國立交通大學

電子工程學系

碩士論文

數位電視廣播之符號邊界偵測和佈散領航碼同步設計

Design of Symbol Boundary Detection and Scattered Pilot Synchronization for DVB-T/H

指導教授：周世傑 博士

研究生：劉瑋昌

中 華 民 國 九 十 五 年 七 月

數位電視廣播之符號邊界偵測和佈散領航碼同步設計

Design of Symbol Boundary Detection and Scattered Pilot Synchronization for DVB-T/H

研究生：劉瑋昌　　Student：Wei-Chang Liu

指導教授：周世傑 博士　　Advisor：Dr. Shyh-Jye Jou

國立交通大學
電子工程學系

碩士論文

A Thesis
Submitted to Institute of Electronics Engineering
College of Electrical Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
In
Electronics Engineering

July 2005

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

數位電視廣播之符號邊界偵測和佈散領航碼同步設計

研究生：劉瑋昌　　指導教授：周世傑 博士

國立交通大學電子工程學系

摘要

在地面數位電視廣播和手持式數位電視廣播系統中，盲目的傳輸的符號模式、保護區間長度和符號邊界偵測佔了很重要的角色。同時快速佈散領航碼同步也必須被用來增進系統同步的時間。這篇論文提出了一個使用單一硬體、無除法的地面/手持式數位電視廣播盲目傳輸符號模式、保護區間長度和符號邊界偵測合併架構。藉著使用龍捲風式記憶體存取法，本架構節省了 33%的記憶體面積和 41.6%的偵測時間。同時也提出一個兩階的快速佈散領航碼同步方法以增加快速佈散領航碼同步演算法的可靠性。藉著預先存入通道估測所需之佈散領航碼，整體的快速佈散領航碼同步時間可以被縮減 97.06%到 100%。最後，一個三階的解映射和無除法通道補償器合併架構也被展現在此論文。

Design of Symbol Boundary Detection and Scattered Pilot Synchronization for DVB-T/H

Student：Wei-Chang Liu Advisor：Dr. Shyh-Jye Jou

Institute of Electronics Engineering

National Chiao Tung University

Abstract

In DVB-T/H broadcasting system, blind mode, Guard Interval (GI) length detection and coarse symbol synchronization plays important roles to estimate the symbol parameters before TPS is decoded. Also fast scattered pilot synchronization is necessary to reduce the timing of synchronization. In this thesis, a single hardware and division-free architecture modified from Normalized-Maximum-Correlation (NMC) architecture for DVB-T/H mode/GI length detection and coarse symbol synchronization is proposed. By adopting the twister memory access method and a sequential mode detection scheme, the architecture reduces 33% memory cost and at most 41.6% detection latency. A two-stage fast scattered pilot synchronization scheme is also proposed to increase the reliability of scattered pilot synchronization. A channel estimation pilots pre-filling skill is added to the two-stage fast scattered pilot synchronization scheme. Thus the scattered pilot synchronization timing is reduced by at best 100% to 97.06%. Furthermore, a three-stage demapping embedded with division-free frequency domain equalizer (FEQ) scheme is also presented.

誌謝

首先必須感謝周世傑老師，在我大一、二的時候啟蒙了我進入電子組這個領域。在研究所的兩年生涯中，給與我許多珍貴的知識和鼓勵。這兩年來，老師不僅在研究上使我受益良多，在為人處事上、態度上、研究的精神上有給予我許多難得的寶貴意見。

再者感謝劉建男老師和李進福老師，培養了我這方面的才能。對我的諄諄教誨令我永遠銘記在心，使我能夠在這裡繼續努力。

又君、儒遠、俊豪，你們不止是大學時代的戰友，同時也是實驗室令人尊敬的學長姐，給予我許多有用的建議，也陪伴我一路成長至今。

實驗室的伙伴們，偷偷龍、丸子、momo、老大、儷蓉、小朱、誌華、一哥、Nirvana、阿賢、奕瑋、志雄、spice、俊男、阿樸、俊誼、晉欽和小趙。與你們在一起的時間充滿著歡樂和汗水。

Vinson、文良、黃 yaya、昭龍、飯團和大師兄。這些自大學時代就與我一起互相勉勵的伙伴們，沒有你們的支持我的研究所生涯也不會過得如此精彩。

研究所生涯中最重要的伙伴，庭禎、琪耀和一心。感謝你們在 DVB 研究上和我一起努力打拼，並且提供我很多有用的意見和想法。

最後要感謝我的父母和女友的支持，使我可以安心無虞的度過這光輝的兩年歲月。

Content

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview of Digital TV Broadcasting

Recently, thanks to the fast growing of VLSI technology, information technology and transmission technology, digital television (DTV) systems are going to replace traditional analogue television systems. By using digital compression, digital television systems can provide a HDTV (High Definition TV) program or four to six SDTV (Standard TV) programs in the same traditional channel bandwidth (6MHz). Digital television systems also have the ability to eliminate common analog broadcasting phenomenon such as "ghosting", "snow" and static noises in audio. Therefore, digital television systems provide a high quality image and sound for consumers.

In the last few years, many standards for Digital TV Broadcasting are proposed with different performance of digital signal transmission. Such as DVB-T [1] (Digital Video Broadcasting-Terrestrial, Europe), ATSC [5] (Advanced Television System Committer, USA), ISDB-T [6] (Integrated Services Digital Broadcasting-Terrestrial, Japan). All above were recognized by ITU (International Telecommunication Union). For the portable devices, mobile phone and mobile TV for example, handheld standard such as DVB-H [2] (Digital Video Broadcasting-Handheld) is proposed by European Telecommunication Standard Institute (ESTI). Currently, another handheld standard T-DMB [7] (Terrestrial Digital Multimedia Broadcasting) has been to put into use in a number of countries. Table 1-1 lists the comparisons of above standards.

Table 1-1 Comparisons of different broadcasting standards

Standard	DVB-T/H	ATSC	ISDB-T	T-DMB
Modulation	COFDM	8-VSB	COFDM	COFDM
Video	MPEG-2	MPEG-2	MPEG-2	H.264
Audio	ACC	AC-3	ACC	BSAC
Bandwidth	5/6/7/8MHz	6MHz	6MHz	1.5MHz

1.2 DTV System in Taiwan - DVB-T/H

ATSC has once adopted as Taiwan's DTV standard in 1998. But by the effort of CTV, PTS, FTV, TTC and CTS, DVB-T was finally adopted to be the standard in year 2001. In year 2005 DVB-H was also adopted as mobile TV standard. As a result, DVB-T/H has become a hot topic in Taiwan. In May 2005, DTV system was broadcasted in west Taiwan. The E-Taiwan proposal plans start to retrieve analog channels in 2006. In 2010, TV broadcasting will become all in digital [8] [9].

Digital Video Broadcasting (DVB) was published by Joint Technical Committee (JTC) of European Telecommunication Standard Institute (ESTI), European Committee for Electrotechnical Standardization (CENELEC) and European Broadcasting Union (EBU). DVB family includes DVB-S (Satellite) [3], DVB-C (Cable) [4], DVB-T (Terrestrial) [1] and DVB-H (Handheld) [2]. DVB-T was published in 1997 with COFDM (Coded Orthogonal Frequency Division Multiplexing) transmission technique. Now there are many countries, such as UK, Germany, Norway, Australia, South Africa, India and Taiwan use DVB-T as their DTV transmission standard.

Though DVB-T can be applied in mobile environment, the power consumption is not good enough for portable devices. As a result, DVB-H was proposed for low power design and adopted for portable devices.

1.3 Motivation

Since the government of Taiwan adopted DVB-T as DTV standard and decides to revoke all analog channels in 2010, DVB-T products have become a very hot research topic now. Some products on market are based on DSP and others on ASIC [10]~[13]. For the purpose to be compatible with DVB-H, a low power version must be proposed. Since DVB-T/H receiver design is very important for portable devices, the ASIC design approach is preferred because it has lower power consumption than DSP based approach.

As many recently communication system such as ADSL (Asymmetric Digital Subscriber Line), DAB (Digital Audio Broadcasting) and WLAN (Wireless Local Area Network), DVB-T/H also uses OFDM technology. The benefits of OFDM includes high spectrum efficient by orthogonal technology, ability to against multipath interference (Fading) by inserting guard interval and cyclic prefix and can be transmitted in single frequency network (SFN). However time-variant channel will cause a loss in sub-carrier orthogonality and system performance due to the sub-carriers are spaced too close.

In broadcasting environment, DVB-T/H must solve the following problems. Transmission mode and Guard Interval (GI) mode detection is an important step at the beginning of system. Wrong transmission or GI mode decision will cause the whole system failed as a result of sending incorrect data to Fast Fourier Transform (FFT) module and get incorrect outputs. Accurate symbol boundary detection also plays an important role in DVB-T/H system. Without accurate symbol boundary, the sub-carriers may have a phase rotation or loss their orthogonality. Since DVB-T/H suffers from multipath interference and time-variant channel, the system need to extract the inserted scattered pilots and use the scattered pilots to compensate the

sub-carriers degradation by a frequency domain equalizer (FEQ). Scattered pilots synchronization (SPS) is a way to determine the current scattered pilot mode and reduce the timing to demapping. Other synchronization schemes such as carrier frequency offset (CFO) and sampling clock offset (SCO) are referred in [14].

This thesis proposes an efficient memory shared and division-free synchronization architecture which reduces the memory storage requirement. A Maximum-Correlation (MC) and Normalized-Maximum-Correlation (NMC) compromised architecture is proposed in this thesis to detect the transmission mode and guard interval length accurately and easily. Since Power-Based (PB) scattered pilot synchronization is weak in against noise, a two-stage scattered pilot synchronization scheme with channel estimation scattered pilots pre-filling method is also proposed in this thesis to improve the reliability and timing. Finally, a three-stage demapping architecture with embedded division-free FEQ is designed and it makes the whole architecture proposed in this these is a division-free architecture.

1.4 Thesis Organization

The following of this thesis is organized as follows. In chapter 2, OFDM will be introduced briefly and DVB-T/H technology will be discussed. Chapter 3 describes the effect of incorrect transmission/GI mode and detected boundary offset. Moreover, some algorithms, simulation and architectures for mode/GI detection and coarse symbol synchronization will be compared. A blind mode/GI and boundary detection scheme will be proposed. Algorithms, performance simulation and proposed scheme for scattered pilot synchronization, channel estimation and demapping will be discussed in Chapter 4. The architecture and hardware design will be presented in chapter 5. Finally, chapter 6 will give our conclusions and future works.

Chapter 2

OFDM and DVB-T/H Technology

2.1 Concept of OFDM

In mid 60s, frequency division multiplexing (FDM) was published. In FDM, multiple signals are sent at the same time with different sub-channels. OFDM [15] is based on this idea and uses orthogonal technique to overlap the sub-channels to send more signals. By overlapping sub-channels in orthogonal frequencies, OFDM is able to carry more information than FDM with the same bandwidth. The sub-channels in FDM and OFDM are shown in Fig. 2.1 and Fig. 2.2.

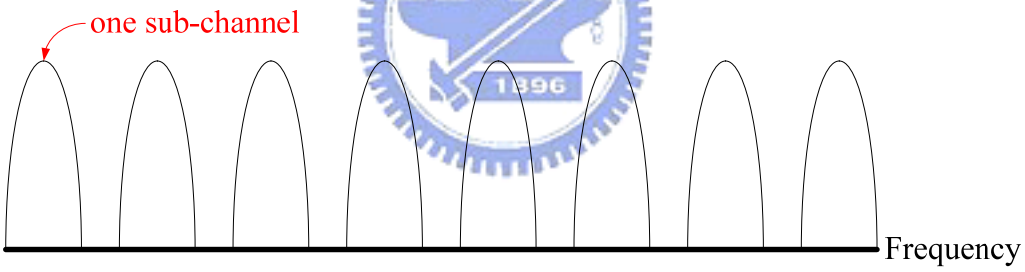


Fig. 2.1 Sub-channels in FDM modulation

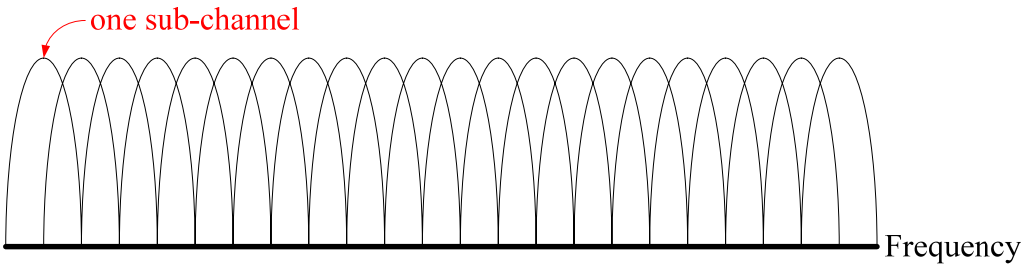


Fig. 2.2 Sub-channels in OFDM modulation

OFDM with channel coding is called COFDM (Coded OFDM). The COFDM technique adopted by DVB-T, has the ability to reduce the ICI (Inter-Carrier Interference) and ISI (Inter-Symbol Interference) by inserting guard intervals (GI), a

copy of the last period of the same OFDM symbol, between successive OFDM symbols. Since the multipath effect will cause the ISI phenomenon, to insert a guard interval is able to prevent the damage from other sub-paths. With inaccurate symbol detection, system has a high probability to find a wrong boundary which may allocate near the correct one. Owing to the difficulty of finding accurate boundary with multipath effect, system must tolerate to the inaccurate symbol boundary location. Therefore, the guard interval is filled by the copy of last period of a symbol. By using this characteristic, the detected boundary which locates before the correct one is able to decode correctly. Since the detected boundary locates before the correct one, according to OFDM mathematical function, the transmitted pattern looks like a cyclic shift without destroying the orthogonality of sub-carriers and will only cause a phase rotation. But if the detected boundary locates after the correct one, the sub-carriers will lose its orthogonality and lead to a wrong decoded answer. The received symbols with multipath effect are shown in Fig. 2.3.

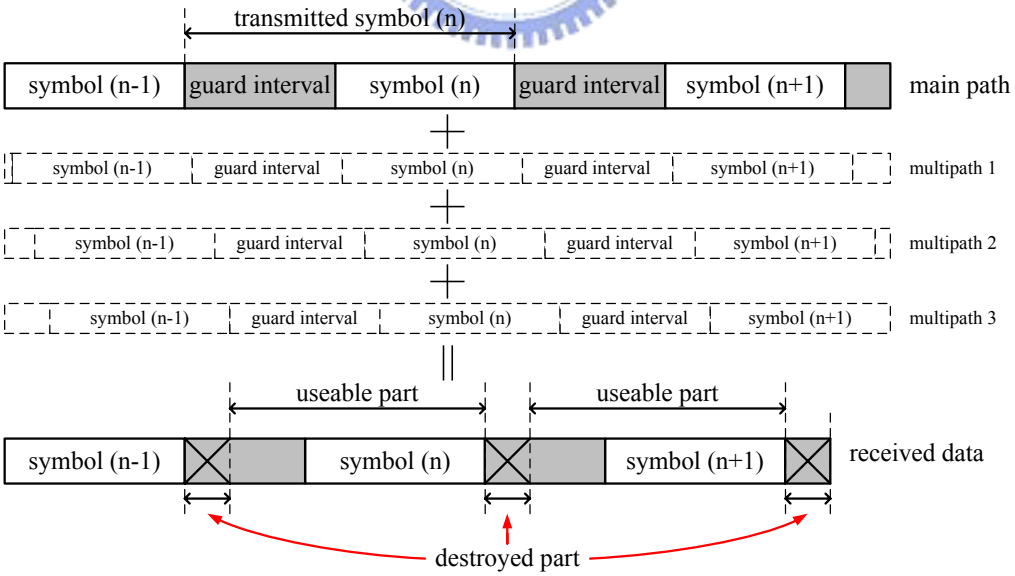


Fig. 2.3 Received OFDM symbols with multipath effect

2.2 DVB-T/H Technology

2.2.1 MPEG-2 Source Coding and Multiplexing

DVB-T is a broadcasting system based on OFDM modulation technology. Generally, this system can be divided into two parts, transmitter and receiver. In Fig. 2.4, a DVB-T transmission system block diagram is presented. MPEG-2 source coding and multiplexing multiplexes video, audio and data into an MPEG-2 program stream.

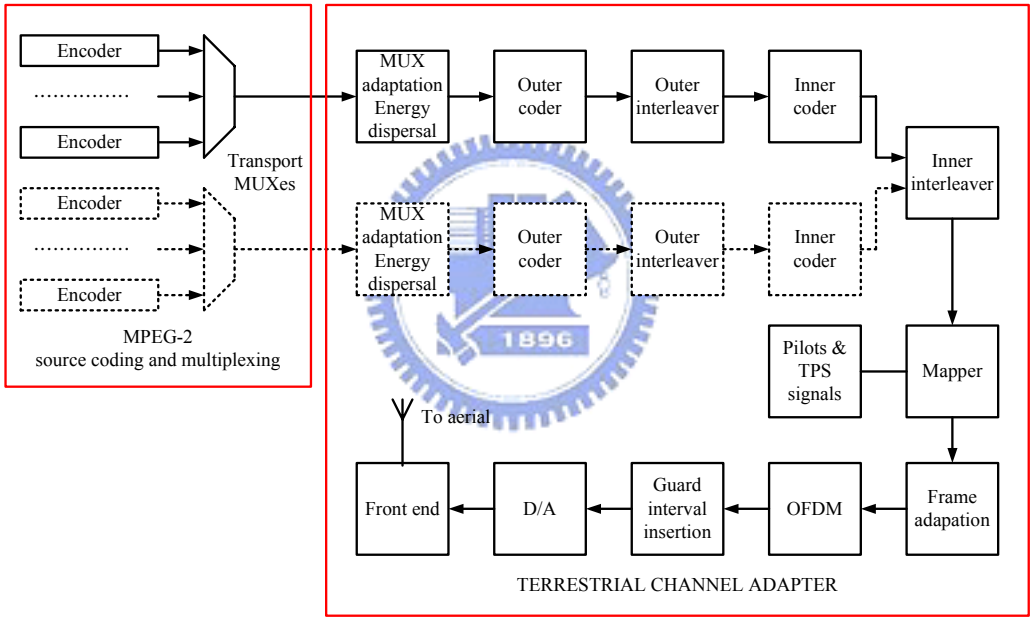


Fig. 2.4 Functional block diagram of DVB-T transmission system

2.2.2 Channel Coding

Channel coding includes randomizer, outer coder, outer interleaver, inner coder and inner interleaver. The MPEG-2 transport stream will be separated into high-priority (real line in Fig. 2.4) and low-priority (dotted line in Fig. 2.4). Therefore, in a small size monitor or weak signal environment, the receiver can switch a HDTV program into a SDTV program.

The MPEG-2 transport stream is organized as fixed length packets (118 bytes) and decorrelated by the block “MUX adaptation and energy dispersal”. Considering about the MPEG-2 Transport Stream have a probability to be a long 1 or 0 sequences, there will be some problem in synchronization. Doing Exclusive-OR operation with PRBS sequences can efficiently reach energy dispersal and randomize the MPEG-2 Transport Stream sequences. Fig. 2.5 illustrates the energy dispersal schematic diagram.

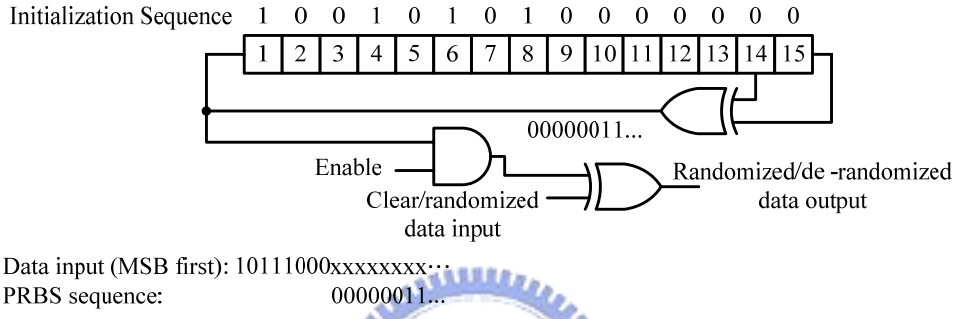


Fig. 2.5 Scrambler/descrambler schematic diagram

Outer coder uses a nonbinary block code, Reed-Solomon RS (240,188, t=8) shorten code, which have an ability to correct up to eight errors, with generator polynomial as Eqn. (2.1) is the usually used coding scheme recently.

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1 \qquad (2.1)$$

A convolutional byte-wise interleaving with depth I=12 is applied to make the transmitted data sequence being rugged to long sequences of errors. While transmission in air, a suddenly short period interference may occur because of many know or unknown reasons. Though RS have an ability to correct no more than 8 errors, the interference above is possibly lead to more than 8 errors in a RS package and unable to correct. At this time the outer interleaver opposites to this situation by interleaving the data in a symbol to other symbols. Fig. 2.6 is the conceptual diagram of the outer interleaver and deinterleaver.

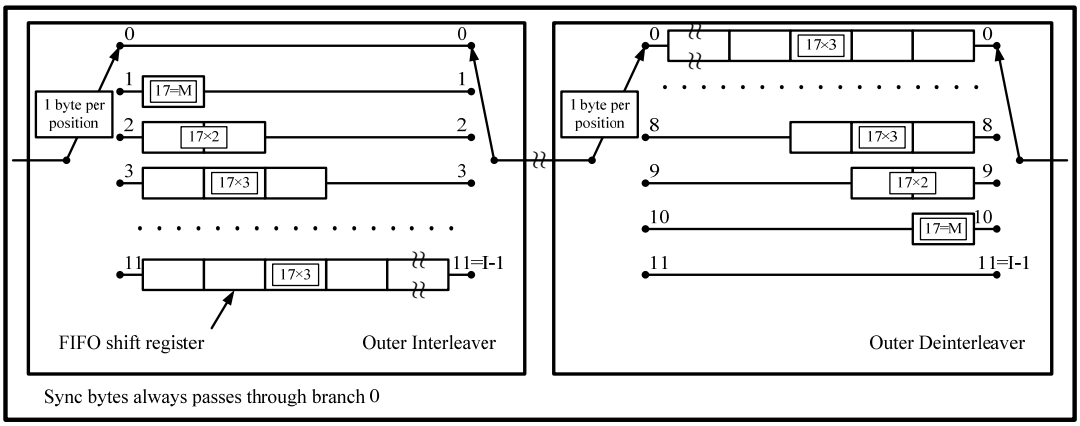


Fig. 2.6 Conceptual diagram of the outer interleaver and deinterleaver

For the purpose to have better Bit-Error-Rate (BER), a punctured convolutional encoder is cascaded with five valid coding rates: 1/2, 2/3, 3/4, 5/6, and 7/8. Higher coding rate means lower correction ability. Fig. 2.7 is an example of 1/2 coding rate which means one of two bits is useful.

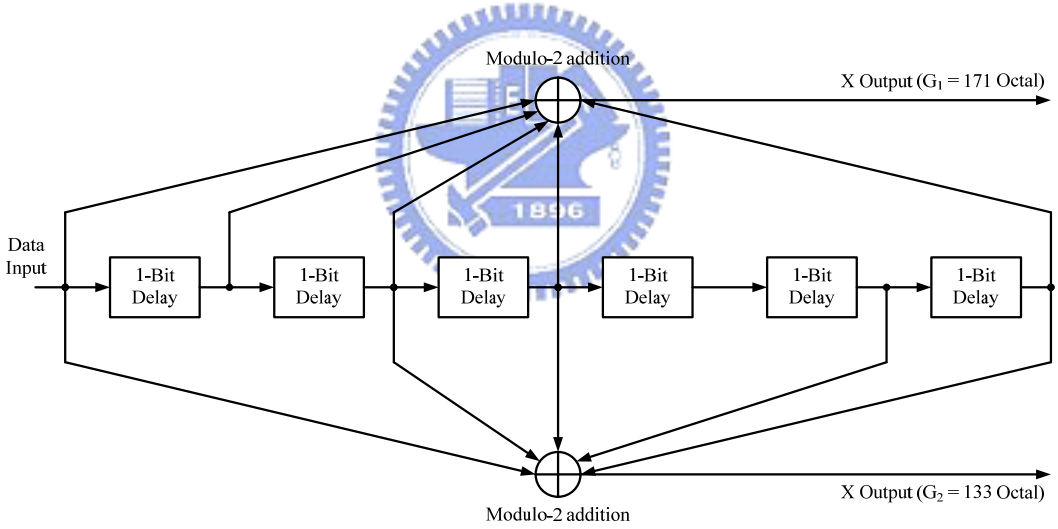


Fig. 2.7 The mother convolution code of rate 1/2

Finally, a block based bit-wise inner interleaver is used to against the Viterbi output burst errors.

2.2.3 Mapper & Frame

After two level channel coding, all data carriers in an OFDM symbol will be mapped onto QPSK, 16-QAM, 64-QAM, non-uniform 16-QAM or non-uniform

64-QAM by the mapper. DVB-T provides 2 transmission modes, 2K mode and 8K mode. In DVB-H a 4K transmission mode is specially provided. Each frame consists of 68 OFDM symbols and contains scattered pilot cells, continual pilot carriers and TPS carriers. Four frames constitute one super-frame.

2.2.4 Reference Signals

Reference signals can be classed as three types, scattered pilot cells, continual pilot carriers and TPS carriers. All of them are inserted in all OFDM symbols. Scattered pilot cells do not really have a fixed position in a symbol but others do. The positions scattered pilot cells locates are as Eqn. (2.2) shows.

$$\{k = K_{\min} + 3 \times (l \mod 4\} + 12p \mid p \text{ integer}, p \geq 0, k \in [K_{\min}; K_{\max}] \tag{2.2}$$

where k is the frequency index of the sub-carriers and l is the time index of the symbols. And it's easy to seem that, scattered pilot cells' location will be the same every four symbol. For different transmission mode, K_{max} is different. Eqn. (2.3) shows how to generate scattered pilot cells' value.

$$\begin{aligned} \mathrm{Re}\{c_{m,l,k}\} &= 4/3 \times 2(1/2 - w_k) \\ \mathrm{Im}\{c_{m,l,k}\} &= 0 \end{aligned} \tag{2.3}$$

where w_k is the k^{th} data generated by PRBS.

In summary, Fig. 2.8 shows the distribution of scattered pilots. According to regular fixed located characteristic, scattered pilots can be used for channel estimation. By using timing interpolation, channel response can be reduced to 1/3.

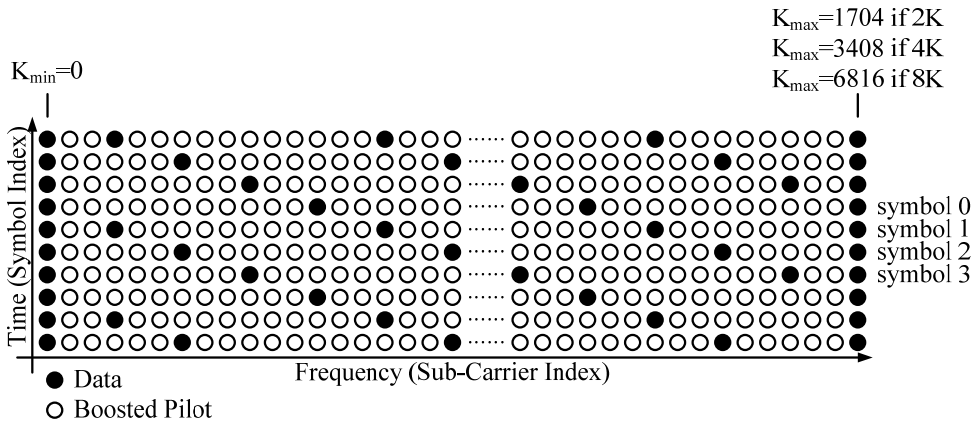


Fig. 2.8 Distribution of scattered pilots

The locations of continual pilot carriers are listed in Table 2-1. The values of continual pilots are generated like that of Eqn. (2.3). The main function of continual pilots is to track carrier frequency offset.

Table 2-1 Continual pilot carrier position

Continual Pilot Indices for Continual Pilot Carriers													
2K mode													
0	48	54	87	141	156	192	201	255	279	282	333	432	450
183	525	531	618	636	714	759	765	780	804	873	888	918	939
1137	1140	1146	1206	1269	1323	1377	1491	1683	1704				
4K mode													
0	48	54	87	141	156	192	201	255	279	282	333	432	450
183	525	531	618	636	714	759	765	780	804	873	888	918	939
1137	1140	1146	1206	1269	1323	1377	1491	1683	1704	1752	1759	1791	1845
1860	1896	1905	1959	1983	1986	2037	2136	2154	2187	2229	2235	2322	2340
2418	2463	2469	2484	2508	2577	2592	2622	2643	2646	2673	2688	2754	2805
2811	2814	2841	2844	2850	2910	2973	3027	3081	3195	3387	3408		
8K mode													
0	48	54	87	141	156	192	201	255	279	282	333	432	450
183	525	531	618	636	714	759	765	780	804	873	888	918	939
1137	1140	1146	1206	1269	1323	1377	1491	1683	1704	1752	1759	1791	1845
1860	1896	1905	1959	1983	1986	2037	2136	2154	2187	2229	2235	2322	2340
2418	2463	2469	2484	2508	2577	2592	2622	2643	2646	2673	2688	2754	2805
2811	2814	2841	2844	2850	2910	2973	3027	3081	3195	3387	3408	3456	3462
3495	3549	3564	3600	3609	3663	3687	3690	3741	3840	3858	3891	3933	3939
4026	4044	4122	4167	4173	4188	4212	4281	4296	4326	4347	4350	4377	4392
4458	4509	4515	4518	4545	4548	4554	4614	4677	4731	4785	4899	5091	5112
5160	5166	5199	5253	5268	5304	5313	5367	5391	5394	5445	5544	5562	5595
5637	5643	5730	5748	5826	5871	5877	5892	5916	5985	6000	6030	6051	6054
6081	6096	6162	6213	6219	6222	6249	6252	6258	6318	6381	6435	6489	6603
6795	6816												

TPS pilots plays an important rule in DVB-T because of there is no any handshaking before communication. TPS pilots carry 68 different messages in a frame and the messages are listed in Table 2-2.

Table 2-2 TPS signaling information and format

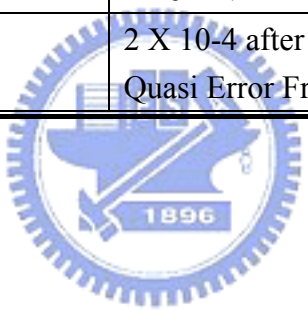
Bit Number	Purpose/Content
S0	Initialization
S1 ~ S16	Synchronization word
S17 ~ S22	Length indicator
S23 , S24	Frame number
S25 , S26	Constellation
S27, S28, S29	Hierarchy information
S30, S31, S32	Code rate, HP stream
S33, S34, S35	Code rate, LP stream
S36, S37	Guard interval
S38, S39	Transmission mode
S40 ~ S47	Cell identifier
S48 ~ S53	Reserved
S54 ~ S67	Error protection

2.2.5 DVB-H Particular

For portable devices, time slicing technology is employed to reduce power consumption. For the purpose to improve the system performance in mobile environment, forward error correction for multiprotocol encapsulated data (MPE-FEC) is adopted with powerful channeling and time interleaving. Since 8K mode has better performance in large single frequency network (SFN) but worse in against Doppler Effect and 2K mode has better performance in against Doppler Effect but not suitable for large SFN, a comprised 4k mode is proposed. Overall, the specification of DVB-T/H is listed in Table 2-3.

Table 2-3 Specification of DVB-T/H

Transmission mode	2K, 4K, 8K
Number of useful sub-carriers	1705, 3409, 6817
Number of continual pilots	45, 89, 177
Number of scattered pilots	141, 282, 564
Number of TPS pilots	17, 34, 68
Radio frequency (MHz)	45~860
Guard interval	1/4, 1/8, 1/16, 1/32
Bandwidth (MHz)	5, 6, 7, 8
Elementary period (us)	7/40, 7/48, 7/56, 7/64
Channel model	Rayleigh, Ricean
Forward error correct	Convolution code with puncturing Reed Solomon Code (204,188)
Constellation	QPSK, 16QAM, 64QAM, non-uniform 16QAM, non-uniform 16QAM
Required BER	2 X 10-4 after Viterbi decoder Quasi Error Free after Reed Solomon



Chapter 3

Symbol Synchronization Algorithms

Fig. 3.1 illustrates the block diagram of DVB-T baseband inner receiver and all synchronization processes in digital domain. The block diagram contains mode/GI & symbol boundary detection, carrier synchronization loop, sampling synchronization loop, frequency domain channel estimation/compensation and hard demapper. This thesis will design the architecture based on this block diagram and focuses on the highlighted blocks.

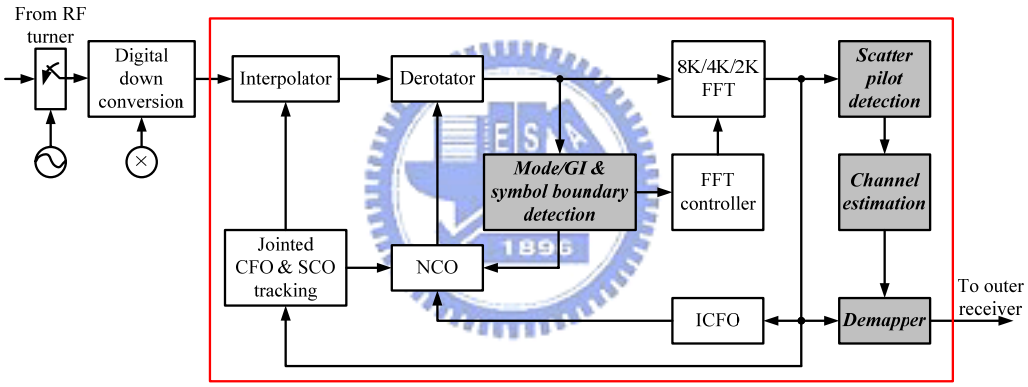


Fig. 3.1 Block diagram of DVB-T baseband inner receiver

Timing synchronization plays an important role in digital communication systems. Without accurate timing synchronization process, the systems will fail to work in the beginning or get unreliable outcome. Timing synchronization includes mode/GI detection, coarse symbol synchronization (CSS), scattered pilot synchronization (SPS), carrier frequency offset (CFO) and sampling clock offset (SCO) issues. The last two topics have been discussed in [14]. This chapter focuses on mode/GI detection and coarse symbol synchronization problems. These two jobs should be down first in receiver. The FFT window has to decide the window length and locations with the correct transmission mode, guard interval length and symbol

boundary information. Otherwise, the feedback loops will have no idea to recover CFO and SCO and so do those parts behind inner receiver.

The goal of mode/GI detection is to get precise mode/GI parameters of transmitted symbols. By using the detected transmission mode, the system is able to set a FFT window, which length equals to transmission mode. Transmission mode and guard interval parameters make the system being able to compute boundaries behind the first boundary detected by coarse symbol synchronization. After mode/GI detection, coarse symbol synchronization process will adopt the parameters from mode/GI detection to do more accurate symbol boundary detection for the purpose to reduce timing offset effects and avoid ISI effect. Moreover, the results from mode/GI detection and coarse symbol synchronization make the system have enough information to determine the FFT window locations.

3.1 Mode/GI Detection

It seems that the system can get the information of transmission mode and guard interval length from TPS pilots discussed in 2.2.4. But without these messages, how can the systems set a correct FFT window length and correct FFT window locations? That means FFT has no idea to start to work and leads to no TPS information. This phenomenon will become a vicious cycle and system will never start to work. In order to make the FFT start to work, it's necessary to do blind mode/GI detection before other processes of inner receiver.

3.1.1 Introduction to Mode/GI Detection

Mode/GI detection algorithms are usually similar to coarse symbol synchronization theorems. There are many methods to detect the mode/GI. For example, [16] proposed a blind transmission mode detection process, [17] modified a

mode/GI jointed detection process based on [16] and a two-stage mode/GI detection is discussed in [18].

a) Mode Detection

The basic idea of mode detection is to use the characteristic of the inserted guard intervals, a copy of tail in OFDM symbols. After surviving from fading channel, the guard interval part will still have a high correlation with symbol's tail. Fortunately, other parts will have low or even no correlation with guard interval. Thus, for a 2K transmission mode, the mode detector shall find the correlation of *r(n)* and *r(n-2K)*, where *r(n)* is the n^{th} received signal. For the purpose to ensure the correlation result, to accumulate the correlation results between $r(n) \times r(n-2K)$ and $r(n+64) \times r(n-2K+64)$ is necessary. Therefore, 2K+2×64, 4K+2×64 and 8K+2×64 long moving windows are required to detect the 2K/4K/8K transmission modes for DVB-T/H mode detection. Fig. 3.2 illustrates a simple diagram of the correlation results of 2K/4K/8K mode detection windows under 8K transmitted symbols. As the figure shows, only the 8K mode window is possibly located at the region of guard interval and symbol's tail at the same time and will have prominent correlation results. Eqn. (3.1) presents the computation required in Fig. 3.2 mathematically.

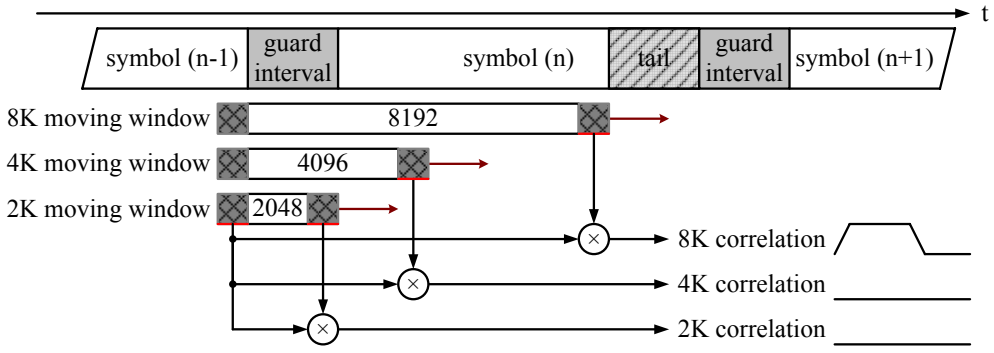


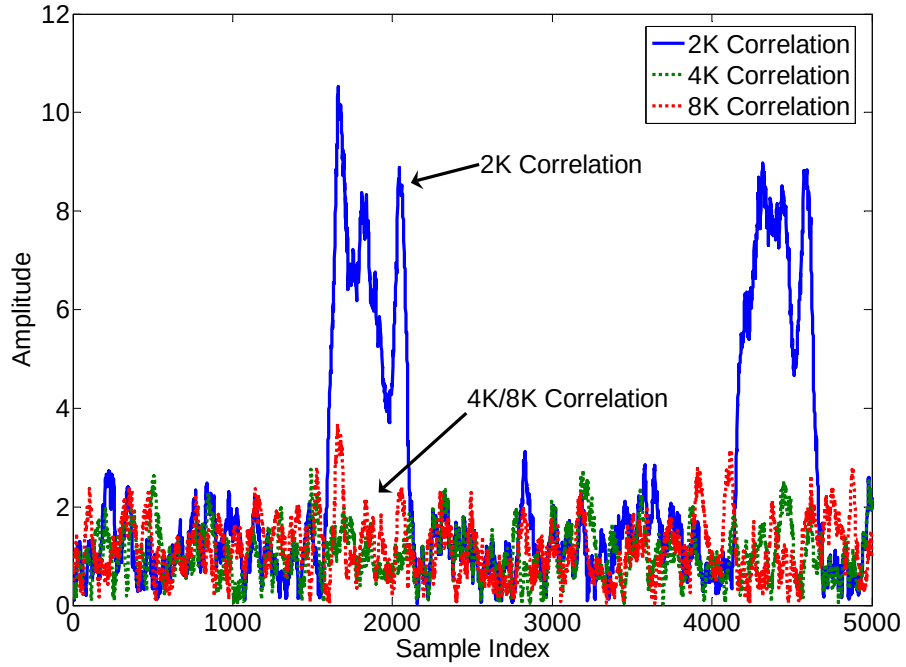
Fig. 3.2 2K/4K/8K correlation under 8K mode

$$x(n) = \left| \sum_{i=0}^{63} r^*(n-i) \times r(n-i-N) \right| \qquad (3.1)$$

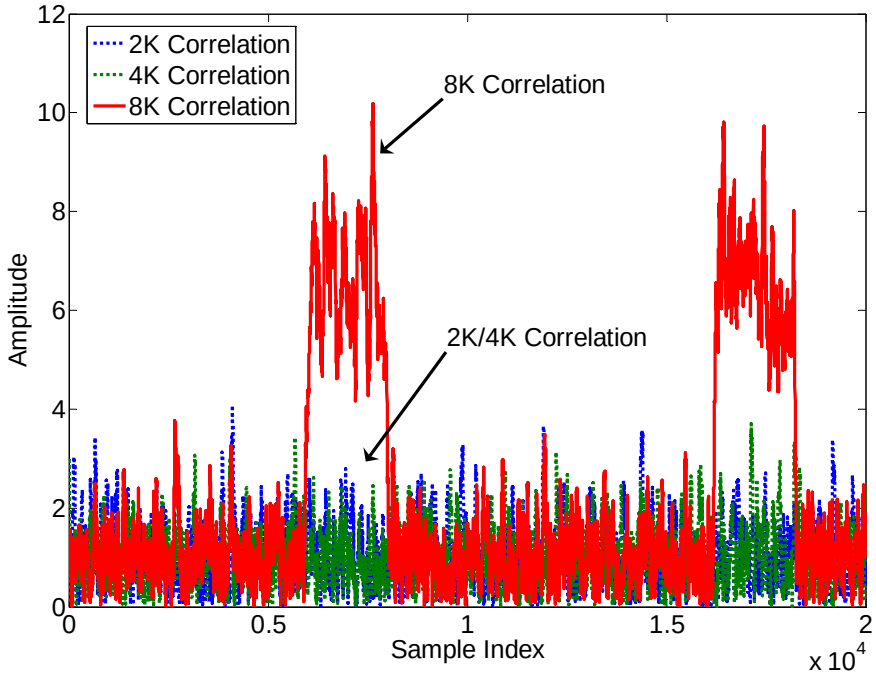
where N is the delay-line length which value will be 2K, 4K or 8K according to different transmission mode, *r(n)* is the received nth signal. Eqn. (3.2) is a modified version of Eqn. (3.1). By using an integration length equals to the minimum guard interval length of each tested transmission mode, the detector is able to have a reliable correlation result.

$$x(n) = \left| \sum_{i=0}^{\frac{N}{32}-1} r^*(n-i) \times r(n-i-N) \right| \tag{3.2}$$

Fig. 3.3 (a) and (b) show the different correlation results under different transmission modes with 1/4 guard interval, 12dB SNR, CFO=23.33 and surviving from Rayleigh channel. In Fig. 3.3 (a) only 2K correlation results have apparent plateaus and Fig. 3.3 (b) has the same situation for 8K correlation results.



(a)



(b)

Fig. 3.3 2K/4K/8K correlation results under (a) 2K (b) 8K transmission mode

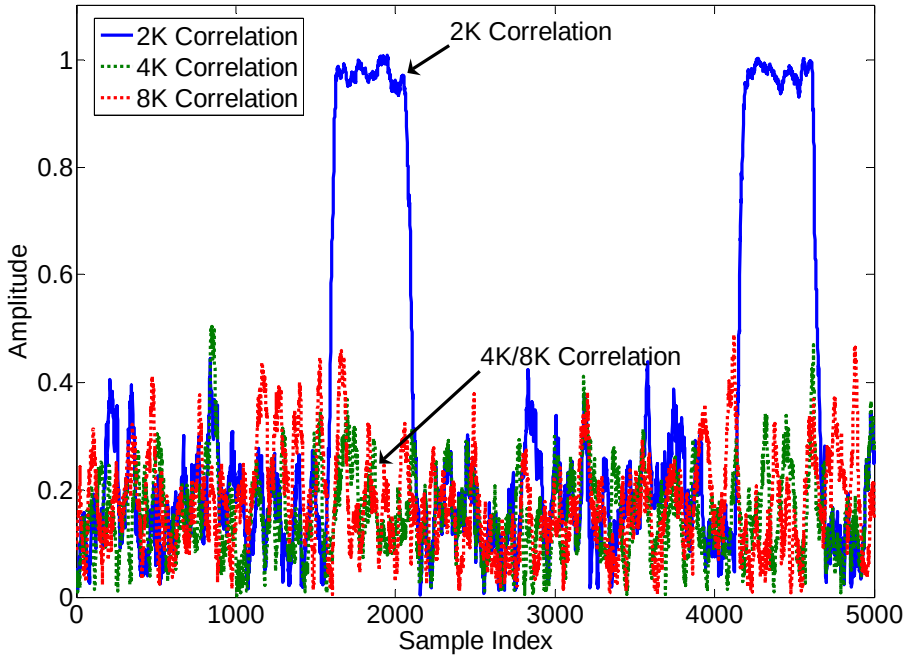
Even though Eqn. (3.1) or Eqn. (3.2) have the ability to distinguish the transmission mode, there is still some aliasing peaks occurred due to channel noise and may possibly lead to a wrong detected mode. For example, in Fig. 3.3 (a), there is an aliasing peak near sample index 3000 of 2K correlation results which is almost as high as the lowest plateau value near to sample index 2000 for 2K. Thus it is a problem to decide the peak or plateau threshold since there is no flat plateau and unitary plateau value.

To eliminate the effect from channel noises and ease the decision of the threshold, [17] used a normalized method to detect the transmission mode shown in Eqn. (3.3).

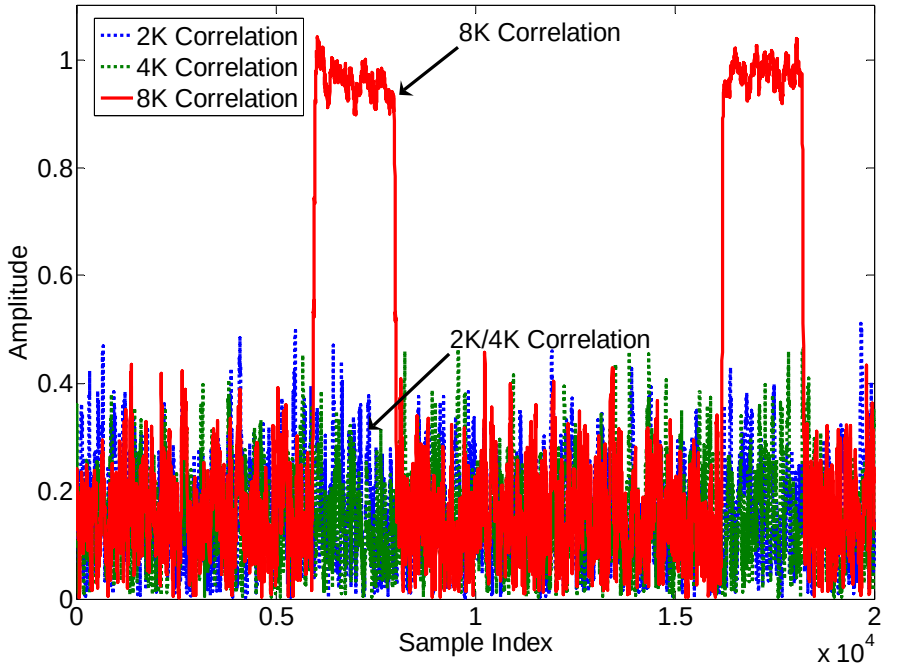
$$x(n) = \frac{\left| \sum_{i=0}^{\frac{N}{32}-1} r^*(n-i) \times r(n-i-N) \right|}{\left| \sum_{i=0}^{\frac{N}{32}-1} r^*(n-i) \times r(n-i) \right|} \tag{3.3}$$

The correlation result will be normalized to 1 theoretically by dividing the power

term. Fig. 3.4 (a) and (b) are the results of Eqn. (3.3), using the same pattern with Fig. 3.3. The plateau is ideally close to “1” and no other aliasing correlation results are higher than “0.707”. The characteristic of the normalized flat plateau can be also used to calculate the guard interval length and it will be discussed later.



(a)



(b)

Fig. 3.4 2K/4K/8K normalized correlation results under (a) 2K (b) 8K mode

b) Guard Interval Length Detection

The guard interval length will be detected after transmission mode. With an incorrect guard interval length, FFT won't get correct patterns and sub-carriers after FFT won't be the same with transmitted. Fig. 3.5 illustrates the situations of incorrect guard interval length. In Fig. 3.5 (a), a smaller guard interval length is detected and the second FFT window has a probability to get signals form the ISI destroyed region. The third FFT window gets some signals from the correct symbol, some from ISI destroyed region and others from previous symbol, this will lose the orthogonality of OFDM symbols and FFT outputs are absolutely incorrect. FFT windows in Fig. 3.5 (b) also get incorrect signals either.

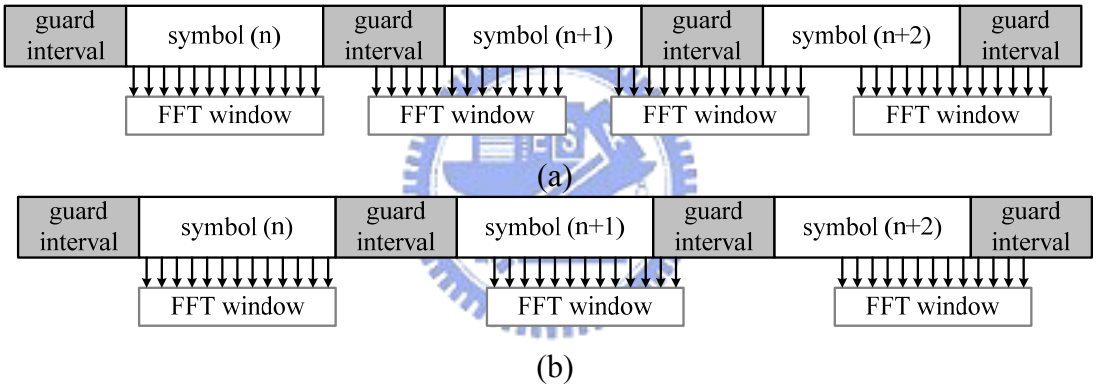


Fig. 3.5 Detected guard interval length is (a) smaller (b) larger than transmitted

[18] used the minimum guard interval length, which is 1/32 of transmission mode, and accumulate the results using different delay-line length windows, 1/32, 1/16, 1/8 and 1/4. Only the correct guard interval mode will have a maximum peak after different length accumulation. [17] adopted an simple and less delay-line method to calculate the guard interval length. This method just calculates the length of plateau period and that will approximate to "guard interval length subtracts mode/32".

3.1.2 Proposed Mode/GI Scheme

The normalized method seems very easy to realize and compute the guard

interval length, but a divider wastes large power and area. Two-stage mode/GI detection proposed in [18] needs extra delay-lines which size will be from 2K to 8K. This is another penalty. As a result, a modified method based on normalized mode/GI detection is proposed in this thesis. Since a threshold value is determined to define the plateau region, it implies that all *x(n)* which are bigger than the pre-defined threshold belongs to the plateau. Therefore, two key observations below can be found:

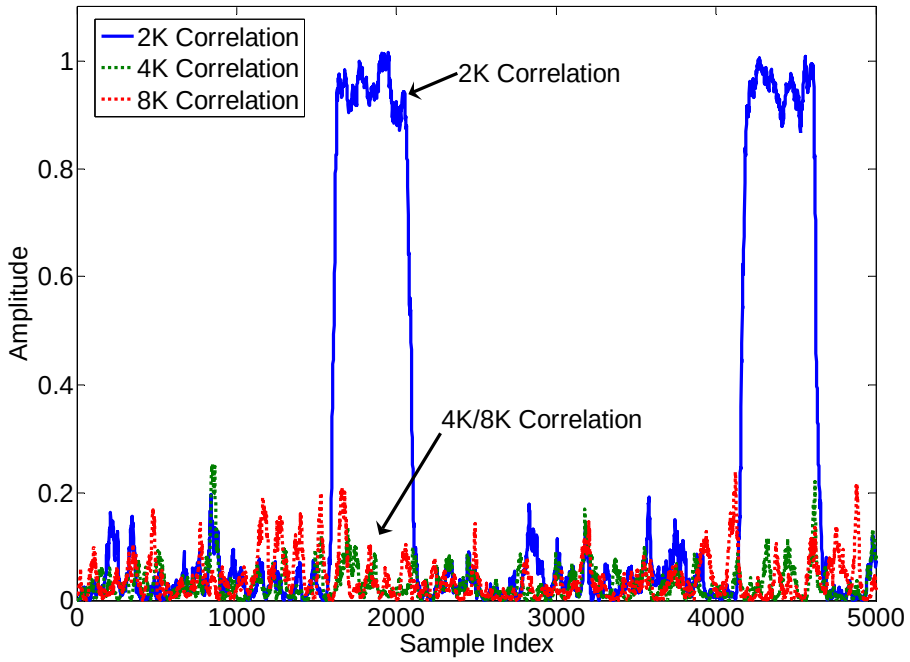
1) If the plateaus exist, the transmission mode will be the same with the tested mode.

2) The period of the plateau represents the guard interval length.

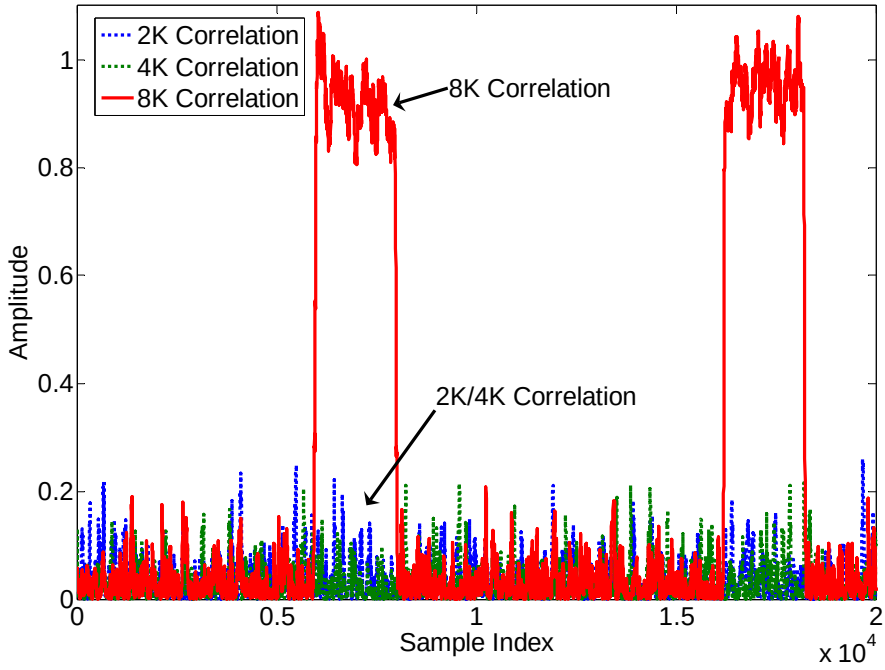
Using the two key observations above, if *x(n)* is bigger than the threshold means *x(n)* belongs to the plateau region. Now the threshold is defined as 0.707 and the derivation of using a subtractor to replace the divider is shown in Eqn. (3.4).

$$
\begin{aligned}
& x(n) \in plateau \quad if \ \frac{\left|\sum_{i=0}^{\frac{N}{32}-1} r^*(n-i)\cdot r(n-i-N)\right|}{\left|\sum_{i=0}^{\frac{N}{32}-1} r^*(n-i)\cdot r(n-i)\right|} \geq 0.707 \\
& \Rightarrow \left|\sum_{i=0}^{\frac{N}{32}-1} r^*(n-i)\cdot r(n-i-N)\right| \geq 0.707 \times \left|\sum_{i=0}^{\frac{N}{32}-1} r^*(n-i)\cdot r(n-i)\right| \\
& \Rightarrow \left|\sum_{i=0}^{\frac{N}{32}-1} r^*(n-i)\cdot r(n-i-N)\right|^2 \geq 0.707^2 \times \left|\sum_{i=0}^{\frac{N}{32}-1} r^*(n-i)\cdot r(n-i)\right|^2 \\
& \Rightarrow \left|\sum_{i=0}^{\frac{N}{32}-1} r^*(n-i)\cdot r(n-i-N)\right|^2 \geq 0.5 \times \left|\sum_{i=0}^{\frac{N}{32}-1} r^*(n-i)\cdot r(n-i)\right|^2 \\
& \Rightarrow \left|\sum_{i=0}^{\frac{N}{32}-1} r^*(n-i)\cdot r(n-i-N)\right|^2 - 0.5 \times \left|\sum_{i=0}^{\frac{N}{32}-1} r^*(n-i)\cdot r(n-i)\right|^2 \geq 0
\end{aligned}
\tag{3.4}
$$

First, move the denominator of the left term to the right term. Thus, a divider is replaced by a multiplier. Second, as the result of calculating the absolute value for complex numbers is too complicated to implement, squaring both sides of the equation is used to replace the absolute value calculating. In fact as Fig. 3.6 shows, modified from Fig. 3.4, square operation eliminates the noises. Then move the right term to left, this action makes the comparator replaced by a subtractor. Finally, the square of threshold becomes "0.5" that means only a bit shift instead of a multiplier is able to accomplish this job. Overall, the transmission mode can be tested by observing whether the "result >=0" and guard interval length can be detected by computing the period of the "result >=0".



(a)



(b)

Fig. 3.6 2K/4K/8K squared normalized correlation under (a) 2K (b) 8K mode

3.1.3 Performance Simulation

Fig. 3.7 illustrates the error rate under different threshold of the proposed mode/GI detection. The simulation environment is 1000 2K transmission mode symbols with 1/4 guard interval, 12dB SNR, 23.33 sub-carriers CFO and surviving from Rayleigh channel. As the simulation shows, the error rate of the proposed mode/GI detection method under the threshold 0.5 to 0.8 is zero. That means the pre-defined threshold value 0.707 locates at the reliable region. Because of noise and channel effect, the normalized correlation results are closing to 0.9 instead of 1. Therefore, the reliability decreases while the threshold is defined close to 0.9. For threshold smaller then 0.5 cases, the noise will influence the detection result.

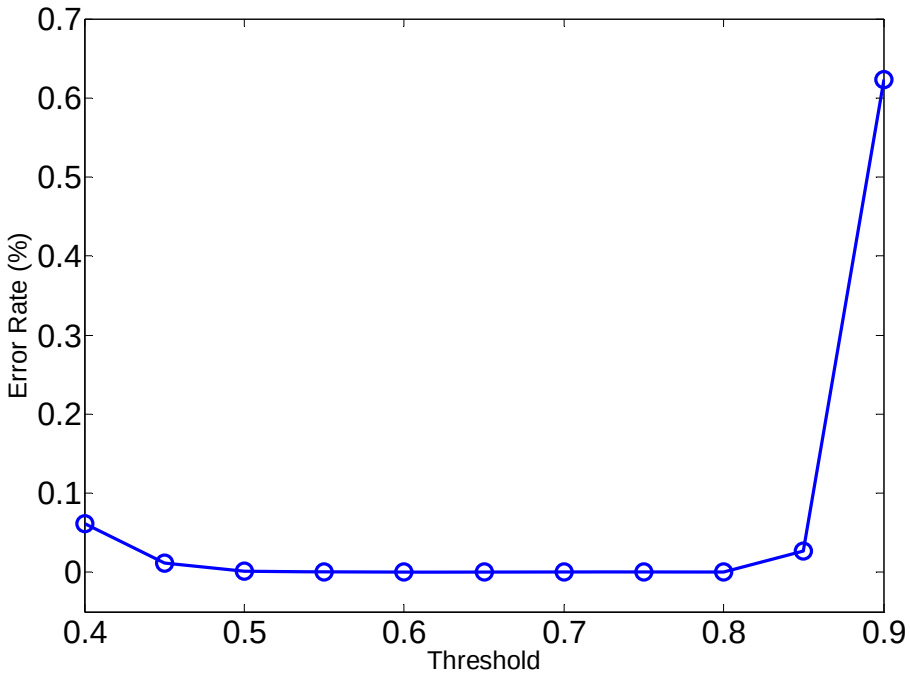


Fig. 3.7 Error rate under different threshold of proposed mode/GI detection

3.2 Coarse Symbol Synchronization

Coarse symbol synchronization (CSS) is also named symbol boundary detection. The goal of symbol boundary detection is to find out boundaries of transmitted symbols. Coarse symbol synchronization starts to work after finishing the mode/GI detection. With the information of transmission mode and guard interval length, symbol boundary detection has enough information to detect boundaries. After the first symbol boundary is detected, the system will use transmission mode, guard interval length and boundary location to derive the successive symbol boundaries.

In order to make the FFT windows locate on correct locations, symbol boundary detection needs to solve some problems. Since DVB-T signals are transmitted in SFN (Single Frequency Network), for a receiver there will have many signals from different transmitters with discordant delay time as Fig. 3.8 illustrates. The multipath effect leads to the same symbol with different delays overlaps and hard to detect the correct boundary of main path. Channel noises also reduce the reliability of coarse

symbol synchronization algorithms. The solution will be discussed in section.

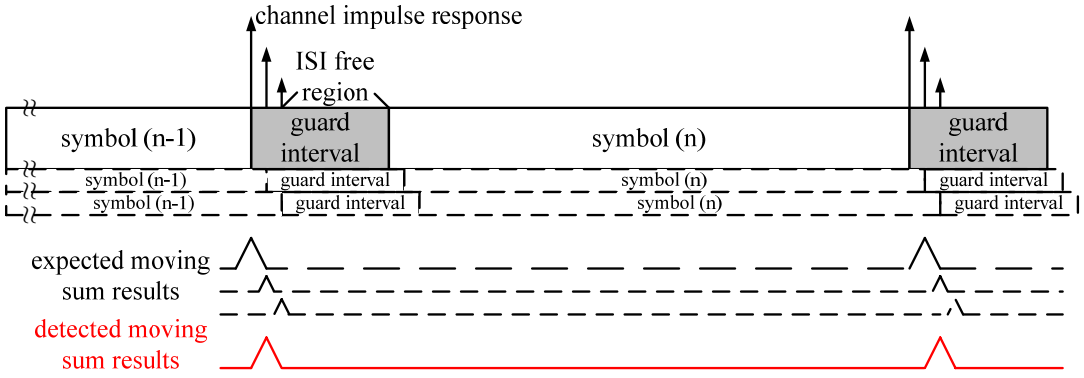


Fig. 3.8 Effect of multipath fading

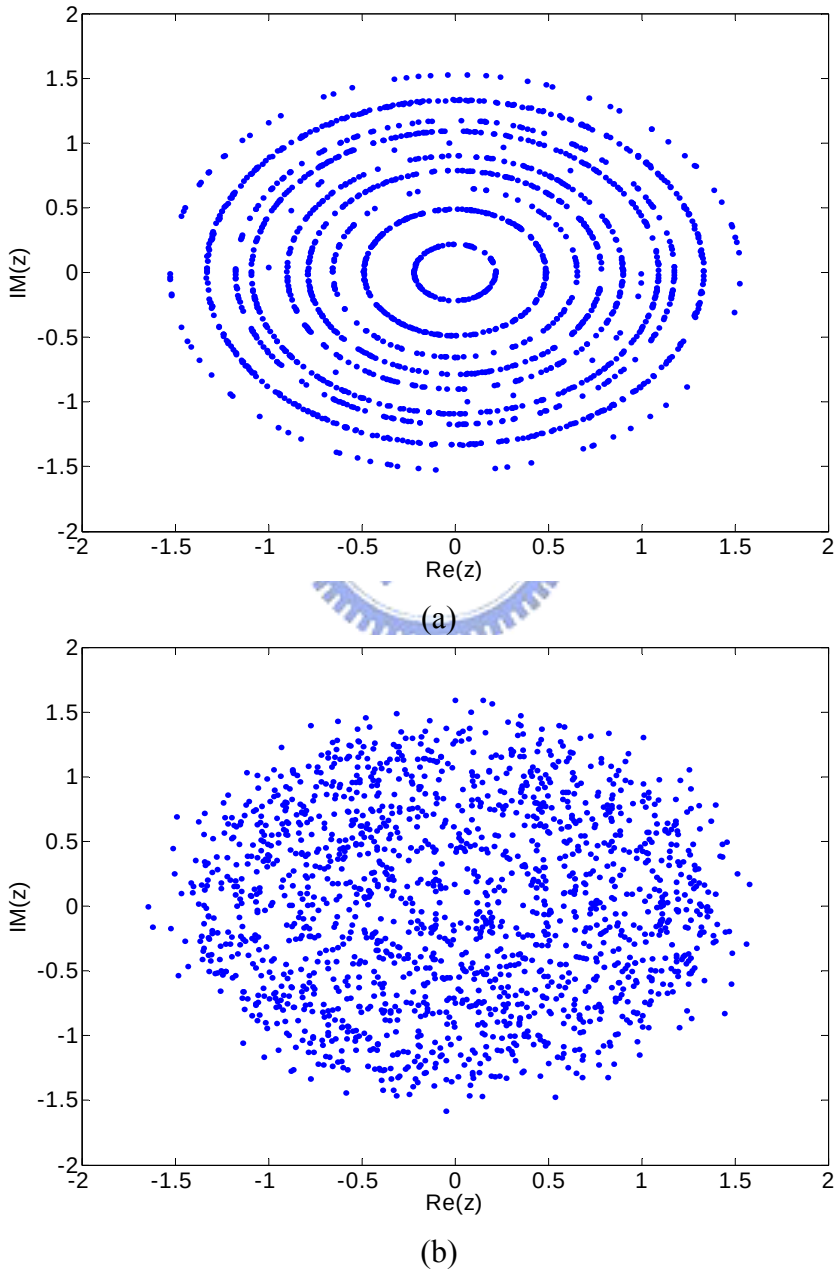
3.2.1 Effect of Symbol Timing Offset

Before starting to introduce to coarse symbol synchronization algorithms, the effect of symbol timing offset must been derived. Symbol timing offset means the detected boundary does not locate on the true symbol boundary. This phenomenon is due to multipath effect, channel noise and aliasing. Two possible cases will occur, later or earlier than the true symbol boundary. Thanks to cyclic prefix, an earlier boundary location only causes a phase rotation which is able to be compensated by frequency domain equalization. The effect of the offset is derived in Eqn. (3.5).

$$
\begin{aligned}
\hat{X}(k) &= X(k-\varepsilon) \\
&= \sum_{n=0}^{N-1} x(n-\varepsilon) e^{j2\pi k\frac{n}{N}} \\
&= \sum_{n=0}^{\varepsilon-1} x(n-\varepsilon) e^{j2\pi k\frac{n}{N}} + \sum_{n=\varepsilon}^{N-1} x(n-\varepsilon) e^{j2\pi k\frac{n}{N}} \\
&= \sum_{n=0}^{\varepsilon-1} x(n+N-\varepsilon) e^{j2\pi k\frac{n+N-\varepsilon}{N}} e^{j2\pi k\frac{\varepsilon}{N}} + \sum_{n=\varepsilon}^{N-1} x(n-\varepsilon) e^{j2\pi k\frac{n-\varepsilon}{N}} e^{j2\pi k\frac{\varepsilon}{N}} \\
&= \sum_{n=0}^{N-1} x(n) e^{j2\pi k\frac{n}{N}} e^{j2\pi k\frac{\varepsilon}{N}} \\
&= X(k) e^{j2\pi k\frac{\varepsilon}{N}}
\end{aligned}
\tag{3.5}
$$

where $X(k)$ is the respected result of FFT output and ε is the number of offset samples.

The case of earlier offset only leads to a phase rotation. But the derivation above only works when ε is not too large to make FFT window locates on ISI region. Otherwise, just like the later case, FFT window will get signals from previous symbol which has no orthogonality with the current symbol. Fig. 3.9 represents the sub-carriers after FFT by early and late cases in constellation map. The early case leads to a phase rotation in constellation map while the late case leads to mix-up in constellation map.



(a)

(b)

Fig. 3.9 Sub-carriers of (a) early case and (b) late case for timing offset

3.2.2 Coarse Symbol Synchronization Algorithms

Coarse symbol synchronization aims at finding a rough symbol boundary. As a result of DVB-T system uses broadcasting technique to transmit signals in air, maximum-likelihood algorithms are not suit for DVB-T. Thanks to cyclic prefix again, like mode/GI detection, many coarse symbol synchronization algorithms are based on using the correlations between guard interval and symbol's tail. There are three common algorithms, which are maximum correlation (MC) [19], normalized maximum correlation (NMC) [17] and minimum mean square error (MMSE) in [20]. In the following will compare their performance.

a) Maximum Correlation (MC)

$$K_{est} = \arg \max_{n} \left| \sum_{i=0}^{N_g - 1} r^*(n-i) \times r(n-i-N) \right| \tag{3.6}$$

Similar to mode/GI detection, maximum correlation algorithm uses guard interval length detected by mode/GI detection as N_g to detect the maximum peak value. Unlike mode/GI detection, there will only be a maximum peak exist in stead of a plateau. Ideally, when the moving sum window goes into guard interval region, correlation will start to grow up. When the moving sum window exactly fits the whole guard interval region, a maximum peak occurs. Correlation value will decrease when the moving sum window start to leave the guard interval region.

b) Normalized Maximum Correlation (NMC)

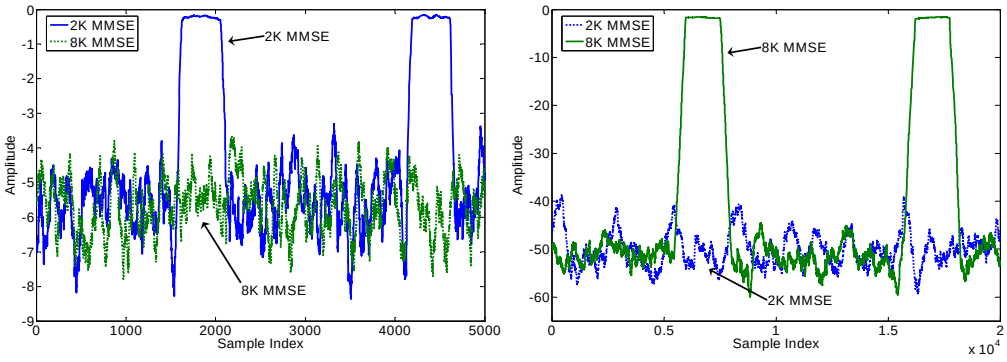
$$K_{est} = \arg \max_{n} \frac{\left| \sum_{i=0}^{N_g - 1} r^*(n-i) \times r(n-i-N) \right|}{\left| \sum_{i=0}^{N_g - 1} r^*(n-i) \times r(n-i) \right|} \tag{3.7}$$

Normalized maximum correlation algorithm is similar to maximum correlation algorithm. By dividing its own power term, the peak is normalized to "1" ideally. A normalized peak makes the threshold easy to define and that's why the modified normalized maximum correlation is adopted as mode/GI detection algorithm. But a penalty of an extra moving sum delay-line for power term and a divider are the disadvantages.

c) Minimum Mean Square Error (MMSE)

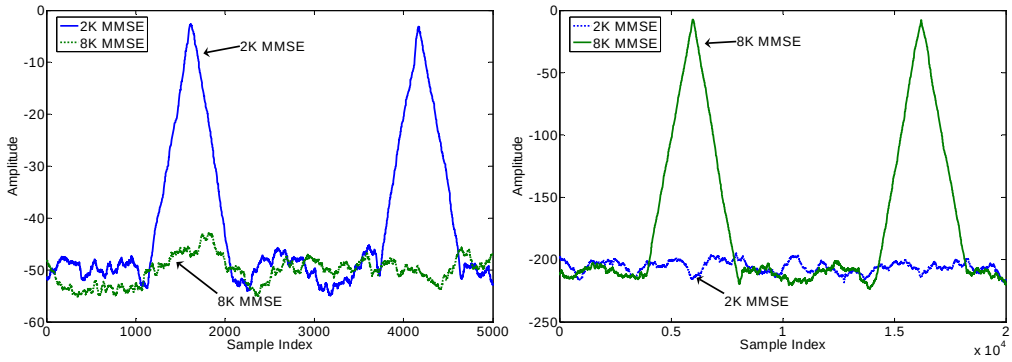
$$K_{est} = \arg \max_{n} \left(\left| \sum_{i=0}^{N_g-1} r^*(n-i) \times r(n-i-N) \right| - \frac{1}{2} \sum_{i=0}^{N_g-1} \left(|r(n-i)|^2 + |r(n-i-N)|^2 \right) \right) \quad (3.8)$$

This algorithm is similar to ML algorithm in [21], but the received signals replace the look-up table. It also likes a change of normalized maximum correlation algorithm. Since the maximum value of NMC is close to "1", that means the numerator substrates the denominator will close to "0". The characteristic of plateau, which is close to "0", also can be used in mode/GI detection. But without a normalized threshold, MMSE is hard to tell from the plateau of 8K mode from bottom of other modes. As shown in Fig. 3.10 (a) and (b), the plateau value of 8K mode is closed to the bottom of 2K mode. Fig. 3.11 (a) and (b) show the boundary detection results of MMSE algorithm and the peak characteristic is similar to MC and NMC. The patterns of Fig. 3.10 and Fig. 3.11 are the same with Fig. 3.3.



(a) (b)

Fig. 3.10 MMSE mode/GI detection under (a) 2K (b) 8K mode



(a) (b)

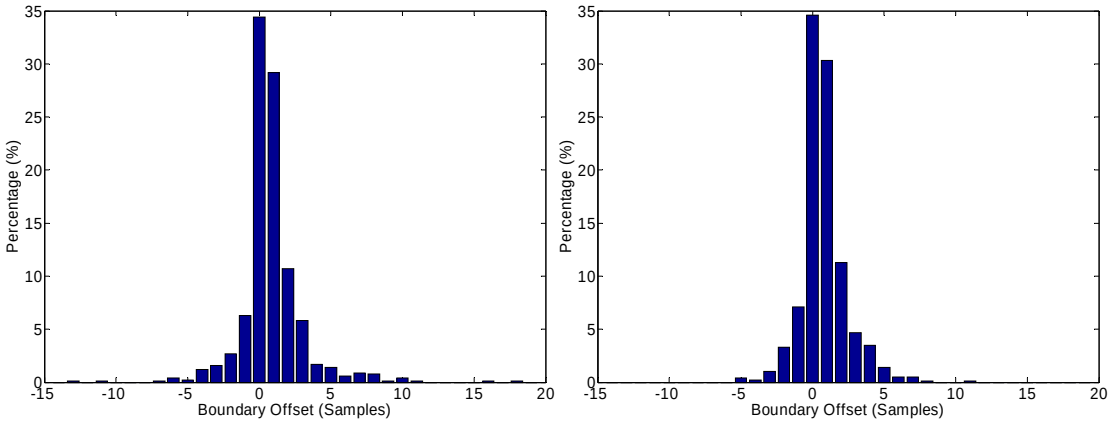
Fig. 3.11 MMSE boundary detection under (a) 2K (b) 8K mode

As a result, NMC is adopted as mode/GI detection algorithm in this thesis because of the normalized plateau characteristic.

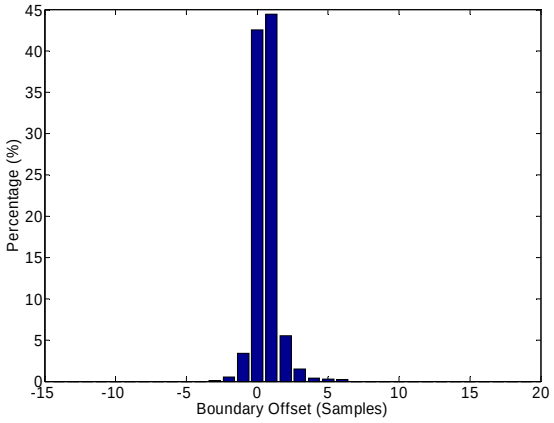
3.2.3 Performance Simulation and Comparisons

a) Accuracy Simulation

The symbol boundary location accuracy of MC, NMC and MMSE is simulated to compare the performance. The transmitted 1000 symbols will survive from two kinds of channels, Rayleigh and Ricean, with CFO equals to 23.33 sub-carriers, 2K transmission mode, 1/4 guard interval length, 50Hz Doppler spread and 12dB SNR. As Fig. 3.12, Fig. 3.13 and Table 3-1 shows, MMSE owns the best accuracy. The accuracy of MC and NMC are closely.

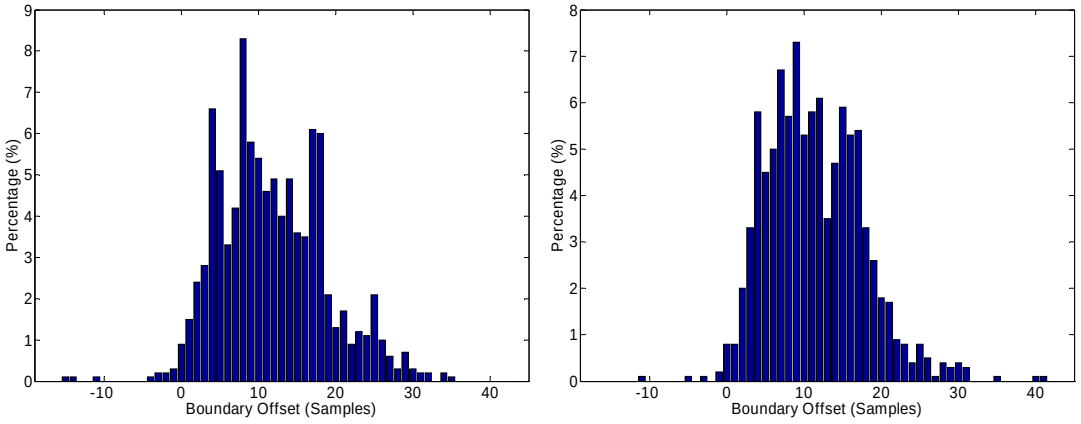


(a) (b)

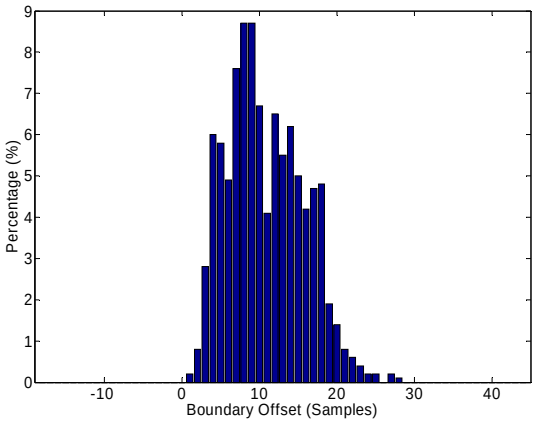


(c)

Fig. 3.12 (a) MC, (b) NMC and (c) MMSE boundary offset distribution @ Ricean



(a) (b)



(c)

Fig. 3.13 (a) MC, (b) NMC and (c) MMSE boundary offset distribution @ Rayleigh

Table 3-1 MC, NMC and MMSE boundary average/peak offset @ Ricean/Rayleigh

Name	MC (Average/Peak Offsets)	NMC (Average/Peak Offsets)	MMSE (Average/Peak Offsets)
Ricean	0.8051/18	0.7768/11	0.6020/7
Rayleigh	11.7687/36	11.4111/41	10.8273/29

b) Hardware Complexity

For 2K/4K/8K applications in DVB-T/H, all of the algorithms above must have at least an 8K long correlation delay-line, which is used to store the *r(n-8K)* signal and also can be used to store *r(n-2K)* and *r(n-4K)* signals, and a 2K (a quarter of 8K) long moving sum delay-line, which is used to integrate the 2K moving sum correlation values for 8K mode with 1/4 GI length, as Fig. 3.14 illustrates. Due to the correlation expression, a complex multiplier is necessary. Then two pairs of adders and subtractors, one for real part and another for image part, are used to do moving sum calculation. Finally, an additional complex multiplier is required to do the square operation. For NMC, an extra 2K moving sum delay-line, complex multiplier, adder and subtractor are needed for its power term denominator. The square operation of power term only needs a multiplier as a result of the power term is integer. Further more, a divider is needed for normalization operation. MMSE needs an extra complex

multiplier and adder comparing to the NMC denominator components and the divider is replaced by a subtractor. In summary, Table 3-2 lists the components required.

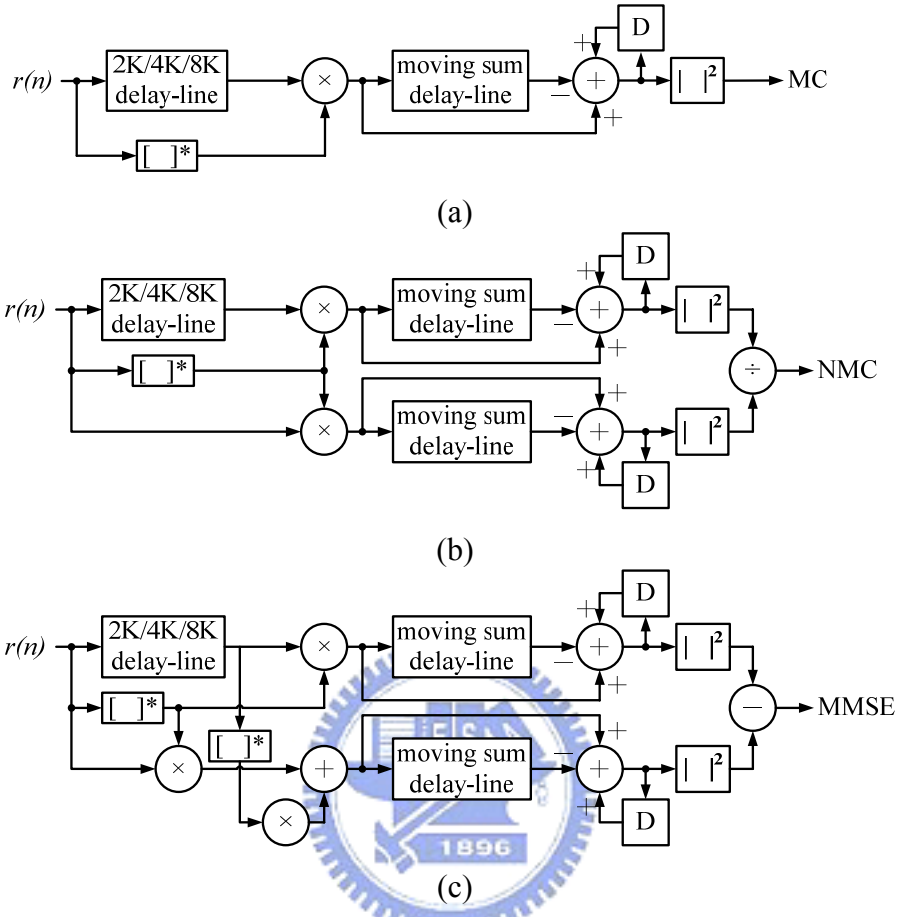


Fig. 3.14 Block diagram of (a) MC (b) NMC and (c) MMSE

Table 3-2 Components required for MC, NMC and MMSE

Name	Correlation/Moving Sum Delay-Line	Complex Multiplier	Adder/ Subtractor	Multiplier/ Divider
MC	8K/2K	2	2/2	0/0
NMC	8K/2K×2	3	3/3	1/1
MMSE	8K/2K×2	4	4/4	1/0

The reason why MC is adopted as symbol boundary detection algorithm in this thesis is because of the performance is not differ too much to NMC and MMSE, but it needs the fewest components and no divider.

3.2.4 Proposed Mode/GI and Symbol Boundary Detection Scheme

By observing the mode/GI and boundary detection expression, it is easy to find out their structures are very similar. The mode/GI detection block diagram is illustrated in Fig. 3.15. The architecture is modified from NMC architecture by using the subtraction to replace the division as Eqn. (3.4) derived. As a result, mode/GI and boundary detection is able to share the same hardware to detect mode/GI and symbol boundary by controlling the correlation and moving sum delay-line lengths. After mode/GI detection, controller changes the correlation and moving sum delay-line lengths for symbol boundary detection.

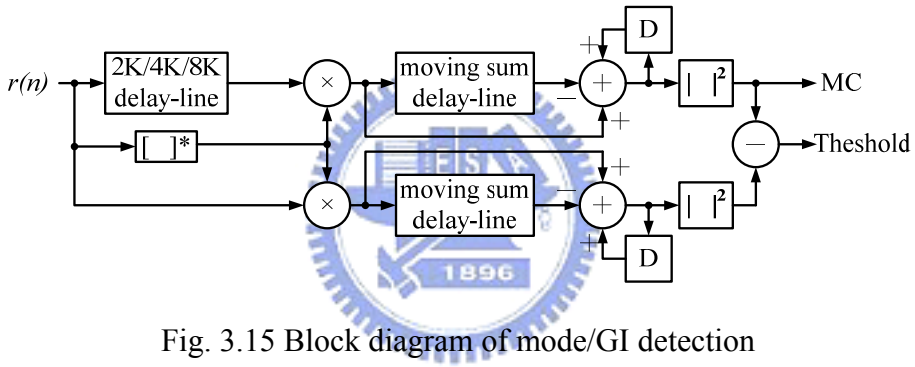


Fig. 3.15 Block diagram of mode/GI detection

Fig. 3.16 illustrates the proposed mode/GI and boundary detection hybrid architecture. In the correlation part, the functions of 2K, 4K and 8K delays for different transmission modes are realized by a 2K/4K/8K triple modes reconfigurable delay-line and defined as correlation delay-line in this architecture. For four guard interval lengths of three transmission modes, there are totally six possible moving sum lengths and realized by two 2K/1K/512/256/128/64 reconfigurable delay-line, which is also named the moving sum (MS) delay-line.

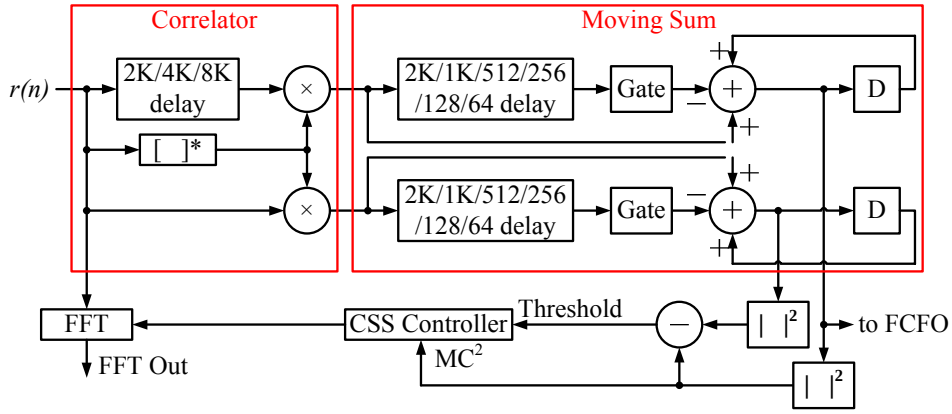


Fig. 3.16 Architecture of mode/GI and boundary detection scheme

The proposed mode/GI and CSS scheme can be divided into three stages, the first stage is mode detection, the second stage is guard interval length calculation and the final stage symbol is boundary detection. As Fig. 3.17 shows, the system first starts to detect the mode beginning from 2K then 4K and finally 8K mode. It spends 2K samples to fill the correlation delay-line and 64 samples (the minimum guard interval length of 2K) to fill the moving sum delay-line and then enter the 2K mode detection. The purpose to choose 2K mode as the first detected mode is it only needs 2K samples latencies to fill the correlation delay-line since others candidates need 4K or 8K samples. By using the twister memory access method based delay-line design to overlap the filling time and the previous mode detection time, the correlation delay-line does not have to refill or replenish when the detection mode is changed. As a result, the latency of the proposed scheme is reduced to only 2K samples.

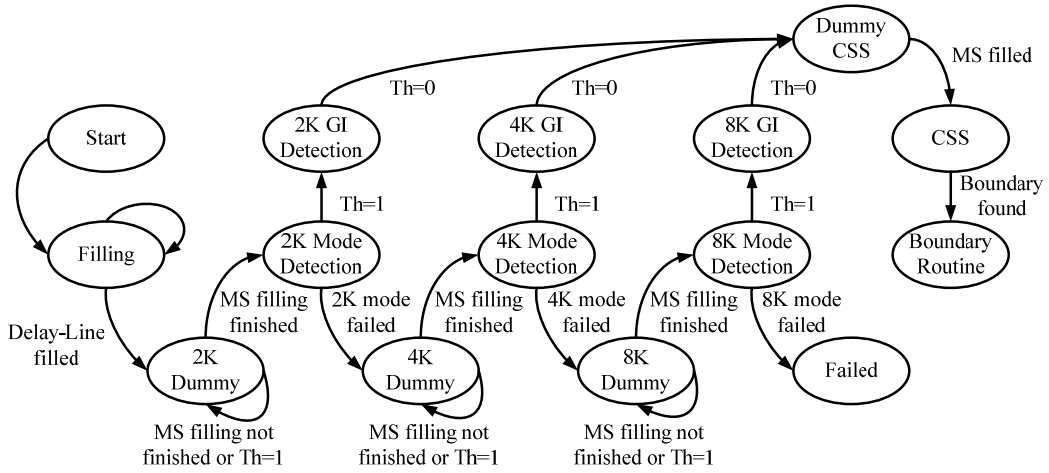


Fig. 3.17 Finite state machine of mode/GI and boundary detection scheme

The pseudo code of the proposed scheme is shown below.

1. *Set the detection mode as 2K and fill the correlation delay-line with 2K samples.*
2. *Fill the moving sum with detection mode/32 samples. If the moving sum is filled and threshold=0, go to Step 3.*
3. *Detect the detection mode for a pre-defined detection period. If the threshold=1, jump to Step 4. Else if the detection counter=detection period, jump to Step 2 and set the detection mode as detection mode×2.*
4. *Count the period length of threshold=1. If the threshold=0, set moving sum length as detected GI length and go to Step 5.*
5. *Fill the moving sum then find the maximum correlation result as boundary.*
6. *Derive the next boundaries using the detected mode, GI length and boundary.*

After a detection period of (1+1/32)×2K ~ 2×(1+1/4)×2K samples, the system knows the transmission mode is not the same with 2K mode and then enters the "4K Dummy State" to do the 4K mode detection. Otherwise, if the threshold changes from low to high during the 2K mode detection period, it means the transmission mode is the 2K mode and so do other detection modes. The system jumps to "GI Detection State" as soon as the threshold turns to high and calculate the period length when it is high. The system period length is regarded as guard interval length. When the moving sum window accidentally locates on any place of the guard interval period initially, the threshold will turn to high before leaving "Dummy State". This situation possibly

leads to an incorrect guard interval length. For the purpose to prevent this situation, the state won't enter "Mode Detection State" before the threshold turns to low. Thus, the system only starts to calculate the guard interval length at the beginning of a plateau.

After mode/GI detection, the system enters the "Dummy Find Boundary State" and uses the detected guard interval length to refill the moving sum delay-line. After refilling the moving sum delay-line, the system has a best MC^2 result. By observing the MC^2 results during plateau, the location of the maximum MC^2 result subtracts half of the guard interval length will be the rough symbol boundary. After all, the system predicts the successive boundaries using the detected mode, GI length and first boundary information.

The required time of the proposed scheme is analyzed below. For the 2K transmission mode, it needs 2K samples to fill correlation delay-line, 64 samples to fill moving sum delay-line, 1 ~ (1+1/32)×2K samples for mode detection, 64~512 samples for calculating guard interval length and (1+1/32)×2K ~ (1+1/4)×2K for boundary detection. Theoretically, 7296 samples is the worst case to finish the mode/GI and symbol boundary detection scheme. The system stays at "2K Mode Detection State" for at least $(1+1/32)\times 2K$ samples to guarantee the possible peak/plateau is detected. Without enough stay time, the detector is possible to miss the peak/plateau. After this, the system jumps to "4K Dummy State" and changes the correlation delay-line to 4K mode and refill the moving sum delay-line. While the mode changes to 4K, it's not necessary to refill or replenish the correlation delay-line because of the signal $r(n\text{-}4k)$ is already in the correlation delay-line. The other procedures are similar to 2K mode and so do 8K mode. Table 3-3 lists the required samples for different transmission modes.

Table 3-3 Timing for mode/GI and boundary detection

Mode	$Mode_{det\_t}$	GI_{det_t}	$Boundary_{det\_t}$	$Total_{det\_t}$
2K	2K+64+(1~2K+64)	64~512	$2K+2K\_GI_{det\_t}$	$2K\_Mode_{det\_t}+2K\_GI_{det\_t}$ $+2K\_Boundary_{det\_t}$
4K	$Max(2K\_Mode_{det\_t})$ +128+(1~4K+128)	128~1K	$4K+4K\_GI_{det\_t}$	$4K\_Mode_{det\_t}+4K\_GI_{det\_t}$ $+4K\_Boundary_{det\_t}$
8K	$Max(4K\_Mode_{det\_t})$ +256+(1~8K+256)	256~2K	$8K+8K\_GI_{det\_t}$	$8K\_Mode_{det\_t}+8K\_GI_{det\_t}$ $+8K\_Boundary_{det\_t}$

Comparing to parallel mode detection shown in Fig. 3.18 (a), (b) and (c), the proposed scheme, shown in Fig. 3.18 (f), has the same latency for 2K transmission mode and an extra delay depends on mode detection period for other modes. But only single hardware is necessary for different mode detection. For the sequential mode detection shown in Fig. 3.18 (d) and (e), the proposed scheme saves 12K sample as a result of using twister memory access based delay-line. In summary, only 384 samples (2.27%) more than parallel 8K mode detection, 12K samples (41.56%) less than sequential mode detection (refill) and 6K samples (26.23%) less than sequential mode detection (replenish).

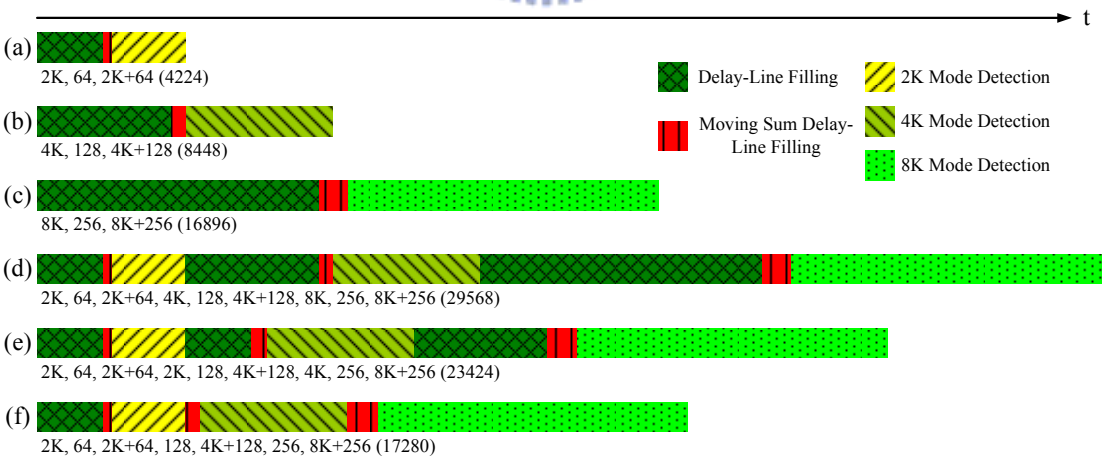


Fig. 3.18 Timing of parallel (a) 2K, (b) 4K, (c) 8K, sequential (d) refill, (e) replenish and (f) proposed mode detection schemes

Fig. 3.19 (a), (b), (c) and (d) show the simulation result of the proposed scheme for 2K and 8K transmission modes in MC^2 and NMC using the same pattern with Fig. 3.3. The aliasing peak is caused by the moving sum doesn't accumulate enough values

and it only occurs at “Dummy States”. As Fig. 3.19 shows, without the assistance of NMC, the correlation values during GI detection period is much smaller than the boundary detection period and the system is not easy to define the GI detection period. With the assistance of NMC, to define the GI detection period and compute the period length becomes much easier for the system. Furthermore, for the purpose to improve the reliability, the threshold detector is realized using 8 states confidence counter. The simulation environment is 2K/8K transmission symbols with 1/4 GI, 23.33 CFO, 12dB SNR and surviving from Rayleigh channel.

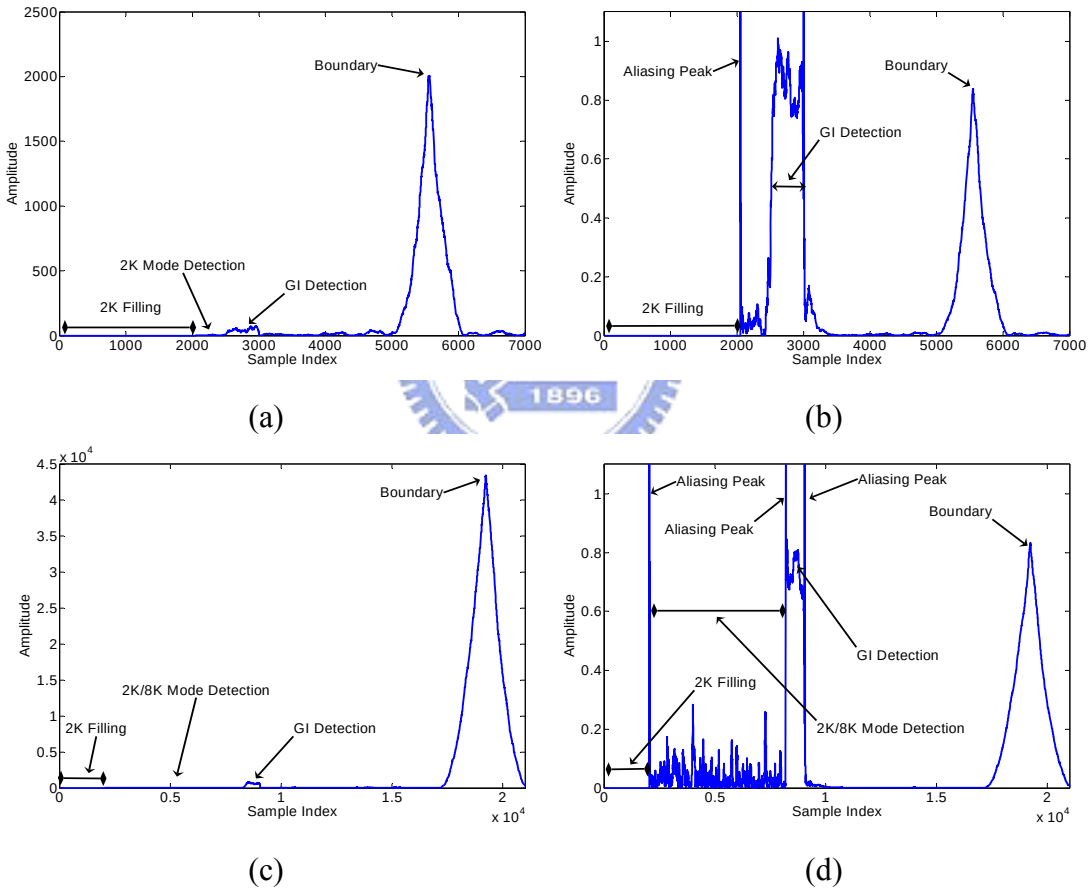


Fig. 3.19 (a) 2K MC^2, (b) 2K NMC, (c) 8K MC^2 and (d) 8K NMC simulation of the proposed blind mode/GI boundary detection scheme

In summary, an efficient sequential mode/GI and boundary detection scheme using normalized plateau to compute the guard interval length with single hardware, which is also used by boundary detection, is proposed in this chapter.

Chapter 4

Channel Estimation Algorithms

This chapter will focus on channel estimation issued. First, the scattered pilot synchronization (SPS), which is also called scattered pilot mode detection, will be discussed for the purpose to extract the correct scattered pilots for channel estimation. After the scattered pilot mode is detected, the channel estimation collects the scattered pilots and using their value to estimate the channel response. Finally, a division-free equalizer and three-stage demapping hybrid architecture will be described.

4.1 Scattered Pilot Synchronization

The inserted scattered pilots are useful for the purpose to estimate the channel response for channel estimation. Unfortunately, unlike the continuous pilots, the distribution of scattered pilots is not fixed with regular locations. As Eqn. (2.2) shows, there are four scattered pilot distribution modes. Typically, the scattered pilot distribution can be known after TPS synchronization. But it takes 17~68 symbols to decode the TPS pilots and it spends too many time. For DVB-H application, the synchronization time is strictly limited. Therefore, using fast scattered pilot synchronization technique to accelerate the synchronization timing is necessary. This section will introduce two fast scattered pilot detection algorithms. Performance simulation and hardware complexity comparison is carried out. Furthermore, an idea of two-stage scattered pilot synchronization scheme with extracted scattered pilots pre-filling, which is required by channel estimation to estimate the channel response, will be described.

4.1.1 Fast Scattered Pilot Synchronization Algorithms

In order to detect the scattered pilot distribution mode as soon as possible, two fast scattered pilot synchronization algorithms were proposed in [22] [23] and will be discussed below.

a) Power-Based Scattered Pilots Mode Detection

The first algorithm is power-based scattered pilot mode detection algorithm (PB) as shown in Eqn. (4.1):

$$
\begin{aligned}
PS_1(n) &= \left| \sum_{p=0}^{p_{\max}} SC(n,12p+12) \cdot SC^*(n,12p+12) \right| \\
PS_2(n) &= \left| \sum_{p=0}^{p_{\max}} SC(n,12p+3) \cdot SC^*(n,12p+3) \right| \\
PS_3(n) &= \left| \sum_{p=0}^{p_{\max}} SC(n,12p+6) \cdot SC^*(n,12p+6) \right| \\
PS_4(n) &= \left| \sum_{p=0}^{p_{\max}} SC(n,12p+9) \cdot SC^*(n,12p+9) \right| \\
Scattered \quad Pilot \quad Mode &= \max(PS_P(n)); p \in \{1,2,3,4\}
\end{aligned}
\tag{4.1}
$$

where p_{max} is 141 for 2K, 283 for 4K and 567 for 8K mode, *SC(n,m)* is the m^{th} sub-carrier of the n^{th} OFDM symbol. The power-based algorithm uses the characteristic of scattered pilots' boosted power by 4/3. It's easy to see that only the correct scattered pilots have the largest `power due to its boosted power. By observing the power accumulation of four possible scattered pilot distributions, the scattered pilot mode can be detected.

b) Correlation-Based Scattered Pilots Mode Detection

The second algorithm is correlation-based scattered pilot mode detection algorithm (CB). By using the similar skill as Eqn. (4.1) but multiplying the same

sub-carriers located four OFDM symbols before instead of itself. This method is stronger to against noise. But if Doppler effect changes the channel response significantly after four symbols time, the performance of this algorithm will be affected by Doppler effects. Eqn. (4.2) presents the correlation-based scattered pilot mode detection algorithms mathematically.

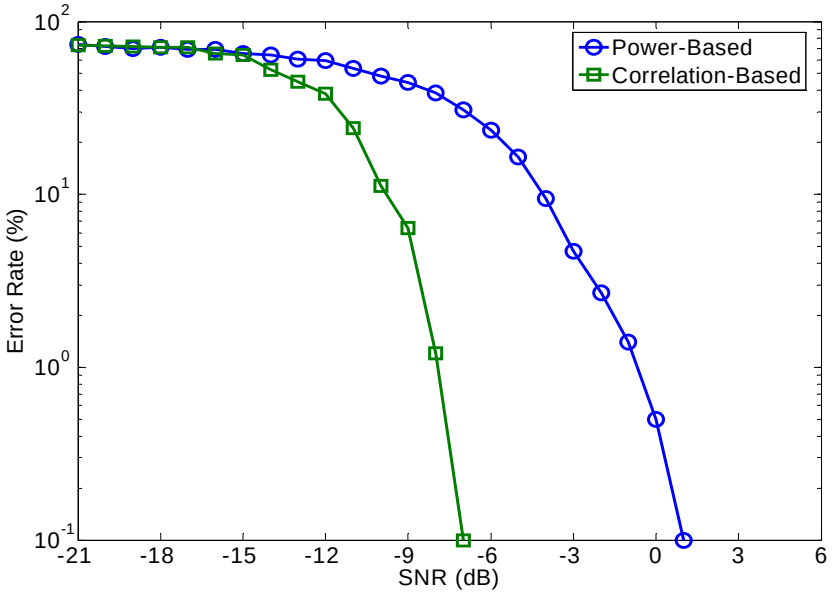
$$
\begin{aligned}
CS_1(n) &= \left| \sum_{p=0}^{p_{\max}} SC(n,12p+12) \cdot SC^*(n-4,12p+12) \right| \\
CS_2(n) &= \left| \sum_{p=0}^{p_{\max}} SC(n,12p+3) \cdot SC^*(n-4,12p+3) \right| \\
CS_3(n) &= \left| \sum_{p=0}^{p_{\max}} SC(n,12p+6) \cdot SC^*(n-4,12p+6) \right| \\
CS_4(n) &= \left| \sum_{p=0}^{p_{\max}} SC(n,12p+9) \cdot SC^*(n-4,12p+9) \right| \\
Scattered \quad Pilot \quad Mode &= \max(CS_p(n)); p \in \{1,2,3,4\}
\end{aligned}
\tag{4.2}
$$

Comparing to the second method, the first one is better in against Doppler Effects and doesn't require a latency of four symbols due to multiplying sub-carriers itself. This implies that the power-based algorithm is four symbols time faster than the correlation-based and requires no sub-carriers storage memory to save the scattered pilots of four distributions four symbols before current symbol. Though the scattered pilot storage can be done by using the channel estimation storage memories and does not cause an additional hardware overhead, the ability to against Doppler effects and fast to get the result makes power-based scattered pilot mode detection algorithm the choice in scattered pilot mode detection.

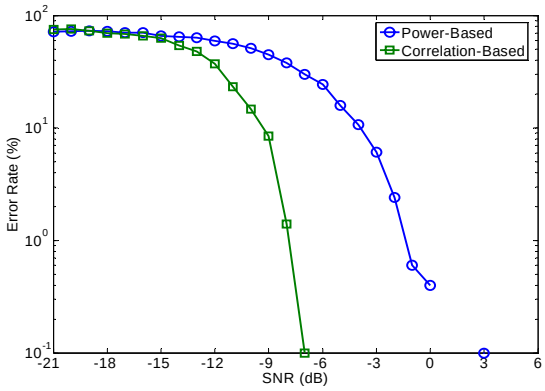
4.1.2 Performance Simulation and Comparisons

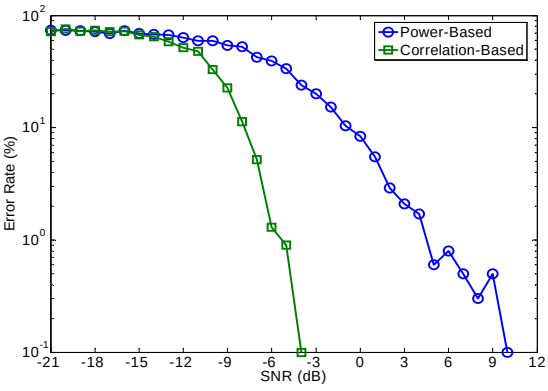
a) Performance Simulation

Fig. 4.1 shows the error rate of the above two algorithms under different Signal-to-Noise Ratio (SNR). The environment is 1000 2K transmission mode symbols with 1/4 guard interval length, Rayleigh channel, 10 samples earlier than correct boundary. The residual CFOs after fractional and integer CFO compensation are (a) zero (b) 0.03 sub-carrier and (c) 0.33 sub-carrier. As shown in Fig. 4.1, the error rate of power-based algorithm starts to increase when SNR below 2dB and the correlation based algorithm starts to increase when the SNR below -6dB. Generally, as [1] defined, the SNR is bigger than 3 dB and the accidentally scattered pilot mode detection error is only need to be concerned.



(a)





(b) (c)

Fig. 4.1 Error rate of two SPS algorithms versus SNR with (a) 0 (b) 0.03 (c) 0.33 CFO

b) Hardware Complexity

The architecture of power-based and correlation-based scattered pilot mode detection is shown in Fig. 4.2.

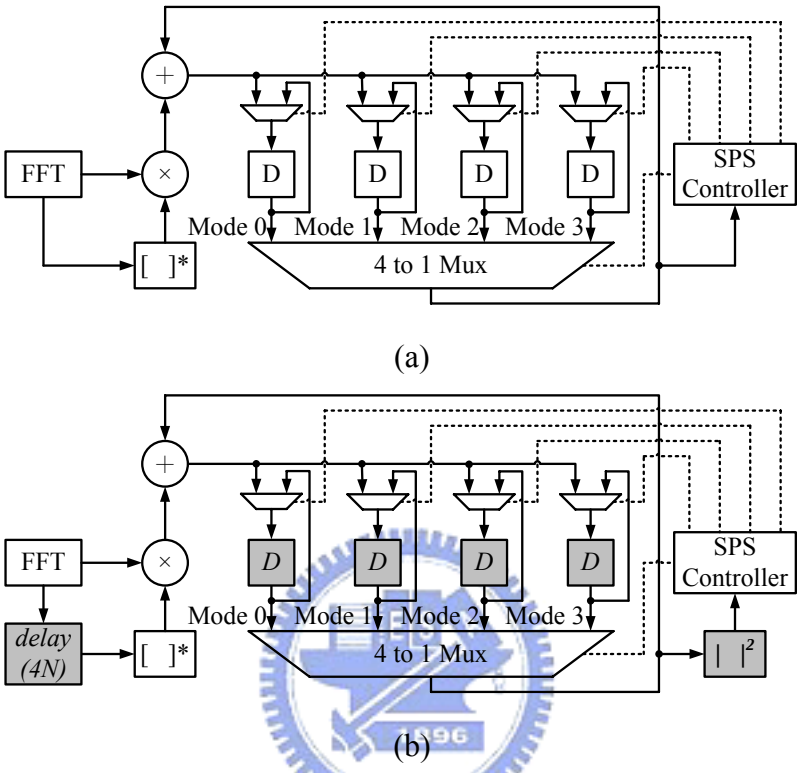


Fig. 4.2 Architecture of (a) power-based (b) correlation-based SPS

The power-based algorithm needs a complex multiplier to have a correlation with the conjugate of itself, an adder to do summation and four register groups to hold the values. The correlation-based algorithm needs two adders due to the correlation expression output is a complexity number. This also means the size of register groups is double. Also an additional complex multiplier is required to do square operation.

Table 4-1 shows the statistic of the two algorithms. Correlation-based algorithm needs more area and more time to figure out the answer than power-based algorithm. Therefore, the algorithm with low area cost and fast to detect the mode, which is power-based algorithm, is adopted as scattered pilot mode detection algorithm in this thesis.

Table 4-1 Statistic of power-based and correlation-based architectures

Name	Hardware	Time	Other
Power-Based	Complex Multiplier×1 Adder×1 Registers Groups×4	1 Symbol	Better for Doppler
Correlation-Based	Complex Multiplier×2 Adder×2 Registers Groups×8 Delay Element×1	5 Symbols	Better for noise

4.1.3 Proposed Two-Stage Scattered Pilot Synchronization Scheme

Generally, under the specified SNR, both the power-based (PB) and correlation-based (CB) scattered pilot synchronization algorithms are seldom fail. However, an unexpected fail may occur. As a result, a two-stage scattered pilot synchronization scheme illustrated in Fig. 4.3 is proposed to improve the reliability. The two-stage scheme will do scattered pilot synchronization process twice, the first time is to detect the scattered pilot mode and the second time is to ensure the result of the first one. If the detected mode from the second time is not the same as the predicted mode to the first time, the system will repeat the two-stage scattered pilot synchronization process until they are the same.

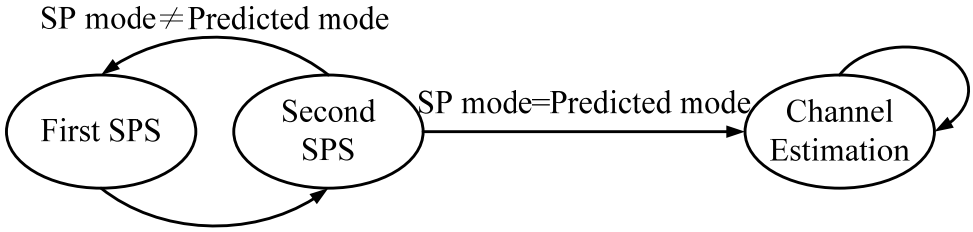


Fig. 4.3 Two-stage scattered pilot synchronization scheme

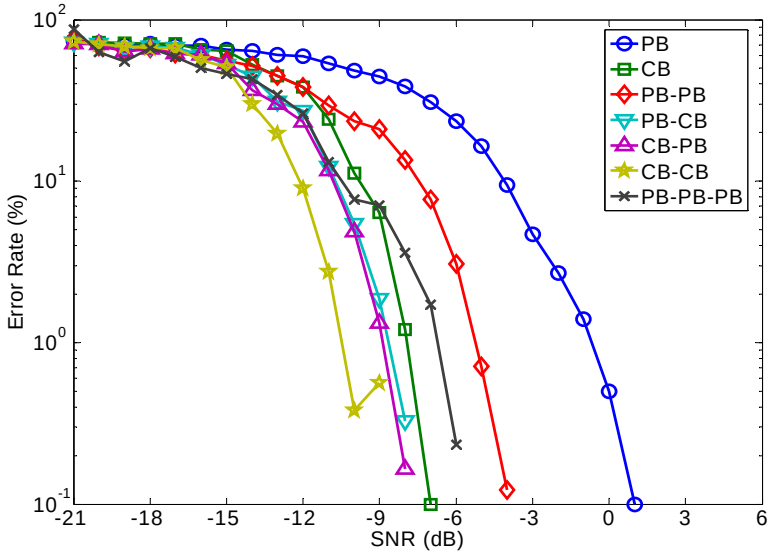
The first and second scattered pilot synchronization processes can be constructed by the combination of PB-PB, PB-CB, CB-PB or PB-PB. Table 4-2 lists the symbol time required in the scattered pilot synchronization time for PB, CB and four combinations above.

Table 4-2 Latency of different combination

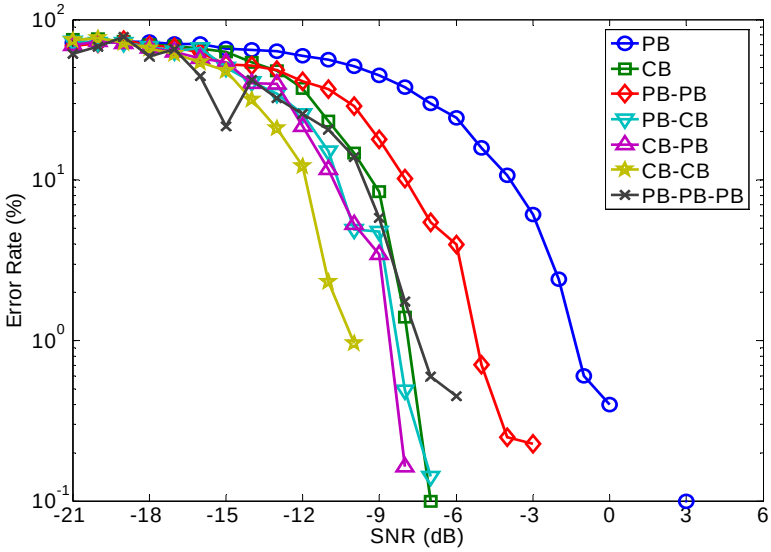
	One stage		Two stage			
	PB	CB	PB-PB	PB-CB	CB-PB	CB-CB
No Error	1	5	2	5	6	6
An Error			4	10	12	12

The PB-PB does the first time SPS process at the 1st symbol and the second time SPS at the 2nd symbol. The PB-CB does the first time SPS process at the 1st or the 4th symbol and the second time SPS at the 5th symbol. The CB-PB does the first time SPS process at the 5th and the second time SPS at the 6th symbols. Finally, the CB-CB does the first time SPS process at the 5th and the second time SPS at the 6th symbols. As shown in Table 4-2, the penalty of an error occurrence is to repeat the two-stage scattered pilot synchronization process again which will leads to at least one (PB-PB) to six (CB-PB or CB-CB) symbol delay to channel estimation process.

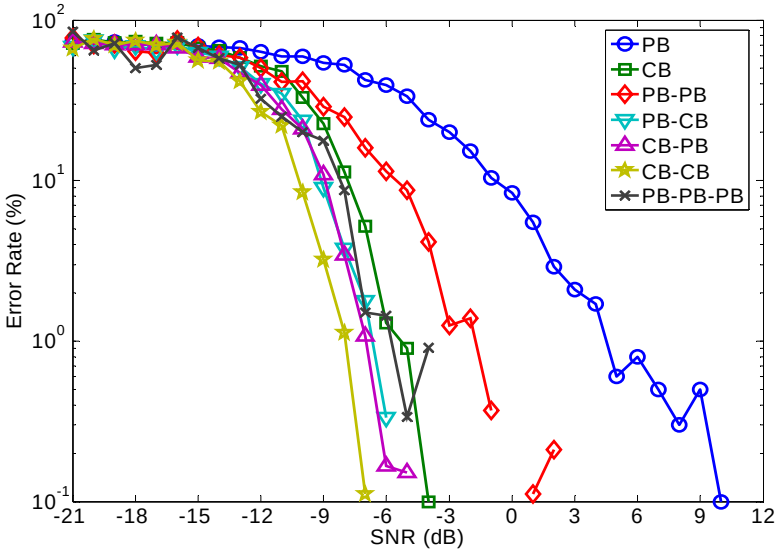
For the purpose to prove the proposed two-stage scattered pilot synchronization scheme really improves the reliability of the original scattered pilot synchronization algorithms, a scattered pilot detection error rate simulation is shown in Fig. 4.4. The testing environment is as the same with Fig. 4.1. For the two-stage scattered pilot synchronization scheme, the second stage tests all 1000 symbols and inverse infers the symbols that the first stage tests. Therefore, the two-stage scattered pilot synchronization scheme tests all 1000 symbols overall. Due to the two-stage scattered pilot synchronization scheme gives up the debatable answers, the error rate is calculated as (correct number of the detection result pass the two-stage scattered pilot synchronization scheme) over (number of the detection result pass the two-stage scattered pilot synchronization scheme). Moreover, a three-stage PB^3 SPS scheme is added in the simulation. As the simulation shows, the three-stage PB^3 scheme is very close to single stage CB. Therefore, using three-stage or more stages PB schemes to approximate the performance of single stage CB is worth to be considered.



(a)



(b)



(c)

Fig. 4.4 Error rate of SPS algorithms versus SNR with (a) 0 (b) 0.03 (c) 0.33 CFO

The simulation result shown in Fig. 4.4 proves the two-stage PB-PB scattered pilot synchronization is really has a better performance comparing to single PB scattered pilot synchronization. Since CB has better ability to against noise, the two-stage scattered pilot synchronization curves with at least one CB show a lower error rate than single CB or PB. Therefore, the two-stage scattered pilot synchronization scheme truly has an improvement on reliability.

4.1.4 Proposed Scattered Pilots Pre-Filling Scheme

As the latencies referred in Table 4-2, in order to improve the reliability, the two-stage scattered pilot synchronization scheme increases one to six symbol latencies to start channel estimation. For the purpose to reduce or eliminate the latency listed in Table 4-2, a channel estimation storage elements pre-filling scheme is proposed below:

1. *1st SPS: Fill the scattered pilots of four possible locations into four pilot storage elements individually.*
2. *Set predicted mode = mod(1st SPS detected mode+1,4).*
3. *Fill the predicted mode pilots into the storage element next to previous one.*
4. *Set predicted mode = mod(predicted mode+1,4).*
5. *Check if the state is 2nd SPS go to 6, else go to 3.*
6. *Check if the predicted mode = 2nd SPS detected mode, go to 7. Else, go to 1.*
7. *Fill the remaining storage elements and start channel estimation.*

The principle of channel estimation scattered pilots pre-filling scheme is to store all pilots of the four possible modes during the first scattered pilot synchronization process. Then use the detected mode to predict and store the later symbols' scattered pilots until the second scattered pilot synchronization process ensure the mode detected in the first scattered pilot synchronization process is correct or not. An example of two-stage PB-PB scattered pilot synchronization process with scattered pilots pre-filling for 2-D channel estimation, which requires at least six symbols'

scattered pilots to be stored and will be referred later, is shown in Fig. 4.5. Where an error occurs and *SP(n,m)* represent the mode m scattered pilot distribution of symbol n.

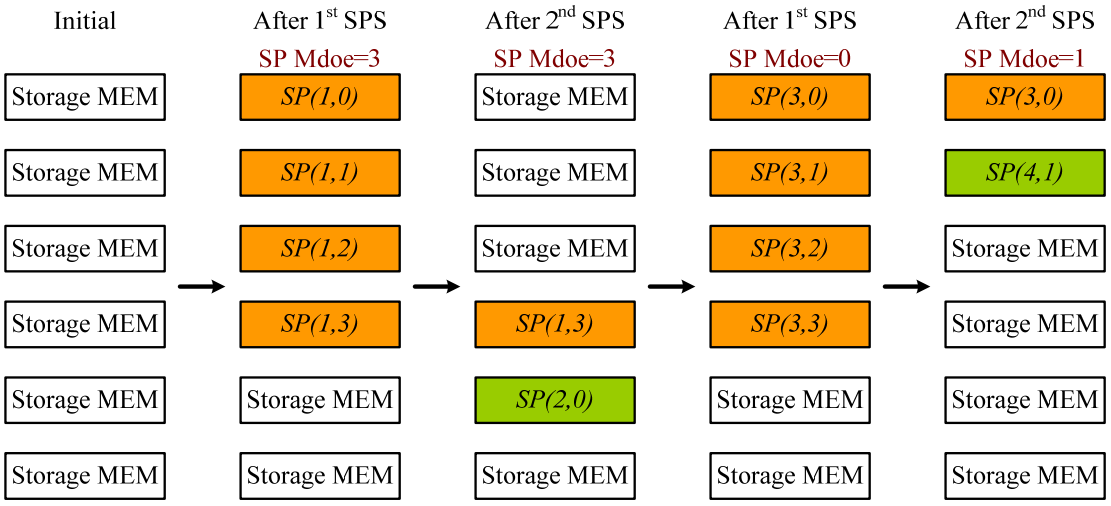


Fig. 4.5 Example of pre-filling scheme with one error occur

Overall, Table 4-3 lists the required timing of channel estimation pre-processing (timing for SPS and collecting scattered pilots) and different schemes in 2-D channel estimation. Although the error probability under specified SNR is a visual zero from [23], Fig. 4.3 and Fig. 4.4, the two-stage scheme is mainly proposed to prevent from an unexpected accident error. Therefore, at most one error is supposed to occur in Table 4-3. Of course, the scattered pilots pre-filling scheme is not only proposed for two-stage scattered pilot synchronization scheme and 2-D channel estimation but also capable for other scattered pilot synchronization schemes and channel estimation algorithms.

Table 4-3 Summary of different SPS schemes for 2-D channel estimation

Scheme	Error	Pre-Filling	CE Pre-Processing Time
PB		No	7 Symbols
		Yes	6 Symbols
CB		No	11 Symbols
		Yes	10 Symbols
PB-PB	No	No	8 Symbols
		Yes	6 Symbols
	Yes	No	10 Symbols
		Yes	8 Symbols
PB-CB	No	No	11 Symbols
		Yes	6 Symbols
	Yes	No	16 Symbols
		Yes	11 Symbols
CB-PB	No	No	12 Symbols
		Yes	10 Symbols
	Yes	No	18 Symbols
		Yes	16 Symbols
CB-CB	No	No	12 Symbols
		Yes	10 Symbols
	Yes	No	18 Symbols
		Yes	16 Symbols

To sum up, the channel estimation scattered pilot filling time is overlapped with scattered pilot synchronization by using the proposed scattered pilot pre-filling scheme. Therefore, a more reliable and fast scattered pilot synchronization scheme is proposed.

4.2 Frequency Domain Channel Estimation

The frequency domain channel estimation is aimed at estimating both the channel response and phase error as a result of symbol timing offset by using the inserted scattered pilots. In this section, three kinds of channel estimation algorithms, 1-D channel estimation [14] [26], 2-D channel estimation [13] [24] [25] and 2-D predictive channel estimation [26], will be discussed and compared.

3.4.1 Frequency Domain Channel Estimation Algorithms

The basic idea of channel estimation (CE) is to use the known value of inserted scattered pilots to estimate the channel responses. Three estimation algorithms will be discussed below.

a) 1-D Channel Estimation

The one dimensional channel estimation algorithm is the simplest of the three algorithms. Since a scattered pilot is inserted every twelve sub-carriers in a symbol, the easiest way is using the two neighboring scattered pilots to interpolate the channel response of the eleven sub-carriers between them. Unfortunately, the variation in frequency domain is very serious and eleven sub-carriers space is too large to lead to an inaccuracy approximation. As Fig. 2.8 shows, the scattered pilots have three sub-carriers space shift for each successive symbol. Due to this characteristic, the space of the two neighboring scattered pilots will reduce to two by collecting the scattered pilots of four continuous symbols. This method supposes the time domain variation during four symbols is very small or even none. As Fig. 4.6 illustrates, the n^{th} symbol collects the scattered pilots from the $(n-3)^{th}$ to n^{th} symbols and uses frequency domain interpolation operation to approximate the channel response in the middle of the two scattered pilots as Fig. 4.7 shows. For the purpose to reduce the hardware complexity, the linear interpolation is adopted as frequency domain interpolation.

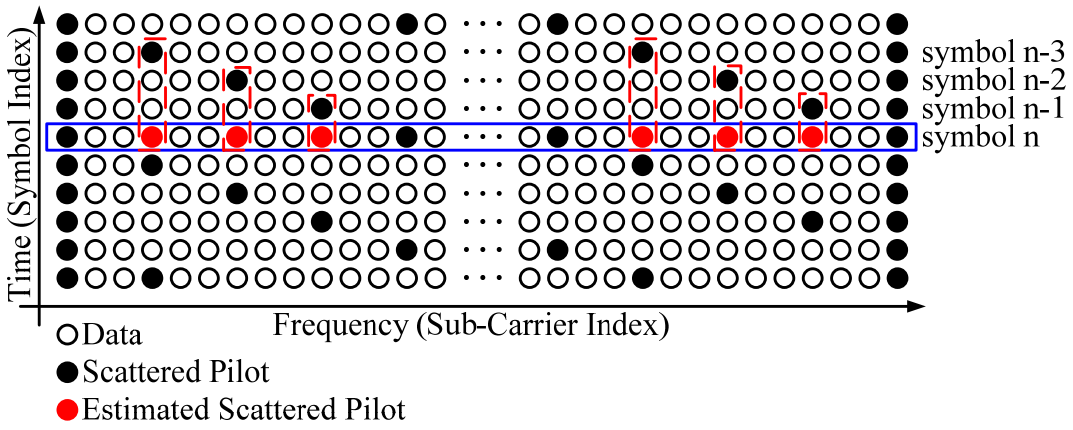


Fig. 4.6 Stored scattered pilots of 1-D channel estimation

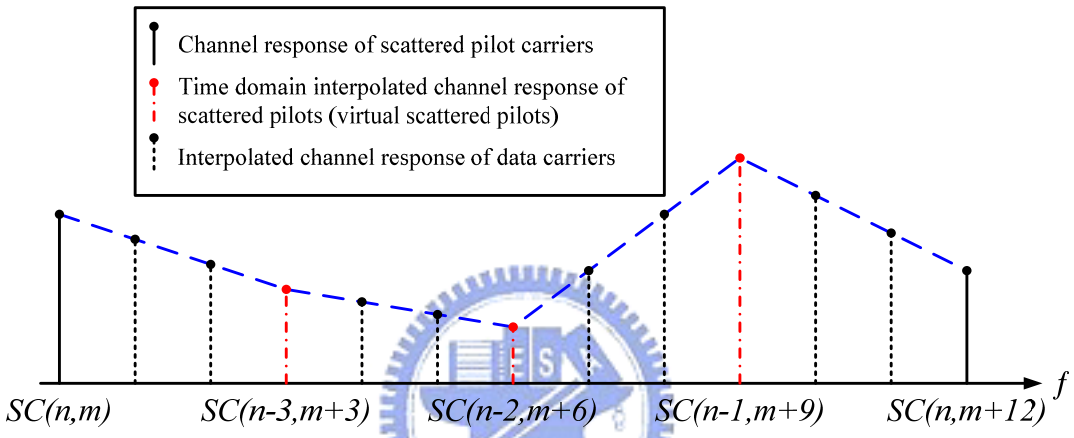


Fig. 4.7 Frequency domain linear interpolation of 1-D channel estimation

The 1-D channel estimation is described mathematically in Eqn. (4.3).

$$
\begin{aligned}
CR(n,m) &= \frac{SC(n,m)}{SC_{\exp}(m)} \\
CR(n,m+3) &= \frac{SC(n-3,m+3)}{SC_{\exp}(m+3)} \\
CR(n,m+6) &= \frac{SC(n-2,m+6)}{SC_{\exp}(m+6)} \\
CR(n,m+9) &= \frac{SC(n-1,m+9)}{SC_{\exp}(m+9)}
\end{aligned}
\tag{4.3}
$$

where the $CR(n,m)$ is the mth channel response in symbol n, $SC(n,m)$ belongs to scattered pilots and $SC_{exp}(m)$ is the mth expect value.

This method comes with a risk. If the channel response in time domain changes serious or not as small as expected, the scattered pilots those don't belong to current

symbol are not adequate to represent the channel response of current symbol. As a result, the wrong channel response is possibly leads to a worse performance. Therefore, the 1-D channel estimation algorithm only suits for time-invariant channel.

b) 2-D Channel Estimation

The 2-D channel estimation algorithm can take the time domain variation into consideration. In order to conquer the variation in time domain, the 2-D channel estimation uses the scattered pilots before and after the current symbol to interpolate a virtual scattered pilot of current symbol instead of collecting scattered pilots from the other three previous symbols. As Fig. 4.8, Fig. 4.9 and Fig. 4.10 shows, the 2-D channel estimation first collect the scattered pilots from seven continuous symbols and doing time domain linear interpolation to approximate the virtual scattered pilots. Then use the virtual scattered pilots and scattered pilots itself to do frequency domain interpolation. While the channel response changes linearly or not very serious during continuous five symbols, the 2-D channel estimation algorithm will get the best performance. But the 2-D channel estimation needs to store the sub-carriers at least three symbols before to get enough scattered pilots to do time domain interpolation. This means a large storage element is required, which is at least 6817×3 sub-carriers for 8K transmission mode.

錯誤! 物件無法用編輯功能變數代碼來建立。

Fig. 4.8 Stored scattered pilots of 2-D channel estimation

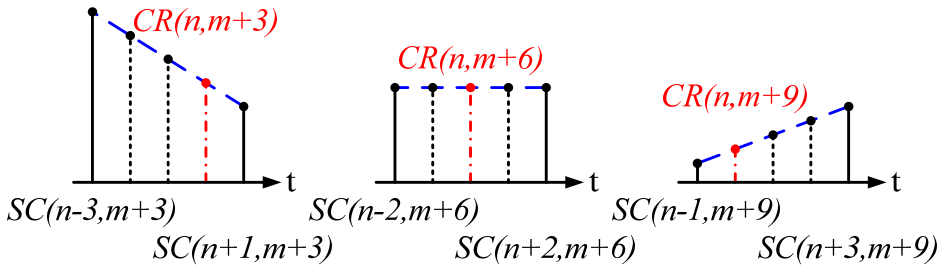


Fig. 4.9 Time domain linear interpolation of 2-D channel estimation

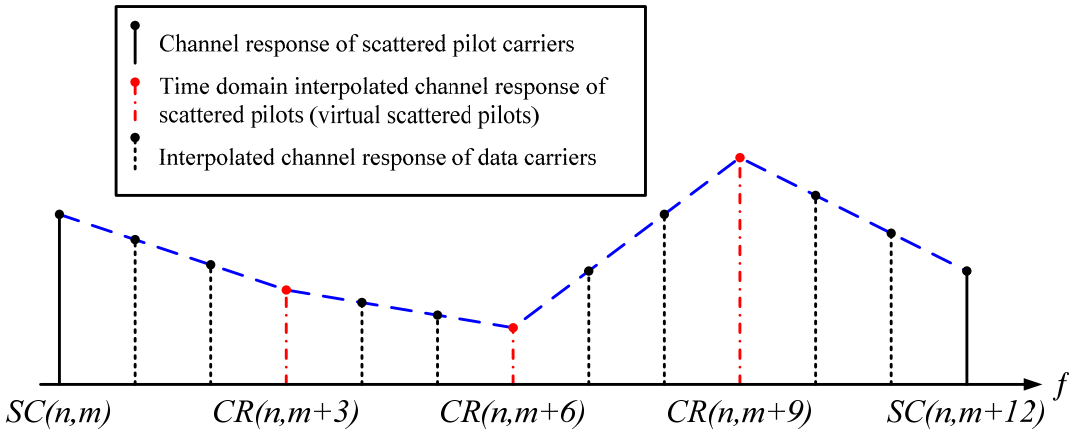


Fig. 4.10 Frequency domain linear interpolation of 2-D channel estimation

The 2-D channel estimation is described mathematically in Eqn. (4.4).

$$
\begin{aligned}
CR(n,m) &= \frac{SC(n,m)}{SC_{\exp}(m)} \\
CR(n,m+3) &= \frac{SC(n-3,m+3)\times 1 + SC(n+1,m+3)\times 3}{4\times SC_{\exp}(m+3)} \\
CR(n,m+6) &= \frac{SC(n-2,m+6)\times 2 + SC(n+2,m+6)\times 2}{4\times SC_{\exp}(m+6)} \\
CR(n,m+9) &= \frac{SC(n-1,m+9)\times 3 + SC(n+3,m+9)\times 1}{4\times SC_{\exp}(m+9)}
\end{aligned}
\qquad (4.4)
$$

c) 2-D Predictive Channel Estimation

For the purpose to solve the requirement of huge storage memories and keep the characteristic to conquer the time domain variation at the same time, a predictive 2-D channel estimation algorithm is offered by using time domain external interpolation instead of internal interpolation [26]. The 2-D predictive channel estimation uses two scattered pilots before current and doing external interpolation to predict the channel response of current symbol as shown in Fig. 4.11, Fig. 4.12 and Fig. 4.13. As a result, the storage element for three symbols' sub-carriers is saved and an extra storage element for scattered pilots is required. Comparing to the sub-carriers storage element, the storage element for additional scattered pilots is much smaller. But the predictive external time domain interpolation won't work as good as the internal time domain

interpolation because of the uncertain of prediction. Due to this reason, the 2-D predictive channel estimation is not as good as 2-D channel estimation but better than 1-D channel estimation in time-variant channel.

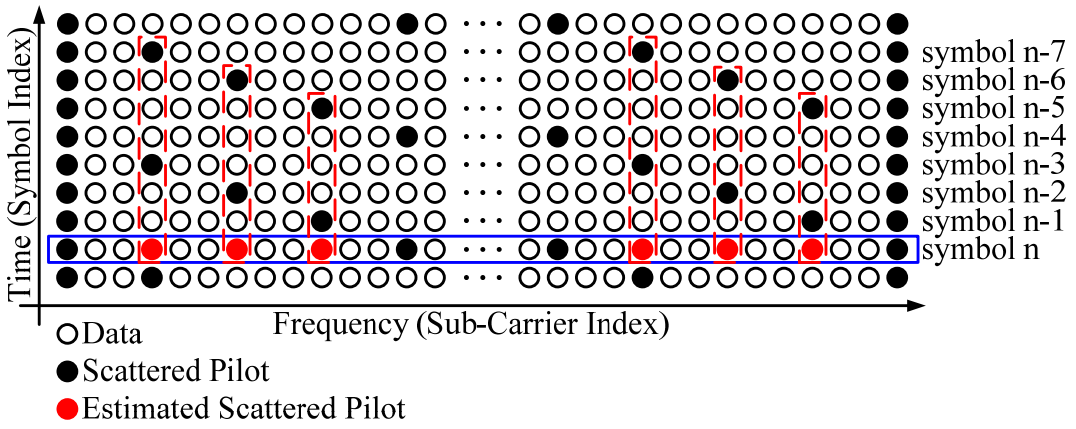


Fig. 4.11 Stored scattered pilots of predictive 2-D channel estimation

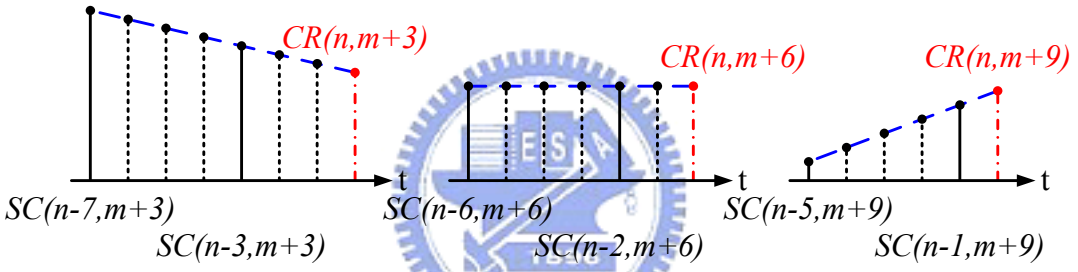


Fig. 4.12 Time domain linear interpolation of predictive 2-D channel estimation

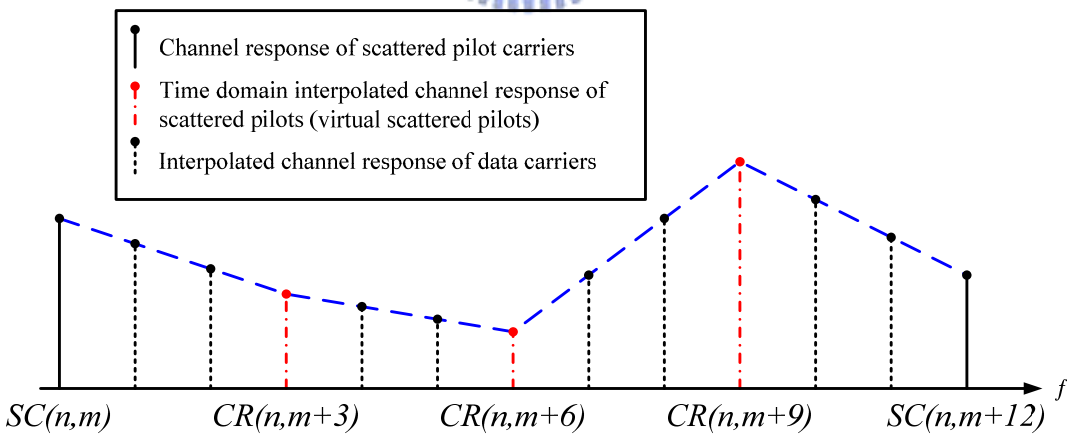


Fig. 4.13 Freq. domain linear interpolation of predictive 2-D channel estimation

The Predictive 2-D channel estimation is described mathematically in Eqn. (4.5).

$$
\begin{aligned}
CR(n,m) &= \frac{SC(n,m)}{SC_{\exp}(m)} \\
CR(n,m+3) &= \frac{SC(n-3,m+3)\times 7 - SC(n-7,m+3)\times 3}{4\times SC_{\exp}(m+3)} \\
CR(n,m+6) &= \frac{SC(n-2,m+6)\times 6 - SC(n-6,m+6)\times 2}{4\times SC_{\exp}(m+6)} \\
CR(n,m+9) &= \frac{SC(n-1,m+9)\times 5 - SC(n-5,m+9)\times 1}{4\times SC_{\exp}(m+9)}
\end{aligned}
\tag{4.5}
$$

3.4.2 Performance Simulation and Comparisons

The performance simulation results are shown in Fig. 4.14. The simulation environment is 2K transmission mode symbols with 1/4 guard interval, zero CFO, and surviving from Rayleigh channel with (a) 0Hz Doppler spread (b) 70Hz Doppler spread. Under the static channel, the predictive 2-D channel estimation algorithm is the worst because of the noise leads to a wrong predictive channel response. For dynamic channel, the predictive 2-D channel estimation algorithm is better then the 1-D channel estimation algorithm because of the channel varies. Overall, the 2-D channel estimation algorithm has the best performance.

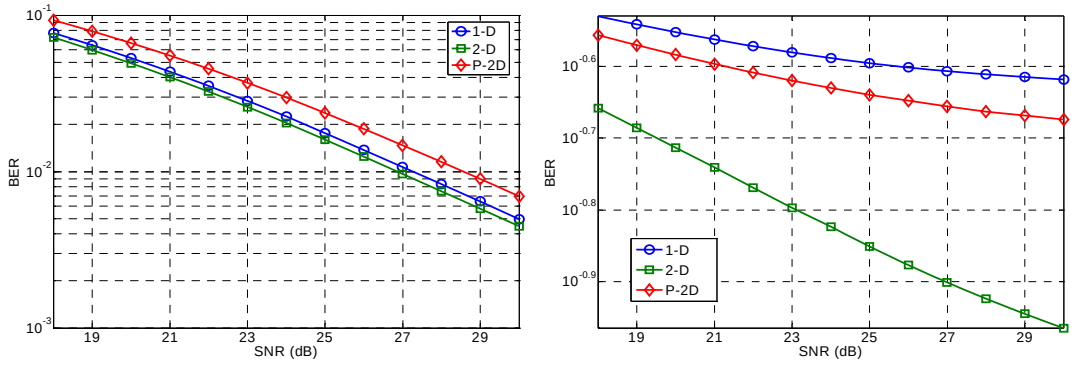


Fig. 4.14 BER under (a) static and (b) dynamic Channel

The hardware of the three algorithms is listed in Table 4-4.

Table 4-4 Required storage elements of channel estimation algorithms

	1-D	2-D	Predictive 2-D

	Channel Estimation	Channel Estimation	Channel Estimation
Scattered Pilot Storage Element	1+142×3 (427) 1+284×3 (745) 1+568×3 (1705)	1+142×3 (472) 1+284×3 (745) 1+568×3 (1705)	1+142×7 (996) 1+284×7 (1738) 1+568×7 (3978)
Data Pilot Storage Element	0	1705×3(5115) 3408×3(10224) 6816×3(20448)	0
Overall	427/745/1725	5587/10969/22153	996/1738/3978

The 1-D algorithm theoretically needs to store four symbols' scattered pilots. The scattered pilot number is 142/284/568 for 2K/4K/8K defined in [1]. The number for Mode 0 scattered pilots is one more than others. The $(n-3)^{th}$ symbol's scattered pilots are only used to compensation for the n^{th} symbol and can be overwrote by the n^{th} symbol's scattered pilots after used. Therefore, the maximum scattered pilot storage is reduced from four to three symbol's scattered pilots. The 2-D algorithm needs to store additional three symbols' pilots, which includes the scattered pilots. Therefore, the 2-D algorithm needs to store scattered pilots from the $(n-1)^{th}$ symbol to the $(n-3)^{th}$ symbol and all pilots from the $(n)^{th}$ symbol to the $(n+2)^{th}$ symbol. For predictive 2-D algorithm, it only needs to store scattered pilots form the $(n-1)^{th}$ symbol to the $(n-7)^{th}$ symbol and no data pilot is required to be stored.

Overall, the 2-D algorithm has the best performance with an unreasonable large storage elements requirement. The predictive 2-D algorithm has a better performance than 1-D algorithm under dynamic channel and a little worse performance under static channel. The storage elements required for predictive 2-D algorithm is less than that of the storage elements of mode/GI and boundary detection process. Thus, the predictive 2-D channel estimation algorithm is adopted as the channel estimation algorithm in this thesis.

4.3 Channel Compensation and Demapping

Intuitively, the channel compensation is implemented by using a divider, where the denominator is the value of estimated channel responses and the numerator is the received sub-carriers, and the output is for the demapper. The hard demapping process uses the result from divider and finds the distance to each point in QAM constellation map. After this, the point with minimum distance represents the result of hard demapping.

In [27] and [28], a divider free idea is referred. The idea is to multiply the denominator to both sides as shown in Eqn. (4.6).

$$X(m) = \frac{SC(n,m)}{CR(m)} \Rightarrow SC(n,m) = X(m) \times CR(m) \tag{4.6}$$

where *X(m)* is supposed to be the answer of *SC(n,m)/CE(m)* which is unknown yet.

Since the value of points in constellation map is known, replacing the *X(m)* by using all possible points in constellation map and finding the distances to all points are practicable. The point in constellation map with the minimum distance indicates that it is the most like-hood answer. Then the demapper outputs the demapping result which the point represents. This derivation can be thought as magnifying the constellation map by a scale which value equals to the denominator. However for the 64-QAM application in DVB-T, using exhaustive method means at least 63 comparators and 64 multipliers are necessary. Therefore, in [28] a three-stage demapping algorithm is jointed to reduce the hardware cost. By using dichotomy method, the *X(m)* can be replaced by axes and middle line of two neighboring points. Therefore the system needs few reference values stored, three multipliers and three comparators. Eqn. (4.7) illustrates the idea of Eqn. (4.6) in detail.

$$
\begin{aligned}
&\frac{SC_{re}(n,m)+iSC_{im}(n,m)}{CR_{re}(m)+iCR_{im}(m)} = X_{re}(m)+iX_{im}(m) \\
&= \frac{\left[SC_{re}(n.m)+iSC_{im}(n.m)\right]\times\left[CR_{re}(m)-iCR_{im}(m)\right]}{\left[CR_{re}(m)+iCE_{im}(m)\right]\times\left[CR_{re}(m)-iCR_{im}(m)\right]} \\
&= \frac{\left[SC_{re}(n.m)+iSC_{im}(n.m)\right]\times\left[CR_{re}(m)-iCR_{im}(m)\right]}{CR_{re}(m)^2+CR_{im}(m)^2} \\
&\Rightarrow \left[SC_{re}(n.m)+iSC_{im}(n.m)\right]\times\left[CR_{re}(m)-iCR_{im}(m)\right] \\
&\quad = \left[X_{re}(m)+iX_{im}(m)\right]\times\left[CR_{re}(m)^2+CR_{im}(m)^2\right]
\end{aligned}
\tag{4.7}
$$

let

$$F_1 = \left[SC_{re}(n.m)+iSC_{im}(n.m)\right]\times\left[CR_{re}(m)-iCR_{im}(m)\right]$$

$$F_2 = \left[CR_{re}(m)^2+CR_{lm}(m)^2\right]$$

where $SC_{re}(n,m)$ and $SC_{im}(n,m)$ are the real part and image part of $SC(n,m)$ and so do $CR_{xx}(m)$ and $X_{xx}(m)$. There are two possible choices to cancel the denominator. One is to multiply the denominator to the right side directly. Therefore, the left side is a complex number and right side becomes to a complex number multiplies a complex number. The second choice is to multiply the conjugate of denominator to the numerator and denominator. After multiply operation, the numerator is still a complex number but the denominator turns to an integer. Then move the integer denominator to right side and the operation is done. This transfer operation makes a complex division turn to two integer divisions and then makes the right side multiply the denominator as a scale. Comparing to the first one, the second operation uses an integer multiplier, which can be simplified by several adders, to scale the constellation map instead of a complex multiplier. Therefore, the power consumption and hardware cost are able to be reduced in the hard demapping process which will be discussed later.

After partitioning the equation into real and image parts, the demapping process is able to demap the real part and image part separately. In [1], all possible situations were described and the values of all positions were also defined. As a result, all the

possible middle line values are stored previously. Fig. 4.15 represents the values of F_1 in Eqn. (4.7).

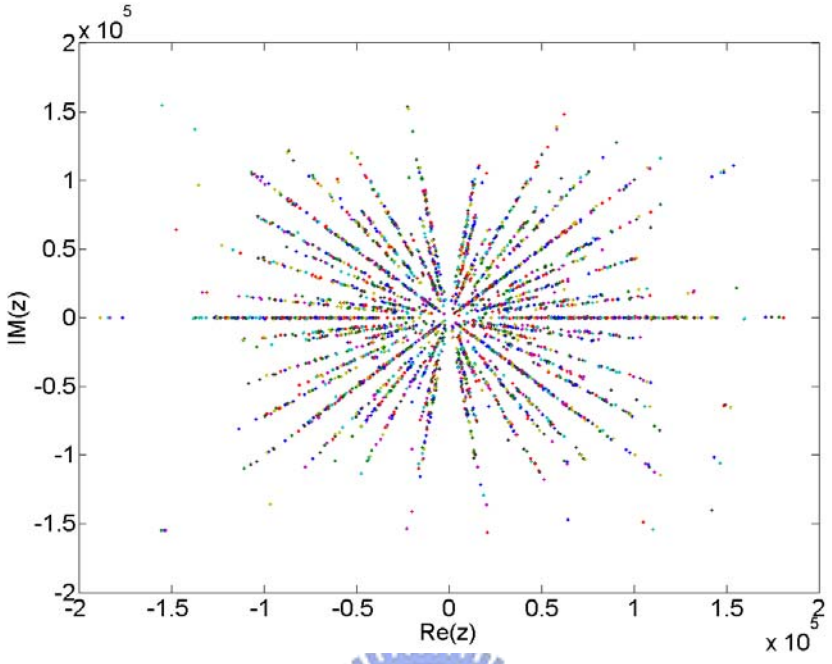


Fig. 4.15 Division simplification results in QAM

A divider free and fast three-stage real and image individual hard demapping process is proposed below:

1. *Get the sign of Re: if Re<0 y0=1, else y0=0*
2. *If constellation mode is 16-QAM or 64-QAM go to 3, otherwise go to 7.*
3. *If 16-QAM B=α+1, else B=α+3*
4. *If |Re|>B×NF×INT y2=0, else y2=1.*
5. *If constellation mode is 64-QAM go to 6, otherwise go to 7.*
6. *If y2=0, B=B+1: if |Re|>B×NF×INT y4=0, else y4=1*
   else B=B-1: if |Re|>B×NF×INT y4=1 else y4=0
7. *y0,y2,y4 is demapped.*

Repeat the same procedure to demap the y1,y3,y5.

In this procedure, the *Re* is the real part of the numerator, *INT* is the denominator F_2 of Eqn. (4.7), *NF* is the normalized factor referred in [1] and *B* is the decision boundary (the X-axis/Y-axis value between two neighboring points in X-axis/Y-axis aspect). As a result of the derivation above, the system can magnify the constellation map by a scale which value equals to the denominator *(INT)*. Then detect what region in the magnified constellation map does the F_1 locates on.

Fig. 4.16 (b) is the block diagram of proposed three-stage hard demapper, where the function of stage 1 is the as same with Eqn. (4.7), which will first examine the sign of F_1 to decode the y0 and y1. Because of the first reference values are X axe and Y axe, the multiplier is able to be saved. After stage 1, stage 2 uses the information from stage 1 to choose a decision boundary value from Fig. 4.17 (a). The system will tell the F_1 value is bigger or less then the value of decision boundary multiplies the denominator *(INTxBxNF)* to decide the region F_1 locates. Stage 3 repeats the actions of stage 2 and uses the information from stage 2 and decision boundary shown in Fig. 4.17 (b).

As shown in Fig. 4.16 (a), before the constellation mode is detected by TPS decoder, the system use stage 1 to decode y0 and y1 as QPSK mode. If the constellation mode is 16-QAM, stage 2 starts to decode y2 and y3. Stage 3 only works at 64-QAM mode. By using this hierarchical decoding scheme, the system can disable the useless stages to reduce the unnecessary power consumption. For example, stage 2 and stage 3 are disabled while doing QPSK demodulation.

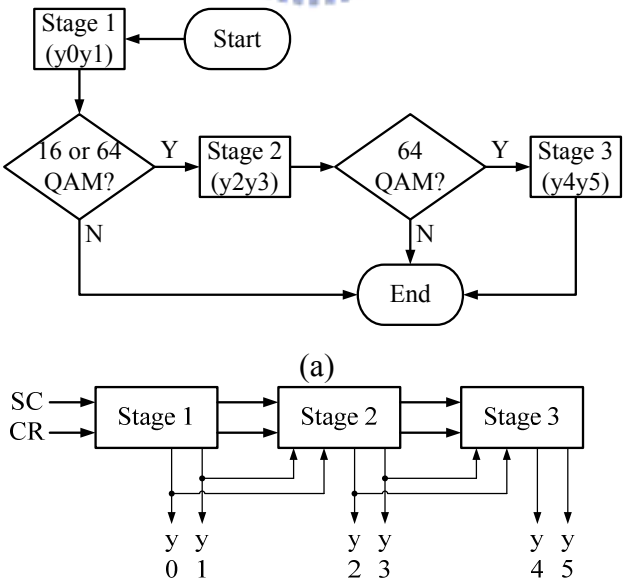
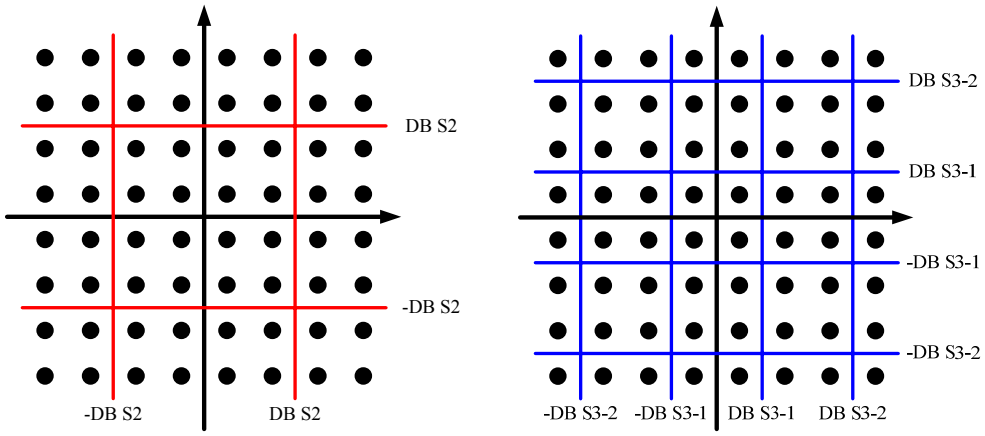


Fig. 4.16 (a) FSM and (b) block diagram of hard demapping



(a) (b)

Fig. 4.17 Decision boundary of (a) stage 2 and (b) stage 3

Fig. 4.18 is an example of the three-stage demapping process. The three-stage demapper demapps the real and image parts individually and using their intersection to finding the demapping results.

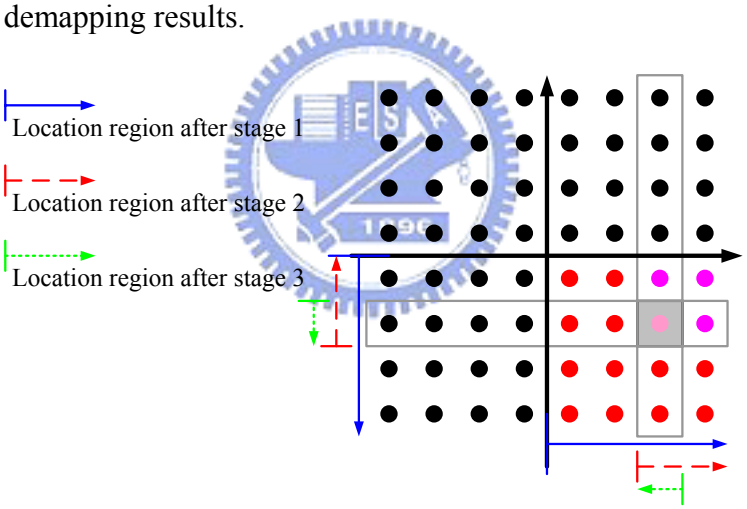


Fig. 4.18 Example of three-stage demapping process

4.4 Timing Synchronization Scheme

This section illustrates the demodulation flow of the proposed DVB-T inner receiver. First, the system starts with doing mode/GI, symbol boundary detection and fractional carrier frequency offset detection. After 2~3 symbols, the mode/GI and symbol boundary are detected and FFT controller uses the information to decide the FFT window length and location. After FFT begins to output useful results, ICFO

spends two symbols to detect and repair the integer carrier frequency offset. After this, the SCO and residual CFO tracking loops start to work. The scattered pilot mode detection spends one symbol to detect scattered pilot mode. After the scattered pilot mode is detected, the system spend five symbols to finishing the scattered pilot filling then start channel estimation and demapping processes.

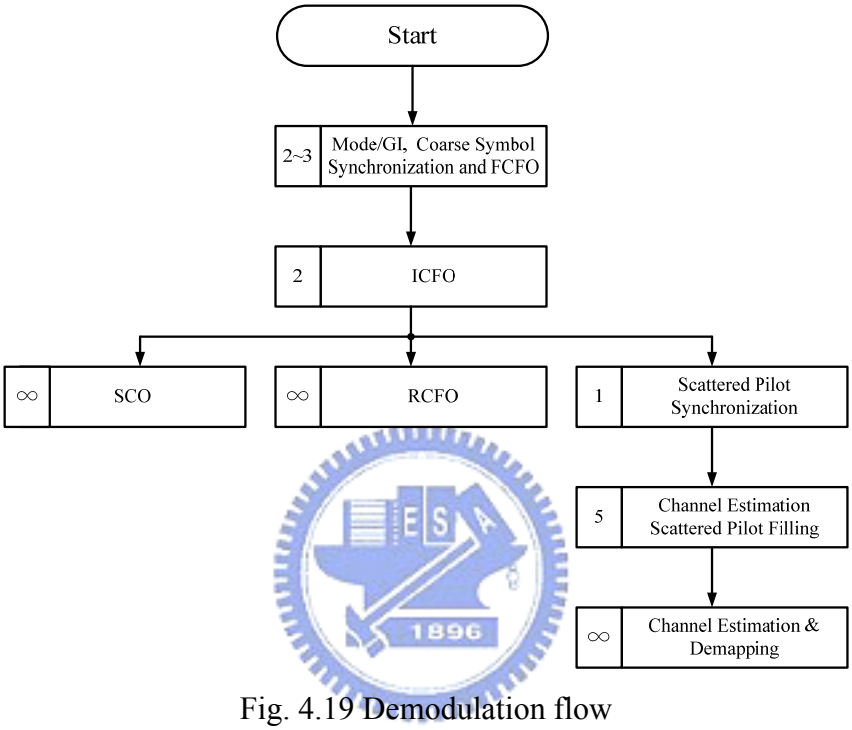


Fig. 4.19 Demodulation flow

Chapter 5

Architecture Design and Implementation

5.1 Architecture of DVB-T Inner Receiver

Fig. 5.1 is the proposed DVB-T inner receiver architecture and the highlighted blocks are the design targets in this chapter. This chapter will focus on mode/GI & symbol boundary detection, scattered pilot mode detection, channel estimation and demapper design. As the demodulation flow shown in Fig. 4.19, the architectures can be divided into three different timing sequences. According to this characteristic, hardware sharing and low power design will also be discussed in this chapter.

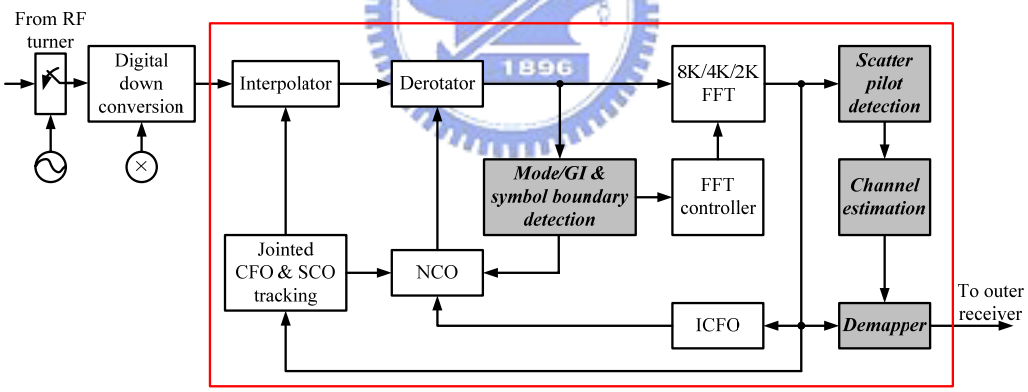


Fig. 5.1 Architecture of DVB-T inner receiver

5.2 Mode/GI & Symbol Boundary Detection

In this section, the mode/GI and boundary detection architecture shown in Fig. 3.16 will be realized by two major blocks, memory-based delay-lines and correlation block. First, the correlation and moving sum delay-lines will be implemented using memories. And a twister memory access method will be described. Second, the

correlation part will be simplified.

5.2.1 SRAM Based Delay-Line

Intuitively, the delay-line can be realized using shift registers. For DVB-T application, to realize a longest 8K and shortest 2K correlation delay-line with shift register is very ridiculous. Using the SRAM module to accomplish the delay-lines helps to reduce power consumptions and area very large. Since the SRAM is adopted to realize the delay-line, there are two kinds of SRAM modules to be the candidates, single port SRAM and dual port SRAM offered by TSMC or UMC. Using an 8K indexes dual port SRAM to realize the 8K delay-line is the intuitive way. But the hardware cost for a dual port SRAM is still too large. Therefore, the design can be realized using two 4K single port SRAM with interleaved read/write access to replace the dual port SRAM. Fig. 5.2 shows the read/write operation for 8K application of dual port and single port SRAM-based delay-line.

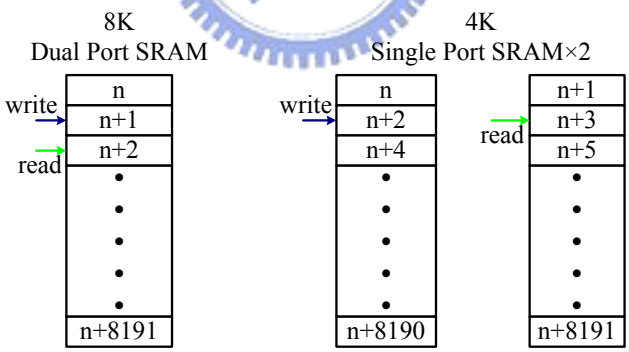


Fig. 5.2 R/W operation of dual and single port SRAM-based delay-line

In this thesis, the correlation delay-line and moving sum delay-line are composed by 1K single port SRAM modules for the purpose to share the memory elements with channel estimation. Moreover, in order to used the SRAM modules efficiently and reduce the delay-line filling time while the mode changes, a twister circular memory access operation method is proposed. Generally, for the 2K application, the delay-line uses two of the eight 1K SRAM modules or the lower part of the 4K SRAM modules

as shown in Fig. 5.3. As a result, the *r(n-2K)* is overwritten as soon as *r(n)* is written. When the detection mode changes to 4K, the oldest data in delay-line is 2K samples ago that means another 2K samples must be filled or replenished. This movement causes a 2K samples waiting time and while changing from 4K to 8K application another 4K waiting time is required. For the 2K and 4K application, not all of the SRAM modules or memory cells are accessed. As a result, there comes the power distribution issue.

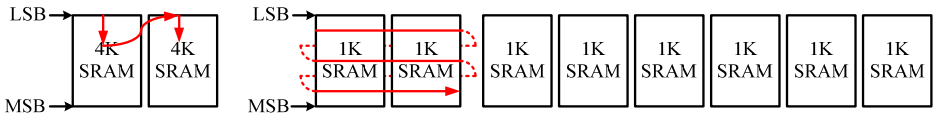


Fig. 5.3 2K delay-line application

The twister circular memory access operation method has the ability to overcome the waiting time and power distribution problems. As Fig. 5.4 shows, there is only one writing mode and the write process writes the data using all of the 1K SRAM modules as access direction in spite of the detection mode. There are three reading mode, 2K-1, 4K-1 and 8K-1 later than writing address. The read process changes the read access according to the different detection mode. When the detection mode changes form 2K to 4K at signal *r(n)*, signal *r(r-4K)* is still stored in delay-line because of the special writing behavior. Therefore, no additional waiting time is necessary to refill or replenish the delay-line and each memory cell is in probability to be accessed fairly.

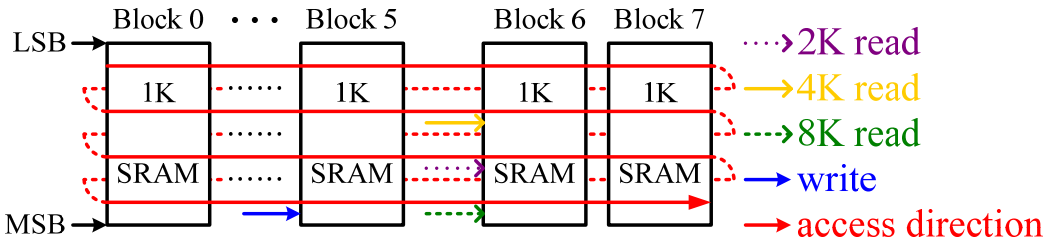


Fig. 5.4 Twister memory access operation

According to simulation results, six bits for *r(n)* is enough to detect a maximum correlation result. For the purpose to support the twelve bits word-length of FFT

output as defined by [14], the I/O bits for SRAM are defined as twelve bits. For correlation delay-line, the real part and image part will be stored together as twelve bits. For moving sum delay-line, the input bits are twelve bits due to the multiply operation and the twelve bits are wrote to the same 1K SRAM. The upper moving sum delay-line needs to store the complex result for 2K long therefore it needs four 1K SRAM modules. The lower one only needs to store a real number for 2K long and two 1K SRAM module is enough. Overall, totally (8+4+2)×1K×12bits=168Kbit cells are required. Overall, the area comparison for four correlation delay-line is listed in Table 5-1.

Table 5-1 Area of four correlation delay-line realization methods

	Shift Register	8K Dual Port SRAM	4K Single Port SRAM×2	1K Single Port SRAM×8
Area	5.56 mm^2	1.56mm^2	0.74mm^2(0.37×2)	1.04 mm^2(0.13×8)

In summary, the area of eight 1K single port SRAM modules saves 50% area than an 8K dual port SRAM. This is a great gain though its area is bigger than two 4K single port SRAM. The twister circular memory access operation method not only save the waiting time problem but also reduce the power distribution problem with a little control penalty.

5.2.2 Correlation Circuits

As shown in Fig. 3.16, the correlation part includes a complex multiplier and a multiplier. Eqn. (5.1)shows that the multipliers of the complex multiply operation can be reduced by additional two adders and a subtractor [29]. Therefore four multipliers and two adders are reduced to three multipliers and five adders as shown in Fig. 5.5. Therefore, more than 12% area is saved and 4% power consumption is improved in complex multiplier design.

$$
\begin{aligned}
&(A+Bi)\times(C-Di)\\
&=(AC+BD)+i(BC-AD)\\
&=(AC+BD)+i\left[(A+B)\times(C-D)-AC+DB\right]
\end{aligned}
\tag{5.1}
$$

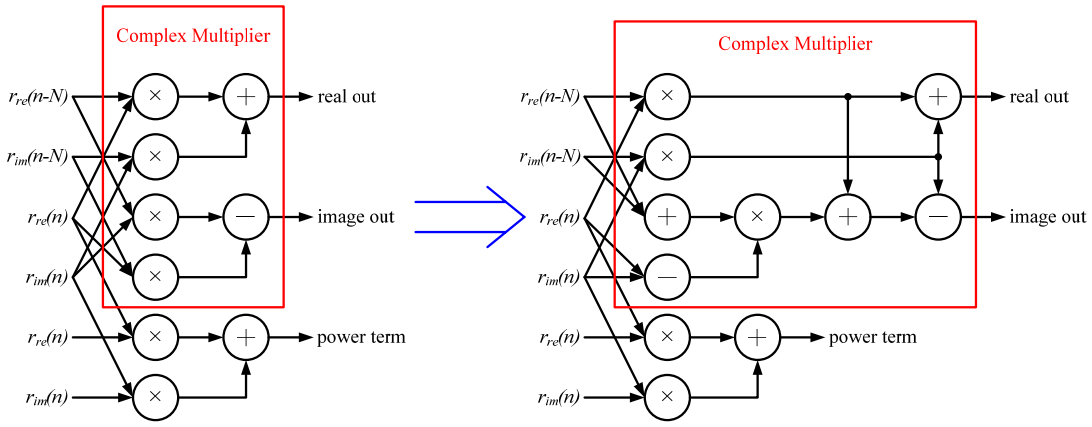


Fig. 5.5 Complex multiplier reduction

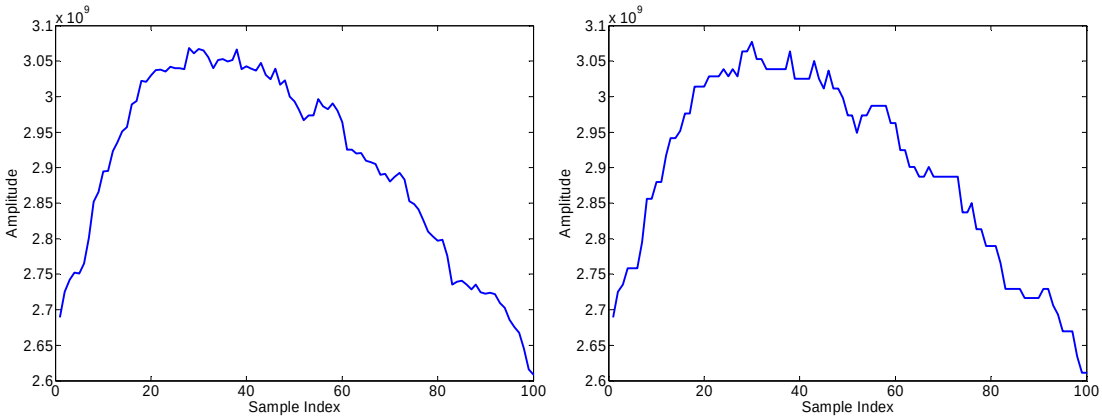
5.2.3 Moving Sum Circuits

As shown in Fig. 3.16, the two output of Gate block at the output of moving sum delay-line is set to zero at “Dummy State” for the purpose to do the initial integration. After this, the moving sum delay-line output value pass the Gate block and the value from “D” will subtract it to do the moving sum function. The square operation is realized by a complex multiplier and an integer multiplier.

5.2.4 Finite Word-Length Simulation

In order to prevent from the word-length is going to be bigger and bigger, the finite word-length simulation is necessary. For the purpose to reduce the multiplier size of square operation, the input to register “D” is the target. Due to the value of integration of moving sum is not necessary 23 bits (12 bits with 2K integration length), the MSB bit is cut and the word-length is reduced from 23 to 19 after maximum value simulation. As a result of the LSB bits are too small to influence the result after square operation, the input of square only take [18:8] 11 bits of “D”. Thus, the multiplier size

is successively reduced from 23×23 to 11×11. A piece of MC^2 result around peak period before and after finite word-length truncation is shown in Fig. 5.6. The detail architecture of the proposed blind mode/GI and boundary detection is shown in Fig. 5.7.



(a) (b)

Fig. 5.6 MC^2 results (a) before (b) after finite word-length simulation

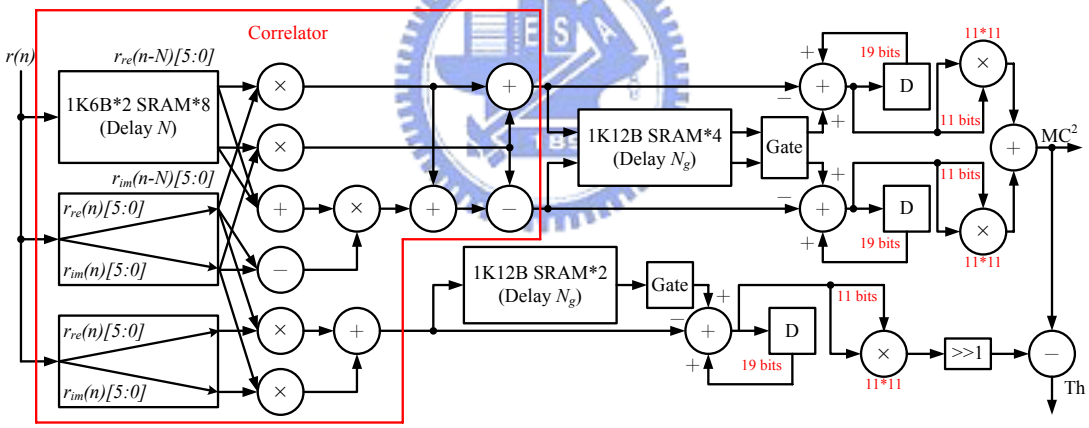


Fig. 5.7 Improved architecture of blind mode/GI and boundary detection

5.3 Scattered Pilot Synchronization

The block diagram architecture of scattered pilot mode detection is shown in Fig. 4.2. The complex multiplication is realized by two multipliers and an adder. As a result of the structure is the same with the square operation in Fig. 5.7, this part is shard from the blind mode/GI and boundary detection after the boundary is detected. The MUXs in Fig. 4.2 is replaced by using gated clock to reduce the power

consumption and area cost.

Since the inputs of the multiplier are the same, the sign bit is able to be ignored. After the finite word-length simulation, the complex multiplication output value can be represented using the MSB 7 bits and the accumulation value can be represented using 11 bits. Therefore, a register group is reduced from 34 to 11 bits. Overall, the improved architecture of scattered pilot mode detection is shown in Fig. 5.8.

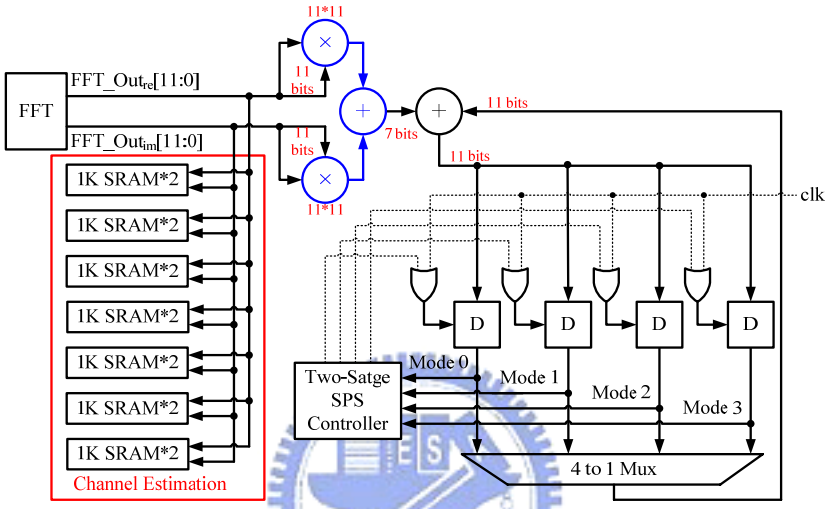


Fig. 5.8 Improved architecture of scattered pilot mode detection

5.4 Frequency Domain Channel Estimation

Since the channel estimation needs to store the scattered pilots of seven symbols, the storage element is shared from the fourteen 1K SRAM modules of mode/GI boundary detection after the boundary is detected. Therefore, the fourteen 1K SRAM module is divided into seven groups and each group consists two 1K SRAM module to store the real and image part of scattered pilots. Each group only stores one symbol's scattered pilots.

Since the scattered pilot is stored by seven 1K×2 SRAM modules, the read/write conflict is possibly occurred. For example, as shown in Fig. 5.9, the gray area means the same address index for the SRAM module. While the lower red scattered pilot

stores to a 1K SRAM module, the previous upper red scattered pilot is overwritten. After two cycles, the channel estimation needs the upper red scattered pilot to do channel estimation and it is covered. Thus, the R/W conflict leads to a wrong scattered pilot is read. In order to prevent this situation, the scattered pilot in the same address index will be pre-read before the new scattered pilot is stored. Therefore, the registers are required to hold the pre-read scattered pilots from the output of SRAM modules.

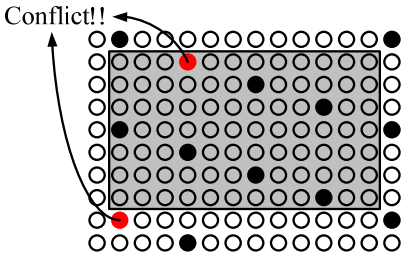


Fig. 5.9 Scattered pilot R/W conflict

The four time domain interpolation equations in Eqn. (4.5) can be modified as shown in Eqn. (5.2) by decomposing the scale numbers. Thus, the multiplication is transferred to shift and addition or subtraction. Therefore, four multipliers are saved.

$$
\begin{aligned}
CR(n,m) &= \frac{SC(n,m)}{SC_{\exp}(m)} \\
CR(n,m+3) &= \frac{(8-1)\times SC(n-3,m+3)-(2+1)\times SC(n-7,m+3)}{4\times SC_{\exp}(m+3)} \\
CR(n,m+6) &= \frac{(4+2)\times SC(n-2,m+6)-(2)\times SC(n-6,m+6)}{4\times SC_{\exp}(m+6)} \\
CR(n,m+9) &= \frac{(4+1)\times SC(n-1,m+9)-(1)\times SC(n-5,m+9)}{4\times SC_{\exp}(m+9)}
\end{aligned}
\qquad (5.2)
$$

Furthermore, the addition and subtraction operation is implemented using CSA (Carry Save Adder) and CPA (Carrier Propagation Adder) to reduce the power consumption and area cost.

The frequency domain interpolation is shown in Fig. 5.10. The non-black CRs are interpolated by the block CR before and after themselves. Therefore, two sample

delay registers is used to hold the black CR before the non-black CR.

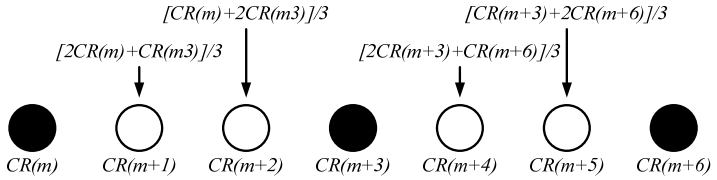


Fig. 5.10 Frequency domain interpolation

Because of there is no multiplication in channel estimation and the finite word-length does not improve too much, the word-length of channel estimation is not truncated. Overall, the architecture of channel estimation is illustrated in Fig. 5.11.

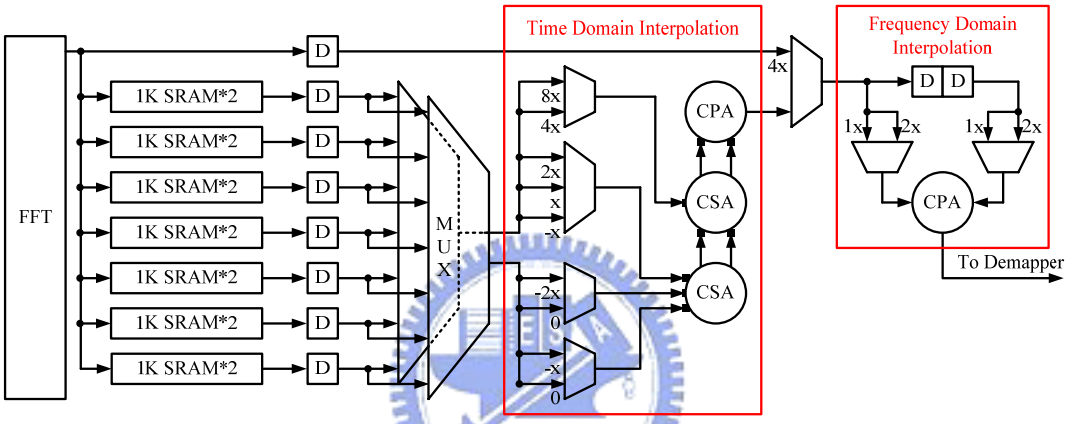


Fig. 5.11 Architecture of channel estimation

5.4 Channel Compensation and Hard Demapper

5.4.1 Hardware Design

The structure of Eqn. (4.7) is as the same as shown in Fig. 5.5. The F_1 is the same with the correlation part and the F_2 with the power term. Therefore, the first stage architecture is shared from the mode/GI and boundary detection after the boundary is detected.

The value of *BxNF* in chapter 4.3 is pre-calculated using the normalized factor, decision boundary and α defined in [1]. According to the simulation result, the *BxNF* can be represented in five bits with only 1% performance loss while the performance

loss of four bits increases very much. Therefore, the possible B×NF values is listed in Table 5-2.

Table 5-2 Possible B×NF values of stage 2

QAM	α	B×NF	R-Bits
16-QAM	1	$2\sqrt{10/10}$	101 00
	2	$3\sqrt{20/20}$	101 01
	4	$5\sqrt{52/52}$	101 10
64-QAM	1	$4\sqrt{42/42}$	101 00
	2	$5\sqrt{60/60}$	101 01
	4	$7\sqrt{108/108}$	101 10

where the B×NF is represented in 5 bits as R-Bits. And the possible B×NF values of stage 3 is listed in Table 5-3. By using CSD technique, the maximum non-zero digital is reduced to three from four. That means the number of adders and power consumption are also reduced.

Table 5-3 Possible B×NF values of stage 3

QAM	α	B×NF	R-Bits	R-Bits(CSD)
64-QAM	1	$2\sqrt{42/42}$	01010	0 0 1 0 1 0
	2	$3\sqrt{60/60}$	01100	0 1 0 -1 0 0
	4	$5\sqrt{108/108}$	01111	0 1 0 0 0 -1
	1	$6\sqrt{42/42}$	11110	1 0 0 0 -1 0
	2	$7\sqrt{60/60}$	11101	1 0 0 -1 0 1
	4	$9\sqrt{108/108}$	11100	1 0 0 -1 0 0

Therefore, the scaling operation is replaced by addition and subtraction. For the purpose to reduce the power consumption and hardware cost, the addition and subtraction are realized by CSA and CPA. Due to the CSD effort, the maximum CSA number is reduce to two from three.

5.4.2 Finite Word-Length Simulation

In order to reduce the multipliers hardware cost of stage 1, the finite word-length is simulated. As shown in Fig. 5.12, the bit-error-rate (BER) increases dramatically

while the bit number is smaller than twelve. Thus, the architecture of stage 1 is illustrated in Fig. 5.13 with the word-length is dramatically reduced. The stage 2 and real part of stage 3 architecture is shown in Fig. 5.14.

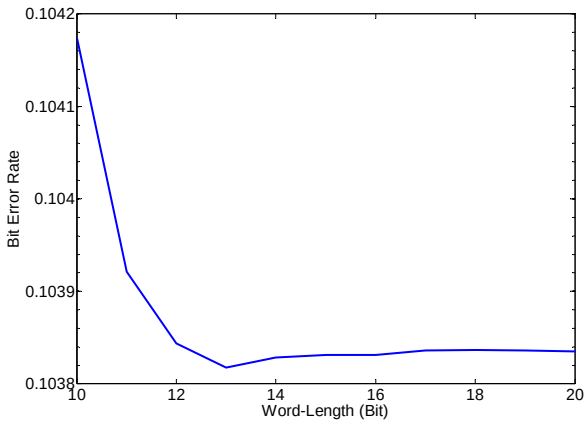


Fig. 5.12 Finite word-length of BER after demapping

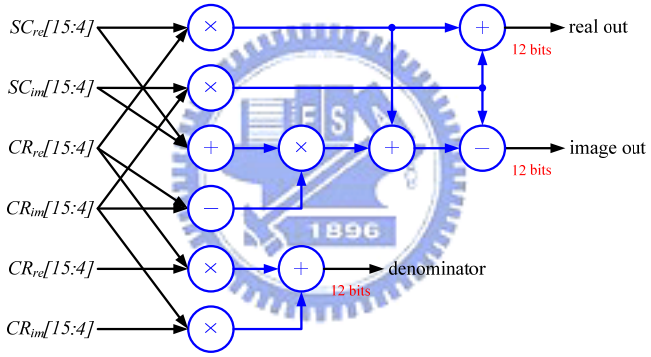
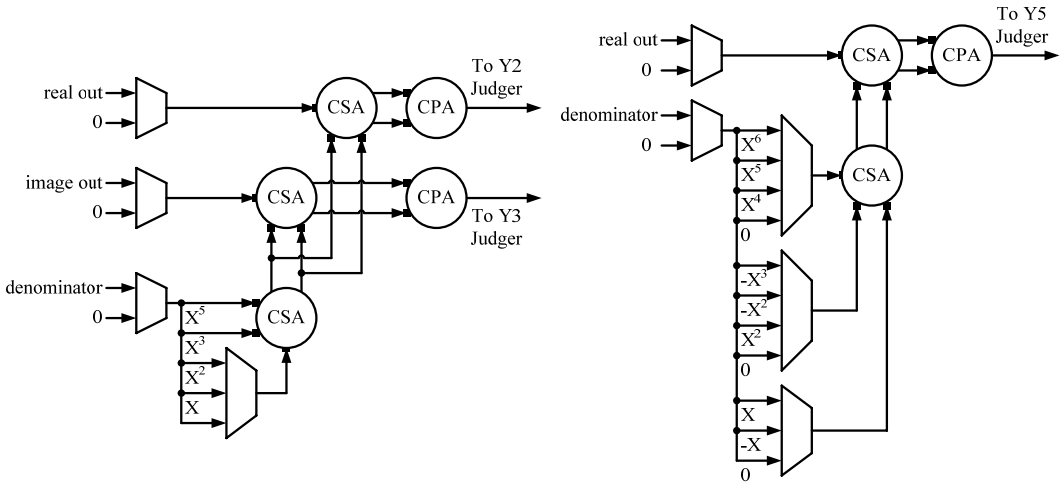


Fig. 5.13 Architecture of stage-1



(a) (b)

Fig. 5.14 Architecture of (a) stage-2 and (b) stage-3

5.5 Design and Implementation Results

The blind mode/GI and boundary detection, scattered pilot mode detection, channel estimation and demapping in this thesis are synthesized using TSMC 0.18 um process. The synthesis tool is Synopsys Design Complier and the operation is set at 10MHz and slow condition. The synthesis results are listed in Table 5-4. Moreover, the statistics of all blocks are listed in Table 5-5.

Table 5-4 Design results

Synthesis Results	
Process	TSMC 0.18 um (1.62 v)
System Required Speed	9.14MHz
Power	3.93mW@10MHz
Maximum Delay/Freq.	23.70ns/42.19MHz
Gate Counts (w/MEM)	209,676(2,054,824um2)
Gate Counts (w/o MEM)	18,385(180,174um2)
Gate Counts (MEM)	191,291(1,874,650um2)

Table 5-5 Statistics of all blocks

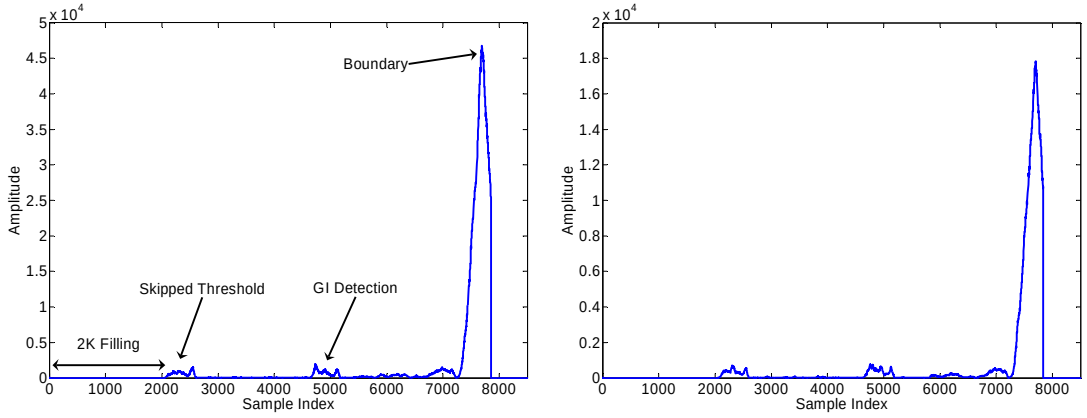
	Before Improved		After Improved		Improve Ratio
	Area	Ratio	Area	Ratio	
CSS Ctrl	20,567um2	0.95%	17,264um2	0.84%	16.06%
CSS Combination	32,379um2	1.50%	17,637um2	0.86%	45.53 %
SPS & CE Ctrl	26,721um2	1.24%	15,624um2	0.76%	41.53%
CE Combination	40,033um2	1.86%	38,783um2	1.89%	3.12%
Hard Demapper	23,118um2	1.07%	17,291um2	0.84%	25.21%
Shared Modules	123,290um2	5.72%	62,693um2	3.05%	49.15%
Others	14,414um2	0.67%	10,882um2	0.53%	24.50%
Total (w/o Mem)	280,522um2	13.02%	180,174um2	8.77%	35.77%

The “CSS Ctrl” and “CSS Combination” are the controller and combinational part of blind mode/GI and boundary detection process, the “SPS & CE Ctrl” and “CE Combination” includes the circuits of scattered pilot mode detection and channel estimation. “Hard Demapper” consists stage 2 and stage 3 of demapping process.

"Share Modules" are organized by the correlation circuits shown in Fig. 5.13 and a complex multiplier, which are shared with blind mode/GI and boundary detection process, scattered pilot mode detection and demapping process. The "Memory Bank" is the fourteen 1KSRAM modules, which takes a large part of area (91%).

With memory sharing, the area of memory is 1.04 mm^2. Without the memory sharing skill, addition two 4K SRAM and two 1K SRAM are required. Therefore, the memory area will increase to 2.56 mm^2. Since the memory takes a large part of the area, to do the memory sharing really helps a lot to improve the area cost (59.38%). The shared modules scheme proposed in this thesis also save large area. Finally, the area without memory is reduced greatly as a result of doing finite word-length truncation.

In order to prove the design really work, the simulation results of RTL and gate level is shown in Fig. 5.15, where the skipped threshold is the protection mechanism to prevent computing the wrong guard interval length.



(a) (b)

Fig. 5.15 MC^2 results of (a) RTL (b) Gate level simulation

Chapter 6

Conclusion and Future Work

In this thesis, a modified architecture of Normalized-Maximum-Correlation (NMC) architecture, which holds the normalized mode/GI detection advantage without a division operation and using the Maximum-Correlation (MC) part to determine the symbol boundary, is proposed. This thesis also proposes an efficient mode/GI and symbol boundary detection scheme. By detecting one transmission mode at a time, the hardware is reduced to a single hardware. For some mode detection method, the delay-line is refilled or replenished while the tested mode changes. To overcome the delay penalty during the tested mode changes, a twister memory access method is offered to eliminate the refilling/replenishing penalty and without the refilling/replenishing penalty the mode/GI and boundary detection timing has a 41.56% (12,288 samples) reduction. Furthermore, the twister memory access method does not only reduce the delay-line refilling/replenishing penalty but also reduce the power distribution problem by accessing all memory blocks fairly. To reduce the unexpected accident scattered pilot synchronization error, a two-stage scattered pilot synchronization scheme, which can be organized by the power-based and the correlation-based or other scattered pilot synchronization algorithms, is proposed to improve the reliability of the scattered pilot synchronization result. With the channel estimation scattered pilots pre-filling scheme, the scattered pilot synchronization timing and channel estimation scattered pilot filling timing is overlapped. The scattered pilot synchronization timing is efficiently reduced from 68 symbols (TPS decoding time) to zero symbol for idea and two symbols if an error

occurs. Therefore, the scattered pilot synchronization timing is reduced by at best 100% to 97.06%. A three-stage division-free demapping scheme is also proposed to save a frequency domain equalizer (FEQ) and realized an all division-free architecture in this thesis. Finally, the memory and multipliers sharing efficiently reduce 43.51% (3,637,517um^2→2,054,824um^2) area cost.

In the future, the channel estimation algorithm and architecture will be improved to against whether static or dynamic channels well with a small hardware cost. The inner receiver architecture , including the mode/GI and boundary detection, Fast Fourier Transform (FFT), scattered pilot synchronization (SPS), frequency domain channel estimation (CE), demapping, carrier frequency offset (CFO) recovering loop and sampling clock offset (SCO) recovering loop, will also be integrated and taped out.

Reference

[1] "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television," *European Telecommunication Standard* EN 300 744 V1.5.1, Nov. 2004.

[2] "Transmission System for Handheld Terminals (DVB-H)," *European Telecommunication Standard* EN 302 304 V1.1.1 Nov. 2004.

[3] "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services," *European Telecommunication Standard* EN 300 421 ed.1, Dec. 1994.

[4] "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for cable systems," *European Telecommunication Standard* EN 300 429 V1.2.1, Apr. 1998.

[5] "ATSC Digital Television Standard," *ATSC Standard* A/53E, Dec 2005.

[6] "Terrestrial integrated services digital broadcasting (ISBD-T)," *ARIB Standard* STD-B31 V1.5, Jun. 2004.

[7] "Digital Multimedia Broadcasting," *Telecommunications Technology Association in Korea*, 2003SG05.02-046, 2003.

[8] http://www.dtvc.org.tw/.

[9] http://info.gio.gov.tw/.

[10] M. Hosemann, G. Cichon, P. Robelly, H. Seidel, T. Drger, T. Richter, M. Bronzel, and G. Fettweis, "Implementing a receiver for terrestrial digital video broadcasting in software on an application-specific DSP," *IEEE SIPS.2004*, Oct. 2004, pp. 53-58.

[11] C. D. Toso, P. Combelles, J. Galbrun, L. Lauer, P. Penard, P. Robertson, F. Scalise, P. Senn, and L. Soyer, " 0.5 um CMOS Circuits for Demodulation and Decoding of an OFDM-Based Digital TV signal Conforming to the European DVB-T Standard," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 11, pp. 1781-1792, Nov. 1998.

[12] Y-J. Chen; Y-C. Lei; T-D. Chiueh, **"**Baseband transceiver design for the DVB-terrestrial standard," *IEEE APCCAS*, Vol. 1, Dec. 2004, pp. 389-392.

[13] 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

[14] T-Z. Wei, "Design of Carrier Recovery for DVB-T Baseband Receiver," Master Thesis, Department of Electronics Engineering, National Central University, Jhongli, Taiwan, Jun. 2005.

[15] R.W. Chang, ”Synthesis of Band-Limited Orthogonal Signals for Multichannel Data Transmission”, *Bell Syst. Tech. J.*, vol.45, pp. 1775-1796, Dec. 1966.

[16] S. Chen, W. He, H. Chen and Y. Lee, “Mode Detection, Synchronization, and Channel Estimation for DVB-T OFDM Receiver,” *IEEE GLOBECOM*, vol. 5, Dec. 2003, pp. 2416-2420

[17] C-W. Kuang, “Timing Synchronization for DVB-T System,” Master Thesis, Institute of Electronics Engineering, National Chao Tung University, Hsinchu, Taiwan, Sep. 2004.

[18] K-H. Lin, “Design of a Baseband Receiver for DVB-T Standard,” Master Thesis, Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan, Jul. 2005.

[19] A. Palin, J. Rinne, “Symbol synchronization in OFDM system for time selective channel conditions,” *IEEE ICECS*, vol. 3, Sep. 1999, pp. 1581-1584

[20] A. Hazmi, J. Rinne, T. Kuusisto, M. Renfors, “Performance evaluation of symbol synchronization in OFDM systems over impulsive noisy channels,” *IEEE VETECS*, vol. 3, May 2004, pp. 1782-1786.

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

[22] L. Schwoerer “Fast Pilot Synchronization Schemes for DVB-H,” *IASTED*, July 2004, pp. 420-424

[23] L. Schwoerer, J Vesma, “Fast Scattered Pilot Synchronization for DVB-T and DVB-H,” Proc. 8th International OFDM-Workshop, Hamburg, Germany, Sept. 2003

[24] F. Eory, “Comparison of adaptive equalization methods for the ATSC and DVB-T digital television broadcast systems,” *IEEE ICCDCS*, Cancun, Mar. 2000, pp. T107/1-T107/7

[25] P. Combelles, C. D. Toso, D. Hepper, D. Le Goff, J.J. Ma, P. Robertson, F. Scalise, D. Soyer, M. Zamboni, “A receiver architecture conforming to the OFDM based digital video broadcasting standard for terrestrial transmission (DVB-T),” *IEEE ICC*, vol. 2, Atlanta, GA, Jun. 1998, pp. 780-785.

[26] T-A. Lin, C-Y. Lee, “Predictive equalizer design for DVB-T system,” *IEEE ISCAS*, vol.2, May 2005, pp. 940-943.

[27] L. Horvath, I. B. Dhaou, H. Tenhunen and J. Isoaho “A Novel, High-Speed, Reconfigurable Demapper-Symbol Deinterleaver Architecture for DVB-T,” *IEEE ISCAS*, Vol. 4, June. 1999, pp. 382-385.

[28] S. A. Rechtel, A. Blaickner, “Efficient FFT and Equalizer Implement for OFDM Receivers,” *IEEE. Trans. Consumer Electronics*, pp. 1104-1107, Sept. 1999.

[29] K-W. Shin, B-S. Song, “A complex multiplier architecture based on redundant binary arithmetic,” *IEEE ISCAS*, Vol. 3, June 1997, pp.1944-1947.