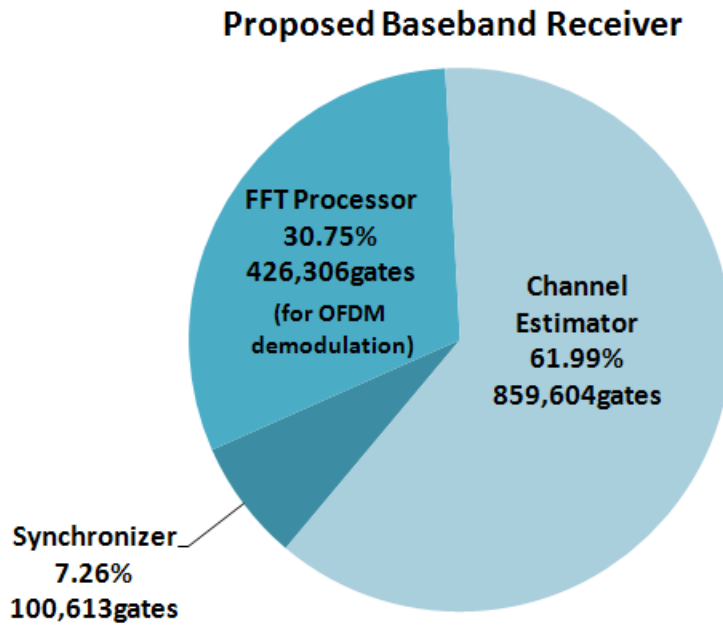


Fig. 6.34 Area proportion of the proposed STBC-OFDM downlink baseband receiver.

single-port register file to have the lower cost of memory area; besides, the match filter design adopting the unsigned calculation can further reduce the area from $150,623 \text{ um}^2$ (signed calculation) to $109,327 \text{ um}^2$, which saves the area of about 27.42%. The area of the FFT processor for OFDM demodulation is about $1,705,228 \text{ um}^2$ and equivalent to 426,306 gates, and that excluding the five memory banks is about $494,499 \text{ um}^2$ and equivalent to 123,625 gates.

The proposed receiver is applied in STBC-OFDM system with two transmit antenna and one receive antenna; hence, the two channel responses transmitted from different antenna must be estimated and STBC decoded with the two received OFDM symbols simultaneously. Since the process of four tracking iterations is enough to achieve an acceptable BER performance as shown in Fig. 6.32, an OFDM symbol time is dominated by the execution time in the initialization stage. Within a time slot (containing two OFDM symbol times), the proposed channel estimator can support up to four tracking iterations in the tracking stage, and the iteration number can be adapted to the vehicle speed. The channel estimator outputs the two decision OFDM symbols in each time slot. Table 6.6 lists the implementation result of the proposed

TABLE 6.6
IMPLEMENTATION RESULTS OF THE PROPOSED CHANNEL ESTIMATOR

Technology	CMOS 90nm 1P9M, 1 V
Clock Frequency	78.4 MHz
Proposed Channel Estimator	
SRAM Size (bits)	195.2K
Total Area Including SRAM (gates)	859,604 (3.43mm ²)
Power (mW)	43.71
Proposed Channel Estimator Excluding the FFT/IFFT Module	
SRAM Size (bits)	7.8K
Total Area Including SRAM (gates)	281,226 (1.12 mm ²)
Power (mW)	13.97

channel estimator. The area of the proposed channel estimator is about 3,438,417 μm^2 and equivalent to 859,604 gates. Without the FFT/IFFT module, the area is only 1,124,907 μm^2 and equivalent to 281,226 gates. In 16-QAM modulation, the uncoded throughput for this receiver design can support up to 27.32 Mbps which is the number of bit transmission in a frame divided by the time duration of a frame.

Fig. 6.35 illustrates the area proportion of each block in the proposed channel estimator. The preamble match and LS estimators are combined to be the PREA/LS module, the SMPIC-based decorrelator and path decorrelator are merged to be the SMPIC/HESS module, and the FFT and IFFT sharing the same design resources are denoted as the FFT/IFFT module. Final channel response update and store are designed to be the END/PROCESS module. As shown in Fig. 6.35, the FFT/IFFT module used for the proposed channel estimator take 65.95% area of the channel estimator, but other parts only cost 34.05%. Therefore, the FFT/IFFT module dominates the complexity of the channel estimator implementation. The ratio of combinational and non-combinational area is about 4:6; in other words, the use of

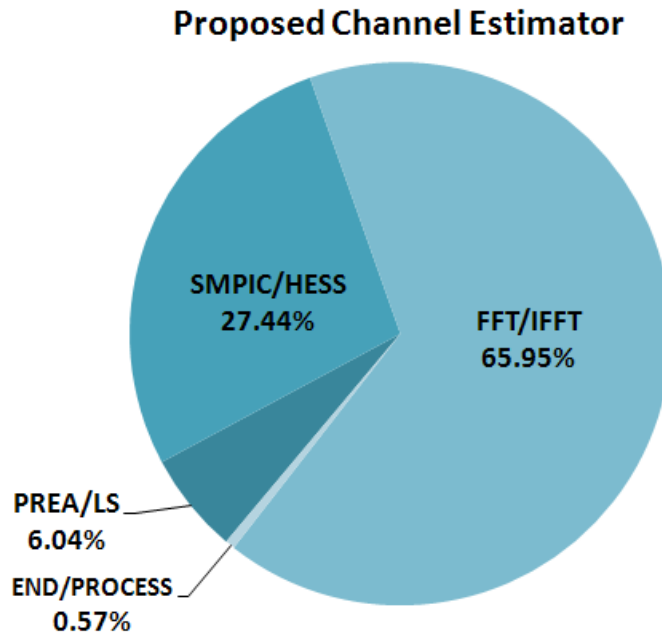


Fig. 6.35 Area proportion of the proposed two-stage channel estimator.

memory in this design occupies about 60% of overall design. Only partial outputs of IFFT and partial inputs of FFT are required in the proposed channel estimation method. Therefore, in the future, the FFT/IFFT module can be further studied with the partial FFT algorithm [82] to reduce the computational complexity and memory access operations. It can also be improved by the scaling algorithm [83] with shorter word length. When the internal word length of the FFT/IFFT module can be reduced by 3~4 bits, the area cost can be improved by about 10~12%.

Fig. 6.36 illustrates the hardware reduction of the proposed channel estimator. Under the same system timing requirement, the direct implementation of the two-stage channel estimation requires about 1891.2K gates excluding the FFT/IFFT module. By using our proposed scheme and architecture mentioned in Section 4.4 and Section 6.4, the hardware is reduced to only 281.2K gates, which is 14.8% of the original two-stage channel estimator design. The percentage value in the bar denotes the step-by-step hardware reduction of each block as compared with the overall direct implementation architecture. In the initialization stage, the preamble match uses only adders and shifters instead of multipliers, and the SMPIC-based decorrelator

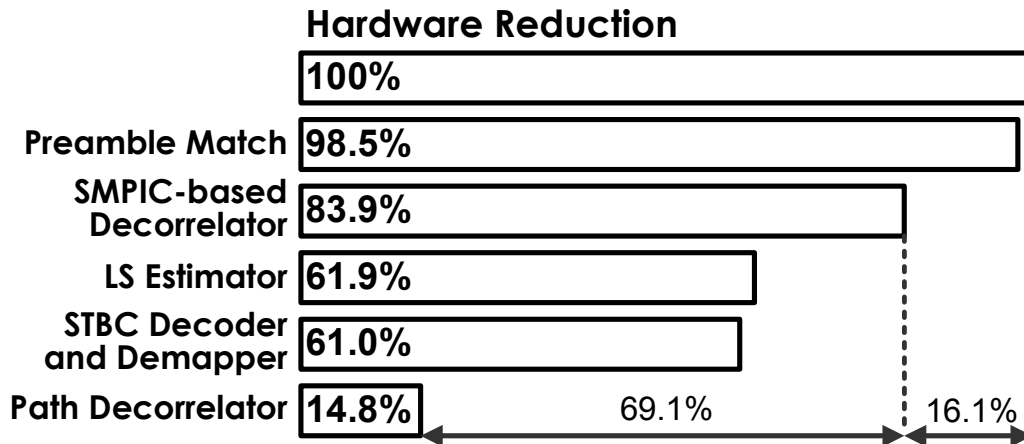


Fig. 6.36 Hardware reduction of the proposed channel estimator.

efficiently reduces the execution cycles by 9.63 times as compared to the MPIC-based decorrelator. Moreover, in the tracking stage, the LS estimator only uses adders and multiplexers instead of complex multipliers and dividers. The implementation of matrix inverse is avoided in both the direct implementation and the proposed channel estimator. Since the Hessian matrix calculator is effectively merged into the path decorrelator, the path decorrelator further avoids a lot of execution cycles to compute each entry of the Hessian matrix and frees to use any multipliers and memory. Furthermore, the path decorrelator uses only adders and multiplexers instead of complex multipliers for matrix multiplication.

In summary, the interpolation-based channel estimation methods have the advantage of low implementation cost since they do not require FFT and IFFT to operate in transform domain. However, their disadvantage is difficult to estimate CSI accurately under the situation of limited pilot subcarriers over doubly selective channels. In contrast, the two-stage channel estimation method has significant performance improvement in outdoor mobile environments, but it requires high hardware cost. For realizing the successful STBC-OFDM systems, the proposed channel estimator effectively improves the design complexity of the two-stage channel estimation with acceptable hardware cost while keeping the performance of the two-stage channel estimation.

TABLE 6.7
COMPARISONS WITH OTHER DESIGN OPERATING AT 200 MHz

	Haene [84]	Our Work
OFDM Symbol Size	64	1024
STBC-OFDM System	not applied	applied
Data Demapper	not included	included
For One Pair Channel Estimation without Data Detection		
Technology	0.25 um	90 nm
SRAM Size (bits)	N/A	98K
Total Area (gates)	60.6K	422.6K

Table 6.7 lists the comparisons between the proposed channel estimator and a similar work [84] implemented by a transform domain ML method. Note that the two designs have distinct functions and a fair comparison between them is hard to carry out. When the operating frequency is up to 200 MHz which is same as the work in [84], the proposed channel estimator requires about 422.6K gates for one pair channel estimation. For comparison, the normalized design efficiency is defined as

$$\text{Design Efficiency} = \text{OFDM Symbol Size} / \text{Gate Count}. \quad (6.16)$$

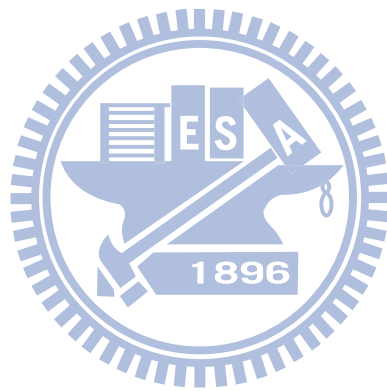
As we can see, the proposed channel estimator has 2.29 times of design efficiency than the work in [84].

6.9 Summary

In this chapter, the implementation of the proposed baseband receiver for STBC-OFDM systems is presented. In the proposed receiver design, we use low cost single-port register file to save about 50% to 60% of the area than the dual-port SRAM solution. The match filter design adopting the unsigned calculation can further save the area of about 27.42% as compared with that adopting signed calculation. The

implementation complexity of the proposed channel estimator is effectively reduced by 85.2% as compared with the direct implementation. Moreover, it has 2.29 times of design efficiency than a similar work in [84]. Fixed-point simulation is used to imitate the actual hardware behaviors. When operating at v_e of 120 and 240 km/hr with E_b/N_0 of 16 dB for QPSK modulation, the proposed receiver can achieve BER of about 10^{-4} and 10^{-3} without using channel coding. This proposed receiver is implemented in 90nm CMOS technology and operated at 78.4 MHz from 1 V supply voltage while drawing 68.48 mW. The design area costs 1,386,523 gates and can support up to 27.32 Mbps (uncoded) downlink data transmission in 16-QAM modulation. With all these features, the proposed receiver can be applied to STBC-OFDM systems for WMAN in fixed and mobile channels.





Chapter 7

Conclusions

In this dissertation, a downlink baseband receiver for STBC-OFDM systems in outdoor mobile channels is proposed. It is successfully applied in IEEE 802.16e. A simple symbol timing detection scheme, a carrier frequency recovery loop modified by the ping-pong algorithm and a robust two-stage DFT-based channel estimation method are implemented in the proposed receiver. Features of the proposed downlink baseband receiver include the following:

- implementation of a downlink baseband receiver applied in an STBC-OFDM system with two transmit antennas and one receive antenna;
- adoption of a simple and robust synchronization mechanism for symbol timing detection and CFO estimation;
- adoption of a high-performance two-stage channel estimation method to provide precise CSI in wireless mobile channels and MIMO channels;
- provision of an efficient channel estimator architecture which improves about 85.2% implementation complexity for low-complexity hardware implementation.

From the system simulation results, we have shown that the carrier frequency detection modified by the ping-pong algorithm can effectively improve the ICFO error probability by 20 times in the weak region. As the proposed two-stage channel estimation method is compared with 2-D interpolation methods, the proposed scheme improves the normalized MSE of about 8.5~10.4 dB in outdoor mobile channels.

When operating at the vehicle speed v_e of 120 and 240 hr/km with E_b/N_0 of 16 dB for QPSK modulation, the proposed channel estimation can achieve BER of about 10^{-4} and 10^{-3} without using channel coding. Moreover, using the convolutional coding specified by IEEE 802.16e, the coded BER of the proposed receiver for 16-QAM modulation can achieve less than 10^{-6} with coded rate of 1/2 at v_e of 120 km/hr. The proposed receiver implemented in 90nm CMOS technology can support up to 27.32 Mbps (uncoded) downlink data transmission in 16-QAM. The design of the proposed receiver has an equivalent gate count of 1,385,523 gates. The chip implementation requires the core size of $2.4 \times 2.4 \text{ mm}^2$ and dissipates 68.48 mW at 78.4MHz operating frequency from a supply voltage 1 V. With verifications through all the design and simulation results, the proposed STBC-OFDM baseband receiver can provide a solid foundation for WMAN receiver design in fixed and mobile wireless communication.



Appendix A:

A DC-balance Low-jitter Transmission Code for 4-PAM Signaling

A.1 Introduction

In a serial link data transmission system, a data clock is not transmitted with the data. In the receiver end, phase noise (jitter) in the generated clocks, phase alignment of the sampling clocks to the received high-speed data and data jitter all influence system performance [85]. Therefore, clock generation and receiver timing recovery are crucial functions in a high-speed signaling system.

8B/10B data transmission method has become the standard for many high-speed serial links today [86]. This method belongs to the physical network layer and is utilized in IEEE 1394 [87], 1/10 Gigabit Ethernet [88][89], Asynchronous Transfer Mode (ATM) [90] and fiber optic transmission link [91]. 8B/10B transmission code includes serial encoding and decoding rules, special characters, and error detection functions. The encoding scheme creates a DC balanced bit stream to ensure an equal number of positive and negative pulses to prevent signal distortions associated with the AC-coupling. Moreover, the scheme guarantees at least one signal transition for five transmitted bits in the symbol stream, and thus provides sufficient transitions for clock recovery and significantly increases the probability of detecting single or multiple errors during data transmission.

The channel (e.g. cable or wire) bandwidth limitation increases the importance of 4-PAM signaling transmission, which only adopts half of the bandwidth of 2-PAM signaling [92]. This dissertation proposes a novel DC-balanced low-jitter transmission code, a 4-PAM symmetric code [93]. The 4-PAM symmetric code preserves all the useful characteristics of 8B/10B code; moreover it can decrease the jitter of the timing transition of the data in the receiver for 4-PAM signaling. The remainder of this appendix is organized as follows. Section A.2 describes the design of the proposed 4-PAM symmetric code. Section A.3 presents the architecture designs and the implementations of the 4-PAM symmetric encoder and decoder. Finally, Section A.4 draws the conclusions.

A.2 Design of the 4-PAM Symmetric Code

A differential 4-PAM stream has three different transition types, as illustrated in Fig. A.1. Of these three transitions, only type 1 makes a transition to the same magnitude but opposite polarity (symmetric transition). A type 1 transition, generates a zero crossing precisely at the midpoint between two symbols, and is therefore the

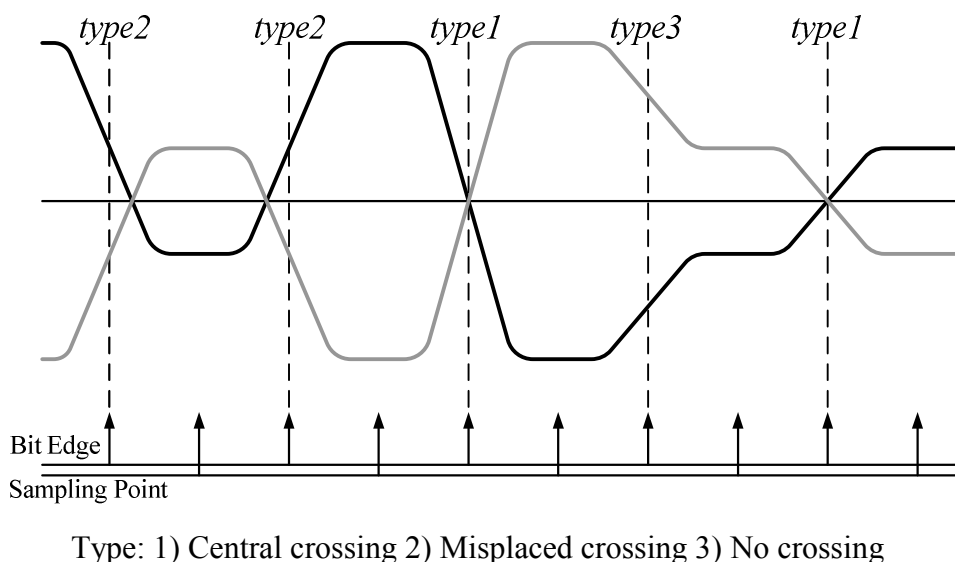


Fig. A.1 Three transition types in the differential 4-PAM symbol stream.

most appropriate for clock recovery. Type 2 and type 3 transitions shall be avoided since they convey incorrect phase information [94].

For two adjacent 4-PAM symbols (ex, symbol a and symbol b), we can have the formula

$$y = [\text{MSB}(a) \oplus \text{MSB}(b)] \cdot [\text{LSB}(a) \oplus \text{LSB}(b)] \quad (\text{A.1})$$

(\oplus is XOR operation; \cdot is AND operation). If $y = 1$, it means that this transition is type 1 transition. Therefore, using (A.1), the receiver can detect the type 1 transition and ignore type2 and type3 transition. To improve the jitter on the data transition, a 4-PAM symmetric code is proposed, providing at least one type 1 transition of the differential 4-PAM signal for every ten transmitted symbols. By the 4-PAM symmetric coding scheme, the differential 4-PAM data streams have sufficient density of type1 transitions and only type 1 transitions perform synchronization in the receiver. By do so, it will reduce the jitter by $\pm 25\%$ of the data transition region (T_s) as illustrated in Fig. A.2.

A.2.1 The Structure of 4-PAM Symmetric Code

The 4-PAM symmetric code translates byte-wide data into a 10-bit serial data stream as the 8B/10B transmission code does. The proposed 4-PAM signaling structure adopts parallel inputs of 16-bit data to two 4-PAM symmetric encoders, each of which has 8-bit data input, as depicted in Fig. A.3. The two encoders encode the MSB and LSB part individually, and output 20-bit data from LSB and MSB encoders outputs. The MSB and LSB output bit are combined to generate one symbol with 4-level in amplitude. A data rate of 10 Gbps requires a line rate of 6.25 GBd (one symbol transmitted two bits). Thus, the output of both the MSB encoder and the LSB encoder consists of two 10-bit data words clocked at 625 MHz (equivalent to 10 Gbps in the source data). Therefore, the 4-PAM symmetric code for 4-PAM transmission requires only half the bandwidth required for 2-PAM transmission.

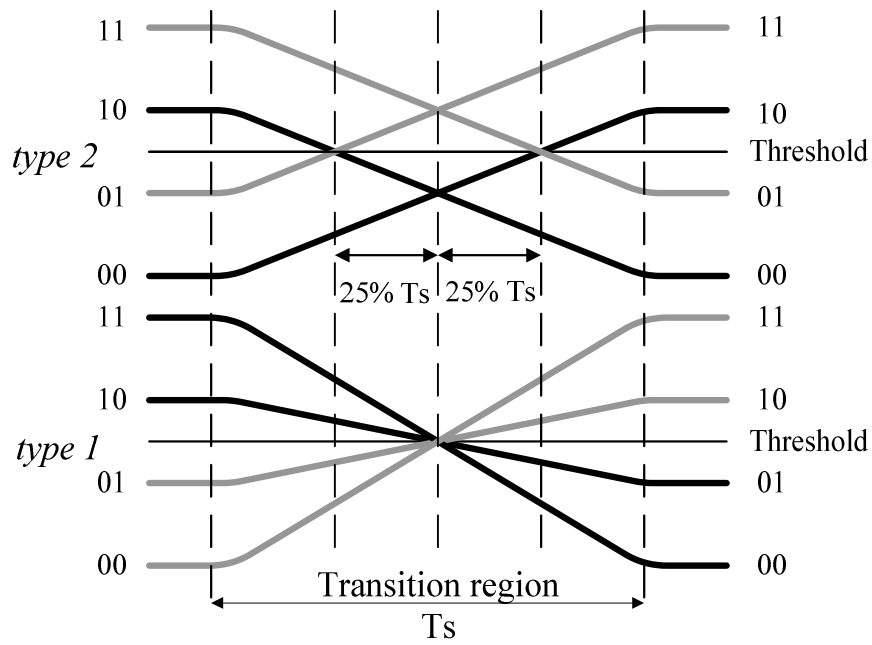


Fig. A.2 Comparisons of type 1 and type 2 transitions in the differential 4-PAM signals.

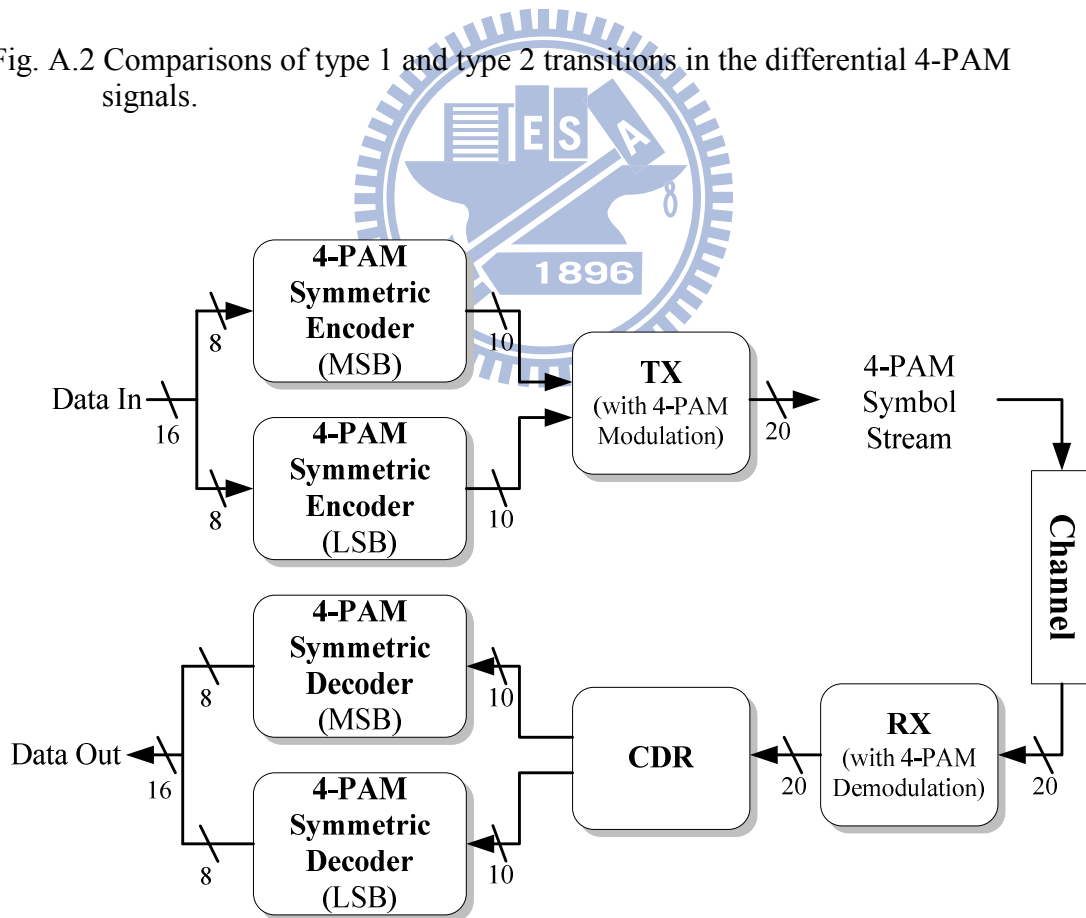


Fig. A.3 Structure of 4-PAM symmetric encoders and decoders for 4-PAM signaling system.

A.2.2 Low-jitter of Data Transitions

In every cycle (10 symbols), the 4-PAM symmetric code guarantees at least one type 1 transition, which is different from that of the 8B/10B encoder in the differential 4-PAM signal. To obtain the type 1 transition of the differential 4-PAM signal, bit 3 and bit 2 of the 4-PAM symmetric code must always be coded for (1,0) or (0,1), as illustrated in Fig. A.4 (a) with $f = \bar{g}$. Combining one MSB bit and one corresponding LSB bit forms one symbol, and then symbol 3 to symbol 2 will be 11 / 00 to 00 / 11 or 01 / 10 to 10 / 01. Therefore symbol 3 to symbol 2 produces a type 1 transition and at least one type 1 transition is generated in every 10 symbols. Furthermore, the third symbol of every 10 symbols occur a type 1 transition and we can obtain the 10-symbols frame synchronization by observing the data stream to find the intervals with highest average density of type 1 transitions, as illustrated in Fig. A.4 (b).

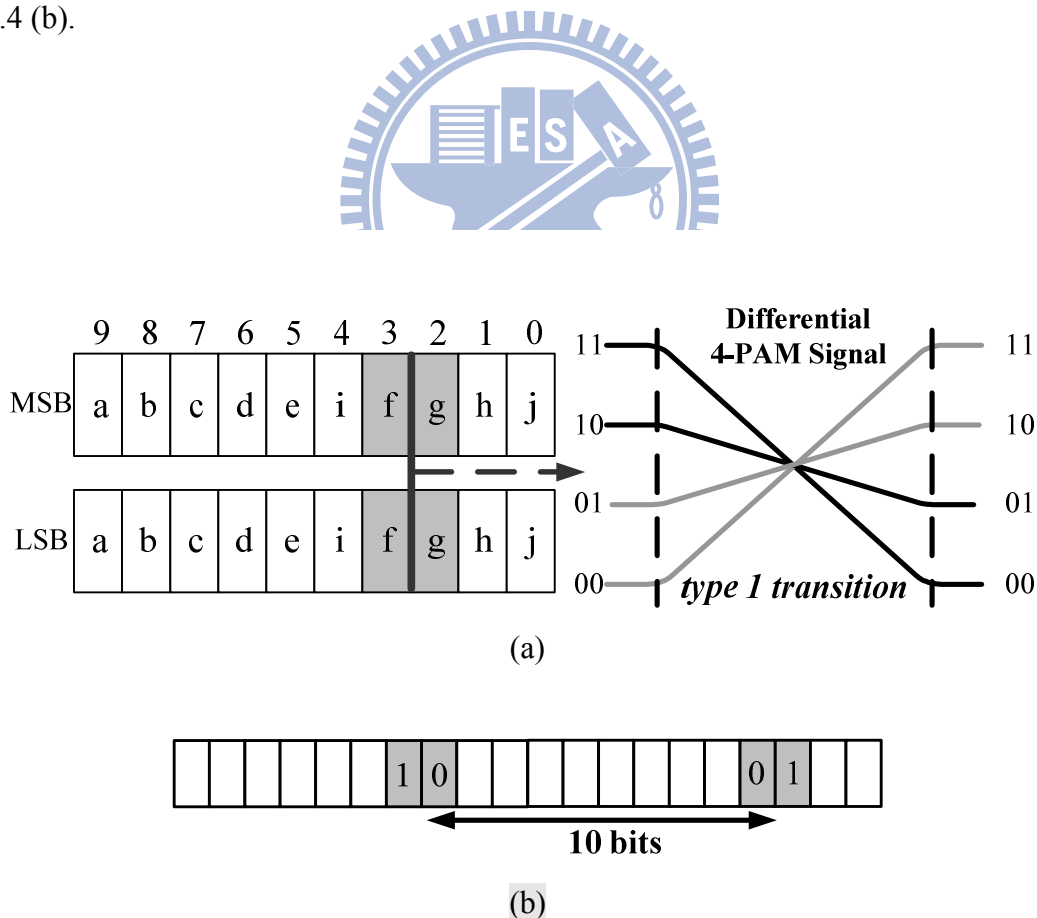


Fig. A.4 (a) Type 1 transition in the 4-PAM symmetric code and (b) frame synchronization.

A.2.3 The Design Concept of Running Disparity

DC balancing is achieved by running disparity. The disparity designates the difference between the numbers of 1s and 0s in a defined block of digits, or the instantaneous deviation from the long-term average value of the running digital sum. The 4-PAM symmetric encoder translates byte-wide data into a 10-bit serial data stream composed of 6-bit and 4-bit sub-blocks. The serial data stream controls the numbers of 1s and 0s in each sub-block transmission to help balance the DC level between the voltage levels denoting 1 and 0. No distinction is made between 6-bit and 4-bit sub-blocks, which are therefore used to compensate each other.

The unit cell concept is used to explain running disparity of the 4-PAM symmetric code. Fig. A.5 plots the disparity (rd) as a function of time or digit intervals (n). For a binary or two-level code (d), each 1s is denoted by a line segment extending over a one-digit interval and rising at a 45° angle, and a 0s is denoted by a falling line. A waveform corresponding to a possible signal during a unit interval is assigned an algebraic value corresponding to its dc component. The 1s and 0s are typically assigned values of +1 and -1, respectively. The formula of disparity is

$$rd(n) = rd(n-1) + (-1)^{d+1} \quad (A.2)$$

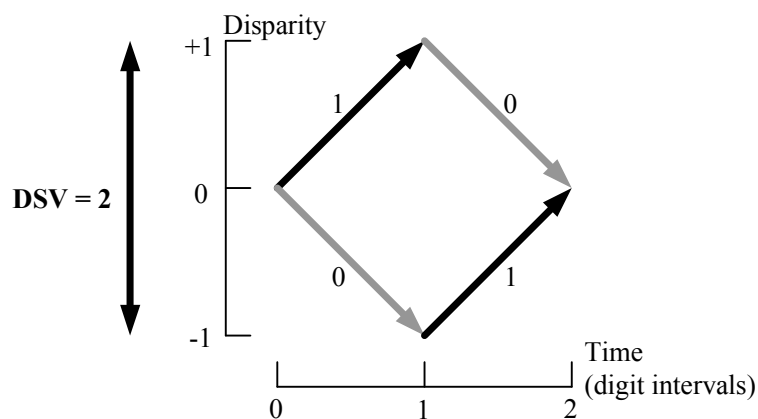


Fig. A.5 Unit cell of running disparity.

The digital sum variation (DSV) is given as the variation in the running disparity sum of the encoded data stream.

Individual 6-bit and 4-bit sub-blocks are complemented in accordance with disparity rules and the difference between the number of 1s and 0s is always 0, ± 2 or ± 4 . The running disparity is positive (+1 or +3) if two or four more 1s than 0s have been transmitted, respectively, and negative (-1 or -3) if two or four more 0s than 1s are transmitted. The running disparity remains unchanged from the previous transmission if the code contains an equal number of 1s and 0s. As illustrated in Fig. A.6, all sub-block boundaries have running disparity values of ± 1 or ± 3 , which are different from the 8B/10B code boundaries of ± 1 . The running disparities of ± 1 and ± 3 make very little difference to the long-term average DC offset. Fig. A.6 clearly demonstrates that the disparity (or DSV) is bounded, and that the run length of the 4-PAM symmetric code is eight, because a fixed type1 transition occurs in the bit 3 and bit 2 positions. The DSV bounded range makes the resultant code DC-balanced,

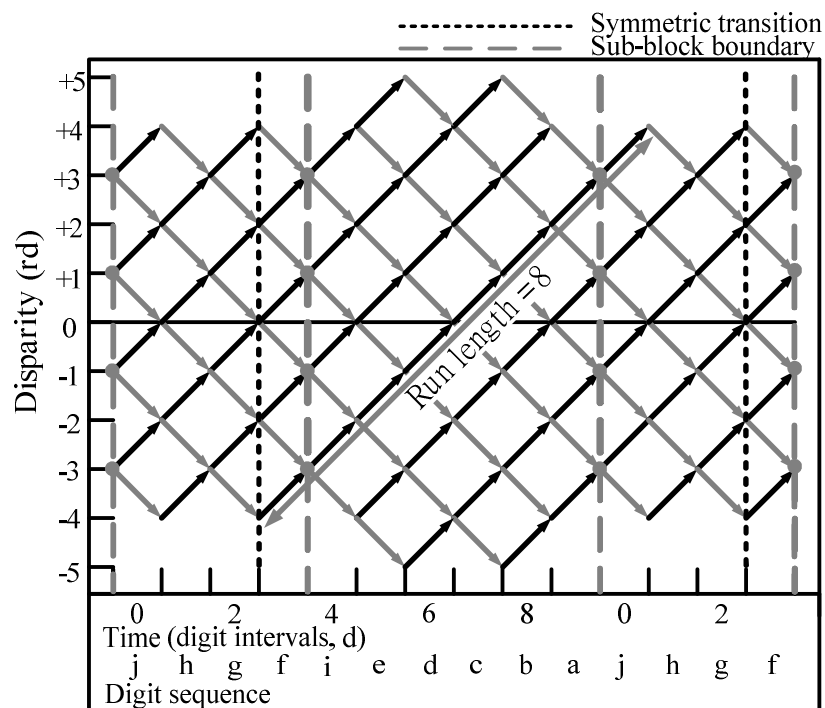


Fig. A.6 Disparity versus time plot.

i.e. zero spectral power at zero frequency, which is one of the most frequently required code characteristics in transmission via AC-coupled channels.

A.2.4 Error Detection

Like the 8B/10B code design concepts, the error detection in the 4-PAM symmetric code can be implemented in two ways. The first approach checks each packet transmitted for redundancy. Additionally, the first approach validates the transmission of each packet, using a Media Access Control (MAC) layer to identify errors based on the start and end delimiters. A delimiter is a special character which characterizes synchronization. The second approach adopts cyclic redundancy checking (CRC) to identify errors on individual 6-bit or 4-bit code blocks. If the code is a legal 4-PAM symmetric code and does not violate the disparity rules, then no errors arise. In addition to 256 data characters, the 4-PAM symmetric code defines twelve out-of-band indicators which are the same as those of the 8B/10B code and are called special control characters. Special characters are typically used for transmitting link diagnostics and code-words such as ABORT, RESET, SHUTDOWN and IDLE.

A.2.5 Comparison of Transmission Codes

Another novel transmission code 8B5Q that encodes 8 bits into five 4-PAM symbols removing all full-swing transitions to reduce the baud rate and achieve low jitter of timing recovery has been proposed in [95]. For the 4-PAM system, Table A.1 gives the comparison of the 8B/10B code and the 8B5Q code with the 4-PAM symmetric code. The 4-PAM symmetric code has the good characteristics, including DC-balanced, finite run-length, special characters and error detection as the 8B/10B code does. For the 4-PAM system, the 4-PAM symmetric code can guarantee sufficient type 1 transitions for low-jitter of timing recovery, but the 8B/10B code does not have such property. The 8B5Q code reduces the jitter of timing recovery for 4-PAM system, but does not have good characteristics as the 8B/10B code and the 4-PAM symmetric code do.

TABLE A.1
8B/10B, 8B5Q AND 4-PAM SYMMETRIC TRANSMISSION CODES
FOR 4-PAM SYSTEM

	8B/10B Code [86]	8B5Q Code [94]	Proposed 4-PAM Symmetric Code [93]
DC-Balanced	Yes	No	Yes
Bounded Disparity	(+1,-1)	No	(+3,+1,-1,-3)
Run Length	5 bits	No	8 bits
Error Detection	Yes	No	Yes
Low Jitter for Timing Recovery	No	Yes	Yes
Frame Synchronization	No	No	Yes

A.3 Architecture and Implementation of 4-PAM Symmetric Encoder / Decoder

The overall block diagram of the 4-PAM symmetric encoder architecture is illustrated in Fig. A.7. The straightforward data flow of 4-PAM makes it easy to pipeline to speed up the encoding rate. In this case, the encoder is partitioned into three stages so that it can operate at 13.1 Gbps. A character is encoded in four clock cycles. The 4-PAM symmetric encoder is clocked by a byte rate clock (*clk*). The *Adder* block estimates the number of 1s in *data_in[7:0]*, as demonstrated in Fig. A.7. The *Disparity* block predicts the next state running disparity state based on the 1s numbers of *data_in[7:0]*.

The *MUX_rd* block specifies the running disparity from the reset mode signals, *int_rd_val[1:0]*, or the *Disparity* block output. The control signal *k_char* denotes whether the *data_in[7:0]* represents data or control information. For encoding, each incoming byte is split into two sub-blocks. The five binary lines *EDCBA* (*data_in[4:0]*) are encoded into the six binary lines *abcdei* by the *abcdei_look_up*

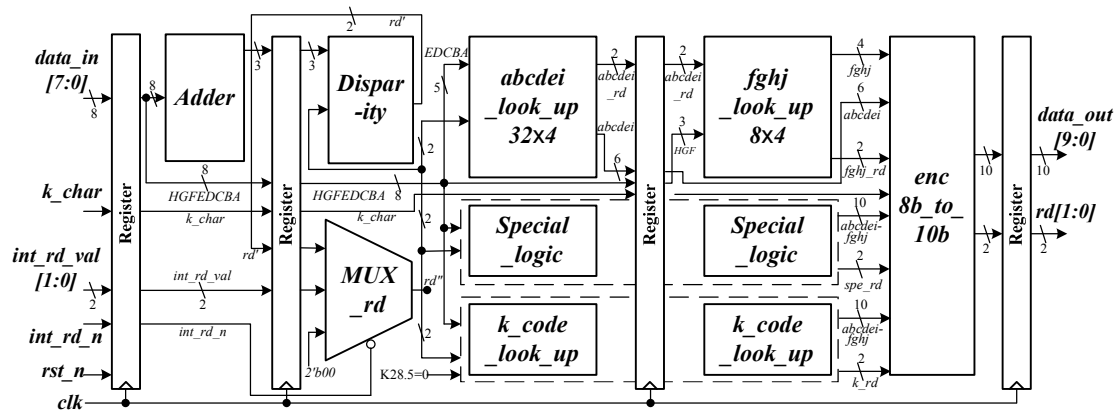


Fig. A.7 Block diagram of the 4-PAM symmetric encoder.

block. Similarly, the three bits HGF ($data_in[7:5]$) are encoded into the four bits $fg'hj$ by the $fg'hj_look_up$ block. The disparity of each sub-block indicates the difference between the numbers of 1s and 0s, where positive and negative disparity numbers represent an excess of 1s and 0s, respectively. The permitted disparity of both $abcdei$ and $fg'hj$ is 0, ± 4 or ± 2 . Therefore, encoding scheme is DC free by complementing the disparity of each sub-block.

The *Special_logic* block encodes the eight bits $FGHEDCBA$ ($data_in[7:0]$) into the ten bits $abcdeifg'hj$ when $FGHEDCBA$ can not directly be encoded by the $abcdei_look_up$ and $fg'hj_look_up$ blocks. When the signal k_char is high, the $k_code_look_up$ block generates special characters. Finally, the $enc8b_to_10b$ block integrates the above results and determines the final outputs ($rd[1:0]$, $data_out[9:0]$).

The overall block diagram of the 4-PAM symmetric decoder architecture is illustrated in Fig. A.8. In this case, the 4-PAM symmetric decoder is pipelined, and therefore a character is decoded in four clock cycles. The 4-PAM symmetric decoder is clocked by a byte rate clock (clk). As demonstrated in Fig. A.8, the *Adder* block is applied to estimate the number of 1s in $data_in[9:0]$, which is employed by the *Disparity* block to predict the next state of running disparity. The *MUX_rd* block determines the running disparity from the reset mode, $int_rd_val[1:0]$ or the *Disparity* block output. For decoding purposes, each incoming 10-bit data is partitioned into two sub-blocks. The *EDCBA_look_up* block then decodes the six binary lines $abcdei$ ($data_in[9:4]$) into five binary lines $EDCBA$. Similarly, the four bits $fg'hj$

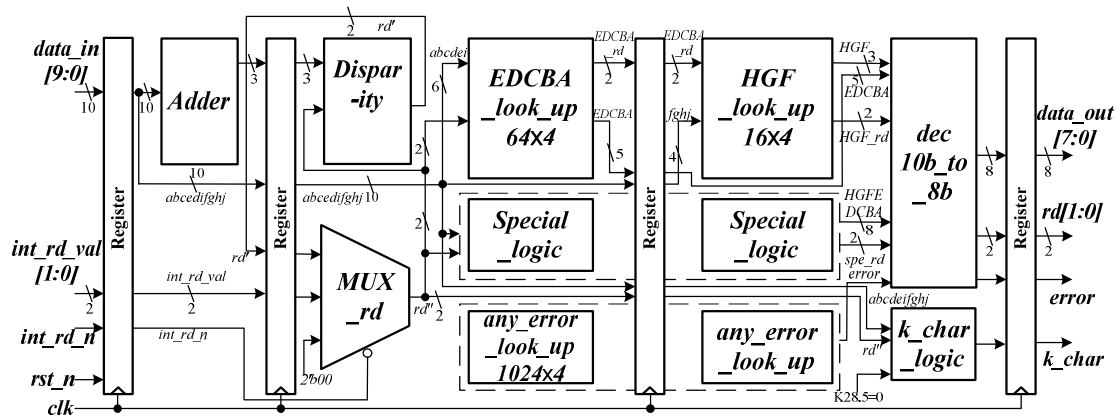


Fig. A.8 Block diagram of the 4-PAM symmetric decoder.

($data_in[3:0]$) are decoded into the three bits HGF by HGF_look_up block. The disparity of each sub-block of the incoming data is the difference between the number of 1s and 0s; positive and negative disparity numbers denote an excess of 1s and 0s, respectively. For both the $abcdei$ and $fghj$, the permitted disparity is 0, ± 2 or ± 4 . The $Special_logic$ block decodes the ten bits $abcdeifghj$ ($data_in[9:0]$) into eight bits $HGFEDCBA$ when they can not be directly decoded by the $EDCBA_look_up$ and HGF_look_up blocks. The $abcdeifghj$ value is uniquely decoded according to only the $HGFEDCBA$ value, without any reference to disparity or other parameters. Although many errors may be identified and sent to the $error$ signal by the $any_error_look_up$ block due to the propagation properties of the 4-PAM symmetric code that, even a single-bit error might not be discovered until several characters after the error is introduced. The k_char_logic block determines whether the $data_in[9:0]$ denotes data or control information and generates the k_char signal. If the k_char signal is high, then the $data_in$ contains a special character, and otherwise it contains a data character. Finally, the $dec10b_to_8b$ block integrates the above results and obtains the final outputs ($rd[1:0]$, k_char , $data_out[7:0]$ and $error$).

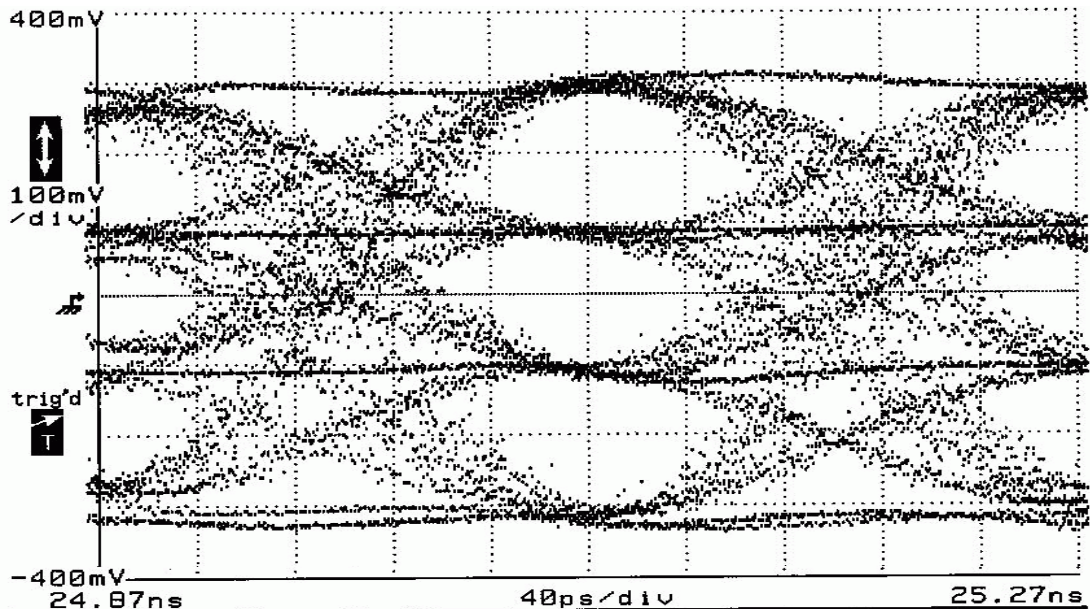
A 4-PAM symmetric encoder / decoder were implemented using UMC 0.18 μm standard cell to synthesize the design. The results are summarized in Table A.2. For the source data rate to achieve 10 Gbps, the area of 4-PAM symmetric encoder / decoder are larger than the 8B/10B encoder / decoder by 1,000~1,500 gate count. The area is measured in equivalents of 2-input NAND gates.

TABLE A.2
SYNTHESIS RESULTS OF THE 4-PAM SYMMETRIC ENCODER / DECODER

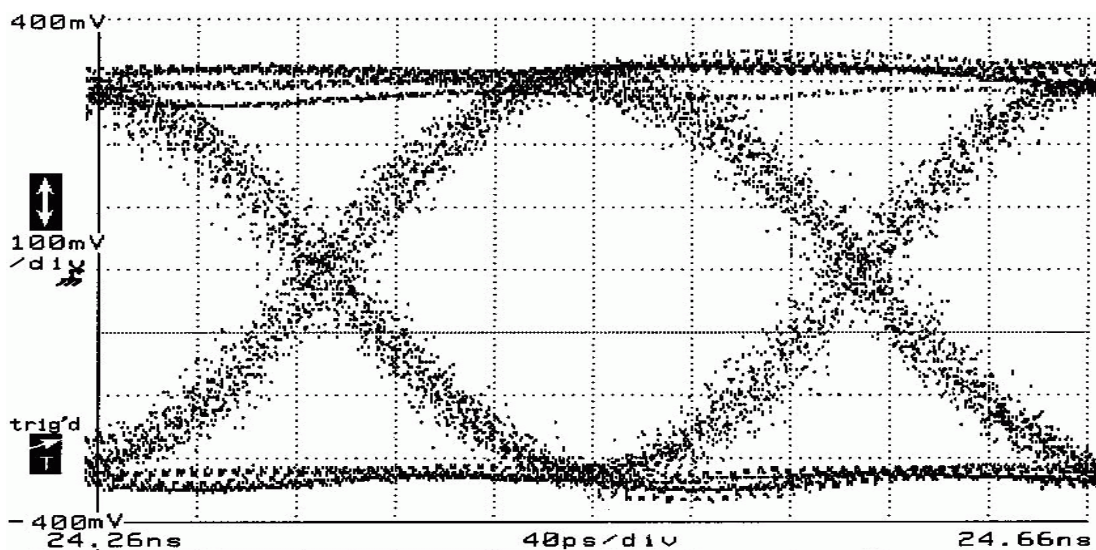
	4-PAM Symmetric Encoder	4-PAM Symmetric Decoder
Technology	UMC 0.18um	
Maximum Clock Rate	819 MHz	704 MHz
Source Data Rate	13.1 Gbps	11.3 Gbps
Area (gates)	2,506	2,915
Power	18.08 mW	16.90 mW

Fig. A.9 shows the measurement results of a devised 4-PAM serial link transmitter [96] outputs with all types of transitions (like the 8B/10B for 4-PAM signaling) and data containing only maximum-swing type 1 transition. The measurements are operated at 10 Gbps and T_s is about 0.6 unit interval (UI). These results demonstrate the peak-to-peak jitter of all types of transitions and the jitter of only maximum-swing type1 transitions are about 0.75 and 0.25 T_s . The measurement results are in agreement with our estimation. Therefore, in the receiver, if the 4-PAM symmetric coding scheme with only type 1 transitions is used, then the timing jitter of the data can be decreased by $\pm 25\%$ T_s . When the data or clock jitter decreases, BER will decrease exponentially [85]. For example, assume the standard deviation of clock jitter is 0.02 UI and T_s is about 0.16 UI. In ideal situation, the data jitter of data stream encoded by 4-PAM symmetric code and un-coded data stream are $0 / \pm 25\%$ T_s . BER are about $10^{-15} / 5 \times 10^{-13}$ for encoded / un-coded data respectively.

The 4-PAM symmetric code can also be used in 2-PAM system (binary) and preserve all useful characteristics of the 8B/10B code. Fig. A.10 displays the frequency responses of random binary data stream encoded by the 4-PAM symmetric code and the 8B/10B code, and demonstrates that the 4-PAM symmetric code, like the 8B/10B code, has zero spectral power at zero frequency. Therefore, the 4-PAM



(a)



(b)

Fig. A.9 Measurement results [96] of (a) all types of transitions and (b) only maximum-swing type1 transitions of the differential 4-PAM signal.

symmetric code is DC-balanced. Although the 4-PAM symmetric code has frame synchronization, the frequency response of the 4-PAM symmetric code does not exhibit any strong tones. The frequency response of the 4-PAM symmetric code is similar to that of the 8B/10B code.

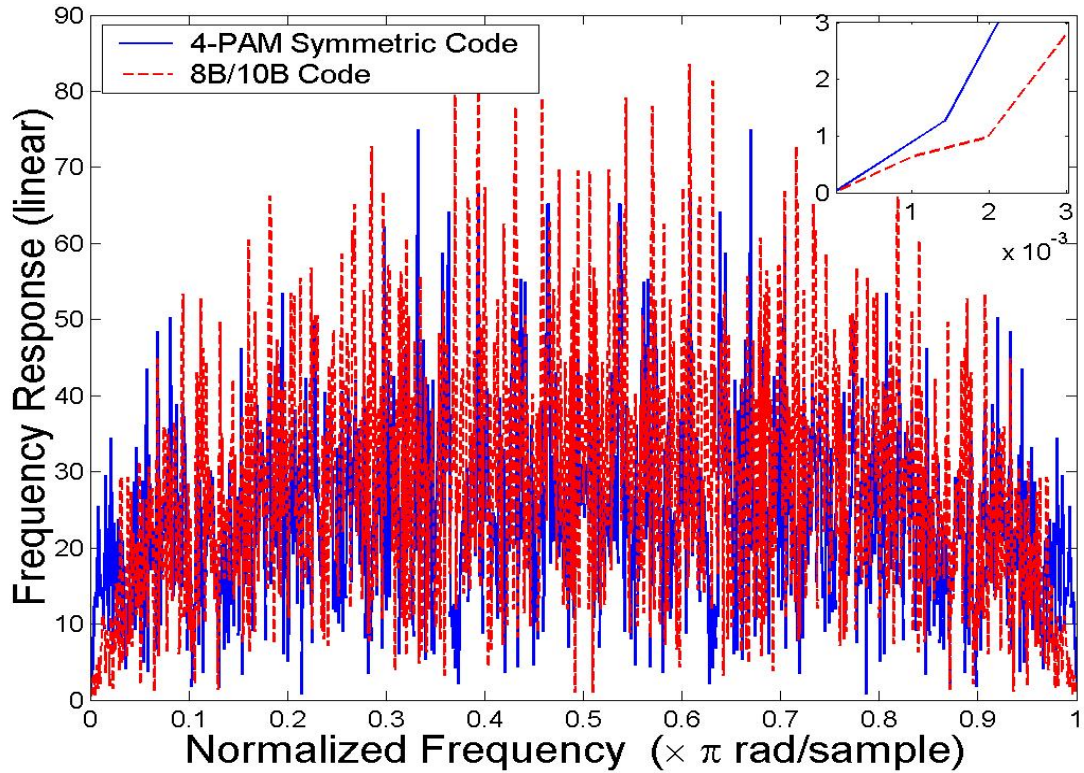


Fig. A.10 Frequency responses of the 4-PAM symmetric code and the 8B/10B code.



A.4 Conclusions

This appendix presents a novel low-jitter transmission code, 4-PAM symmetric code, for 4-PAM signaling in serial links. The proposed method preserves all the good characteristics of 8B/10B code, such as DC-balanced, finite run-length, error detection and special characters. Furthermore, 4-PAM symmetric code guarantees at least one type 1 transition every ten transmitted symbols, and thus can decrease the jitter of transition time by $\pm 25\%T_s$ in 4-PAM system. Using a UMC $0.18\mu\text{m}$ standard cell, 13.1 Gbps / 11.3 Gbps encoding / decoding can be achieved with about 2506 / 2915 gates.

References

- [1] *IEEE standard for local and metropolitan area networks part 15.1: wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)*, IEEE Std. 802.15.1, 2002.
- [2] *IEEE standard for local and metropolitan area networks part 15.3: wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (HR-WPANs)*, IEEE Std. 802.15.3, Sept. 2003.
- [3] *IEEE standard for local and metropolitan area networks part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)*, IEEE Std. 802.15.4, Oct. 2003.
- [4] “IEEE 802.15.1 WPAN press kit,” Jan. 2001. Available: <http://www.ieee802.org>
- [5] *IEEE standard for local and metropolitan area networks part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, IEEE Std. 802.11-2007, July 2007.
- [6] *IEEE standard for local and metropolitan area networks part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: enhancements for higher throughput*, IEEE Std. P802.11n/D9.0, May 2009.
- [7] *IEEE standard for local and metropolitan area networks part 16: air interface for fixed broadband wireless access systems*, IEEE Std. 802.16-2004, Oct. 2004.
- [8] *IEEE standard for local and metropolitan area networks part 16: air interface for fixed broadband wireless access systems amendment 2: physical and*

- medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum*, IEEE Std. 802.16e-2005, Feb. 2006.
- [9] *IEEE standard for local and metropolitan area networks part 20: air interface for mobile broadband wireless access systems supporting vehicular mobility-physical and media access control layer specification*, IEEE Std. 802.20-2008, June 2008.
- [10] *IEEE standard for local and metropolitan area networks part 16: air interface for fixed broadband wireless access systems*, IEEE Std. 802.16-2001, Dec. 2001.
- [11] *IEEE standard for local and metropolitan area networks part 16: air interface for fixed broadband wireless access systems amendment 2: medium access control modifications and additional physical layer specifications for 2-11 GHz*, IEEE Std. 802.16a-2003, Jan. 2003.
- [12] “What is Europe’s ETSI HMAN,” Feb. 2008. Available: <http://www.wimaxforum.org>
- [13] J. G. Andrews, A. Ghosh, and R. Muhamed, *Fundamental of WiMAX*, 1st ed. Prentice Hill, 2007.
- [14] “Mobile WiMAX-part 1: a technical overview and performance evaluation,” WiMAX Forum, Feb. 2006. Available: <http://www.wimaxforum.org>
- [15] “The 802.16 WirelessMAN MAC: it's done, but what is it,” IEEE 802.16-01/58r1, Nov. 2001. Available: <http://wirelessman.org>
- [16] S. M. Alamouti, “A simple transmit diversity technique for wireless communications,” *IEEE J. Select. Areas Commun.*, vol. 16, no. 8, pp. 1451-1458, Oct. 1998.
- [17] V. Tarokh, N. Seshadri, and A. R. Calderbank, “Space-time codes for high data rate wireless communication: performance criterion and code construction,” *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp. 744-765, Mar. 1998.
- [18] Y. Li, N. Seshadri, and S. Ariyavisitakul, “Channel estimation for OFDM systems with transmitter diversity in mobile wireless channels,” *IEEE J. Select. Areas Commun.*, vol. 17, no. 3, pp. 461-471, Mar. 1999.

- [19] M. L. Ku and C. C. Huang, "A complementary codes pilot-based transmitter diversity technique for OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 5, no. 3, pp. 504-508, Mar. 2006.
- [20] B. Song, L. Gui, and W. Zhang, "Comb type pilot aided channel estimation in OFDM systems with transmit diversity," *IEEE Trans. Broadcasting*, vol. 52, no. 1, pp. 50-56, Mar. 2006.
- [21] P. Garg, R. K. Mallik, and H. A. Gupta, "Performance analysis of space-time coding with imperfect channel estimation," in *Proc. IEEE Int. Conf. Personal Wireless Commun.*, Dec. 2002, pp. 71-75.
- [22] *Digital video broadcasting (DVB); framing structure, channel coding and modulation for digital terrestrial television*, ETSI Std. EN 300 744 V1.5.1, Nov. 2004.
- [23] H. Yaghoobi, "Scalable OFDMA physical layer in IEEE 802.16 wireless MAN," *Intel Technology Journal*, vol. 8, no. 3, pp. 200-212, Aug. 2004.
- [24] S. Ahmadi, "Introduction to mobile WiMAX radio access technology: PHY and MAC architecture," Intel Corporation Wireless Standards and Technology Report, Dec. 2006.
- [25] B. D. Van Veen and K. M. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4-24, Apr. 1988.
- [26] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: a fundamental tradeoff in multiple-antenna channels," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, pp. 1073-1096, May 2003.
- [27] B. Vucetic and J. Yuan, *Space-time coding*. John Wiley and Sons, 2003.
- [28] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1456-1467, July 1999.
- [29] K. F. Lee and D. B. Williams, "A space-time coded transmitter diversity technique for frequency selective fading channels," in *Proc. IEEE Workshop on Sensor Array and Multichannel Signal Processing*, Mar. 2000, pp. 149-152.

- [30] J. N. Lin, H. Y. Chen, T. C. Wei, and S. J. Jou, "Symbol and carrier frequency offset synchronization for IEEE802.16e," in *Proc. IEEE Int. Symp. Circuits and Syst.*, May 2008, pp. 3082-3085.
- [31] M. L. Ku and C. C. Huang, "A refined channel estimation method for STBC/OFDM systems in high-mobility wireless channels," *IEEE Trans. Wireless Comm.*, vol. 7, no. 11, pp. 4312-4320, Nov. 2008.
- [32] H. Y. Chen, M. L. Ku, S. J. Jou, and C. C. Huang, "A robust channel estimator for high-mobility STBC-OFDM systems," *IEEE Trans. Circuits and Syst. I*, vol. 57, no. 4, Apr. 2010.
- [33] L. Deneire, P. Vandenameele, L. van der Perre, B. Gyselinckx, and M. Engels, "A low-complexity ML channel estimator for OFDM," *IEEE Trans. Commun.*, vol. 51, no. 2, pp. 135-140, Feb. 2003.
- [34] M. Morelli and U. Mengali, "A comparison of pilot-aided channel estimation methods for OFDM systems," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3065-3073, Dec. 2001.
- [35] J. H. Park, M. K. Oh, and D. J. Park, "New channel estimation exploiting reliable decision-feedback symbols for OFDM Systems," in *Proc. Int. Conf. Commun.*, June 2006, pp. 3046-3051.
- [36] M. L. Ku and C. C. Huang, "A derivation on the equivalence between Newton's method and DF DFT-based method for channel estimation in OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 7, no. 10, pp. 3982-3987, Oct. 2008.
- [37] T. S. Rappaport, *Wireless communications: principles and practice*. Prentice Hall Inc., 2002.
- [38] X. Cai and G. Giannakis, "Bounding performance and suppressing inter-carrier interference in wireless mobile OFDM," *IEEE Trans. Commun.*, vol. 51, no. 12, pp. 2047-2056, Dec. 2003.
- [39] H. Hijazi and J. Ros, "Polynomial estimation of time-varying multipath gains with inter-carrier interference mitigation in OFDM systems," *IEEE Trans. Vehicular Technology*, vol. 58, no. 1, pp. 140-151, Jan. 2009.

- [40] Y. Mostofi and D. C. Cox, "ICI mitigation for pilot-aided OFDM mobile systems," *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 765-774, Mar. 2005.
- [41] S. Tomasin, A. Gorokhov, H. Yang, and J. P. Linnartz, "Iterative interference cancellation and channel estimation for mobile OFDM," *IEEE Trans. Wireless Commun.*, vol. 4, no. 1, pp. 238-245, Jan. 2005.
- [42] M. Speth, F. Classen, and H. Meyr, "Frame synchronization of OFDM Systems in frequency selective fading channels," in *Proc. IEEE Vehicular Technology Conf.*, May 1997, pp.1807-1811.
- [43] D. Landstrom, S. K. Wilson, J. J. van de Beek, P. Odling, and P. O. Borjesson, "Symbol time offset estimation in coherent OFDM systems," *IEEE Trans. Commun.*, vol. 1, pp. 500-505, 1999.
- [44] J. J. van de Beek, M. Sandell, M. Isaksson, and P. O. Borjesson, "Low complex frame synchronization in OFDM systems," in *Proc. IEEE Int. Conf. Universal Personal Commun.*, Nov. 1995, pp. 982-986.
- [45] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Trans. Commun.*, vol. 42, no. 10, pp. 2908-2914, Oct. 1994.
- [46] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1613-1621, Dec. 1997.
- [47] R. V. Nee, *OFDM for wireless multimedia communication*, 1st ed. Artech House, 2000.
- [48] J. Laiho, A. Wacker, and T. Novosad, *Radio network planning and optimisation for UMTS*. New York: Wiley, 2002.
- [49] W. C. Jakes, *Microwave mobile communications*. New York: Wiley, 1974.
- [50] *Transmission system for handheld terminals (DVB-H)*, ETSI Std. EN 302 304 V1.1.1, Nov. 2004.
- [51] S. Werner, M. Enescu, and V. Koivunen, "Low-complexity time-domain channel estimators for mobile wireless OFDM systems," in *Proc. IEEE*

- Workshop on Signal Processing Systems Design and Implementation*, Nov. 2005, pp. 245-250.
- [52] O. Edfors, M. Sandell, J. J. van de Beek, S. K. Wilson, and P. O. Borjesson, "Analysis of DFT-based channel estimators for OFDM," *Wireless Personal Commun.*, vol. 12, Jan. 2000.
- [53] M. Speth, S. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for OFDM-based broadband transmission part II: a case study," *IEEE Trans. Commun.*, vol. 49, no. 4, pp. 571-578, Apr. 2001.
- [54] T. A. Lin and C. Y. Lee, "Predictive equalizer design for DVB-T system," in *Proc. IEEE Int. Symp. Circuits and Syst.*, May 2005, vol. 2, pp. 940-943.
- [55] J. Rinne and M. Renfors, "Pilot spacing in orthogonal frequency division multiplexing systems on practical channels," *IEEE Trans. Consumer Electronics*, vol. 42, no. 4, Nov. 1996.
- [56] Y. Zhao and A. Huang, "A novel channel estimation method for OFDM mobile communication systems based on pilot signals and transform-domain processing," in *Proc. IEEE 47th Vehicular Technology Conf.*, May 1997, pp. 2089-2093.
- [57] J. J. van de Beek, O. Edfors, M. Sandell, S. K. Wilson, and P. O. Borjesson, "On channel estimation in OFDM systems," in *Proc. IEEE 45th Vehicular Technology Conf.*, Jul. 1995, pp. 815-819.
- [58] Edfors, M. Sandell, J. J. van de Beek, S. K. Wilson, and P. O. Borjesson, "OFDM channel estimation by singular value decomposition," in *Proc. IEEE 46th Vehicular Technology Conf.*, Apr. 1996, pp. 923-927.
- [59] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Trans. Broadcast.*, vol. 48, no.3, pp. 223-229, Sept. 2002.
- [60] T. Mueller, K. Brueninghaus, and H. Rohling, "Performance of coherent OFDMCDMA for broadband mobile communications," *Wireless Personal Commun.*, Kluwer, Netherlands, vol. 2, no. 4, pp. 295-305, 1996.

- [61] "Digital video broadcasting (DVB); implementation guidelines for DVB terrestrial services; Transmission Aspects", ETSI technical report TR 100.190 v1.1.1, Dec. 1997.
- [62] M. H. Hsieh and C. H. Wei, "Channel estimation for OFDM systems based on comb-type pilot arrangement in frequency selective fading channels," *IEEE Trans. Consumer Electronics*, vol. 44, no. 1, pp. 217-225, Feb. 1998.
- [63] H. V. Sorensen, "Efficient computation of the DFT with only a subset," *IEEE Trans. Signal Processing*, vol. 41, no. 3, pp. 1184-1200, Mar. 1993.
- [64] C. M. Chen and Y. H. Huang, "Partial cached-FFT algorithm for OFDMA communications," *IEEE TENCON Region 10 Conf.*, Oct, 2007, pp.1-4.
- [65] L. Erup, F. M. Gardner, and R. A. Harris, "Interpolation in digital modems-part II: implementation and performance," *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 998-1008, Jun. 1993.
- [66] P. Hoeher, S. Kaiser, and P. Robertson, "Two dimensional pilot-symbol-aided channel estimation by Wiener filtering," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, 1997, pp. 1845-1848.
- [67] H. Y. Chen and S. J. Jou, "Novel programmable FIR filter based on higher radix recoding for low-power and high-performance applications," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, Apr. 2007, pp. III-1473-III-1476.
- [68] C. N. Lyu and D. W. Matula, "Redundant binary booth recoding," in *Proc. 12th IEEE Int. Symp. Computer Arithmetic*, Jul. 1995, pp. 50-57.
- [69] P. E. Madrid, B. Millar, and E. E. Swartzlander, "Modified booth algorithm for high radix multiplication," in *Proc. IEEE Computer Design Conf.*, Oct. 1992, pp. 118-121.
- [70] P. M. Seidel, L. D. McFearin, and D. W. Matula, "Secondary radix recoding for higher radix multipliers," *IEEE Trans. Computers*, vol. 54, no. 2, pp. 111-123, Feb. 2005.
- [71] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, "Computation sharing programmable FIR filter for low-power and high-performance applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 2, pp. 348-357, Feb. 2004.

- [72] W. C. Liu, T. C. Wei, and S. J. Jou, "Blind mode/GI detection and coarse symbol synchronization for DVB-T/H," in *Proc. IEEE Int. Symp. Circuits and Syst.*, May 2007, pp. 2092-2095.
- [73] K. Hwang, *Computer arithmetic, principles, architecture, and design*. New York: Wiley, 1979.
- [74] S. Olariu, M. C. Pinotti, and S. Q. Zheng, "How to sort N items using a sorting network of fixed I/O size," *IEEE Trans. Parallel and Distributed Syst.*, vol. 10, no. 5, pp. 487-499, Mar. 1999.
- [75] S. Olariu, M. C. Pinotti, and S. Q. Zheng, "An optimal hardware- algorithm for sorting using a fixed-size parallel sorting device," *IEEE Trans. Computers.*, vol. 49, no. 12, pp. 1310-1324, Dec. 2000.
- [76] C. Y. Huang, G. J. Yu, and B. D. Liu, "A hardware design approach for merge-sorting network," in *Proc. IEEE Int. Symp. Circuits and Syst.*, May 2001, vol. 4, pp. 534-537.
- [77] K. E. Batcher, "On bitonic sorting networks," in *Proc. Int'l Conf. Parallel Processing*, 1990, pp. 376-378.
- [78] L. Horvath, I. B. Dhaou, H. Tenhunen, and J. Isoaho "A novel, high-speed, reconfigurable demapper-symbol deinterleaver architecture for DVB-T," in *Proc. IEEE Int. Symp. Circuits and Syst.*, June 1999, vol. 4, pp. 382-385.
- [79] H. S. Hu, H. Y. Chen, and S. J. Jou, "Novel high-throughput FFT processor for DFT-based channel estimation in IEEE 802.16e," in *Proc. IEEE Int. Symp. VLSI Design, Automation, and Test*, Apr. 2009, pp. 150-153.
- [80] H. Y. Lee and I. C. Park, "Balanced binary-tree decomposition for area-efficient pipelined FFT processing," *IEEE Trans. Circuits and Syst. I*, vol. 54, no. 4, pp. 889-900, Apr. 2007.
- [81] Y. W. Lin and C. Y. Lee, "Design of an FFT/IFFT processor for MIMO OFDM systems," *IEEE Trans. Circuits and Syst. I*, vol.54, no. 4, pp. 807-815, Apr. 2007.
- [82] M. Li, D. Novo, B. Bougard, L. van der Perre, and F. Catthoor, "Generic multi-phase software-pipelined partial-FFT on instruction-level-parallel

- architectures and SDR baseband applications,” in *Proc. Design Automation and Test in Europe*, Mar. 2008, pp. 598-603.
- [83] Y. Chen, Y. C. Tsao, Y. W. Lin, C. H. Lin, and C. Y. Lee, “An indexed-scaling pipelined FFT processor for OFDM-based WPAN applications,” *IEEE Trans. Circuits and Syst. II*, vol.55, no. 2, pp. 146-150, Feb. 2008.
- [84] S. Haene, A. Burg, N. Felber, and W. Fichtner, “OFDM channel estimation algorithm and ASIC implementation,” *IEEE 4th European Conf. on Circuits and Syst. for Commun.*, Jul. 2008, pp. 270-275.
- [85] *Fiber Channel Methodology for Jitter and Signal Quality Specification -MJSQ*, INCITS Std. T11.2/Project 1316 DT/Rev 14, June 2004.
- [86] A. X. Widmer and P. A. Franaszek, “A DC-Balanced, partitioned-block, 8B/10B transmission code,” *IBM J. Research and Development*, vol. 27, no. 5, pp.440-451, Sept. 1983.
- [87] *IEEE standard for a high-performance serial bus*, IEEE Std. 1394-2008, Oct. 2008.
- [88] *Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*, IEEE Std. 802.3-2008, 2008.
- [89] *Media access control (MAC) parameters, physical layer, repeater and management parameters for 10 Gbps operation*, IEEE Std. 802.3ae, 2002.
- [90] P. S. Neelakanta, *A textbook on ATM telecommunications, principles and implementation*. CRC Press, 2000.
- [91] G. P. Agrawal, *Fiber-optic communication systems*. New York: John Wiley & Sons, 2002.
- [92] C. K. K. Yang and K. L. J. Wong, “Analysis of timing recovery for multi-Gbps PAM transceivers,” in *Proc. IEEE Custom Integrated Circuits Conf.*, Sept. 2003, pp. 67-72.
- [93] H. Y. Chen, C. H. Lin, and S. J. Jou, “DC-balanced low-jitter transmission code for 4-PAM signaling,” *IEEE Trans. Circuits and Syst. II*, vol. 53, no. 9, pp. 827-831, Oct. 2006.

- [94] C. K. K. Yang, F. R. Ramin, and M. A. Horowitz, "A 0.5- μm CMOS 4.0-Gbit/s serial link transceiver with data recovery using oversampling", *IEEE J. Solid-State Circuits*, vol. 33, no.5, pp. 713-722, May 1998.
- [95] J. T. Stonick, G. Y. Wei, J. L. Sonntag, and D. K. Weinlader, "An adaptive PAM-4 5-Gb/s backplane transceiver in 0.25- μm CMOS," *IEEE J. Solid-State Circuits*, vol. 38, no.3, pp. 436-443, Mar. 2003.
- [96] C. H. Lin, C. H., C. N. Chen, and S. J. Jou, "Multi-gigabit pre-emphasis design and analysis for serial link," *IEICE Trans. Electronics*, vol. e88-c, no. 10, pp. 2009-2019, Oct. 2005.



Vita

Hsiao-Yun Chen was born in Tainan, Taiwan. She received the B.S. degree in electronics engineering from Feng Chia University, Taichung, Taiwan, in 2002 and the M.S. degree in electrical engineering from National Central University, Chung-Li, Taiwan, in 2004. She received the Ph.D. degree in the Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 2009. Her research interests include baseband signal processing, integrated circuit and system designs for wireless and mobile communications.

