Design and FPGA Implementation of a SISO and a MIMO Wireless System for Software Defined Radio

Peng Dong

A Thesis

In

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science at
Concordia University
Montréal, Québec, Canada

March 2009

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-63333-5
Our file Notre référence
ISBN: 978-0-494-63333-5

ABSTRACT

Design and FPGA Implementation of a SISO and a MIMO Wireless System for Software Defined Radio

Peng Dong

MIMO (Multiple-input Multiple-output) technology combined with space time coding techniques provides significant increase in performance and capacity over an equivalent SISO (Single-input Single-output) system while maintaining the same bandwidth and transmission power. MIMO has emerged as the major breakthrough in recent communication technologies. To migrate from SISO to MIMO system, multiple RF (Radio Frequency) front ends and additional signal processing are required. Software defined radio (SDR) allows MIMO and other evolving techniques to be added to current systems through software update instead of hardware replacement. SDR provides a flexible and economic solution to the system upgrade and migration.

In this thesis, an SDR based SISO system using QPSK modulation scheme is implemented on FPGA. The system produces signal with an intermediate frequency of 25 MHz and throughput of 12.5 Mbps. One carrier recovery and two symbol timing recovery algorithms (Gardner and Maximum Likelihood) are investigated and implemented. A 2x1 MIMO system using Alamouti scheme and CORDIC based carrier recovery is designed as well. The SDR based SISO system can be easily incorporated to the MIMO design. Throughout this thesis, detailed design information is presented along with both computer simulation results and real hardware performance. The comparisons of different algorithms and component structures are also provided. Based on these comparisons, the suitable algorithm or structure according to specific implementation considerations and system requirement can be selected.

The design and implementation are processed based on a system-level design flow. System modeling and simulation are performed using Xilinx's *System Generator for DSP* and Simulink.

After it is mapped to HDL (Hardware Description Language) netlist, the design is synthesized and implemented by Xilinx's *ISE* tool. The generated bit-stream is then downloaded to target FPGA to program the device. The hardware performance is measured by BER (Bit Error Rate) tester, oscilloscope and spectrum analyzer.

This thesis is an initial project for future work of Wireless Design Laboratory at Concordia University. The system realized in this project can be viewed as a base of future MIMO implementation with different number of antennas and advanced signal processing techniques.

Acknowledgements

I would like to take this opportunity to express my sincere appreciation to my supervisor, Dr. Yousef R. Shayan who motivated me in working on this implementation-oriented thesis and also lead me to the path of practical design of wireless system. His direction and support are critical in developing this thesis. He has been a constant source of inspiration, and has provided consistent succors and valuable suggestions throughout this project. Without these help he provided, this work would not have been possible.

Besides, I am particularly grateful to the manager of Wireless Design Lab, Mr. Nick Ierfino. As an expert in radio and embedded system design, he shared the precious experience with me. He also offered helpful assistance during the hardware test of this project. It was a pleasant time to work with him.

I owe the deepest gratitude to my beloved parents. Their continuous encouragement and support make it possible for me to pursue a successful study and happy life in Montreal.

Last but never least, I would like to thank my colleagues in the lab and Miss Xuan Liu for their individual support.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Expansion
ADC	Analog to Digital Convertor
ASICs	Application Specific Integrated Circuits
ASSPs	Application Specific Standard Parts
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CDMA	Code Division Multiple Access
CORDIC	Coordinate Rotation Digital Computer
CR	Carrier Recovery
DAC	Digital to Analog Convertor
DDC	Digital Down-Convertor
DSP	Digital Signal Processing
DSPs	Digital Signal Processors
DUC	Digital Up-Convertor
DVB-S	Digital Video Broadcasting Satellite
FDM	Frequency Division Multiplexing
FPGA	Field Programmable Gate Array
IF	Intermediate frequency
IIR	Infinite Impulse Response
ISI	Inter-Symbol Interference
LSTC	Layered Space-Time Codes
LTE	Long Term Evolution
LUT	Look-Up Table

MAC	Multiply-and-Accumulate
Mbaud	Mega-Symbol per Second
MIMO	Multiple-Input Multiple-Output
ML	Maximum Likelihood
Msps	Mega-Sample per Second
NCO	Numerical-Controlled-Oscillator
OFDM	Orthogonal Frequency Division Multiplexing
PDF	Probability Density Function
PSC	Parallel to Serial Convertor
PSF	Pulse Shaping Filter
PSK	Phase-Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
QoS	Quality of Service
ROM	Read Only Memory
SDR	Software Defined Radio
SISO	Single-Input Single-Output
SNR	Signal to Noise Ratio
SPC	Serial to Parallel Convertor
SQRC	Square Root Raised Cosine
STBC	Space-Time Block Codes
STC	Space Time Coding
STR	Symbol Timing Recovery
STTC	Space-Time Trellis Codes
VCO	Voltage-Controlled-Oscillator
VHDL	Very-High-Speed Integrated Circuits or VHSIC HDL

Chapter 1

Introduction

1.1 Background

With the ever increasing demand of wireless and mobile communication, a system that can provide high rate data, voice, image, video and other multimedia capabilities is highly required. As a major breakthrough in recent communications technologies, Multiple-input Multiple-output (MIMO) technology provides significant increase in system capacity and performance by means of using multiple antennas at transmitter and/or receiver and space time coding technique. Much higher data rate, better Quality of Service (QoS) and enhanced transmission reliability can be achieved in a MIMO system compared with the traditional Single-input Single-output (SISO) system. MIMO has attracted great interest from academia to industry for the last decade, and has become the foundation for next-generation wireless communication systems.

However, these benefits are obtained at the expense of multiple RF (Radio Frequency) front-ends and additional signal processing required for space time coding and decoding. To migrate from SISO to MIMO system, traditional hardware intervention results in high costs and low flexibility in supporting multiple waveform standards [1]. A cost effective MIMO system can be realized by means of software defined radio (SDR) technology. Quoted from SDR forum [2], the term SDR is defined as "*Radio* in which some or all of the physical layer functions are *software defined*". In other words, most of the signal processing on physical layer is implemented through configurable software or hardware operating on programmable processing technologies. The new signal processing techniques, air interface protocols and functionalities can be upgraded through software instead of a complete hardware replacement. As a result, SDR allows the MIMO and future technology to be added in current systems via simple software update. SDR provides an inexpensive solution of building multi-mode, multi-band and multifunctional wireless communication devices [2]. Due to its flexibility and cost-efficiency, this technology brings considerable benefits to product manufacturers, service providers and users. In the recent years, SDR has been widely used in the areas of cellular system, satellite communication and defense application.

As a requirement of SDR, digital signal processing (DSP) has become a preferred choice for implementation of communication systems, instead of traditional analog signal processing. Thus the sophisticated signal processing tasks, such as error control coding, synchronization, equalization, power control, channel estimation and so on, are all performed on the SDR based platform. Figure 1-1 shows the DSP functions in an SDR based receiver.

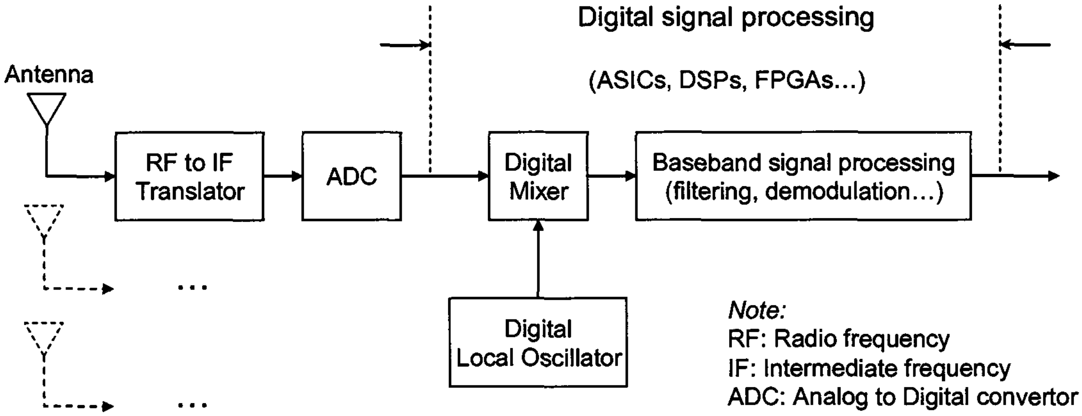

Digital signal processing
(ASICs, DSPs, FPGAs...)
Antenna
RF to IF Translator
ADC
Digital Mixer
Baseband signal processing (filtering, demodulation...)
Digital Local Oscillator
...
...
Note:
RF: Radio frequency
IF: Intermediate frequency
ADC: Analog to Digital convertor

Figure 1-1 Digital signal processing in SDR based receiver

The hardware implementation of an SDR based communication system can be realized by means of semiconductor technology. In the early days, application specific standard parts

(ASSPs), application specific integrated circuits (ASICs), digital signal processors (DSPs) and general-purpose microcontroller were the main solutions of building an SDR platform [3]. During the last decade, field programmable gate array (FPGA) that can offer both high performance and flexibility has become the mainstream among these technology solutions, to meet the design requirement for ever increasing complexity of communication systems.

FPGA is a general-purpose integrated circuit that is programmed by the designer rather than the device manufacturer, which means it can be reprogrammed without changing any component or interconnection at system level even after it has been deployed into a system. Compared with traditional ASICs and DSPs, FPGA features the following advantages [3][4]:

- High-performance and high-speed signal processing capability through parallelism
- Low risk due to the flexible architecture
- Low power consumption and cost
- Completely reconfigurable, allowing design migration for changing system protocols
- Fast time-to-market for industry purpose

In addition, with the help of embedded DSP processors and dedicated multipliers, FPGAs are powerful and suitable for realizing DSP functions.

1.2 Motivation and Contribution of the Thesis

The objective of this thesis is to design and implement an SDR based SISO and MIMO system on FPGA. The topic on design and implementation of communication system has been explained broadly in literature. For example, a QAM (Quadrature Amplitude Modulation) based receiver for SDR is implemented by C. Dick et al [5]. A DVB (Digital Video Broadcasting) receiver is implemented on FPGA by F. Cardells et al [6][7]. The system design using Xilinx's design tool for WCDMA and CDMA2000 base station can be found in [8][9]. The design process using

Altera's design tool is described in [10][11]. On the other hand, some MIMO based implementation and testbeds were claimed [12]-[18].

Among the published work, most of them focus on the algorithm explanation and computer simulation. Few of the work present the detailed design and real hardware performance. The key feature of our work is implementation of an SDR based SISO wireless system on FPGA. A 2x1 MIMO system using Alamouti scheme is designed as well. The SISO system can be easily incorporated into the MIMO design to achieve higher throughput. The specifications of the proposed design are provided along with the real hardware performance. The major contributions of this thesis include:

- An SDR based SISO system using QPSK (Quadrature Phase Shift Keying) modulation scheme is successfully implemented on FPGA. The system has an IF of 25 MHz and a throughput of 12.5 Mbps, which can be up to 15.818 Mbps.
- The specifications of the design and implementation are provided for the proposed SISO system. This design can be used as a base of a MIMO system. Various modulation schemes can be applied without changing most of the components including baseband signal processing and digital up/down conversion. Carrier frequency can be configured as well to fulfill the specific requirement.
- The parallel polyphase structure for interpolation and decimation filter, which is suitable for high data rate, is proposed.
- One carrier recovery algorithm and two symbol timing recovery algorithms are investigated, designed, and implemented. Their performances are also evaluated and compared using both computer simulations and hardware test.
- A study of a 2x1 MIMO system based on Alamouti scheme is made. Detailed design for Alamouti encoder, Maximum likelihood decoder, and carrier recovery using CORDIC algorithm is provided. The SISO system design is flexible and easy to migrate to this MIMO system and future designs.

- The real hardware performance is examined for the proposed SISO system design, and 1.2-dB implementation loss is presented. The BER (Bit Error Rate) performance and hardware utilization are also compared for two different timing recovery algorithms. This provides information to readers of choosing different algorithms based on different design criterion (performance or resources).

This thesis is an initial project for future work of Wireless Design Lab at Concordia University. The lab was established in 2008 to improve Research and Development in areas of digital system design, embedded microcontrollers and wireless technologies. The lab is equipped with full set of hardware and software for design and implementation of wireless communication systems. Numbers of industry-level testbeds are available, such as Virtex-5, Virtex-4, and Virtex-II Pro FPGAs from Xilinx, SignalMaster Quad development platform and dual channel RF transceiver from Lyrtech, and XtremeDSP board from Nallatech. Besides, full set of test equipment are available as well, such as fading channel emulator, BER tester, vector signal generator, vector analyzer, network analyzer, oscilloscope, spectrum analyzer and so forth. Combined with Xilinx's design suit, Wireless Design Lab provides sufficient resources and ideal solution for system design and verification.

1.3 Methodology of Design and Implementation

Traditionally, FPGA based DSP design is realized using standard register transfer level (RTL) flow. At this level, the design is modeled as a combinational circuit separated by registers and a set of transfer functions which describe the data flow between the registers. In addition, two distinct sets of design tools are normally required, one for algorithm development and analysis such as C/C++ and Matlab, and another for hardware synthesis and implementation such as Hardware Description Language (HDL). After manually converting the high level design with floating point representation into hardware model with fixed point representation, the design is simulated at the RTL level. Logic synthesis and physical synthesis are performed afterwards to analyze and verify the design, such as timing and area at gate-level.

Recently, the system level design tool breaks the gap between DSP algorithm design and hardware implementation. In these tools, automatic translation from high level design to RTL model is provided along with auto quantization and timing/area optimization. One example of these tools is Xilinx's *System Generator for DSP* [19]. System Generator is a system-level modeling tool embedded in Simulink for implementing systems in FPGAs. Simulink is an interactive graphical environment for model-based design and multi-domain simulation. System Generator provides libraries of functions and hardware related abstractions that can be used to model a DSP system. Such models are bit and cycle accurate to FPGA hardware. System Generator ensures this by providing automatic code generation from Simulink to a combination of synthesizable HDL and intellectual property (IP) cores. In addition, this software is able to play hardware co-simulation and hardware/software co-design [19] to accelerate the design and simulation process. Not only facilitating the design, it also helps us to focus on the critical part, such as design of DSP algorithm itself. As a result, System Generator is an ideal tool for system design and implementation.

Figure 1-2 shows the design and implementation flow of our system, where three major steps are involved. First, the design is modeled using functional blocks provided by System Generator and Simulink. Computer simulations are then performed for algorithm verification. The design specification can be determined and optimized in this step, such as signal precision, filter length, quantization level, and so forth.

After VHDL (Very-High-Speed Integrated Circuits, or VHSIC HDL) code is generated by System Generator, the design is imported to ISE tool to verify and implement the design at RTL level. Logic synthesis is performed to translate RTL module to an optimized gate-level netlist based on timing and area constraints. Physical synthesis including placement and routing is performed afterwards. Routing delays are back annotated to the gate-level netlist for timing analysis. Finally, a bitstream is generated to program the FPGA. Small modification on the design should be made in this step based on the synthesis and timing analysis results.

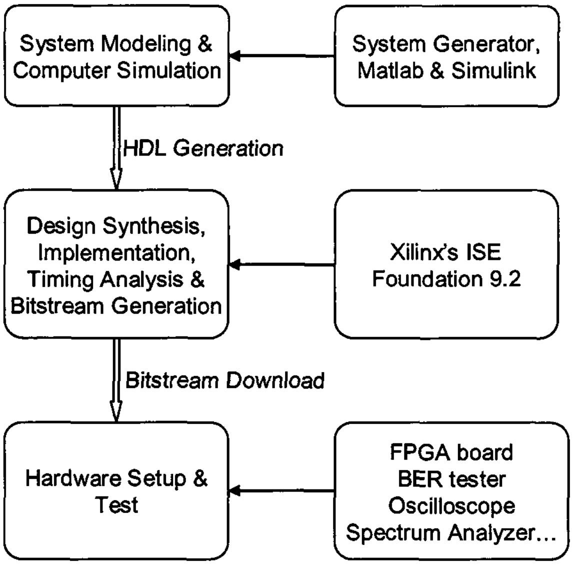

System Modeling & Computer Simulation
System Generator, Matlab & Simulink
HDL Generation
Design Synthesis, Implementation, Timing Analysis & Bitstream Generation
Xilinx's ISE Foundation 9.2
Bitstream Download
Hardware Setup & Test
FPGA board
BER tester
Oscilloscope
Spectrum Analyzer...

Figure 1-2 Design and implementation flow, and related software and hardware

The last step is hardware setup and test. The generated bitstream is now downloaded to the target FPGA. After the board is configured, we measure the system performance using BER (Bit Error Rate) tester, oscilloscope and spectrum analyzer.

1.4 Thesis Organization

This thesis is organized as follows.

- In chapter 2, overview of a typical digital communication system is first given. A SISO system based on our design is then introduced, and the components in transmitter and receiver are explained in detail. During the description of each component, theoretical background is first provided. Design and implementation issues are immediately followed, as well as showing design models with System Generator blocks. This method is also applied to the further chapters. Interpolation and decimation filter design is emphasized in chapter 2, and two filter architectures based on parallel polyphase structure are proposed.

- In chapter 3, the synchronization issues including carrier recovery and symbol timing recovery for a SISO system are discussed. Both floating-point and fixed point simulations are done at this stage to verify and analyze the synchronization algorithm. In symbol timing recovery design, two algorithms are explained and implemented. Simulation results are also presented for comparison between them.
- In chapter 4, we focus on the design and implementation of a MIMO system. An overview of multipath fading channel is first provided. The detail design of Alamouti encoder, Maximum likelihood decoder and carrier recovery using CORDIC algorithm are then explained. Simulation of carrier recovery design is also performed and analyzed.
- In chapter 5, the SISO system design is fully implemented on the FPGA, and the hardware test results including signal observation, BER performance and resource utilization are presented and analyzed. In addition, we briefly describe the FPGA board and test equipment we use.
- Chapter 6 concludes this thesis, and provides some recommendations for future work.

Chapter 2

Design and Implementation of a SISO System

2.1 A Typical Digital Communication System

A model for a typical digital communication system can be categorized into three fundamental parts, transmitter, channel and receiver. The information source for such a system is in the form of binary data, i.e., “0” and “1”. In the beginning of transmission, the binary data is source-encoded or compressed to eliminate the redundant information as much as possible. Then channel coding, which is known as error control coding, is applied to introduce some controlled redundancy into the data stream for the purpose of protecting against channel induced errors. At the same time, every certain number of data bits are grouped together preparing for the digital modulation. In the end of transmitter, those bit groups are modulated to digital symbols, and mapped onto analog waveform for transmission over physical channel.

In practice, a channel could be a wire, a fiber optic cable, free space, or a variety of other models. Each of these has different characteristics affecting the transmitting signal differently. Normally, two major sources of channel interference have to be considered in a wireless environment, which are multipath fading and noise. While the former is a result of scattering, refraction, and reflection from terrestrial objects [21], the latter is mainly due to the thermal noise existing in front end receiver electronics, such as bandpass filter. Both of them cause signal amplitude and phase distortion. Furthermore, the noise can be model as an additive white Gaussian noise (AWGN) channel which has a uniform power spectral density and a Gaussian distributed amplitude [20]. Various other sources of channel corruption exist, and have to be taken into account when necessary. These factors include the movement between transmitter and receiver, number of antennas, number of users, and so on [21].

The ultimate task of receiver is to extract the original transmitted data from the received signal that is corrupted by the channel. In order to do so, all the coding, modulation and signal processing performed in the transmitter should be reversed at the receiver. To accomplish this task, knowledge of channel characteristic, original data clock and other information of transmitted signal are always required. The performance of the system is qualified by how much of the transmitted data can be successfully reconstructed by the receiver. This is measured with respect to the error probability by comparing the original data with the reconstructed data. The probability depends on many factors including modulation schemes, channel types, signal to noise ratio (SNR) and so forth. In order to achieve an acceptable error level, channel and transmitted signal information should be understood as much as possible in the receiver.

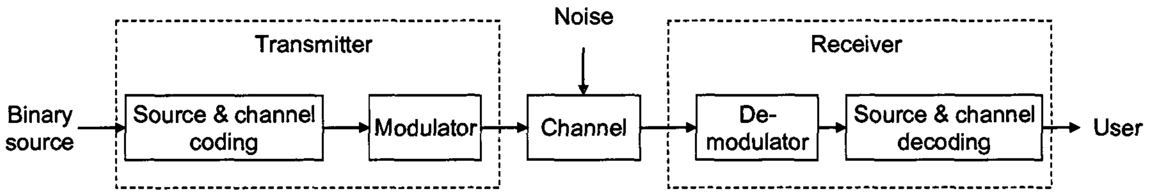


Figure 2-1 Basic components of a digital communication system

Throughout the rest of this chapter, we will decompose the basic elements in a typical communication system, and introduce each component individually. The detailed design and implementation models of each component in both transmitter and receiver are given for the SISO system design.

2.2 Overview of a SISO System Design

Figure 2-2 shows the general components of our SISO system design. In the transmitter side, the binary source with a bit rate of 12.5 Mbps is first modulated using QPSK (Quadrature Phase Shift Keying) scheme. The generated QPSK symbols with rate of 6.25 Mbaud (Mega-symbol per second) are fed into a pulse shaping filter, and oversampled by a factor of 2, resulting in a signal with sample rate of 12.5 Msps (Mega-sample per second). The following interpolation filter and digital up-convertor (DUC) upsample the signal to 100 Msps, and translate it from baseband to an intermediate frequency (IF) of 25 MHz. A digital to analog convertor (DAC) then converts the digital IF signal to analog domain for transmitting through an AWGN channel.

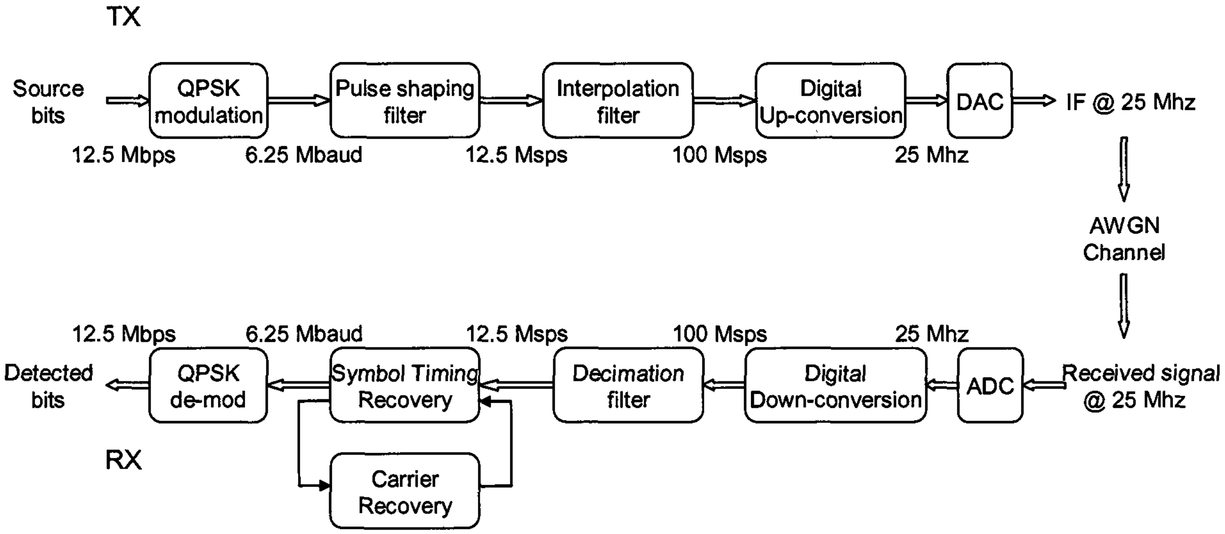


Figure 2-2 Block diagram of proposed SISO system design @ IF of 25 MHz

At the receiver side, an analog to digital convertor (ADC) first converts the received signal back to digital domain. The signal is then down-converted to baseband and down-sampled by decimation filter to a rate of 12.5 Msps, which is 2 samples per symbol. These samples are fed into a feedback loop to perform symbol timing recovery (STR) due to the fact that receiver sampling clock is not synchronous with symbol rate. A matched filter used for optimizing the signal to noise ratio (SNR) is embedded in the STR loop, and not shown in the figure. The output of STR loop is fed into carrier recovery (CR) loop to compensate the residue carrier frequency and phase offset. In the end, QPSK de-modulator detects the recovered symbols, and maps them back to bits based on the QPSK mapping scheme. The bit error rate (BER) performance then can be measured by comparing the reconstructed bits with original source bits.

Synchronization including symbol timing and carrier recovery is one of the most challenging tasks in the system design. In this chapter, we focus on the design and implementation with perfect timing and carrier matching between transmitter and receiver. Synchronization and relative design issue will be introduced and discussed in detail in the next chapter.

2.3 Baseband QPSK Modulator

2.3.1 Background

Phase-shift keying (PSK) is a digital modulation scheme in which the data is modulated by the phase difference compared with a reference signal. QPSK (Quadrature-PSK) has four phase states separated by 90 degrees, which are 45, 135, -135, and -45 degrees on the constellation diagram (Figure 2-3). QPSK is a bandwidth-efficient modulation scheme. With the same bandwidth, it can support twice the bit rate than BPSK (Binary-PSK). QPSK is widely used in the existing communications system, including CDMA (Code Division Multiple Access), wireless local loop, and DVB-S (Digital Video Broadcasting Satellite) [21]. When coherent detection is used, the theoretical BER expression for QPSK over AWGN channel is given by [20]

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{2.1}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-x^2/2\right) dx$, for $x > 0$. Figure 2-4 illustrates the theoretical BER performance of QPSK modulation over AWGN channel.

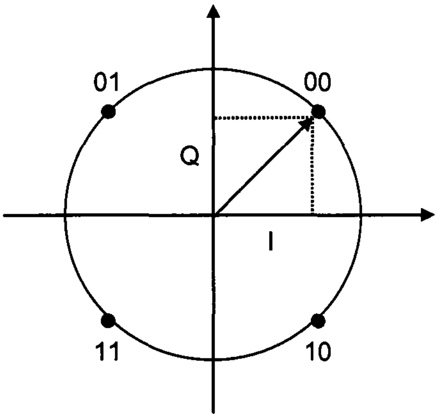


Figure 2-3 QPSK constellation

Bit set	I value	Q value	Phase (degree)
00	1	1	45
01	-1	1	135
11	-1	-1	-135
10	1	-1	-45

Table 2-1 QPSK symbol mapping with Gray coding

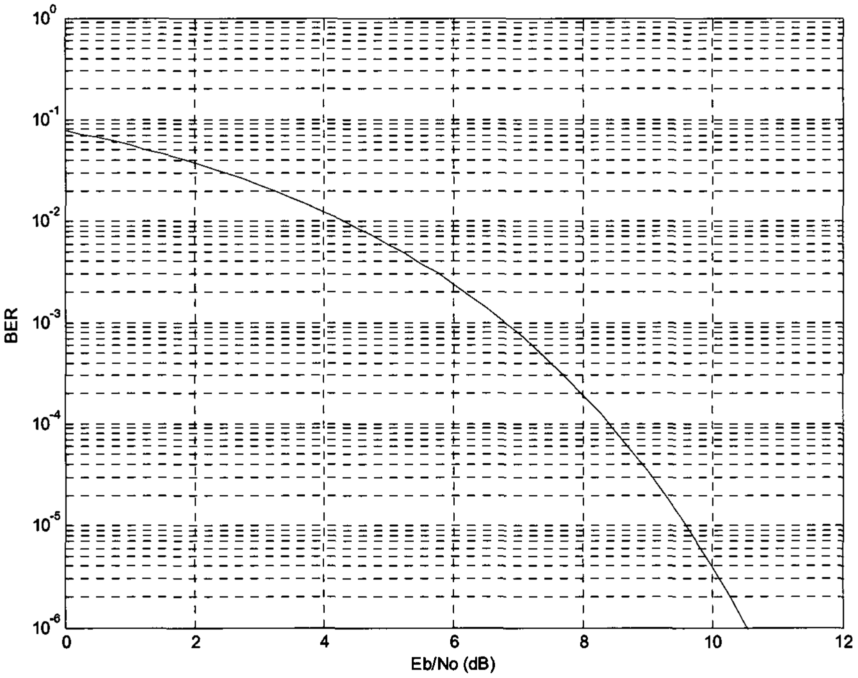


Figure 2-4 Theoretical BER performance of QPSK over AWGN channel

2.3.2 Design and Implementation

Figure 2-5 shows the structure of baseband QPSK modulator, which consists of a serial to parallel converter and 2 look-up tables (LUT) implemented by 2 read only memories (ROMs). The converter groups every 2 bits, and converts them to an index pointing to the corresponding QPSK symbols. Two ROMs are needed to store the in-phase (I) and quadrature (Q) values. Gray coding [20] is also applied to the mapping process to obtain better performance. The bit to symbol mapping is shown in Table 2-1.

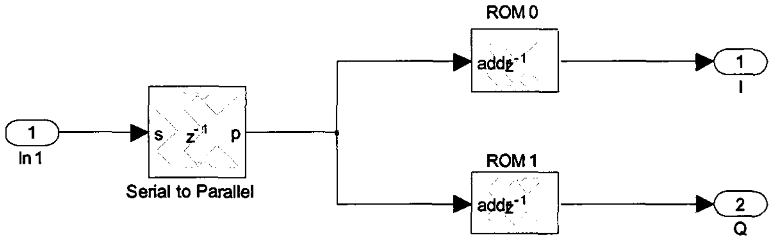


Figure 2-5 Baseband QPSK modulator

2.4 Pulse Shaping Filter and Interpolation Filter

2.4.1 Pulse Shaping Filter

In digital transmission, the binary source is a series of rectangular pulses. However, directly transmitting these pulses causes an infinite frequency span, which is not acceptable in a band-limited system. Pulse shaping filter (PSF) is responsible for shaping the pulses to satisfy the bandwidth requirement. From time domain point of view, a PSF should have zero crossing at each sampling time to avoid inter-symbol interference (ISI). This implies that the fundamental shapes of the pulses are such that they do not interfere with each other. From frequency domain point of view, the magnitude of signal outside the filter's passband should decay rapidly, so that the bandwidth of filtered signal is strictly limited [23]. Raised Cosine filter satisfies these two conditions, and is widely used in system design. In practice, two Square Root Raised Cosine

(SQRC) filters are placed both in transmitter and receiver in order to have an overall raised cosine response of the system. The SQRC filter in the transmitter is known as pulse shaping filter, whereas the one in the receiver is called matched filter (MF). Such a combination guarantees maximum SNR and minimum ISI for signal detection after matched filtering.

The frequency response of SQRC filter is given by [20]

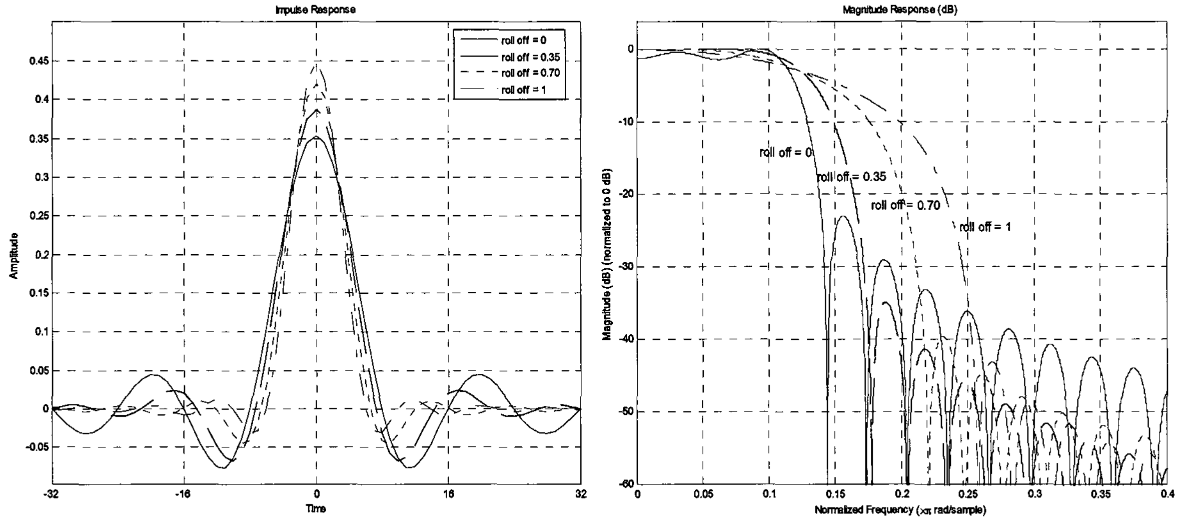
$$H(f)=\begin{cases} 1 & |f|\le\dfrac{1-\beta}{2T_s} \\ \sqrt{\cos\left(\dfrac{\pi T_s}{2\beta}\left(|f|-\dfrac{1-\beta}{2T_s}\right)\right)} & \dfrac{1-\beta}{2T_s}\le|f|\le\dfrac{1+\beta}{2T_s} \\ 0 & |f|>\dfrac{1+\beta}{2T_s} \end{cases} \tag{2.2}$$

and the impulse response of SQRC filter is given by

$$h(t)=\frac{4\beta}{\pi\sqrt{T_s}}\frac{\cos\left((1+\beta)\pi\dfrac{t}{T_s}\right)+\sin\left((1-\beta)\pi\dfrac{t}{T_s}\right)\Big/4\beta\dfrac{t}{T_s}}{1-\left(4\beta\dfrac{t}{T_s}\right)^2} \tag{2.3}$$

where T_s is symbol period, and β is called roll-off factor ($0\le\beta\le1$), which determines the excess bandwidth of shaped signal and decay rate of stopband signal. Figure 2-6 shows the impulse and frequency response of a 65-tap SQRC filter with different roll-off factors. When $\beta=0$, the frequency response (Figure 2-6(b)) has a nearly rectangle form in passband, which offers the most efficient use of bandwidth. However, this gives the slowest decay rate or ripple attenuation in time domain (Figure 2-6(a)). As β increases, the occupied bandwidth increases as well, but with benefit of more rapid decay rate and smaller ripple in pass band. Small roll-off factor is always desired for a band limited system. However, frequency response will become sharper as β decreases, and more coefficients are needed to meet the requirement of passband ripple and stopband attenuation, which results in increasing complexity of filter implementation.

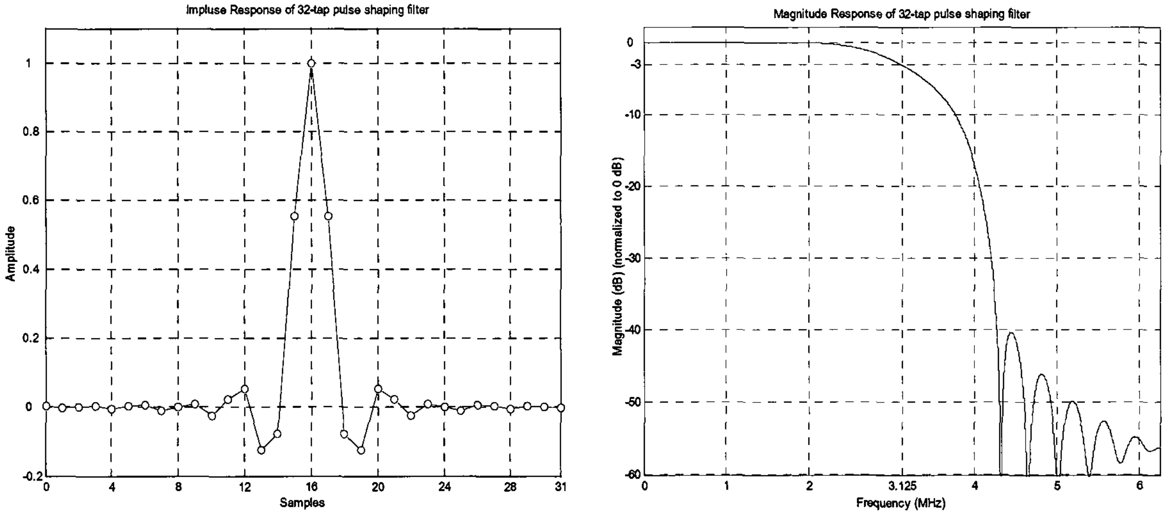
In practice, a roll-off factor in range from 0.15 to 0.5 is chosen as a compromise between bandwidth efficiency and implementation complexity.



(a) (b)

Figure 2-6 Impulse (a) and frequency (b) response of a SQRC filter with different roll-off factors

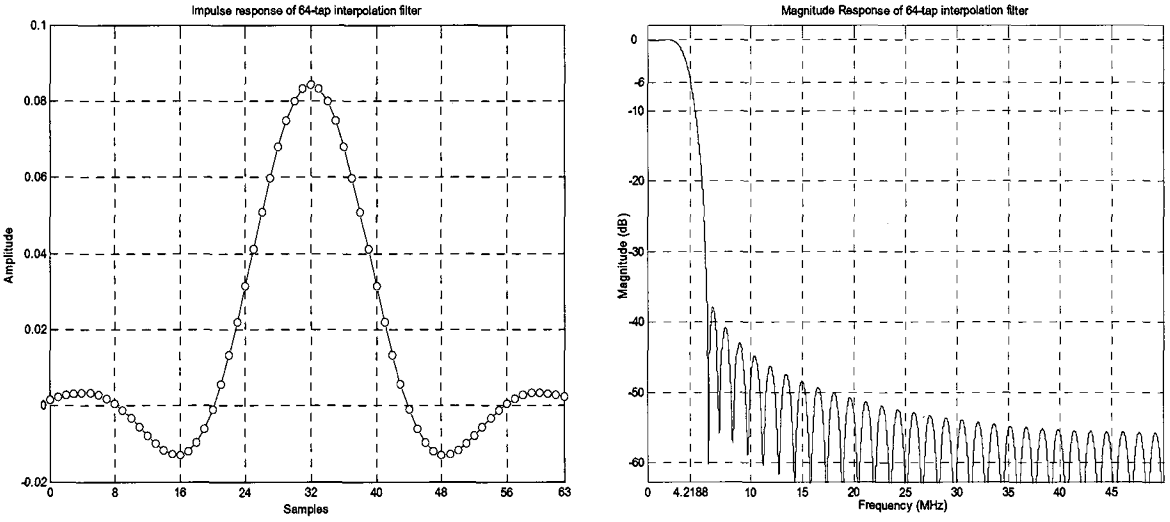
To fulfill the Nyquist sampling criterion, the filter has to operate at a rate of no less than twice the symbol rate. Oversampling with more than two samples per symbol is desired in practical design [24]. Since the oversampling can be accomplished by interpolation filter to be discussed later, we simply increase the symbol rate by a factor of 2 via pulse shaping filter. The roll-off factor is chosen as 0.35, and the filter impulse response is designed to span 16 symbols to obtain a smooth passband and significant stopband attenuation. Then a SQRC filter working at $6.25 \times 2 = 12.5$ MHz with $16 \times 2 = 32$ taps, or coefficients is to be build. After generating coefficients using Hamming window method from Matlab filter design tool, we examine the normalized impulse response and spectrum of SQRC filter. As shown in Figure 2-7 (b), the spectrum has a 3-dB cut-off frequency of half of symbol rate, which is $6.25 / 2 = 3.125$ MHz, and signal out of this band decays rapidly. The shaped baseband signal has a bandwidth of $3.125 \times (1 + 0.35) = 4.2188$ MHz. Please note that the spectrum shown in the figure is within single sided frequency range of $[0, F_s/2]$, where F_s is filter's sampling frequency, and the double sided signal bandwidth is 8.4275 MHz. The architecture structure of pulse shaping filter will be discussed after interpolation filter is introduced.



(a)

(b)

Figure 2-7 Impulse (a) and magnitude (b) response of a 32-tap pulse shaping filter



(a)

(b)

Figure 2-8 Impulse (a) and magnitude (b) response of a 64-tap interpolation filter

2.4.2 Interpolation Filter

2.4.2.1 Background

One of the purposes of SDR is to move the digital signal processing and software controlled sections as close to the antenna as possible [2]. As a result, spectral translation from baseband to IF is intended to perform in digital domain rather in analog domain. In order to accommodate a relatively high IF, or carrier frequency f_c, system is always working at a much higher sampling rate f_s compared with the symbol rate. This is due to the fact that carrier samples should be taken at a rate identical to signal sampling rate f_s, so that the carrier and signal can be fed into a digital mixer. Therefore, upsampling original data by a certain number is necessary for signal transmission. Figure 2-9 shows the signal spectrum after upsampling by a factor of 2. We notice that the upsampled signal has periodic spectrum images located at multiple of sampling rate due to sampling theory, and only the spectrum centered at DC interests us. Therefore, a low pass filter should be placed after up-sampler to remove the unwanted spectrum images, and interpolation filter normally refers to an up-sampler followed by an anti-imaging low pass filter.

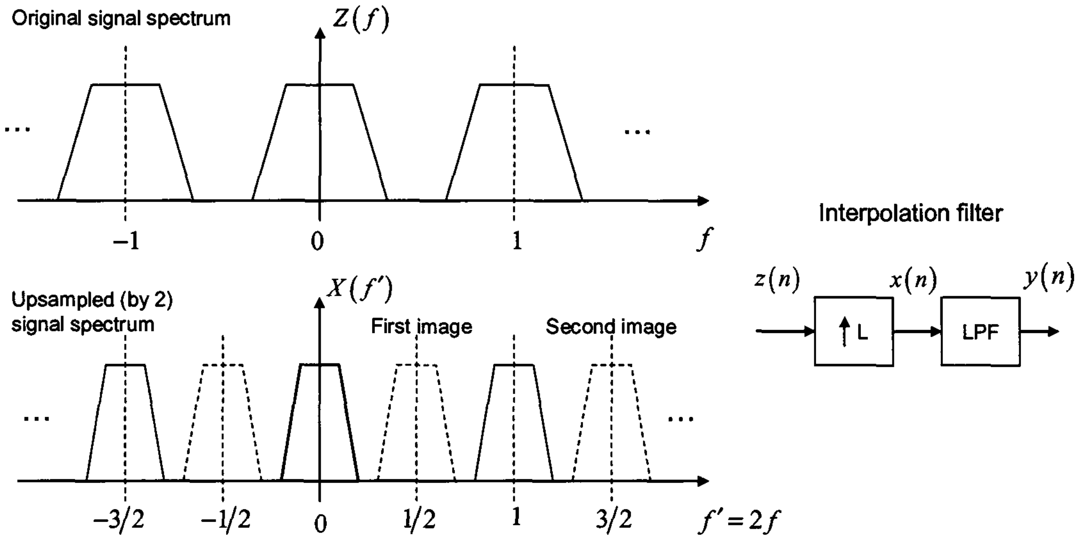


Figure 2-9 Upsampled signal spectrum and interpolation filter

2.4.2.2 Design and Implementation

When it comes to the structure of interpolation filter, polyphase partition [24] is always a good choice. Since the upsampled signal is actually a signal with zeroes inserted between original samples, these zeroes obviously do not contribute to the filtering process, i.e., multiplication and addition. Let us now examine how polyphase can be applied to the interpolation design. First, recall the digital filter transfer function

$$y(n) = x(n) \otimes h(n) = \sum_{k=0}^{N-1} x(n-k)h(k) \tag{2.4}$$

where $h(n)$ is the filter's impulse response with $N = S \cdot L$ taps. Here, we assume that the filter length N can be divided to L groups length S, where L is the upsampling factor. If $x(n)$ is an upsampled version of $z(n)$, the upsampling process can be modeled as

$$x(n) = \begin{cases} z(n/L), & \text{if } n/L \text{ is an integer} \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

Now, substitute $k = r \cdot L + \lambda$ in Eq. (2.4), where r and λ are both integers, and $0 \le \lambda < L-1$. Then we have the following expression,

$$y(n) = \sum_{\lambda=0}^{L-1} \sum_{r=0}^{S-1} x(n-(r \cdot L + \lambda)) \cdot h(r \cdot L + \lambda) \tag{2.6}$$

Substituting $k = r \cdot L + \lambda$ in Eq. (2.5) results in

$$x(n-k) = \begin{cases} x(n\text{-}(r \times L + \lambda)) = z(m\text{-}r), & \text{when } n = m \times L + \lambda \\ 0, & \text{otherwise} \end{cases} \tag{2.7}$$

Using both Eq. (2.6) and (2.7), we can have the expression as

$$\begin{aligned} y(m) &= \sum_{\lambda=0}^{L-1} \sum_{r=0}^{S-1} z(m-r) \cdot h(r \cdot L + \lambda) \\ &= \sum_{\lambda=0}^{L-1} \sum_{r=0}^{S-1} z(m-r) \cdot h_\lambda(m) \end{aligned} \tag{2.8}$$

where $h_\lambda(m)$ is actually a sub-filter deriving from original filter with a phase index λ, and the interpolation filter is successfully decomposed by polypahse structure.

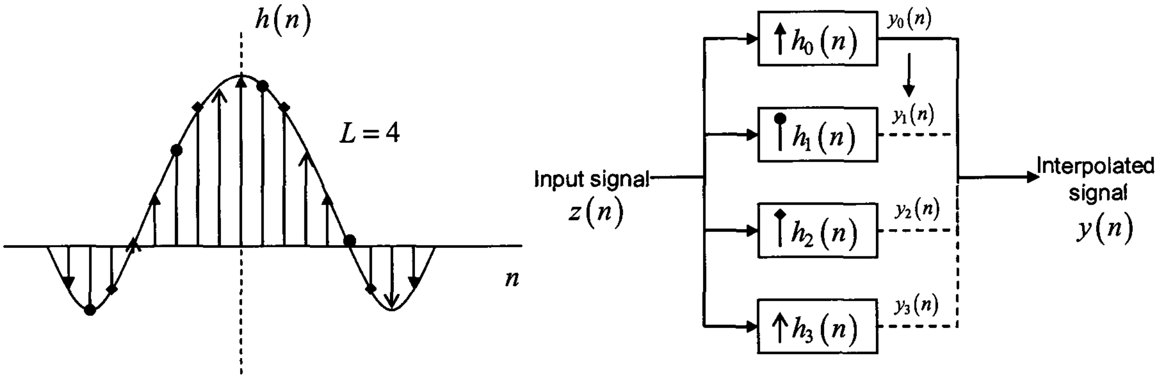


Figure 2-10 Polyphase partition for interpolation filter when $L=4$

As shown in Figure 2-10, the polyphase structure splits the original filter into L sub-filters, each with length of $S=N/L$, and impulse response of $h_\lambda(n)=h(n\cdot L+\lambda)$. Specifically, the sub-filter h_0 contains the coefficients $h(L)$, $h(2L)$,$h(3L)$,... the sub-filter h_1 contains the coefficients $h(L+1)$, $h(2L+1)$, $h(3L+1)$,... and so forth. These filters filter the incoming signal in turn at upsampling rate. The results are summed up to form the interpolation output. By doing this, no unnecessary calculation is performed, and the overall computation savings can be$N-N/L$ per output compared with traditional filter structure of direct-form and transpose-form. Any design method for low pass filter can be applied to interpolation design [24]. Here, we are aiming to build a 64-tap interpolation filter with sampling frequency of $12.5\times 8=100$ MHz and 6-dB cut off frequency of 4.2188 MHz, which equals baseband signal bandwidth after pulse shaping. The filter coefficients are generated using Kaiser Window method from Matlab filter design tool. The impulse and frequency response of interpolation filter are shown in Figure 2-8. As seen in Figure 2-8(b), the stopband attenuation is 40 dB, which also meets our requirement.

The polyphase interpolation filter based on our design is shown in Figure 2-11, where it is composed of delay line, coefficients ROMs, multiplier-and-adder tree and a free running counter. The depth of each ROM implies the number of sub-filters, or upsampling factor, while the number of ROMs represents the length of each sub-filter. For example, if the original filter has 64 taps, and signal is to be upsampled by 8, then 8 ROMs each with depth of 8 are needed. Coefficients h_0 to h_7 are stored in ROM 0, h_8 to h_{15} are stored in ROM 1… and h_{56} to h_{63} are stored in ROM 7. The counter operates at upsampling rate, which is 8 times incoming data rate, and works as the pointer of the ROM content. Since 8 sub-filters are used, the counter counts from 0 to 7 with unit step, cycles through all sub-filters at data rate, and then starts over.

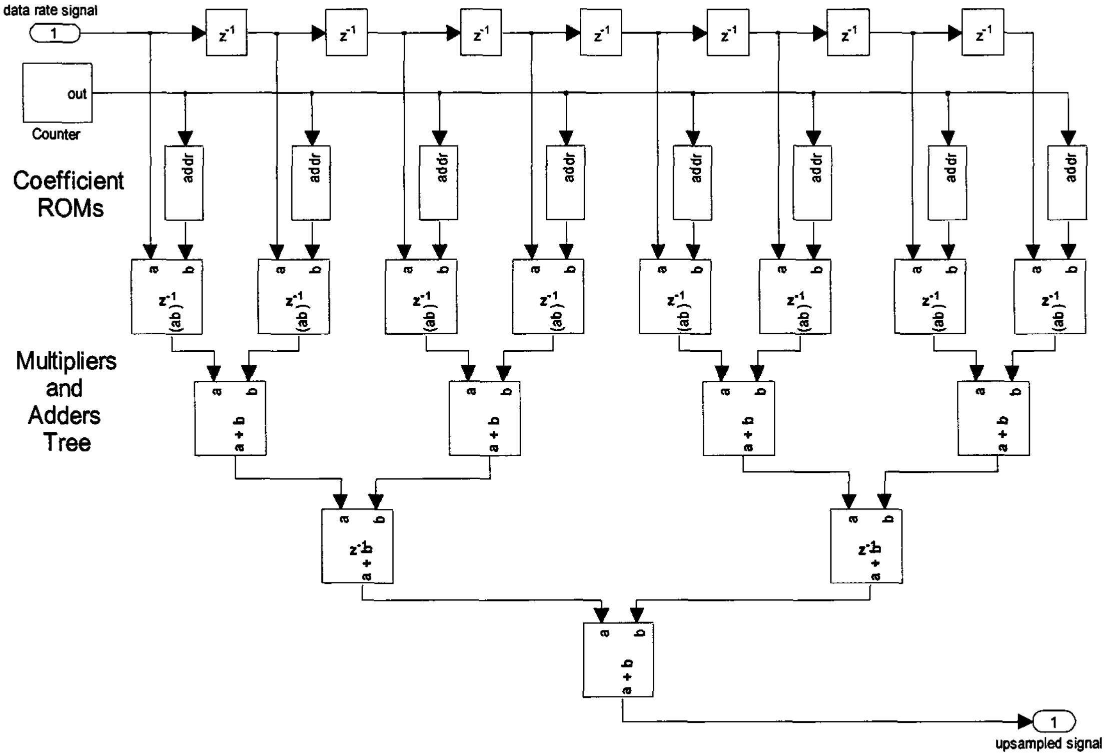


Figure 2-11 Parallel structure for polyphase interpolation filter

The polyphase filter we proposed here and other filter design in the rest of the system are all based on parallel structure. A more hardware-efficient serial structure exists, where only a multiplier and accumulator are needed compared with multiplier-and-adder tree in parallel structure. It is also called Multiply-and-Accumulate (MAC) operation [24][36]. However, the processing rate of MAC depends on the filter length, that is, MAC works at 64 times data rate if the original filter has 64 taps. Therefore, if the processing speed is limited, only very low throughput can be tolerant when MAC filter is used. In order to transmit a relatively high rate data, considering of the clock limitation of the platform, the parallel structure is more suitable for our application.

The pulse shaping filter is realized by the polyphase structure as well. System Generator provides many different filter design cores, such as filter compliers, distributed arithmetic FIR, MAC filter and so on [19]. They can also be applied to our design. However, the use of these existing cores normally cost more hardware utilization, which should be noticed in an area-saving design.

2.5 Digital Up and Down Conversion

2.5.1 Background

The up conversion of baseband signal, also called spectral translation, is performed in a communications system for two main reasons. First of all, transmission at baseband or relatively low frequency will require extremely large antenna size and signal bandwidth [5], which is strictly limited by device vendors. Secondly, the radio spectrum, which is at high frequency, can be shared by multiple users through use of frequency division multiplexing (FDM) [5]. Therefore, signal's central frequency is usually moved by an independent sinusoid carrier from DC to higher frequency for radio transmission. In a practical radio transmitter which is shown in Figure 2-12, the baseband I and Q signals are up-converted by mixing with a sinusoid carrier generated from a local oscillator (LO) with phase of 0 and 90 degrees, respectively. As a result, the I and Q signals are orthogonal and do not interfere with each other. When combined, they are summed to a composite output signal. By doing so, two independent signal components are modulated onto a single carrier wave, which can be split back into the independent components in the receiver.

Denote the baseband I and Q signal as $x_I(t)$ and $x_Q(t)$, respectively. The generated carrier wave for I and Q branches are $\cos(2\pi f_c)$ and $-\sin(2\pi f_c)$, respectively. Then the up-converted signal can be expressed as

$$r(t) = x_I(t)\cos(2\pi f_c) - x_Q(t)\sin(2\pi f_c) \tag{2.9}$$

Eq. (2.9) implies that the baseband signal is modulated onto the carrier. If complex multiplication is performed for signal $x_I(t) + x_Q(t)j$ and carrier $\cos(2\pi f_c) - \sin(2\pi f_c)j$, only the real part of the product is taken to form the signal for actual transmission. This process is illustrated in Figure 2-12, which also includes the down conversion in the receiver side. In practice, a frequency translation from baseband to an IF is performed before transmission on RF. Digital up-convertor (DUC) is such a circuit to accomplish this baseband to IF translation in digital domain, and also to ease the rest analog circuit design.

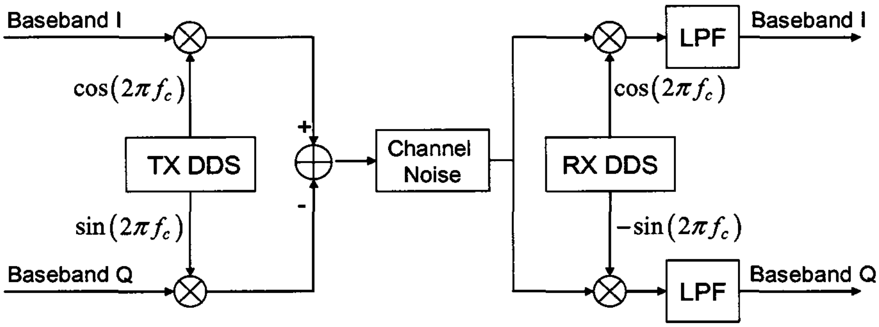


Figure 2-12 Digital up and down conversion

Before a signal is demodulated, the receiver must remove the complex envelope of the carrier, and translate the signal spectrum from IF back to baseband for the further digital signal processing. When this step is realized in digital domain, it is called digital down conversion (DDC). As shown in Figure 2-12, the received signal is mixed with the reference carrier, which is a coherent replica of transmitter carrier generated from local oscillator with a 90-degree phase shift. The composite signal is thus split back into I and Q components which are independent and orthogonal to each other. Recall that in the absence of any channel impairment the received IF signal is given by Eq. (2.9). After down-conversion, the I ($y_I(t)$) and Q ($y_Q(t)$) signal can be expressed as

$$\begin{aligned} y_I(t) &= r(t)\cdot\cos(2\pi f_c) \\ &= \left[x_I(t)\cos(2\pi f_c) - x_Q(t)\sin(2\pi f_c)\right]\cdot\cos(2\pi f_c) \\ &= x_I(t) + \text{high frequency components}, \\ y_Q(t) &= r(t)\cdot\sin(-2\pi f_c) \\ &= \left[x_I(t)\cos(2\pi f_c) - x_Q(t)\sin(2\pi f_c)\right]\cdot\sin(-2\pi f_c) \\ &= x_Q(t) + \text{high frequency components}. \end{aligned} \tag{2.10}$$

Since we are only interested with the DC part, the high frequency items and other distortion can be successfully removed by a proper low pass filter. After that, the original baseband signal is extracted from the down-converted signal. Furthermore, in order to obtain a precise DC part, the receiver carrier should be exactly the same as transmitter carrier, which means, same frequency and same offset. If such a requirement is not satisfied, carrier recovery should be performed in the receiver side, and will be discussed later.

2.5.2 Design and Implementation

In digital system design, local oscillator is realized by a direct digital synthesizer (DDS), and the DUC is composed of a DDS and a mixer. The mixer consists of 2 multipliers and a subtractor, which corresponds to transmitter blocks in Figure 2-12. The DDS synthesizes a discrete-time representation of a sinusoidal waveform, and can be used to generate the sinusoidal carrier with high frequency resolution and desired spectral purity. By using DDS, the phase, frequency and amplitude of carrier can be precisely controlled by the DSP algorithm with high speed.

The DDS is realized by a phase accumulator, a quantizer, and a sinusoid look-up table [25]. The accumulator computes a phase value with high precision from phase increment. Phase quantization is done by truncating the accumulator output, and provides a relatively low precision signal to save the memory of look-up table. The output of quantizer is the index mapped to sinusoid samples stored in the look-up table. The whole operation can be illustrated in Figure 2-13, where phase increments around a circle corresponds sample advanced through a sinusoid waveform. Different samples standing along the waveform are represented by accumulated phase values. Obviously, more samples are taken from the sinusoid, higher phase precision is needed. So, the design problem addressed here is the determination of phase increment value and phase precision. Moreover, the quantization stage produces some unwanted spurious spectral components, known as spurs, in the DDS output. So, the desired spur level should also be taken into account in the DDS design. First of all, we determine the value for carrier frequency. In practice, the IF is normally chosen as one quarter of DDS clock to lower the content precision of lookup table in DDS. If $f_c = 1/4 f_{DDS}$, during each sinusoid cycle, the cosine output is [1 0 -1 0], and the sine output is [0 1 0 -1]. As a result, only 2 bits are needed to represent the content of look-up table, which are -1, 1 and 0. Furthermore, the digital mixer can be realized by a multiplexer instead of a multiplier. In our application, the DDS works at system master clock of 100 MHz, and IF is chosen as 25 MHz.

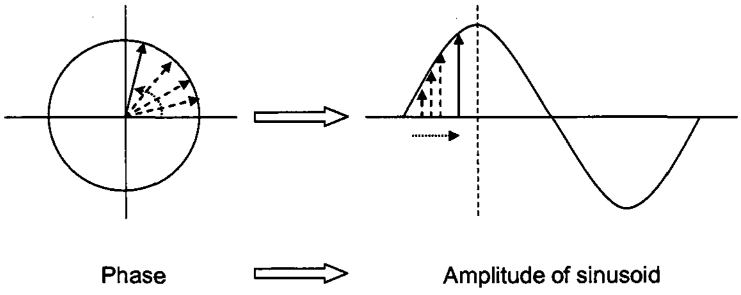


Figure 2-13 Phase to amplitude conversion in DDS

In order to generate a DDS output (f_{out}) of 25 MHz with frequency resolution (Δf) of 1 Hz, and DDS clock (f_{DDS}) is 100 MHz, the bit-precision of phase accumulator is calculated as

$$\begin{aligned} B_{acc} &= \log_2\lceil f_{DDS}/\Delta f\rceil = \left\lceil \log_2\left(100\times10^6/1\right)\right\rceil \\ &= 27 \text{ bits} \end{aligned} \tag{2.11}$$

where the $\lceil x \rceil$ denotes the ceiling operation. The required phase increment is calculated as

$$\begin{aligned} \Delta\theta &= \frac{f_{out}2^{B_{acc}}}{f_{DDS}} = \frac{25\times2^{27}}{100} \\ &= 33554432 \end{aligned} \tag{2.12}$$

which is a decimal constant represented by 27 bits. Furthermore, the previous truncation procedure introduces phase error, and results in the amplitude errors during the phase to amplitude conversion. This kind of errors refers to phase truncation spurs. Since each LUT address bit contributes approximately 6 dB of spur suppression, and in order to provide spur suppression of S dB in the DDS output, $\lceil S/6 \rceil$ bits are needed for the lookup table address. In our design, we choose 14 bits, a $2^{14}=16384$ deep table, to get the desired spur level of -84 dB.

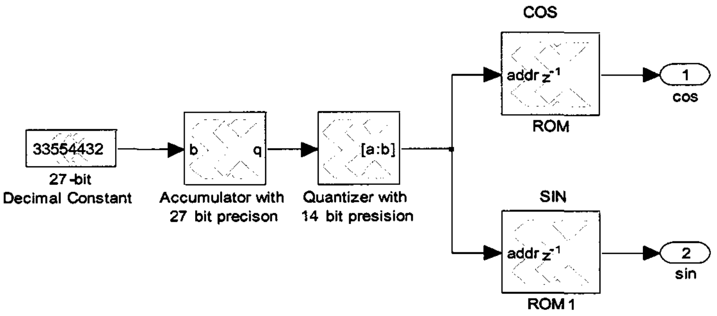


Figure 2-14 DDS block diagram

The DDC circuit is implemented using same structure and specification as DUC design. The phase increment of DDS is also identical to that of DUC in order to generate an exactly same carrier frequency.

2.6 Decimation Filter and Matched Filter

2.6.1 Background

In the receiver side, the signal needs to be down sampled to a proper rate for the following baseband signal processing. Normally, the receiver provides 2 or 4 samples per symbol to the matched filter in order to meet Nyquist criterion. If Nyquist criterion is not satisfied, adjacent spectrum copies located at downsampling rate will interference the desired signal spectrum, and so called Aliasing occurs [24]. Figure 2-15 shows the signal spectrum after downsampled by a factor of 2. To prevent the Aliasing during the downsampling process, a low pass anti-aliasing filter should be placed before down sampler. Decimation filter normally refers to the combination of low pass filter and down-sampler.

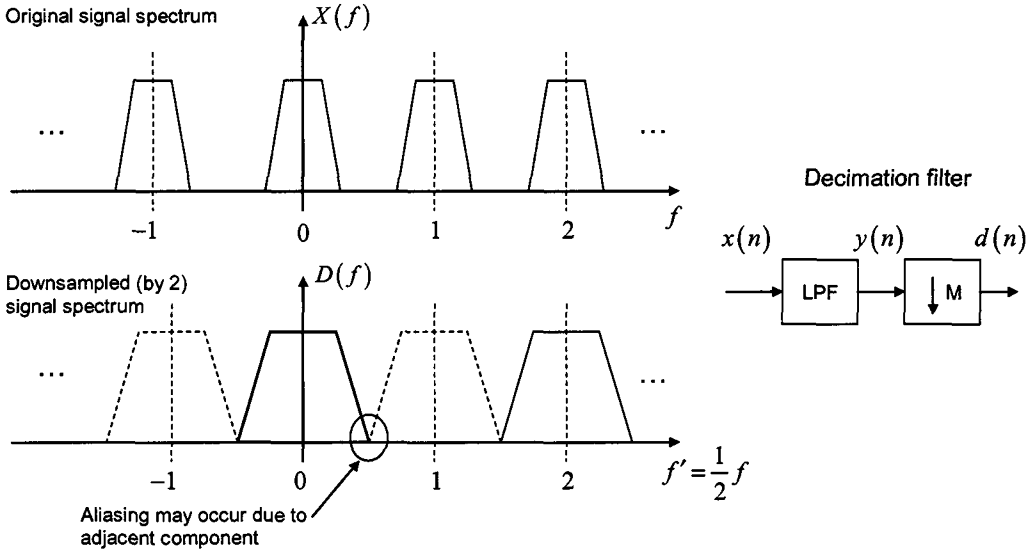


Figure 2-15 Downsampled signal spectrum and decimation filter

All digital communication systems include a low pass filter in the receiver which is intended to perform the matched filtering. The purpose of matched filtering is to minimize the effect of channel noise, and maximize the signal to noise ratio (SNR), so that the samples taken from matched filter output are reliable for the detection stage [20][29]. This is done by shaping the received signal to obtain a matched waveform of received pluses, which is a distorted version of transmitted pulses. As mentioned in the previous section, the matched filter is a SQRC filter. If $p(t)$ is denoted as impulse response of pulse shaping filter, the impulse response of matched filter is $h(t) = p(T-t)$, where T is the symbol period, and $0 \le t \le T$ [20]. Since $p(t)$ normally has a symmetric structure, $h(t)$ is identical with $p(t)$. The output of matched filter is sampled at symbol rate to produce one sample per symbol for the detection path. As long as the sampling rate, or receiver clock, is synchronous with symbol rate, or transmitter clock, the sampling instance can reside at peak of signal pulses, where the value is most reliable for the further processing. The optimum sampling time also corresponds to a clearly-opened eye diagram in the absence of any channel distortion. If such a requirement is not met, symbol timing recovery is necessary and will be discussed in detail in chapter 3.

2.6.2 Design and Implementation

To down sample a signal by a factor of M, only every M_{th} sample will be kept after downsampling, and all the other samples will be thrown away. As the inserted zeroes in the interpolation, the discarded samples that do not contribute to the decimation output can be ignored during filtering process. The polyphase partition is also suitable for decimation filter structure to reduce the unnecessary computation. First, let us derive the decimation equation, and see how polyphase is realized. Digital filter transfer function is re-written as

$$y(n) = x(n) \otimes h(n) = \sum_{k=0}^{N-1} x(n-k)h(k) \tag{2.13}$$

where $h(n)$ is the filter's impulse response with $N = S \cdot M$ taps. If $y(n)$ is to be down-sampled by a factor of M, the process can be modeled as

$$d(n) = y(M \cdot n). \tag{2.14}$$

Now, substitute $k = r \cdot M + \lambda$ in Eq. (2.14), where r and λ are both integers, and $0 \le \lambda < M-1$. Combined with Eq. (2.15), we have the following expression,

$$
\begin{aligned}
d(n) &= \sum_{k=-\infty}^{\infty} h(k)x(M \cdot n - k) \\
&= \sum_{\lambda=0}^{M-1} \sum_{k=-\infty}^{\infty} h(r \cdot M + \lambda)x((n-r)M - \lambda).
\end{aligned}
\tag{2.15}
$$

Then by setting $h_\lambda(m) = h(m \cdot M + \lambda)$ which represents the sub-filter with phase index λ, and $x_\lambda(m) = x(m \cdot M - \lambda)$ which represents the sample set that encounters with the sub-filter, the original decimation filter is successfully decomposed by polyphase structure. The final output is expressed as

$$
\begin{aligned}
d(m) &= \sum_{\lambda=0}^{M-1} \sum_{r=-\infty}^{\infty} h_\lambda(m) \cdot x_\lambda(m-r) \\
&= \sum_{\lambda=0}^{M-1} h_\lambda(m) \otimes x_\lambda(m).
\end{aligned}
\tag{2.16}
$$

As shown in Figure 2-16, the polyphase structure splits the original filter into M sub-filters, each with length of $S = N/M$, and impulse response of $h_\lambda(m) = h(m \cdot M + \lambda)$. By doing this, the unnecessary calculation is prevented. Since only every M_{th} decimation output is reserved, the overall computation savings of polyphase method can be $(M-1)/M$.

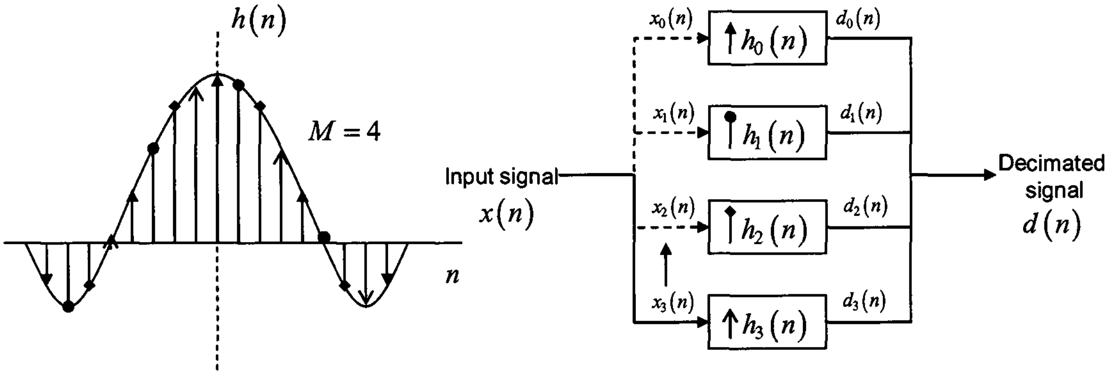


Figure 2-16 Polyphase partition for Decimation filter when $M = 4$

The filter coefficients are generated by Kaiser Window method and identical with the interpolation coefficients, so that it has the same filter properties as interpolation filter. Figure 2-17 shows the parallel polyphase structure for decimation filter. As we can see, it consists of delay line, coefficients ROMs, multiplier and adder tree, free running counter, accumulator and down-sampler. Since the signal is to be downsampled by 8, 8 sub-filters each with 8 taps are used to partition the 64- tap original filter. As we discussed in the interpolation design, parallel structure is also applied here. The number of ROMs represents the length of each sub-filter, and the depth of ROM indicates the number of sub-filters, or simply down-sampling factor. ROM 0 contains coefficients h_0 to h_7, ROM 1 contains h_8 to $h_{15}\ldots$ and ROM 7 contains h_{56} to h_{63}. The free running counter is a decrementing counter working at the incoming data rate. It counts down from 7 to 0 with unit step, and functions as index of sub-filters. Whenever the counter cycles through the ROM content, or every eight clock ticks, the accumulator is enabled to output the desired decimated value. The signal is then downsampled by 8 to ensure that sample time is increased by 8.

The matched filter we designed is a 32-tap SQRC filter, which has the same coefficients and filter properties as the pulse shaping filter in transmitter. It works at two samples per symbol, and outputs one sample per symbol for the detection path. The filter can be realized by the same polyphase structure discussed in decimation design, or other filter design blocks available from System Generator.

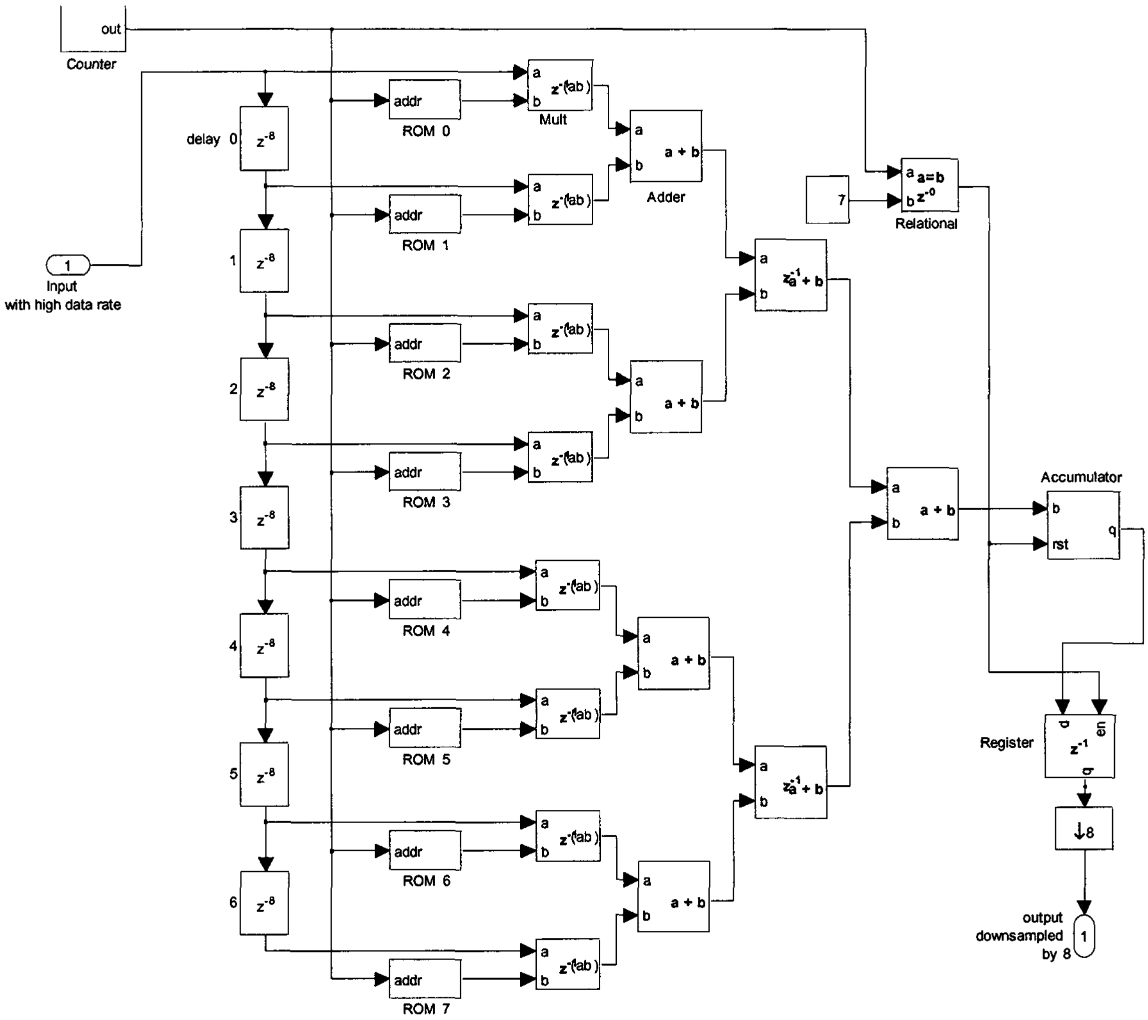


Figure 2-17 Parallel structure for polyphase decimation filter

2.7 Baseband QPSK Demodulation

The detected symbols are converted back to bit group based on QPSK mapping scheme in Table 2-1, that is, if the I and Q values are located in quadrant 1, the detected bits are 00; if they are in quadrant 2, the detected bits are 01, and so forth. The bit sets are then converted to a serial bit stream which can be compared with the original transmitting bits for BER computation. After demodulation, the symbol rate is upsampled by 2, which equals original bit rate. Please note that no channel coding scheme is used in our system. The QPSK demodulator consists of three components, two slicers used for checking the sign of I and Q values, and mapping them to according decision region (quadrant); a look-up table storing the bit values and a parallel to serial convertor. The structure is shown in Figure 2-18.

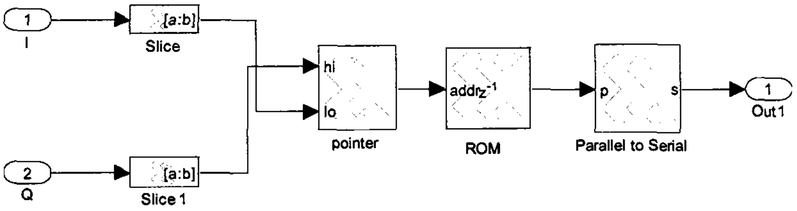


Figure 2-18 QPSK baseband demodulator

Chapter 3

Synchronization for SISO System

Synchronization is critical in receiver design of a communication system. Two common tasks are performed for synchronization between transmitter and receiver, which are clock (symbol timing) recovery and carrier recovery [27][28].

When coherent demodulation is needed, the baseband signal is derived by convolving the received signal with a local reference carrier, which has frequency and phase that match the transmitting carrier. Such an operation performing the carrier matching refers to carrier recovery (CR). On the other hand, the ultimate task of receiver is to produce an accurate replica of transmitting symbol sequence from received signal. In a baseband M-PSK or QAM system, the received signal is passed through a matched filter and then sampled at symbol rate. The optimum sampling instances correspond to the maximum eyes opening and are located at the peaks of signal pulses [27]. It is obvious that the reliability of detection depends on the location of sampling points. Such an operation determining the sample location refers to symbol timing recovery (STR).

In this chapter, we first introduce the theoretical background of carrier and symbol timing recovery, respectively. Design and implementation of these two circuits are then explained in detail. Simulation of one CR algorithm and two STR algorithms are also performed and analyzed, and the BER performance in AWGN channel is also presented.

3.1 Carrier Recovery

3.1.1 Background

As discussed in DDC design, in order to shift the central frequency of received signal from IF to DC, the receiver should introduce a carrier replica matching the frequency and phase with transmitting carrier. In practice, however, the carrier generated from the local oscillator in receiver can not be exactly the same as transmitter carrier due to the drift of internal parameters in different oscillators, or Doppler shift induced by moving objects in multipath fading channel [21][28]. In practice, ±50 ppm (parts per million) frequency accuracy is always reasonable. As shown in Figure 3-1(a), the phase offset rotates the QPSK constellation by a certain angle, and frequency offset results in circular rotation as seen in Figure 3-1(b). Movement of constellation points will introduce cross-talk between I and Q value, and mislead the detection process. From the mathematical point of view, when multiplying a different carrier in receiver, no items with DC frequency will be produced. This means the down-converted signal which should be processed in baseband is not centered at DC. This kind of offset should be tracked and corrected when coherent detection is needed, which arises the requirement of carrier recovery.

The carrier recovery design is based on phase locked loop (PLL) technique [28][46]. PLL is a close loop control system that can control the oscillator to provide a constant phase compared with a reference signal. Figure 3-2 shows a typical PLL which is composed of a phase detector, a loop filter, and a controlled oscillator. Phase detector measures the phase difference between the input signal and a reference signal. Loop filter narrows the bandwidth of phase detector output in order to provide precise control signal to the controlled oscillator, which is used to adjust the phase of the input signal. Furthermore, PLL has two distinct operation modes, acquisition and tracking. The acquisition bandwidth is controlled by the bandwidth of loop filter. The tracking bandwidth implies the range of frequency offset that the loop can follow, and is limited by the control range of the oscillator [46]. PLL technique is widely used in synchronization issue of communication system, and will also be applied to the timing recovery design.

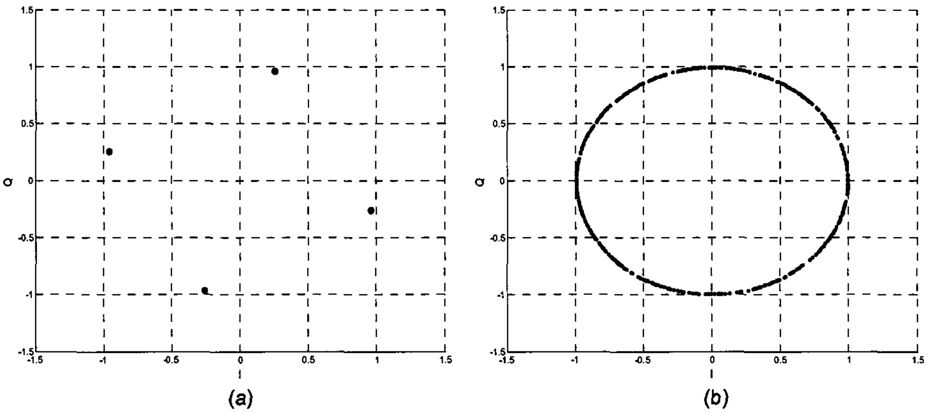


Figure 3-1 Effect of phase (a) and frequency (b) offset on the QPSK signal constellation

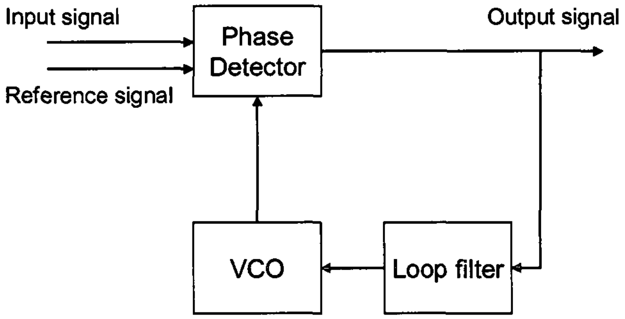


Figure 3-2 Typical PLL block diagram

The block diagram of a feedback carrier recovery loop is shown in Figure 3-3. A coarse digital down conversion using a non-synchronous reference carrier compared with the transmitter carrier is first performed in the receiver side. The carrier loop is responsible for tracking and compensating the residue frequency and phase offset. Similar to the traditional PLL, there are three basic components in CR loop, that are phase error detector, loop filter and control unit. Phase error detector computes the phase error by comparing the received signal and its closest reference signal on the constellation. The loop filter is responsible for tracking both phase and frequency error. It also determines the PLL bandwidth, and provides the control signal to drive the control unit. By measuring the filtered phase error, the control unit can generate a nearly constant phase that feedbacks to original data path, and realizes the phase and frequency adjustment. In analog or hybrid carrier recovery, the control unit is realized by a voltage-controlled-oscillator (VCO), which can be used to adjust the frequency of local oscillator [28]. On the FPGA platform, the frequency of reference carrier in the receiver is made fixed, so a fully digital carrier recovery design is required, and VCO is replaced by a numerical-controlled-oscillator (NCO).

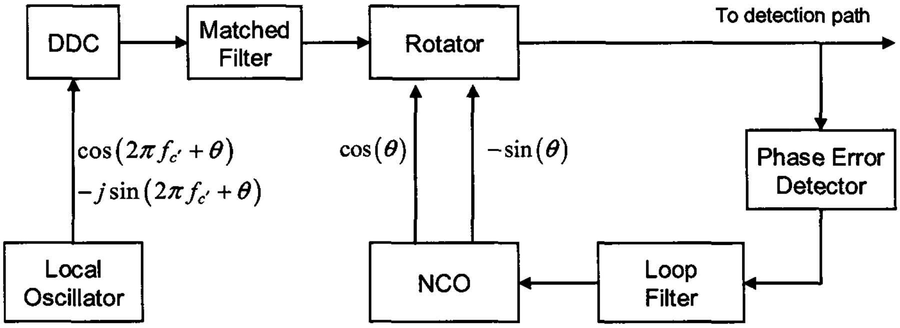


Figure 3-3 Feedback carrier recovery block diagram

3.1.2 Design and Implementation of CR Loop

3.1.2.1 Phase Error Detector

The phase error detector computes the phase difference between the received signal and local carrier replica. For a general M-PSK or QAM receiver, this difference is represented by the angle between the received signal point and the nearest constellation point, i.e., $\arctan(Q/I)$. Directly computing an angle is always with high complexity, however, this process can be simplified for QPSK signal. Recall that when phase offset θ is very small, $\sin(\theta) \approx \theta$, and $\sin(\theta)$ is monotonic increasing with θ when $-90^\circ \le \theta \le 90^\circ$. Therefore sine of phase offset $\sin(\theta)$ is a good approximation of θ when θ is very small [30]. Figure 3-4 shows the received complex signal $x+yj$ and its nearest constellation point $m+nj$ on the QPSK constellation plot. Instead of calculating the angle θ, we calculate $\sin(\theta)$ which is expressed as following.

$$\begin{aligned}\sin(\theta) &= \sin(\alpha-\beta) \\ &= \sin\alpha\cdot\cos\beta-\cos\alpha\cdot\sin\beta \\ &= \frac{my-nx}{\sqrt{\left(x^2+y^2\right)\left(m^2+n^2\right)}}\end{aligned} \tag{3.1}$$

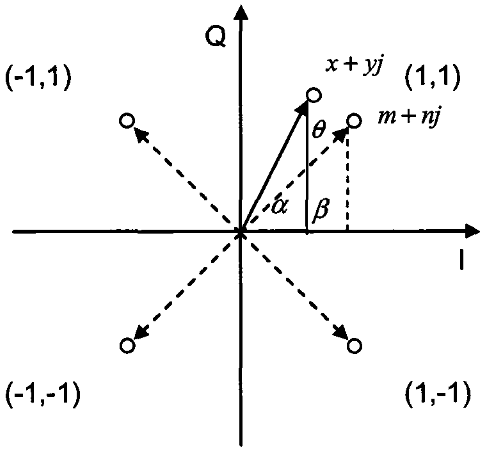


Figure 3-4 Received signal with phase error on QPSK constellation

Since the four constellation points are located at four quadrants, the nearest points are simply computed by taking the sign of I and Q value of received signal. So the expression can be modified as

$$\begin{aligned}\sin(\theta) &= \frac{my-nx}{\sqrt{\left(x^2+y^2\right)\left(m^2+n^2\right)}} \\ &= \frac{sign(x)\cdot y-sign(y)\cdot x}{\sqrt{\left(x^2+y^2\right)\left(m^2+n^2\right)}}\end{aligned} \tag{3.2}$$

Based on above equation, the phase error detector can be realized by two sign detectors, two multipliers and a subtractor. The structure is shown in Figure 3-5. The CR loop with phase error detector of this structure is normally called Costas loop [20]. This sign-detection based design is especially suitable for QPSK signal, however, when modulation order goes high, such as 8PSK, 16QAM and so on, this structure will break down, and more precise algorithm is needed for angle calculation.

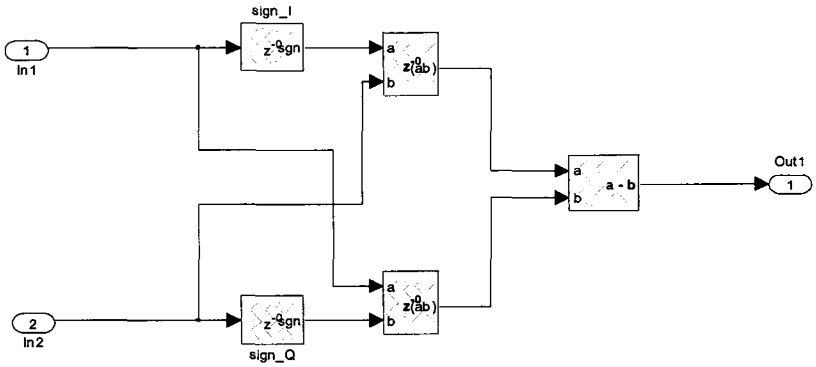


Figure 3-5 Phase error detector for QPSK signal

3.1.2.2 Loop Filter

The loop filter in carrier recovery loop is a proportional-integral IIR (Infinite Impulse Response) filter, where the proportional path tracks the phase error, and the integral path tracks the frequency error [46]. It provides a filtered phase error signal with high precision for NCO processing, and normally 32 bits are required. Figure 3-6 shows the structure of a typical digital loop filter which consists of two multipliers performing multiplication of input signal with filter constant, an accumulator and an adder. As shown in the figure, the upper branch represents the proportional path, while the lower indicates the integral path. Another design issue of loop filter is the determination of loop constant which determines the loop bandwidth, acquisition time and stability of error tracking performance [46]. Larger constant, hence larger loop bandwidth will shorten the acquisition time, but increases spurious noise. In our design, the filter constants are chosen to make the loop bandwidth 0.1% normalized to the symbol rate to make a compromise between acquisition time and noise level.

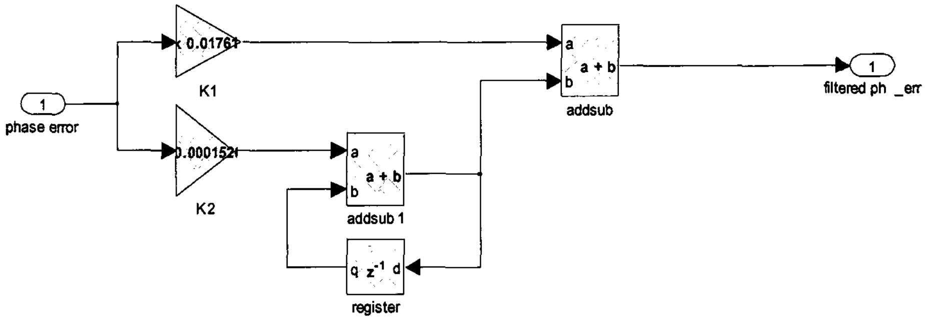


Figure 3-6 Digital loop filter

3.1.2.3 NCO

The NCO mentioned here is actually a phase truncation DDS which has the same structure of DDS used in digital up and down conversion. It consists of a phase accumulator, a quantizer, and a sine/cosine LUT [25]. The input of accumulator is the filtered phase error instead of a pre-calculated constant in DUC/DDC design. In order to save the memory of look-up table, the quantizer truncates the accumulator output to a low precision value, which is used as the address signal for LUT. Similar to DUC/DDC design, in order to obtain a spur level of -84 dB and frequency resolution of 1 Hz, it is required to have a NCO which consists of a 27-bit phase accumulator, a 14-bit quantizer, and two 16384-deep LUTs with output precision of 16 bits. As shown in Figure 3-7, the negation of sine value, i.e., the conjugate of generated complex sinusoid is taken. The reason for using the negated sine is because we assumed a positive (counter-clockwise) angle rotation for received signal in Figure 3-4. Now, we are intended to rotate it back with a negative (clockwise) angle. The DDS works at symbol rate of 12.5 MHz, and sine/cosine samples are stored in a Block RAM [48] in Virtex-4 FPGA.

In order to align the received carrier and local reference carrier, the output of DDS and unrecovered signal are fed into a mixer to perform phase rotation. The mixer is a complex multiplier, and can be realized by two multipliers, one adder and one subtractor.

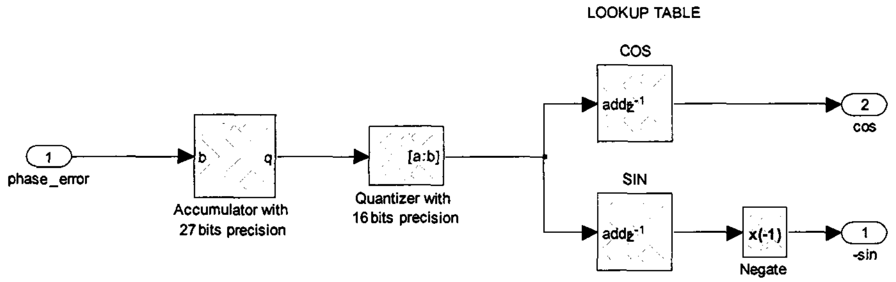


Figure 3-7 NCO block diagram

3.1.2.4 Phase Ambiguity Problem

As the constellation of QPSK signal shown in Figure 3-8(a), we notice that if the signal rotates for 90, 180, or 270 degrees, the carrier recovery loop will lock to a new phase that is 90, 180, or 270 degree offset from the correct phase, or the constellation can simply overlap itself after rotation. This is mainly due to the periodicity of sine function used in the phase error detector and property of QPSK signal. This kind of rotation makes the carrier recovery loop unable to distinguish the reference carrier phase from the correct phase. To solve the phase ambiguity problem, deferential encoding and decoding are applied in our system [20][21]. Instead of mapping each bit set to symbol, differential coding maps the difference of every two consecutive bit sets to according symbols. One of the bit sets is previously coded, and another one is currently un-coded. Since the phase ambiguity does not change the phase difference of two consecutive symbols, the received symbols can still be decoded correctly. Figure 3-9 illustrates the BER performance of coherently detected QPSK signal using differential encoding over AWGN channel [20]. As shown in the figure, the BER of differential coding is normally twice the BER of conventional QPSK, which results in a 0.3 dB performance degradation. However, when SNR is high, this difference is not evident.

The differential encoding can be performed either before or after baseband modulation. The latter method uses a multiplier and an adder/subtractor, while the former mainly uses look-up table which is resource-efficient for FPGA implementation. As a result, we choose the LUT scheme for the design. Let's first make an example here and discuss the structure of encoder and decoder. Assume that the transmitted bits are 01 01 11 01 10, and bit sets *00, 01, 11, 10* are represented for phase offset of *0, 90, 180, 270* degrees, respectively. The encoding process is shown in Table 3-1(a). After that, the encoded bits are mapped to QPSK symbols. The constellation is the same as the traditional one with Gray coding, but has a different implication of transmitted bits (Figure 3-8(b)). The decoding process is shown in Table 3-1(b). As we can see, the bits are correctly decoded.

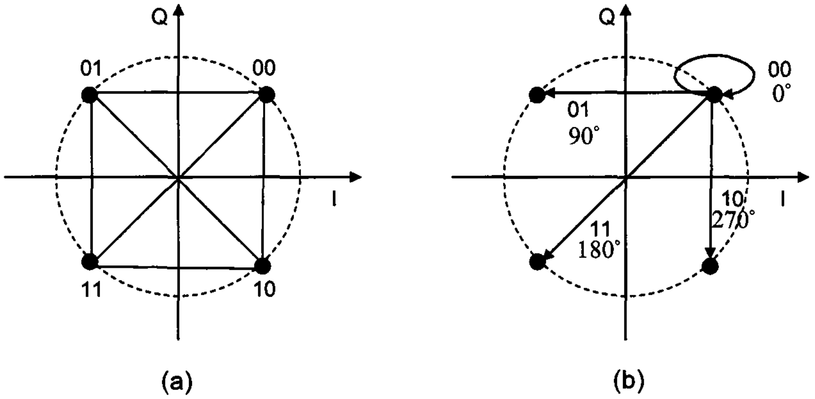


Figure 3-8 QPSK constellation with Gray coding (a) and differential coding (b)

Bit set 1	Bit set 0	Rotation angle 1 (degree)	Encoded bit set
01	00 (default)	0	01
01	01	90	11
11	11	180	00
01	00	0	01
10	01	90	00

(a)

Received bit set 1	Received bit set 0	Rotation angle 2 (degree)	Decoded bit set
01	00 (default)	0	01
11	01	90	01
00	11	180	11
01	00	90	01
00	01	270	10

(b)

Table 3-1 Differential encoding (a) and decoding (b) process

As seen from the table, the encoded bits are 01 11 00 01 00, which are then decoded correctly. Please note that Bit set 1 is the transmitting bits, and Bit set 0 is written by observing the previous encoded bit set. The rotation angle 1 is the angle represented by bit set 0 based on the bit-angle mapping, and the rotation angle 2 is decided by looking at the transition from set 0 to set 1.

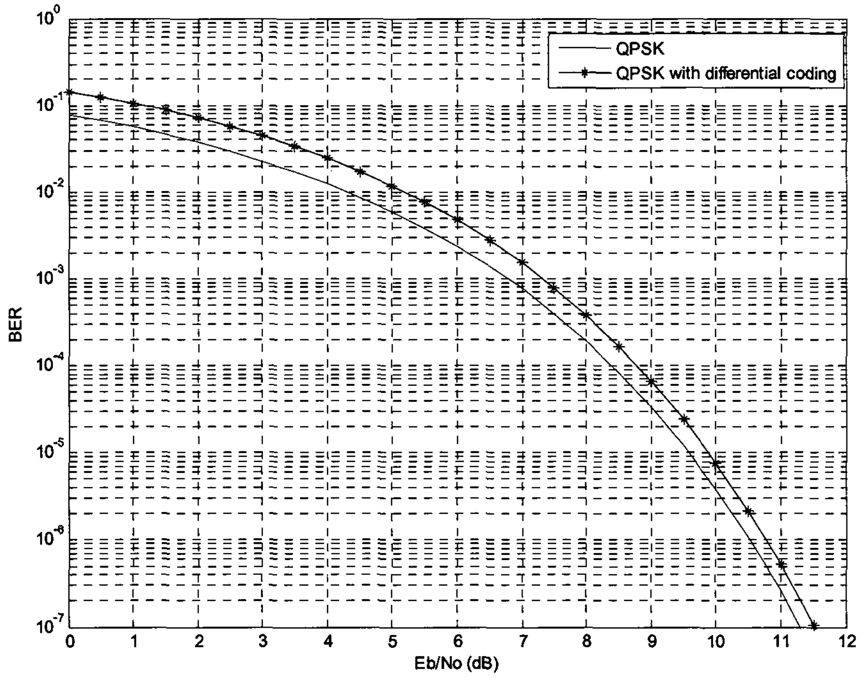


Figure 3-9 BER performance of QPSK with differential encoding over AWGN channel

Since differential coding is applied to our system, a minor change should be made for the block diagram of the SISO system shown in Figure 2-2. In the transmitter, a differential encoder is placed before QPSK modulator, and in the receiver, differential decoder is placed after QPSK de-modulator. The differential encoder can be implemented by a serial to parallel convertor (SPC), a concatenator and a LUT. As shown in Figure 3-10, the SPC groups every two bits to form a bit set. The concatenator connects two consecutive bit sets, and provides the index pointing at the lookup table content. The order of table content is essential, which should be in accord with the mapping scheme explained in Table 3-1. The depth of lookup table is dependent on the modulation scheme. For QPSK signal, a 16-deep ROM is required. The differential decoder is realized by the same components, but with different arrangement (Figure 3-11). Finally, the decoded bits are converted to a serial bit stream through a parallel to serial convertor (PSC).

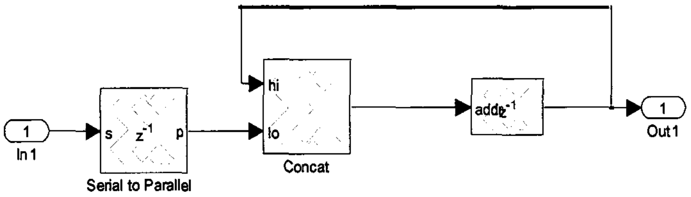


Figure 3-10 Differential encoder

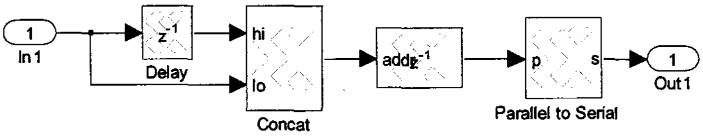


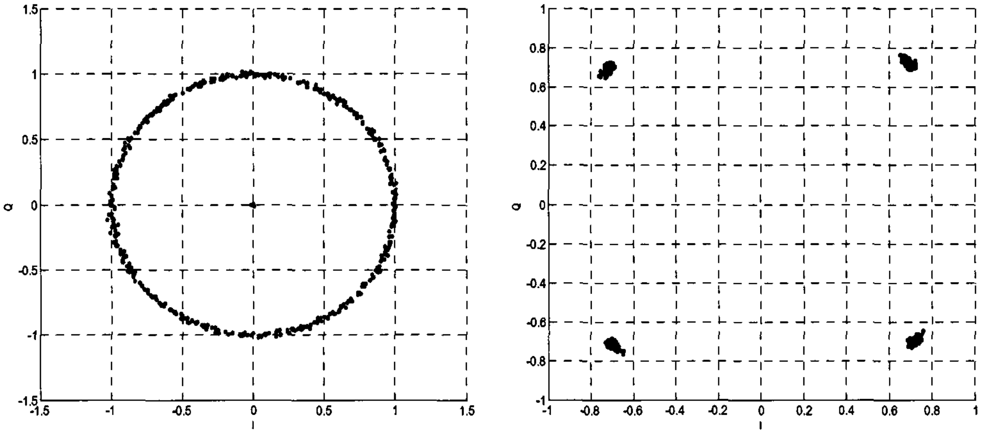
Figure 3-11 Differential decoder

3.1.3 Simulation and Analysis

The simulation of carrier recovery is performed using System Generator blocks with fixed point representation and Simulink blocks with floating point representation. The loop performance is analyzed by observing the constellation plot and phase error signal, and BER versus E_b/N_0 curves are also given with different simulation scenarios. The simulation is based on the following specifications.

- Modulation scheme: QPSK with differential coding
- Oversampling rate: 16
- Roll-off factor: 0.35
- Channel: AWGN
- Normalized frequency offset: 0.001 Hz
- Loop bandwidth: 0.1% normalized to symbol rate.

The received signal constellation before carrier recovery is shown in Figure 3-12(a). As we can see, the frequency offset causes the constellation to rotate. After carrier recovery is performed, last 300 of total received 600 symbols are presented in the plot. It is clear that the loop is locked to a stable status, and four QPSK signal points are presented on the constellation. The phase ambiguity problem is prevented by using differential coding, so that these points seen on the constellation are most reliable for the detection stage.



(a) (b)

Figure 3-12 Signal constellation before (a) and after (b) carrier recovery

A good observation point for watching loop behavior is the phase error as provided into the NCO [46]. Ideally this term should become zero when the loop reaches a stable locked status. The phase error signal in the absence of channel distortion is shown in Figure 3-13. After about 300 symbols, the error signal converges to 0, and keeps up and down in a small range due to the randomness of transmitted signal and spurious noise induced by loop itself. NCO can then provide a nearly constant signal which represents the frequency and phase offset between the transmitter carrier and local reference. The 300 symbols is the acquisition time of our carrier recovery loop. After that, the loop is successfully locked.

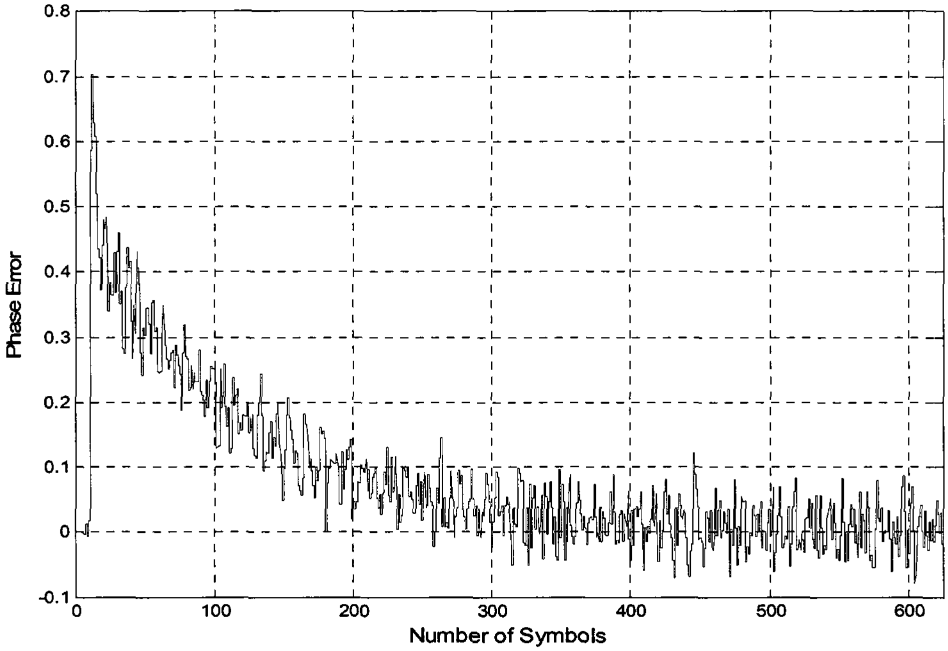


Figure 3-13 Phase error of carrier recovery loop @ 0.001 Hz frequency offset

Furthermore, three BER curves with different simulation scenarios are shown in Figure 3-14. Theoretical curve simply results from BER expression of QPSK signal with differential coding in AWGN channel [20]. The floating-point simulation of a perfect phase-matching system with oversampling factor of 16 is performed as well. As we can see in the figure, the performance agrees with the theoretical result. Finally, assume that system experiences a normalized frequency offset of 0.001 Hz, and carrier recovery is performed in the receiver. The BER curve shows that PLL based carrier recovery algorithm has about 0.3 dB performance degradation than the ideal scenario. The degradation is mainly due to the self noise of the CR loop. We can further conclude that this algorithm is not severely affected by AWGN channel.

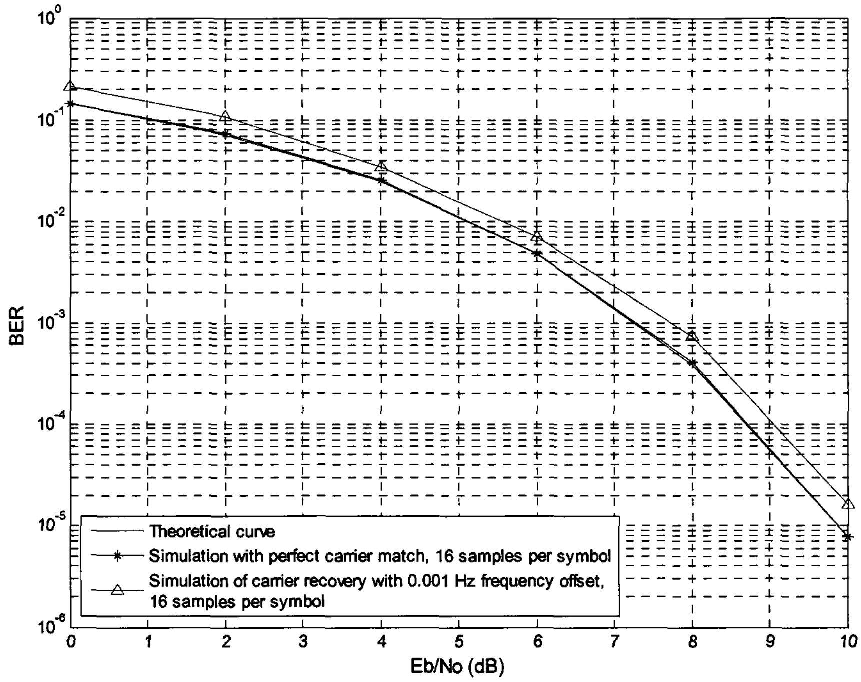


Figure 3-14 BER performance of carrier recovery

3.2 Symbol Timing Recovery

3.2.1 Background

Timing synchronization is essential for a practical communication system. As we know, the analog signal is first fed into an ADC in the receiver side before digital signal processing. In the ideal scenario, the transmitter and receiver share the same clock with exactly the same frequency and phase. If ADC samples the analog signal with a rate satisfying Nyquist criterion, the output of matched filter can be sampled at symbol rate to obtain highest SNR. In practice, however, the transmitter and receiver do not share clock information with each other and some timing offset between the two clocks is inherent. Under this circumstance, the matched filter output introduces ISI, which severely affects the following signal detection, and degrades the system performance. In order to track and compensate for this kind of offset, and sample the matched filter output at correct sampling interval which is synchronous to transmitting symbol rate, symbol timing recovery (STR) is critical in the receiver side.

PLL technique is also applied to STR design, where timing error detector (TED), loop filter and controlled oscillator are three basic components. The timing error detector computes the timing error by comparing current sampling time with the optimum sampling time. The error signal is then filtered and fed into a VCO or NCO for controlling the timing adjustment. The control signal feeds back to the un-recovered signal path to accomplish the adjustment. In the traditional analog receiver, the sampling clock of ADC can be adjusted by a control signal from timing recovery loop to obtain phase alignment with transmitting symbol rate. This method is called synchronous sampling [28], and the block diagram is shown in Figure 3-15. In modern digital receiver, the timing adjustment is fully realized in digital domain. Instead of changing the ADC clock, the optimum sampling instance can be obtained by shifting the received signal forward or backward. This is normally called non-synchronous sampling [28] and the block diagram is shown in Figure 3-16. Consider that the ADC sampling clock is always fixed on the FPGA board, and conventional analog or hybrid timing recovery gives more complex circuit to the receiver design where an additional DAC is needed between STR loop and adjustable ADC. Therefore, we are aiming to implement an all-digital STR loop. Building the circuit fully in digital domain is also a requirement of the SDR based system.

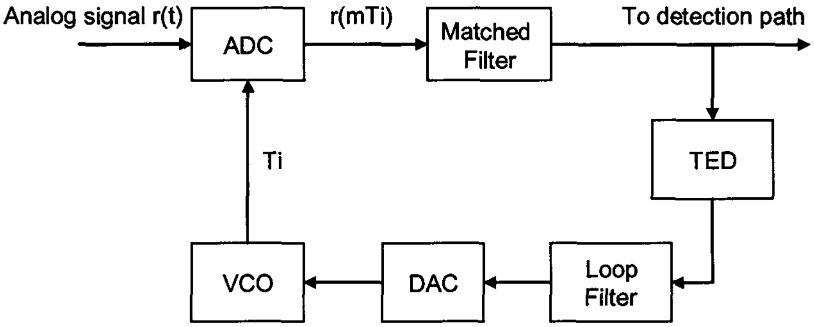


Figure 3-15 Analog synchronous sampling

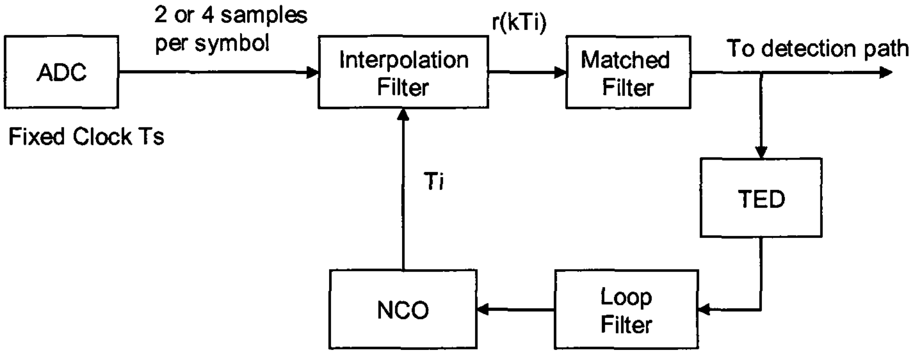


Figure 3-16 Digital non-synchronous sampling

As discussed above, timing adjustment is accomplished by moving the signal to the optimum sampling points rather than changing the physical clock. This requires a certain number of samples presenting between each symbol for the movement. However, as the signal provided into matched filter is normally with two or four samples per symbol, which is far from enough to satisfy the precision requirement of timing adjustment. As a result, interpolating the matched filter input is desired. Interpolation is one of the most popular STR schemes, and is widely used in digital receiver design. Interpolation in timing synchronization has been covered extensively in literature [27][28]. Here, we first review the mathematical model of interpolation devised by Gardner [32].

Designate that T is symbol period, and T_s is receiver sampling interval, then the ratio T/T_s is always irrational due to the fact that T is incommensurate with T_s [32]. The signal appearing at input of STR loop is sampled at least twice the symbol rate in order to satisfy Nyquist sampling theorem. Normally, two samples per symbol are taken from interpolation, and only one of them is chosen for symbol detection based on decision of NCO. Here, we rewrite the interpolation equation presented in [32]. The interpolant signal, which is the output of interpolation filter, is given by

$$y(kT_i)=\sum_m x(mT_s)h_I(kT_i-mT_s) \tag{3.3}$$

where T_i is interpolant interval, $x(mT_s)$ is the signal at input of timing recovery loop with sampling rate of $1/T_s$, and $h_I(nT_s)$ is impulse response of interpolation filter. Furthermore, by using the definition of basepoint index [32]

$$m_k = \mathrm{int}[T_i/T_s] \tag{3.4}$$

and fractional interval

$$\mu_k = kT_i/T_s - m_k \tag{3.5}$$

Then Eq. (3.3) can be modified as below

$$y(kT_i) = \sum_{i=I_0}^{I_1} x((m_k - i)T_s)h_I((i+\mu_k)T_s) \tag{3.6}$$

where the filter length is $I = I_1 - I_0 + 1$. Eq. (3.6) is the well known interpolation equation, and Figure 3-17 illustrates the relation between interpolant interval T_i and receiver sample time T_s. Recall that we are shifting the signal instead of changing the physical clock, the sample period of interpolator input and output are actually the same. However, the output is flip-flopped by a signal strobe coming from NCO. A new interpolant is computed and output only when the strobe goes high. Therefore, the average of interpolant interval T_i represents the estimated symbol time. When the loop is locked, T_i should be nearly a constant, and synchronous with symbol time.

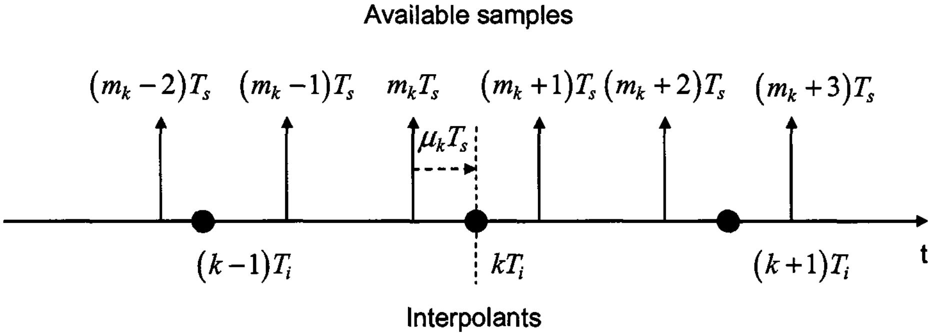


Figure 3-17 Sample time and interpolant time relation

3.2.2 Main Components in STR loop

In a PLL based STR design using interpolation technique, interpolator/matched filter, timing error detector, loop filter and NCO are the key components of STR loop. In this section, these components are discussed except for loop filter, which has been introduced in carrier recovery design. Moreover, two different algorithms for timing error detection are emphasized. The detailed design and implementation of STR loop will be introduced in the next section.

3.2.2.1 Interpolator

In the widely proposed STR schemes using interpolation technique, the matched filter is placed either inside or outside the loop, and different methods for interpolation filter design are also well introduced [27][28][33][41]. In our design, the matched filter and interpolation filter are rolled into a same stage where a so called interpolated matched filter is used for the timing adjustment [35][42]. The new matched filter is actually an upsampled version of original matched filter, and each interpolant between each sample can represent a different timing offset with respect to symbol interval. When polyphase structure is further applied to the filter design, each sub filter corresponds to a different timing offset. In other words, selecting the correct sub filter means making the correct timing adjustment.

Let us now examine how polyphase can be applied to the timing recovery design. Recall the polyphase issue we discussed in interpolation filter design, the impulse response of sub-filter is expressed as

$$h_\lambda(n) = h(n \cdot L + \lambda) \tag{3.7}$$

where L is upsampling factor, and λ is an integer with $0 \le \lambda < L-1$. As we can see, sub-filter can be derived from original filter with a phase index λ. If L sub-filters are used for polyphase structure of timing recovery, the impulse response of m_{th} $(0 \le m < L)$ sub-filter can be expressed as

$$h_m(nT_s) = h_I\left(nT_s + ((m/L)T_s)\right) \tag{3.8}$$

where $h_I(nT_s)$ is impulse response of original interpolation filter. Then the output of the m_{th} sub-filter is given by

$$\begin{aligned} y_m(kT_i) &= \sum_{i=-2P}^{2P} x\left((n-i)T_s\right)h_I\left(\left(i + \frac{m}{L}\right)T_s\right) \\ &= y\left(nT_s + \frac{m}{L}T_s\right) \end{aligned} \tag{3.9}$$

where P is the symbol span of original interpolation filter, and oversampling factor is 2. Since the interpolants are computed and output at time $kT_i = (m_k + \mu_k)T_s$, we can conclude that as long as $n = m_k$, the following equation is established

$$\begin{aligned} y_m(kT_i) &= y\left(m_kT_s + (m/L)T_s\right) \\ &= y\left(m_kT_s + \mu_kT_s\right) \end{aligned} \tag{3.10}$$

Eq. (3.10) clearly shows that the ratio m/L plays the same role as μ_k. Furthermore, instead of computing explicitly, basepoint m_k is indicated by the signal strobe of NCO in order to align the correct input sample set with the filter coefficients set.

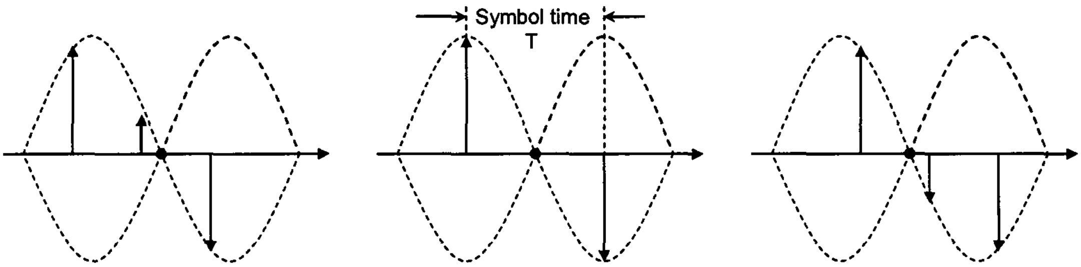
3.2.2.2 Timing Error Detector

Timing error detector (TED) computes the timing error based on the comparison of current sample timing and estimated symbol timing. Two common structures exist for timing detection using PLL technique, that are Decision Directed (DD) structure which uses the data decision to perform detection, and Non-data Aided (NDA) structure, where the detection is independent of the data decision [28]. Different types of TED designs are introduced broadly in literature. Here, we discuss and analyze two algorithms mostly used in hardware implementation for timing error detection which are Gardner algorithm [31] and Maximum Likelihood (ML) technique [26][35]. Both of them are NDA detectors.

When Gardner detector is used, the timing loop looks at the zero crossing point of eye diagram, and requires two samples per symbol to form the timing error for each symbol. Specifically, if there is a sign transition between two symbols, the average mid-way sample is zero in the absence of timing offset, which is shown in Figure 3-18(b). When timing error occurs, the product of slope of the two samples and mid-way samples gives timing error information, and one of the two samples becomes the symbol strobe, which is used for detection. If no sign transition exists between two symbols, the strobe values are the same and mid-way sample is simply ignored. The error signal of Gardner's detector for QPSK modulation is given as [31]

$$\begin{aligned} e(t) &= x_I(t-1/2)\left[x_I(t)-x_I(t-1)\right]+x_Q(t-1/2)\left[x_Q(t)-x_Q(t-1)\right] \\ &=\Re\left\{x(t-1/2)\left[x^*(t)-x^*(t-1)\right]\right\} \end{aligned} \tag{3.11}$$

where $x(t)$ and $x(t-1)$ are samples spacing by symbol time, and $x(t-1/2)$ is mid-way sample.



Timing is advanced
(a)

Timing is optimum
(b)

Timing is retarded
(c)

Figure 3-18 Timing relation on the eye diagram

Gardner's detector is immune to the carrier offset, thus timing recovery can be entirely performed prior to carrier recovery [29]. However, the performance depends on the roll-off factor β [31]. When $\beta<1$, the zero crossing condition does not occur at middle of two symbol-strobe points. The movement around midway point causes self-noise, which disappears only if $\beta=1$.

As β decreases, the performance of Gardner's algorithm becomes worse. As a result, Gardner's detector is not suitable for application of low roll-off factor, or system with bandwidth-efficiency requirement. In the coming section of simulation and analysis, we will examine the performance of Gardner's detector with different values of β.

On the other hand, ML detector measures the timing error by computing the slope (or derivative) of consecutive samples in the receiver. As we know, if samples are taken at peak of signal pulse of the matched filter output, the correlation function that results from the convolution process of matched filter has a zero derivative (Figure 3-19(b)). Otherwise, the derivative will be either positive or negative when sample is early or late, respectively. Therefore timing error can be simply measured by the derivative. However, with the same timing offset, no matter it is early or late, positive samples and negative samples will give different slopes, which means derivative alone does not provide sufficient information to determine if the timing should be advanced or retarded. Therefore, the sign of samples should be taken into account in the timing error detection process. The error signal of ML detector can be expressed as [26]

$$e(t)=\begin{cases} m(t)\cdot\hat{m}(t)\cong m(t)\cdot\left[m(t+\Delta t)-m(t-\Delta t)\right], & \text{when SNR is low} \\ sign\left[m(t)\right]\cdot\hat{m}(t)\cong sign\left[m(t)\right]\cdot\left[m(t+\Delta t)-m(t-\Delta t)\right], & \text{when SNR is high} \end{cases} \qquad (3.12)$$

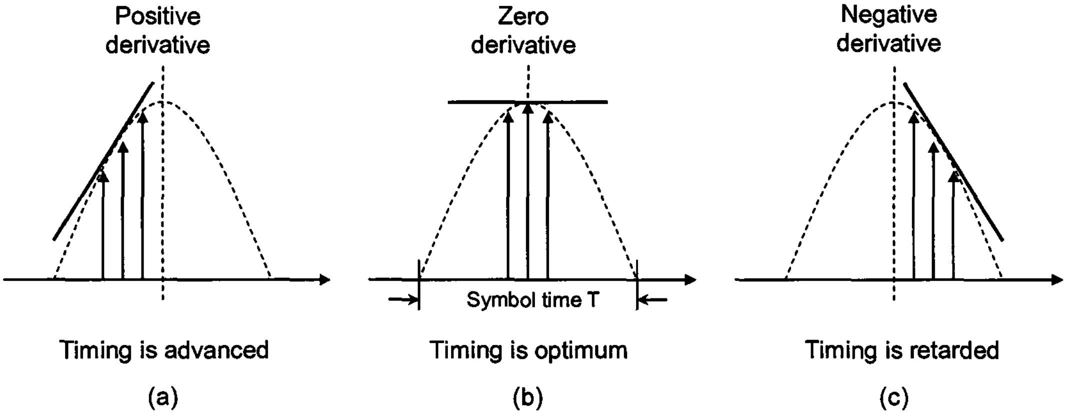


Figure 3-19 Relation of timing and derivative using ML/ELG detector

The ML timing recovery process minimizes the above error, which is the product of matched filter output $m(t)$ and its derivative $\hat{m}(t)$. We also note that when SNR is very low, the sample itself is more reliable rather than its sign. In practical design, we do not bother the switching and simply take the sign value for the whole SNR range [26]. The early-late-gate (ELG) algorithm gives approximation of ML approach and is widely used in the early days. It normally operates with three samples per symbol time, where two of them are time advanced and delayed respectively to an intermediate sample point. By comparing the difference between the 'early' and 'late' samples, this algorithm can form the first central difference of matched filter output [26][29]. In modern receiver design, instead of using two additional sub filters to form the early and late sample, a single filter whose impulse response is given by the derivative of matched filter's impulse response is used [26]. As a result, three filter stages are replaced by two filters which are matched filter and its derivative filter, for each timing error calculation. These two filters are both working at two samples per symbol, and the product of their output forms the timing error during each symbol time. Compared with Gardner's detector, ML detector is not independent with carrier offset but gives better SNR and BER performance. When ML timing error detector is applied in the receiver, carrier recovery loop should be placed inside the timing recovery loop, which is shown in Figure 2-2.

3.2.2.3 NCO

The task of NCO is to determine basepoint index m_k and fractional interval μ_k. As mentioned before, the former represents the correct set of signal samples, and the latter indicates the correct set of filter coefficients. The NCO is actually a decrementing modulo-1 counter. The NCO output can be written as

$$c(n+1)=\left[c(n)-w(n)\right]\bmod 1 \tag{3.13}$$

where $w(n)$ is a control word containing the filtered timing error and a constant, which affect the free running period of NCO. If no timing error appears at input of NCO, its period is totally determined by the constant. Since the counter decreases itself a value of w every sampling time T_s, underflow will occur every $1/w(n)$ clock ticks. Therefore, the NCO period, which represents the interpolant interval, is $T_i = T_s/w(n)$, The underflow condition corresponds to the optimum sampling instance, and also indicates that a new interpolant should be computed and output, so that basepoint index m_k is determined. The fractional interval is given by [32]

$$\mu_k = \frac{c(m_k)}{1 - c(m_k + 1) + c(m_k)} = \frac{c(m_k)}{w(m_k)} \tag{3.14}$$

To avoid division in the equation, recall that $T_i = T_s/w(n)$, $1/w(n)$ can be replaced by a normalized value $\varepsilon \cong T_i/T_s$. The expression then can be modified as [32]

$$\mu_k = \varepsilon \cdot c(m_k) \tag{3.15}$$

As we can see, μ_k is determined by multiplying a normalized value with the NCO content at underflow condition. When polyphase structure is used, calculating μ_k is equivalent to calculating m/L. The result is quantized to indicate a sub filter index. The quantization degree is determined by the number of sub-filters L.

3.2.3 Design and Implementation of STR Loop

In this section, the detailed design and implementation issues for the main components of STR loop are discussed. The loop architectures for two different timing error detectors are also provided.

3.2.3.1 Interpolated Matched Filter

The parallel polyphase structure is applied to interpolated matched filter design. The sub filter index is provided by NCO instead of a free running counter, and the filter output is enabled by a strobe signal indicating underflow condition of NCO content. As mentioned in [40], in order to obtain an acceptable implementation loss of 0.12 dB, which is due to timing quantization in NCO, the desired normalized timing resolution is $1/33$ for 4-QAM modulation with a roll off factor of 0.3. Since our system is based on QPSK with a roll off factor of 0.35, we choose 32 sub filters for the implementation. Note that increasing the timing resolution can not compensate other kinds of implementation loss. The original 32-tap matched filter is then upsampled by 32 to produce an interpolated matched filter with 1024 coefficients. The coefficients are generated by Matlab filter design tool, and stored in 32 LUTs each with depth of 32.

3.2.3.2 Timing Error Detector

We designed two timing error detectors as previously discussed. When ML detector is used, a derivative matched filter realized by the same parallel polyphase structure is required, and the coefficients are obtained by convolving the matched filter coefficients with sequence $\begin{bmatrix}-1 & 0 & 1\end{bmatrix}$. Both of these two filters are working at two samples per symbol, and the product of their output gives the estimated timing error. The sign of matched filter output is taken for large SNR approximation. Figure 3-20 shows the structure of STR loop using ML detector. On the other hand, the use of Gardner's detector reduces hardware utilization since only a single matched filter is necessary as compared to two in ML detector. Figure 3-21 shows the structure of STR loop using Gardner's detector. Based on Eq. (3.11), the implementation of Gardner's detector is relatively easy, where delay blocks, 2 multipliers, and 3 adders/subtractors can realize the function. The circuit is shown in Figure 3-22.

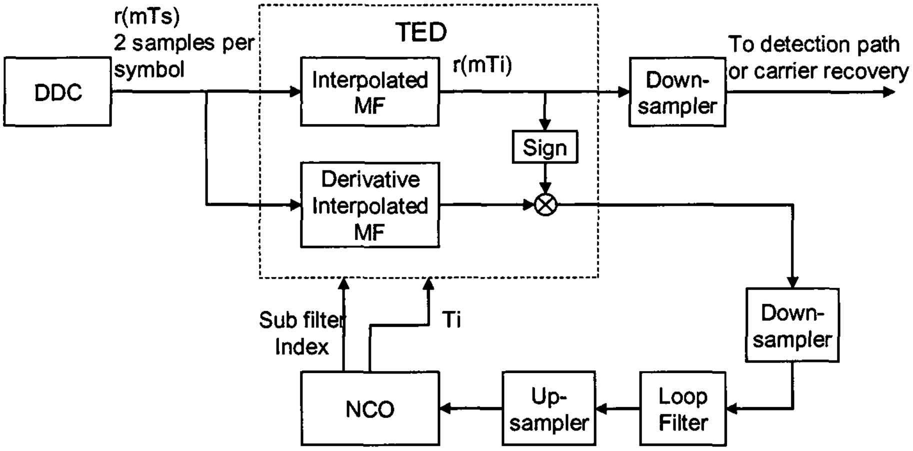


Figure 3-20 Symbol timing recovery using ML timing error detector

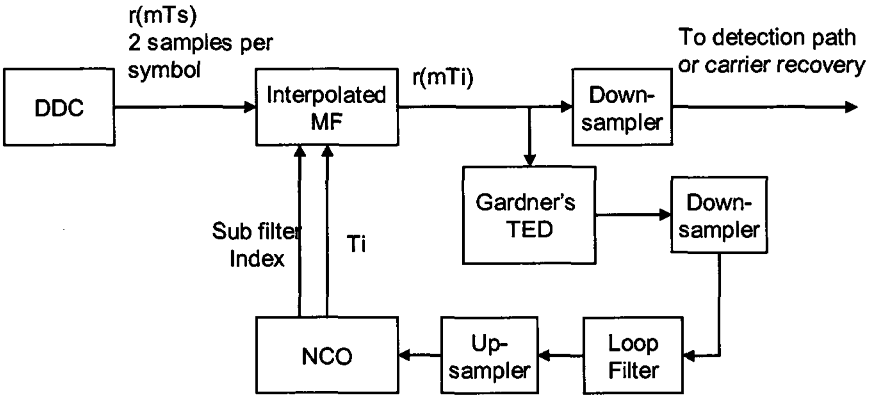


Figure 3-21 Symbol timing recovery using Gardner's timing error detector

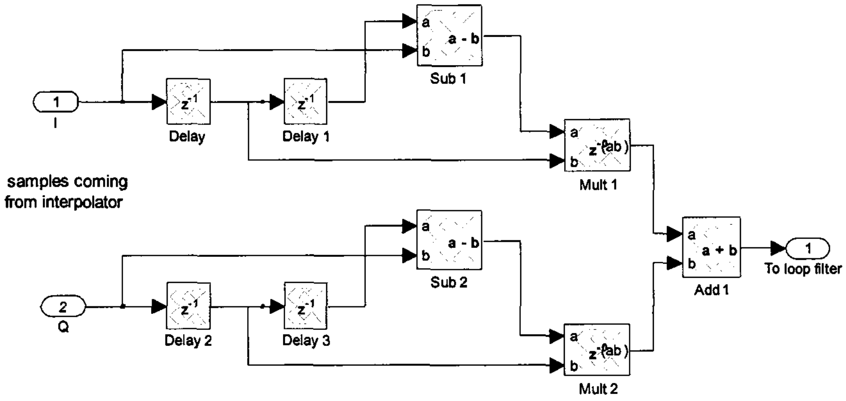


Figure 3-22 Gardner's timing error detector

3.2.3.3 Re-samplers, Loop Filter and NCO

Recall that the sampling rate T_s is incommensurate with the interpolant interval T_i [32], and the interpolants kT_i have been mapped onto receiver time scale kT_s, which is shown in Figure 3-17. As we can see from the time scale relation, a new sample is read into the interpolation filter at sampling time T_s, while a new output is calculated only at time $m_k T_s$. Since the timing error detector provides an estimated timing error to loop filter at symbol rate $T \approx 2T_s$, a down-sampler which is placed after timing error detector selects every other interpolation output to ensure that the error signal is updated once per symbol [27]. In order to accommodate with the NCO processing speed, which equals receiver sampling rate $1/T_s$, the output of loop filter is again upsampled to generate samples with rate of $1/T_s$. The down-samplers and up-samplers are illustrated both in Figure 3-20 and 3-21. The down-sampler in detection path is responsible for providing one sample per symbol to detection stage.

The loop filter design is similar to that described in carrier recovery design. Here we still use an IIR filter which can realize a standard proportional and integral control [46]. Loop characteristics are determined by the loop constants, which are chosen to make the loop bandwidth 0.3% normalized to symbol rate to make a compromise between acquisition time and spurs noise level. The structure of loop filter has been shown in Figure 3-6.

Figure 3-23 shows the structure of NCO which mainly consists of an accumulator, an underflow detector and a quantizer. The control word constant is chosen as 0.5, so that the NCO underflows every $2T_s$, on average. Underflow condition enables the output of interpolated matched filter and its derivative filter. The content of accumulator is multiplied by the normalized value of 2, and further quantized to a 5-bit integer which corresponds to the use of 32 sub-filters in polyphase structure.

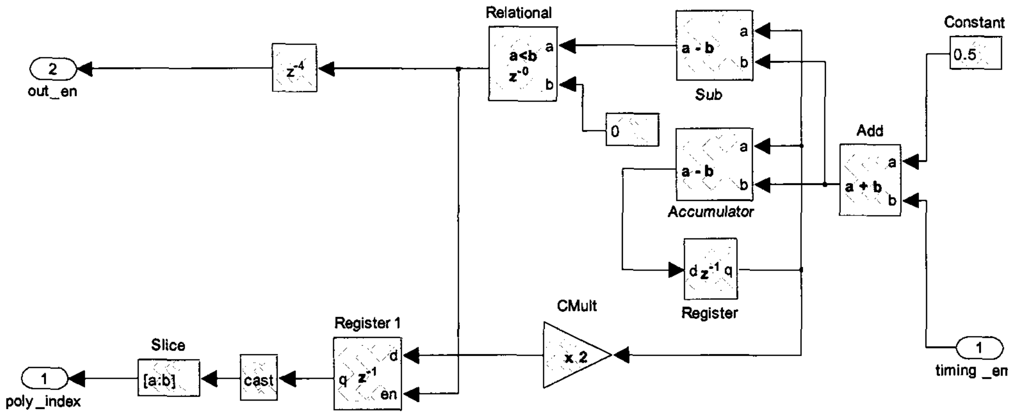


Figure 3-23 NCO in symbol timing recovery loop

3.2.4 Simulation and Analysis

In this section, simulation results of timing recovery using two different TED designs are presented. Comparison of Gardner's detector with different roll-off factors is also made. The loop performance is analyzed by observing the constellation plot and timing error signal, and BER versus E_b/N_0 curves are also given for various simulation scenarios. The simulation is based on the following specifications.

- Modulation scheme: QPSK with differential coding
- Oversampling factor: 16
- Roll-off factor: 0.35 and 0.7
- Channel: AWGN
- Normalized timing offset: 25% of symbol time, i.e., $0.25T$
- Loop bandwidth: 0.3% normalized to symbol rate.

Figure 3-24(a) illustrates the constellation of detected symbols with timing offset of $0.25T$ where timing recovery is not performed. The symbol points are messed up and drift into other quadrants, and four QPSK symbol points can not be clearly seen in the plot. As a result, detection errors will occur with high probability. After symbol timing recovery is performed, last 500 of totally received 700 symbols are presented in Figure 3-24(b). It is clear that the loop is locked and the expected four points are clearly seen on the constellation.

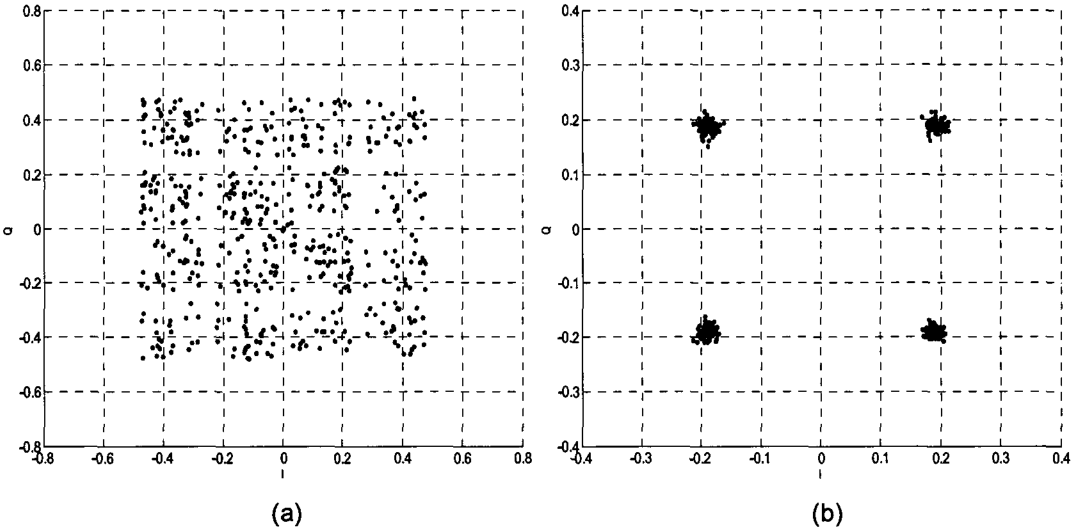


Figure 3-24 QPSK constellation before (a) and after (b) timing recovery

When using ML detector, the timing error signal and sub-filter index signal in the absence of channel distortion is shown in Figure 3-25 and 3-26, respectively. After about 200 symbols, the error signal converges to 0, and fluctuates in a small range around zero due to the randomness of transmitting data as well as spur noise of loop itself. If symmetric data is transmitted, a smooth line will be expected on the figure. As shown in Figure 3-26, after about 200 symbols, the NCO content which represents the sub-filter index remains nearly constant at 16 because of the 25% of symbol timing offset. The acquisition time of this ML based STR loop is 200 symbols. After that, the loop is successfully locked.

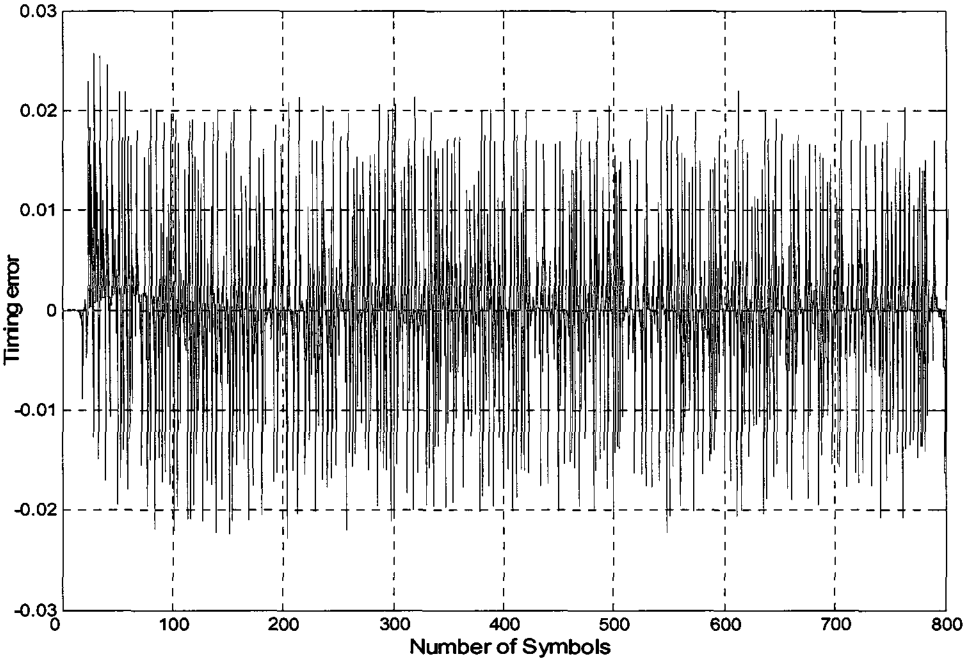


Figure 3-25 Timing error of STR loop using ML detector

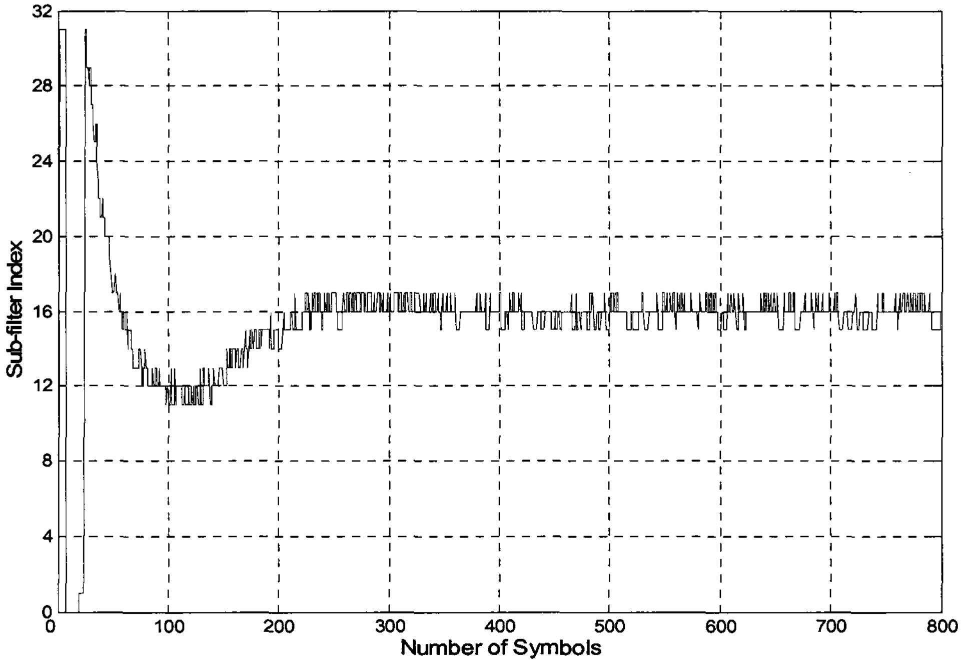


Figure 3-26 Sub-filter index of STR loop using ML detector

When Gardner's detector is used, the sub-filter index signal with roll-off factor β of 0.35 and 0.7 are shown in Figure 3-27 and 3-28, respectively. With $\beta = 0.35$, the acquisition time of STR loop is about 500 symbols. β of 0.35 is a common choice in practical design considering of bandwidth efficiency and simplicity of receiver circuit. When β is increased to 0.7, the loop only takes 200 symbols to stay in lock. As a conclusion, the acquisition time of timing loop becomes shorter as β increases. Moreover, Gardner's detector is best suited to signal with excess bandwidth in range of 40% -100% which is representative of satellite communications [31].

Three BER tests with different simulation scenarios are performed with results shown in Figure 3-29. Theoretical curve results from BER expression of QPSK signal with differential coding in AWGN channel [20]. The next curve is for the floating-point simulation with oversampling factor of 16, and both timing and carrier phase are perfectly matched between transmitter and receiver. As seen in the figure, its performance agrees with theoretical curve. Finally, assume that transmitting signal experiences a timing offset of $0.25T$, and timing recovery using ML detector is performed in the receiver. The BER curve shows that the ML based timing recovery design has less than 0.2 dB performance degradation than the perfect timing situation.

Figure 3-30 illustrates the BER comparison of two timing error detector, which are ML detector with roll-off factor $\beta = 0.35$ and Gardner's detector with $\beta = 0.35$ and $\beta = 0.7$. Both of them experience a timing offset of $0.25T$ and oversampling factor of 16. As shown in the figure, with the same β of 0.35, ML detector gives better performance than Gardner's detector. However, as β increases, the BER performance of Gardner's detector improves and approaches to ML algorithm. We can therefore conclude that the performance of Gardner's detector is dependent on roll-off factor β, and the BER becomes worse as β decreases. This result also agrees with the previous discussion.

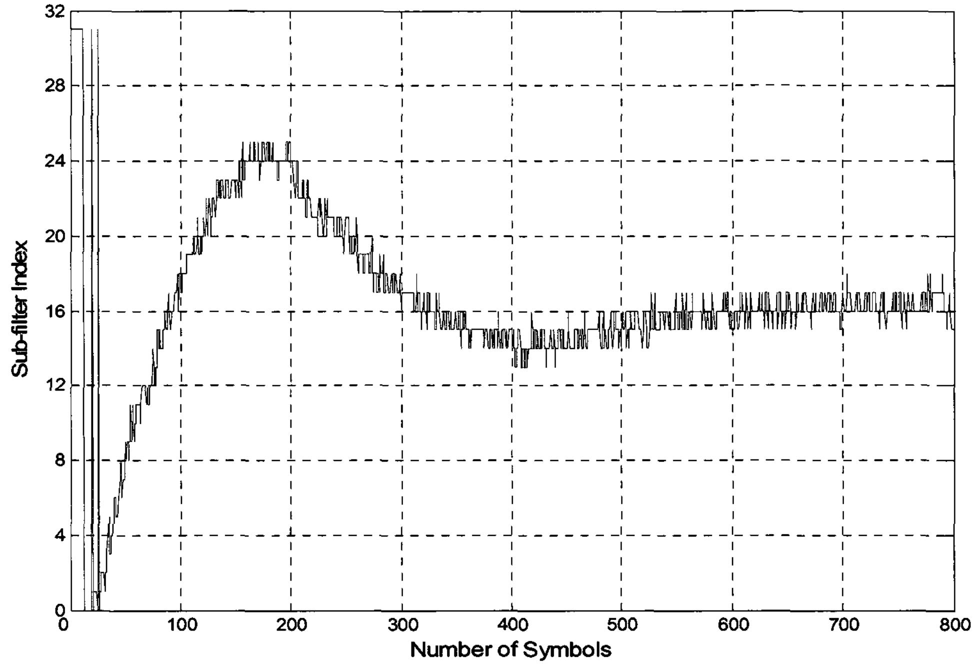


Figure 3-27 Sub-filter index of STR loop using Gardner's detector with $\beta = 0.35$

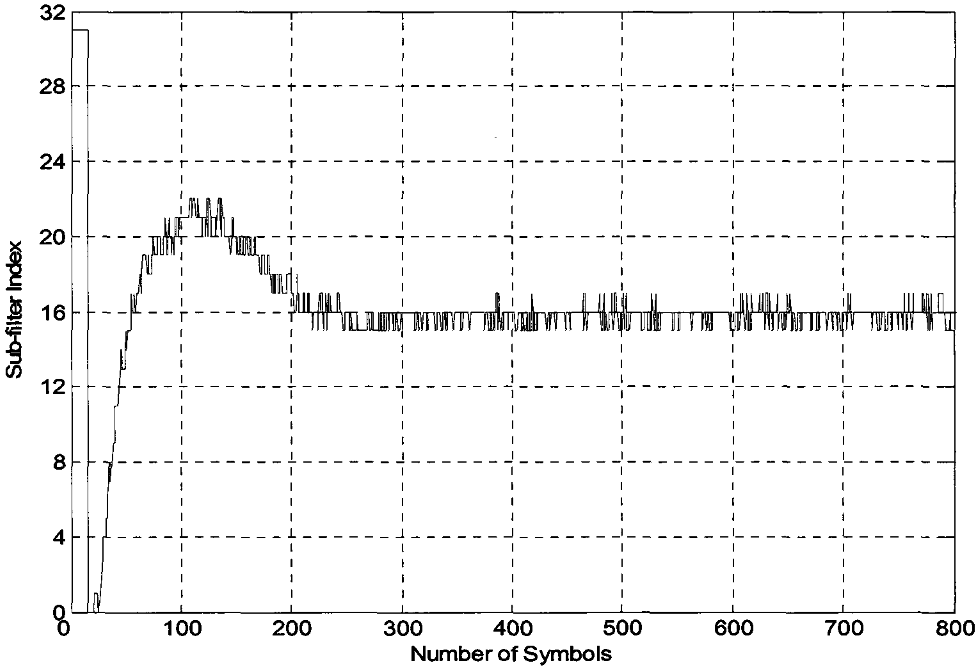


Figure 3-28 Sub-filter index of STR loop using Gardner's detector with $\beta = 0.7$

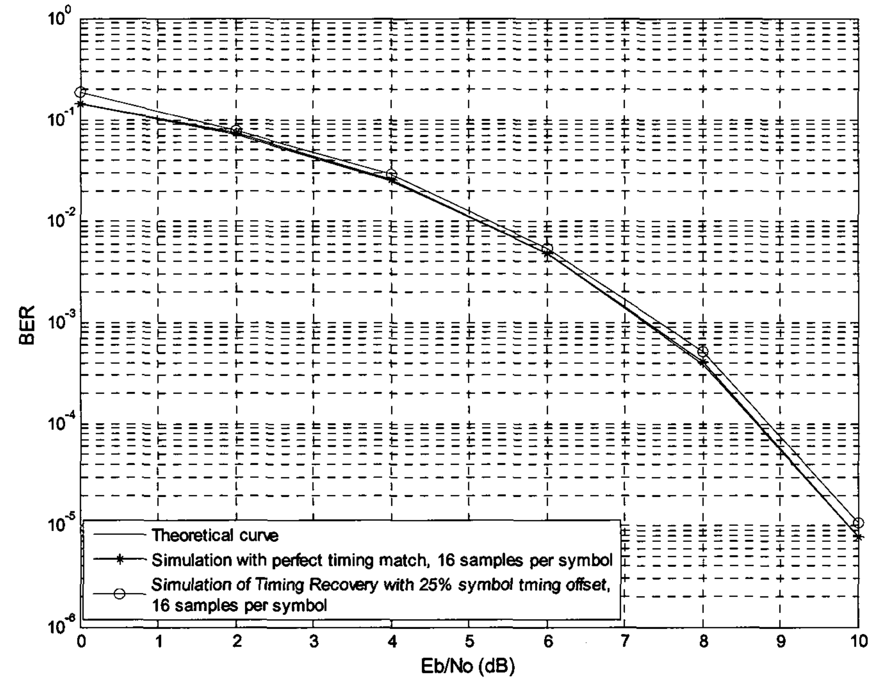


Figure 3-29 BER performance of STR using ML detector

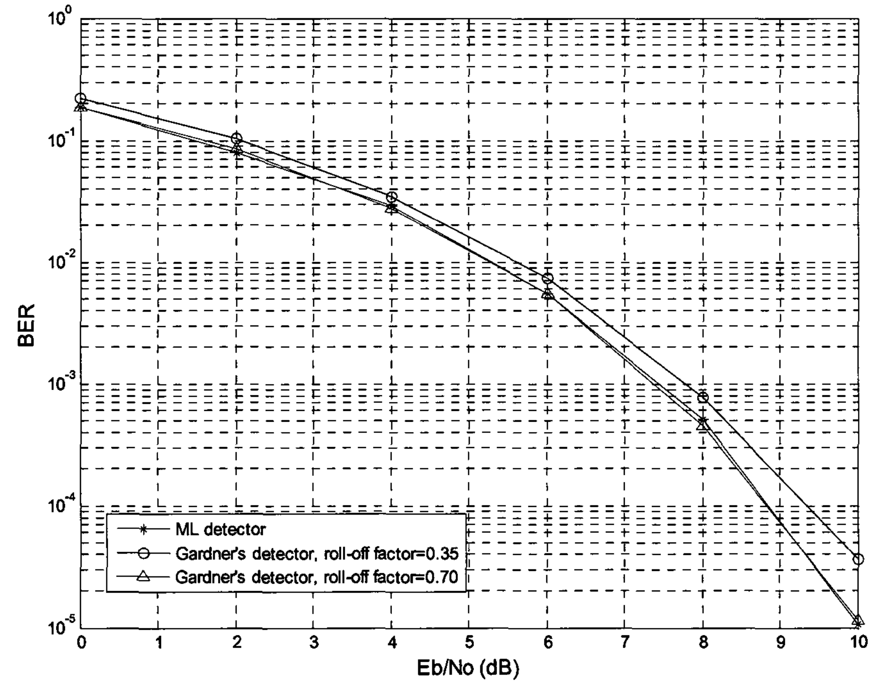


Figure 3-30 BER comparison between ML detector and Gardner's detector

Chapter 4

Design and Implementation of a MIMO System

It has been shown that, by using multiple antennas at transmitter and/or receiver side, multiple-input multiple-output (MIMO) technique provides higher data throughput and improvement in transmission reliability without consuming extra bandwidth or transmission power [38]. MIMO has emerged as one of the most promising technologies for the wireless communication systems.

As we know, the transmitting signal suffers from amplitude and phase distortion due to multipath fading in a wireless environment. Diversity technique is used in wireless systems to combat fading. Diversity in SISO systems can be realized in time or frequency domain, such as channel coding and OFDM (Orthogonal Frequency Division Multiplexing) modulation. On the other hand, MIMO systems utilize antenna (spatial) diversity to combat fading, resulting in a significant increase in channel capacity, hence improvement in SNR and BER performance in the receiver compared with the traditional SISO systems.

To achieve transmit diversity and capacity gains over MIMO fading channel, space time coding (STC) technique is utilized in the transmitter. It can exploit multipath effect and reduce transmission errors by means of coding in both spatial and temporal domain. Specifically, it introduces redundancy in space through multiple antennas, and redundancy in time through channel coding. Therefore, this coding technique enables us to exploit diversity in the spatial dimension, as well as obtain a coding gain. Several STC schemes have been presented in the literature, including space-time block codes (STBC), space-time trellis codes (STTC), layered space-time codes (LSTC) and some concatenated, unitary and differential space-time codes [38].

Alamouti scheme is the first but efficient example of STBC. It can achieve full diversity gain with maximum-likelihood decoding (MLD) in a 2-transmitter 1-receiver system [37]. In this chapter, we will explain the design and implementation of a 2x1 MIMO system using Alamouti scheme.

4.1 Overview of a MIMO System Design

Figure 4-1 shows the general structure of the 2x1 MIMO system based on our design. In the transmitter side, the binary source is first differentially encoded and modulated using QPSK scheme. The generated QPSK symbols are fed into Alamouti encoder with output split into two branches. The coded symbols go through a PSF, and are oversampled by a factor of 2. The PSF is a 32-tap SQRC filter with roll-off factor of 0.35. The following interpolation filter and DUC upsample the signal by a factor of 8, and translate it from baseband to an IF. The IF signal is then converted to analog domain via a DAC for transmission through a multipath fading channel with AWGN. The transmitter structure is almost the same as SISO system, except that two processing branches are needed for the purpose of using two transmitters. In addition, we assume that there is no frequency or phase offset between the two transmitter chains. Therefore, the local oscillators of two transmitters share the same clock and generate the same carriers.

In the receiver side, an ADC first converts the received signal back to digital domain. The IF signal is then down-converted to baseband and down-sampled by a decimation filter with a factor of 8. Symbol timing recovery and carrier recovery are performed afterwards in order to track and compensate the timing and carrier offset, respectively. The rest of the stages are ML decoding, QPSK demodulation, and differential decoding. The channel state information (CSI) is assumed to be known in the receiver, hence simplify the design. The receiver structure is also similar to

SISO system, except for carrier recovery loop, where a different algorithm for phase error detection is used.

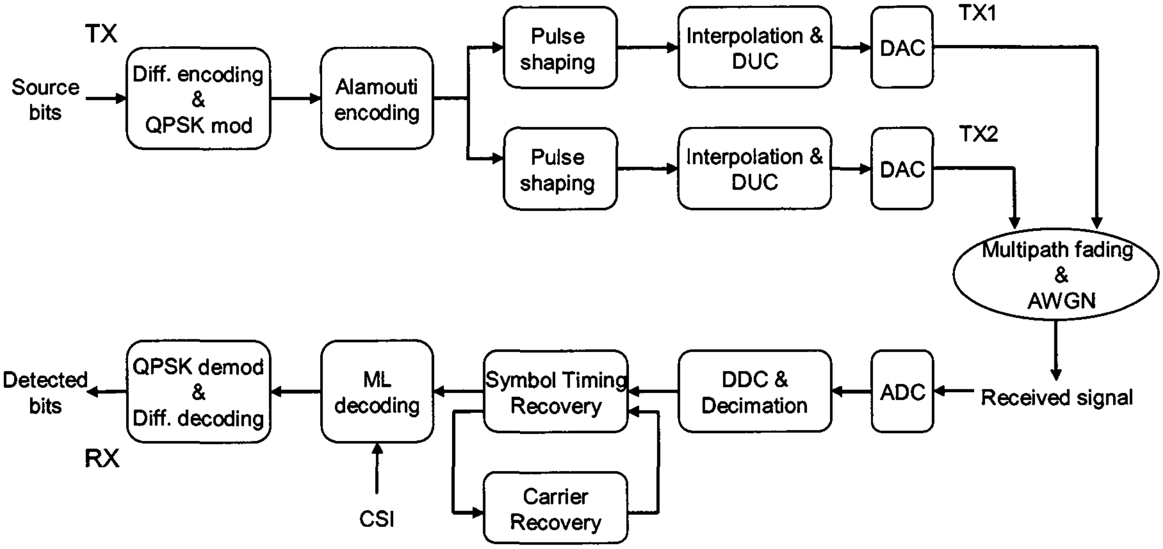


Figure 4-1 Block diagram of proposed MIMO system design

As most of the components have been explained in SISO system design, here we focus on the design of Alamouti encoding, ML decoding, and carrier recovery for the 2x1 MIMO system. The design and implementation issues of these components will be introduced and discussed in detail for the rest of this chapter.

4.2 Alamouti Encoding and ML Decoding

4.2.1 Introduction of Multipath Fading Channel

Before introducing Alamouti scheme, we first review the fading channel model in a wireless environment. When signal is transmitted through a multipath fading channel, its amplitude and phase are distorted due to reflection, refraction and scattering of surrounding objects [21]. Furthermore, the transmitted signal arrives at receiver through different propagation paths, each of which has a relative time delay. As a result, the received signal is composed of a number of scattered waves from multiple propagation paths. Since the signal is spread in time due to the multipath effect, the channel is considered to be time dispersive. A normal measurement for characterizing the propagation delay spread of a multipath channel is the rms delay spread σ_τ, and inverse of delay spread is defined as coherent bandwidth B_C. On the other hand, the Doppler shift due to the motion between transmitter and receiver introduces frequency drift to transmitting signal. In this circumstance, the channel is considered to be frequency dispersive.

Multipath fading channel can be modeled as a complex variable, with Gaussian distribution for both real and imaginary part. In non-line-of-sight (NLOS) environment, where no dominating paths are presented, the fading channel amplitude is Rayleigh distributed with probability density function (pdf) given by [21]

$$P(\alpha)=\begin{cases}\dfrac{\alpha}{\sigma^2}\exp\left(-\dfrac{\alpha^2}{2\sigma^2}\right), & \alpha\geq 0\\ 0, & \alpha<0\end{cases} \tag{4.1}$$

and channel phase is uniformly distributed between $(0,2\pi)$.

In addition, if the channel response keeps constant over a bandwidth that is greater than the signal bandwidth, the transmitting signal undergoes flat fading. In a flat fading channel, the spectral characteristics of the transmitted signal are preserved during the propagation through the channel. For flat fading, the following condition has to be satisfied [21]

$$B_S < B_C \quad \text{and} \quad T > \sigma_\tau \tag{4.2}$$

where B_S is signal bandwidth, and T is symbol time. Recall that B_C and σ_τ are coherence time and rms delay spread, respectively. On the contrary, if above condition is not met, the signal undergoes frequency-selective fading. Since Alamouti scheme and ML decoding are based on a flat fading assumption, we model the channel as flat Rayleigh fading channel.

4.2.2 Introduction of Alamouti Scheme

Alamouti is a simple but efficient STBC scheme. It allows the transmission simultaneously from two antennas with the same data rate as in a SISO system, but increases the transmit diversity from one to two in a flat fading channel. Figure 4-2 illustrates a simplified 2x1 MIMO system using Alamouti scheme. As shown in the figure, two consecutive symbols (x_1, x_2) are first fed into encoder and sequence $(x_1, -x_2^*)$ and (x_2, x_1^*) are then transmitted from antenna 1 and antenna 2 respectively, during two consecutive symbol times, where x^* denotes complex conjugate of symbol x and T is symbol period. After that, the symbols are encoded both in space and time domain, and the transmitting sequences from two antennas are orthogonal to each other.

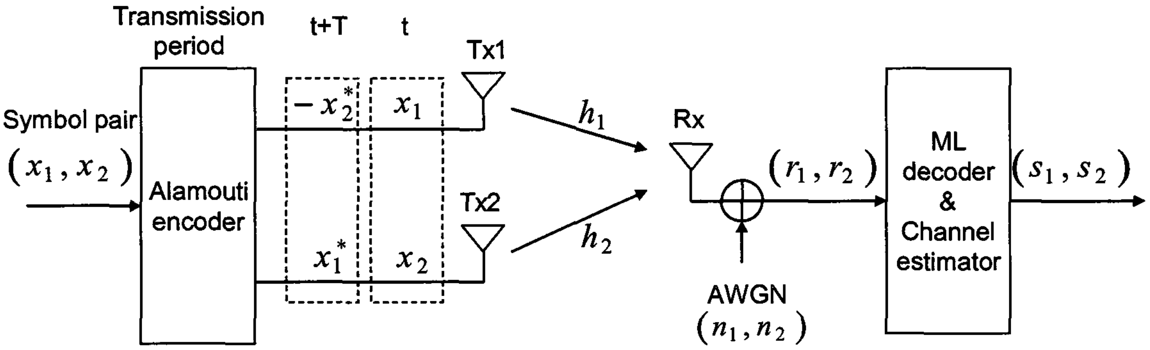


Figure 4-2 Alamouti encoding and ML decoding in a 2x1 MIMO system

Assume that the signal experiences a flat Rayleigh fading channel, and the channel response keeps constant during 2 consecutive symbol times. So the received signal during these 2 symbol periods is given by

$$\begin{aligned} r_1 &= h_1 x_1 + h_2 x_2 + n_1 \\ r_2 &= -h_1 x_2^* + h_2 x_1^* + n_2 \end{aligned} \tag{4.3}$$

where h_1 and h_2 are the fading channel response from two transmitter antennas, and n_1, n_2 are independent complex additive noise with zero mean and power spectral density of $N_0/2$ per dimension. If the channel state information (CSI) is perfectly known at the receiver, maximum likelihood decoding amounts to minimizing the following decision metric [38]

$$\left|r_1 - h_1 x_1 - h_2 x_2\right|^2 + \left|r_2 + h_1 x_2^* - h_2 x_1^*\right|^2 \tag{4.4}$$

over all possible values of x_1 and x_2. The above function can be further decomposed into 2 parts

$$\begin{aligned} &\left|x_1\right|^2\left(\left|h_1\right|^2 + \left|h_2\right|^2\right) - \left(r_1 h_1^* x_1^* + r_1^* h_1 x_1 + r_2 h_2^* x_1 + r_2^* h_2 x_1^*\right) \\ &\left|x_2\right|^2\left(\left|h_1\right|^2 + \left|h_2\right|^2\right) - \left(r_1 h_2^* x_2^* + r_1^* h_2 x_2 - r_2 h_1^* x_2 - r_2^* h_1 x_2^*\right) \end{aligned} \tag{4.5}$$

In expression (4.5), the upper part is only a function of x_1, while the lower one is only a function of x_2. As a conclusion, the 2x1 MIMO system is therefore decoupled into two independent SISO channels, with channel gain of $\left(\left|h_1\right|^2 + \left|h_2\right|^2\right)$. Furthermore, For M-PSK signals, all the constellation points have equal energies, and the first item of Eq. (4.5) $\left|x\right|^2\left(\left|h_1\right|^2 + \left|h_2\right|^2\right)$ can be ignored. Then the decoding process can be simplified as minimizing $\left|x_1 - r_1 h_1^* - r_2^* h_2\right|^2$ to decode x_1, and minimizing $\left|x_2 - r_1 h_2^* + r_2^* h_1\right|^2$ to decode x_2. Therefore, the ML decoding for MPSK signal can be expressed as searching for the closest symbol point to s_1 and s_2 on the constellation, where

$$\begin{aligned} s_1 &= h_1^* r_1 + h_2 r_2^* \\ s_2 &= h_2^* r_1 - h_1 r_2^* \end{aligned} \tag{4.6}$$

For QPSK modulation, decoding s_1 and s_2 can be further simplified by detecting their sign and allocating them to according quadrant. In the section of simulation and analysis, we will examine the performance of Alamouti scheme in a flat fading channel.

4.2.3 Design and Implementation

4.2.3.1 Alamouti Encoder

Let's denote the 2 consecutive QPSK symbol as $x_1 = a + bj$, $x_2 = c + dj$, respectively. Based on the mapping scheme in figure 4-2, the encoded symbols transmitted via antenna 1 in two symbol times is $\left[a+bj,-c+dj\right]$, and the symbols transmitted via antenna 2 is $\left[c+dj,a-bj\right]$. A time division de-multiplexer (TDD) is used to split the modulated signal into two streams, one for odd symbols, and another for even symbols. Negation of real or imaginary part is performed when necessary. The processed symbols are converted back to one stream by a time division multiplexer (TDM) to keep the data rate. Since two transmitters exist, 2 TDDs, 4 TDMs and 2 negation blocks are required to form the Alamouti encoder. The structure is shown in Figure 4-3.

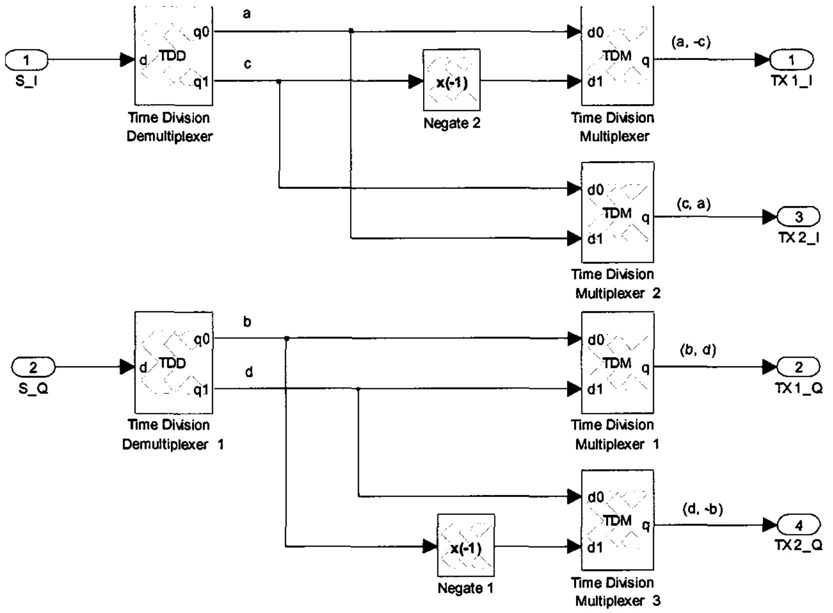

1
S_I
d
TDD
q0
q1
Time Division Demultiplexer
a
c
x(-1)
Negate 2
d0
d1
TDM
q
Time Division Multiplexer
(a, -c)
1
TX1_I
Time Division Multiplexer 2
(c, a)
3
TX2_I
2
S_Q
Time Division Demultiplexer 1
b
d
Time Division Multiplexer 1
(b, d)
2
TX1_Q
Negate 1
Time Division Multiplexer 3
(d, -b)
4
TX2_Q

Figure 4-3 Alamouti encoder

4.2.3.2 ML Decoder

For QPSK signal, the decoding is done by calculating Eq. (4.6), and assigning them to the closet constellation points in the four quadrants. Eq. (4.6) can be expanded using complex number representation. Then we can build the ML decoder based on the resulted expansion. The ML decoder consists of 2 TDDs, 2 TDMs, negation blocks and multiplier-and-adder trees. The channel coefficients can be obtained from a channel estimator. To simplify the design, we assume that these coefficients are perfectly known in the receiver.

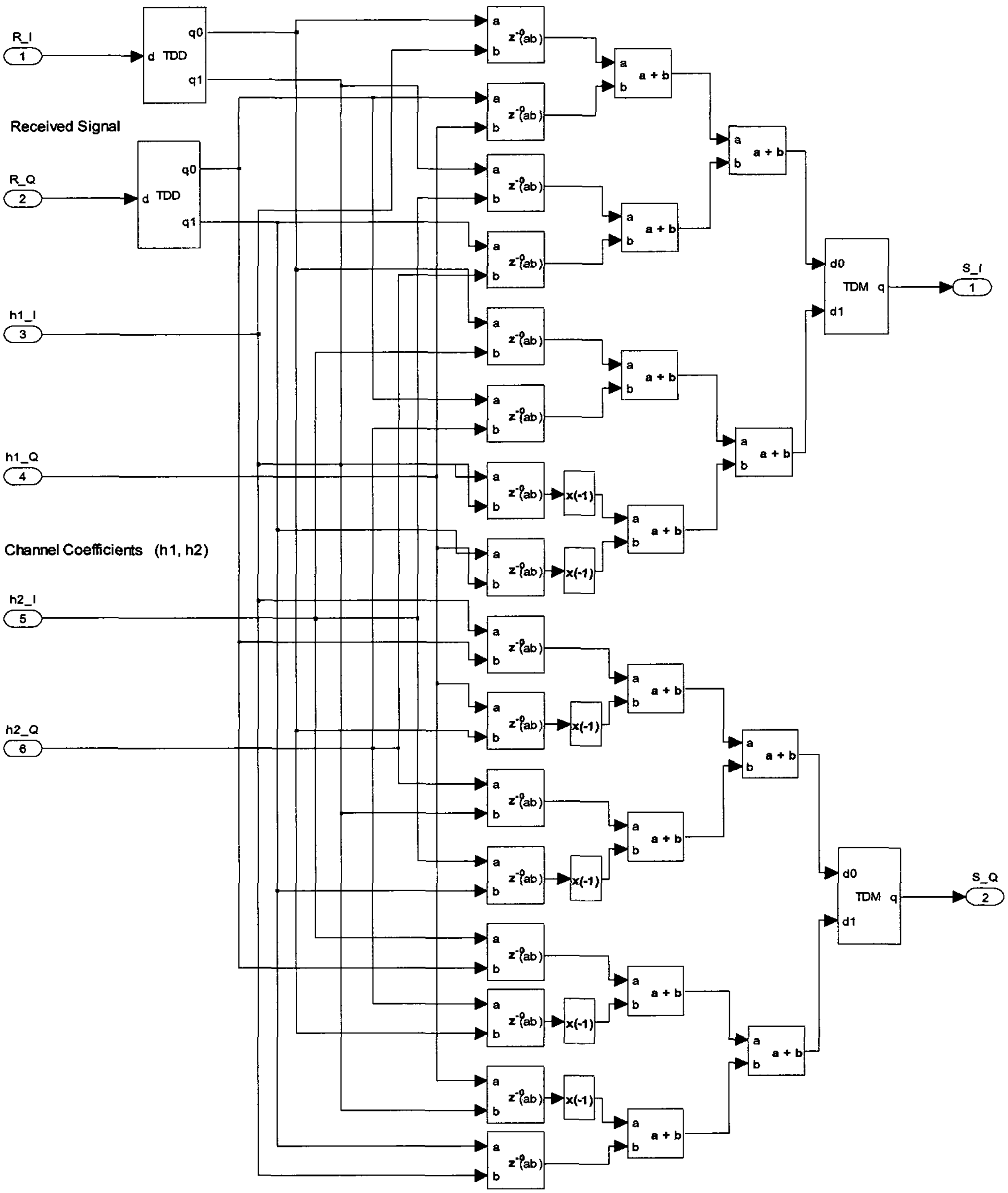


Figure 4-4 ML decoder

4.3 Carrier Recovery for MIMO System

4.3.1 Background

As discussed before, when Alamouti scheme is used, the encoded symbols are either unchanged or in the form of conjugate or negative conjugate of uncoded symbols. The signal constellation of each transmitter has the same four points as conventional QPSK modulation. As a result, the received signal of this 2x1 MIMO system can be viewed as a combination of two independent QPSK signals. Figure 4-5 shows the received constellation using Alamouti scheme in the absence of channel distortion. As we can see, 9 possible signal points are shown in the figure due to the combination mentioned above. Since constellation becomes denser, the phase error detector used in SISO system is no longer suitable for MIMO case. However, the PLL technique is still applied to the design, and coordinate rotation digital computer (CORDIC) algorithm is used to produce detected phase error.

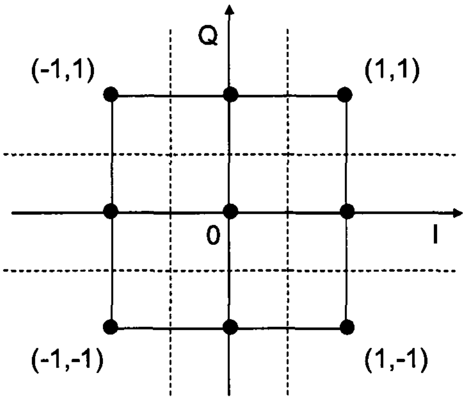


Figure 4-5 Alamouti encoded QPSK signal constellation

CORDIC is an efficient algorithm for calculating hyperbolic and trigonometric functions. In carrier recovery design, it is used for calculating phase difference, which is an arctangent value. The advantage of CORDIC algorithm is that it iteratively computes the rotation of a two-dimensional vector by only using add and shift operations. Since no multiply is performed in this algorithm, CORDIC has been widely used in hardware implementation for digital signal processing and application in SDR [43]. This algorithm is derived from the general rotation equation, that is after rotating a vector (x, y) with an angle ϕ, the resulted vector (x', y') can be expressed as

$$\begin{aligned} x' &= x\cos\phi - y\sin\phi \\ y' &= y\cos\phi + x\sin\phi \end{aligned} \tag{4.7}$$

Eq. (4.7) is equivalent to

$$\begin{aligned} x' &= \cos\phi(x - y\tan\phi) \\ y' &= \cos\phi(y + x\tan\phi) \end{aligned} \tag{4.8}$$

If condition $\tan\phi = \pm 2^{-i}$ (i is a non-negative integer) is satisfied, Eq. (4.8) can be modified as

$$\begin{aligned} x_{i+1} &= K_i\left(x_i - a_i y_i 2^{-i}\right) \\ y_{i+1} &= K_i\left(y_i + a_i x_i 2^{-i}\right) \end{aligned} \tag{4.9}$$

where $a_i = \pm 1, K_i = \cos\left(\arctan\left(2^{-i}\right)\right) = 1/\sqrt{1+2^{-2i}}$, and K_i approaches 0.6073 when i goes to infinity [45]. Based on above discussion, any rotation in range of $[-\pi/2, \pi/2]$ can be realized by iteratively rotating an angle $\arctan\left(2^{-i}\right)$ using only add/sub and shift operation. The CORDIC calculation can be made either in rotation mode or vectoring mode [43][44]. The rotation mode is used for rotating the vector by a specified angle, which is done by initializing an angle accumulator to the specified value, and reducing it to zero through the iteration process. In vectoring mode, the angle accumulator is initialized to 0, and input vector is iteratively rotated toward x coordinate. The angle needed for the vector converging to x coordinate is recorded in the accumulator. Vectoring mode is applied to our carrier recovery design. The mathematical model of iteration process for vectoring mode can be expressed as [45]

- Initialize that $x_0 = x, y_0 = y, \mathrm{z}_0 = 0, i = 0$
- $$\begin{aligned} a_i &= +1, && \text{if } y_i < 0 \\ a_i &= -1, && \text{otherwise} \end{aligned}$$

- $$\begin{aligned} x_{i+1} &= x_i - a_i y_i 2^{-i} \\ y_{i+1} &= y_i + a_i x_i 2^{-i} \\ z_{i+1} &= z_i - a_i \arctan\left(2^{-i}\right) \end{aligned} \tag{4.10}$$

where z_i is the accumulated angle. In our application, it represents the phase error due to the carrier offset. This algorithm rotates the vector clockwise or counter-clockwise by observing the sign of y_i, which converges to 0 through iteration. The performance of CORDIC is dependent on the number of iteration stages, and 3 to 7 stages are normally chosen in practical design. In our design, 5 iterations are used.

4.3.2 Design and Implementation

First of all, the input signal of carrier recovery loop is mapped to its nearest constellation point to calculate both sine and cosine of phase difference. Figure 4-6 shows the received signal $x + yj$ with a phase offset θ compared with its nearest constellation point $m + nj$.

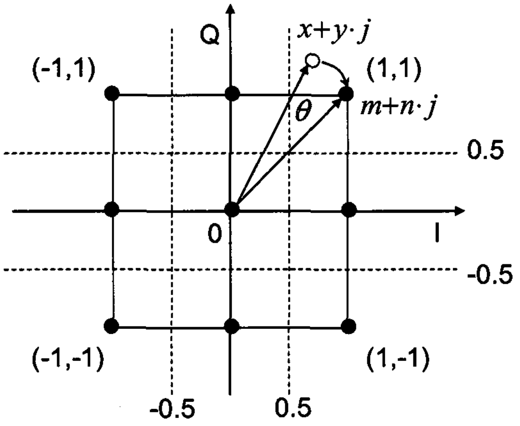


Figure 4-6 Received signal constellation with phase offset

Recall that the sine of phase offset has been derived in chapter 3, which is given as

$$\sin(\theta) = \frac{my - nx}{\sqrt{\left(x^2 + y^2\right)\left(m^2 + n^2\right)}}. \tag{4.11}$$

Cosine value is also computed for use of CORDIC algorithm, and

$$\cos(\theta) = \frac{mx + ny}{\sqrt{\left(x^2 + y^2\right)\left(m^2 + n^2\right)}}. \tag{4.12}$$

The mapping process is based on decision region shown by dashed line in Figure 4-6. Figure 4-7 shows the mapper structure which consists of a multiplexer, a sign detector, a comparator, and a LUT. The multiplexer first maps the signal to its absolute value for comparing with decision boundary value 0.5. At the same time, the sign detector examines the sign of signal. Both components output a Boolean signal used as address of look-up table. Two identical circuits are needed for both I and Q branches. The mapper output and un-mapped signal are fed into a complex multiplier to perform sine and cosine calculation shown in Eq. (4.11) and (4.12). The resulted sine and cosine values are y_0 and x_0, respectively, which are shown in Eq. (4.10).

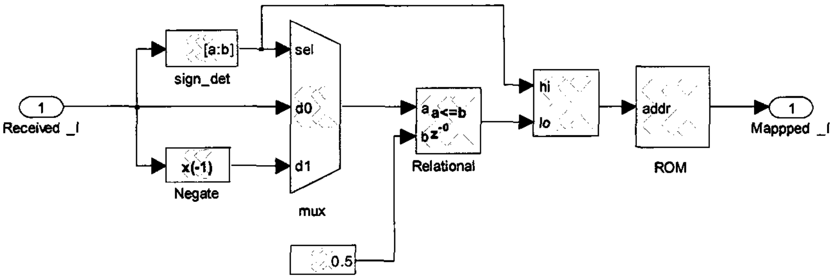


Figure 4-7 Signal mapper in carrier recovery

CORDIC algorithm itself can converge angles between $[-\pi/2, \pi/2]$. In order to support a full range of $[-\pi, \pi]$, the algorithm is expanded to three steps [30]. The vector $(\cos(\theta), \sin(\theta))$ resulting from the previous calculation is first mapped to a specified region $[-\pi/2, \pi/2]$, i.e., quadrant 1 and 4. CORDIC is then performed in the second step through several iterative stages. At the i^{th} stage, the signal vector is rotated for an angle of $\arctan\left(2^{-i}\right)$. As the iteration processes, the rotation angle becomes smaller, and y value of the vector approaches to 0. When the vector converges to the x axis, the overall rotation angle for this convergence is recorded. The final step is to de-map the signal vector to the original quadrant where it belongs to. This is for compensating the reflection effect due to step 1.

Figure 4-8 shows the structure of quadrant mapper used in step 1. As seen in the figure, two slicers are required for sign detection of x (cosine) and y (sine) values. The multiplexer and negation block are responsible for mapping the signal from quadrant 2 and 3 to quadrant 1 and 4. Since cosine value is non-negative for $[-\pi/2, \pi/2]$, the mapping is only applied to cosine path.

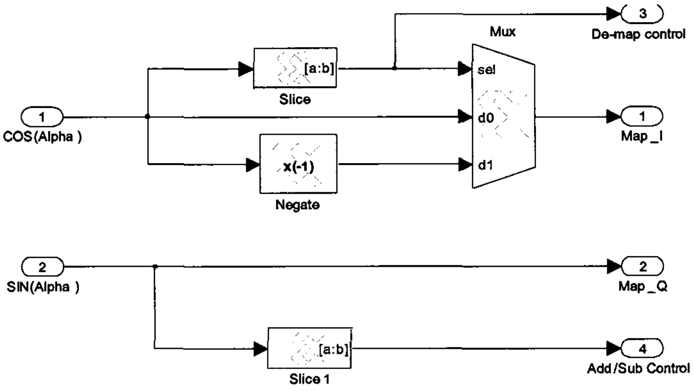


Figure 4-8 Quadrant mapper (step 1) in CORDIC algorithm

Step 2 is the core of CORDIC algorithm where 5 iteration stages are included. Figure 4-9 shows the structure of i^{th} iteration stage [44]. It is mainly composed of two i-bit shifters and three adders/subtractors. Two of the adders/subtractors are used for calculating x and y values of the rotated vector, and another one is to accumulate the rotation angle in radian up to the current stage. The operation of addition or subtraction is dependent on the sign of y_{i-1}, which is the y value of rotated vector from previous iteration stage. Specifically, if $y_{i-1} < 0$, vector is to be rotated counterclockwise, and subtraction is performed in the angle accumulator. If $y_{i-1} > 0$, vector is to be rotated clockwise, and addition is performed in the angle accumulator. Here, vector rotation is realized by an i-bit shifter rather than a traditional multiplier. Furthermore, the rotation angle $\arctan\left(2^{-i}\right)$ for each stage is pre-calculated, and stored in a look-up table.

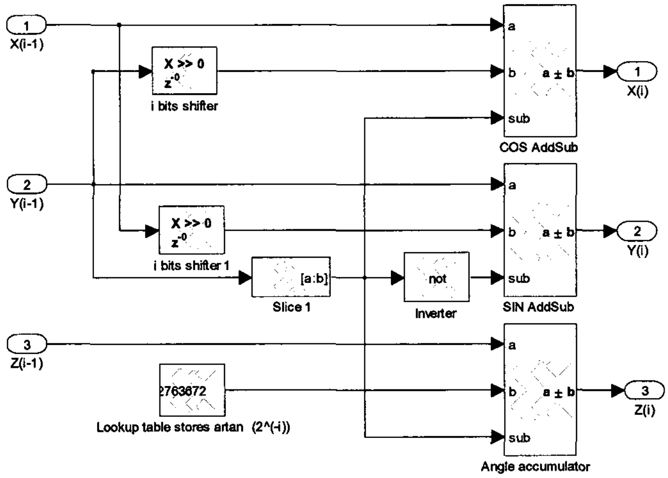


Figure 4-9 i^{th} iteration stage (step 2) in CORDIC algorithm

The quadrant de-mapper used in the last step consists of two multiplexers, a subtractor and a LUT in which π value is stored. As we can see in Figure 4-10, the de-map control is actually the sign of x value of original vector, which is positive if the vector resides in quadrant 1 and 4, and negative in quadrant 2 and 3. If original vector resides in quadrant 2 and 3, quadrant correction should be performed. Specifically, the accumulated angle z_i is subtracted by π to return to quadrant 2, and subtracted by $-\pi$ to return to quadrant 3.

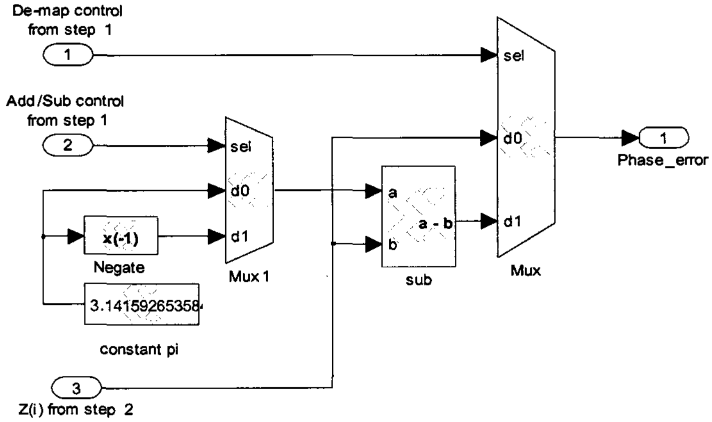


Figure 4-10 Quadrant de-mapper (step 3) CORDIC algorithme

Furthermore, as seen on the constellation of received signal in Figure 4-6, there is one point that overlaps with the origin of coordinates. As a result, the sine and cosine values of this point run to infinity. In this case, the received signal located in the decision region of this point should not be processed to phase error detection in order to prevent the self-error of the algorithm. In addition, phase ambiguity problem can be solved by differential encoding.

The output of CORDIC block is the detected phase error, which is then fed to loop filter and NCO to accomplish phase adjustment. The design and implementation issue of loop filter and NCO has been discussed in detail in SISO system design. Since theses components in our MIMO system have same structure and specification, they are not emphasized here.

4.3.3 Simulation and Analysis

The carrier recovery of MIMO system is simulated using both system generator blocks and Simulink blocks. The loop performance is analyzed by observing the constellation plot and phase error signal. The BER performance in AWGN channel is also given. The simulation is based on the following specifications.

- Modulation scheme: QPSK with differential coding
- Oversampling rate: 16
- Roll-off factor: 0.35
- Channel: Flat Rayleigh fading channel, AWGN
- Normalized frequency offset: 0.001 Hz
- Loop bandwidth: 0.1% normalized to symbol rate.

We first consider the case that receiver carrier is perfectly coherent with transmitter carrier, and a simulation of 2x1 MIMO system using Alamouti scheme is performed. The BER performance is shown in Figure 4-11. In a flat Rayleigh fading channel, where the channel response keeps constant for every 2 symbols, the Alamouti scheme with QPSK modulation performs much better than conventional SISO channel when CSI is known at the receiver. It needs about 10 dB SNR less than SISO to obtain a BER of 10^{-3}. However, two systems have the same BER over AWGN channel, where the channel response keeps constant all the time.

Therefore, use of Alamouti scheme for AWGN channel does not improve the system performance. We can further conclude that MIMO technique is especially suitable for the fading environment, which is more realistic for a wireless transmission.

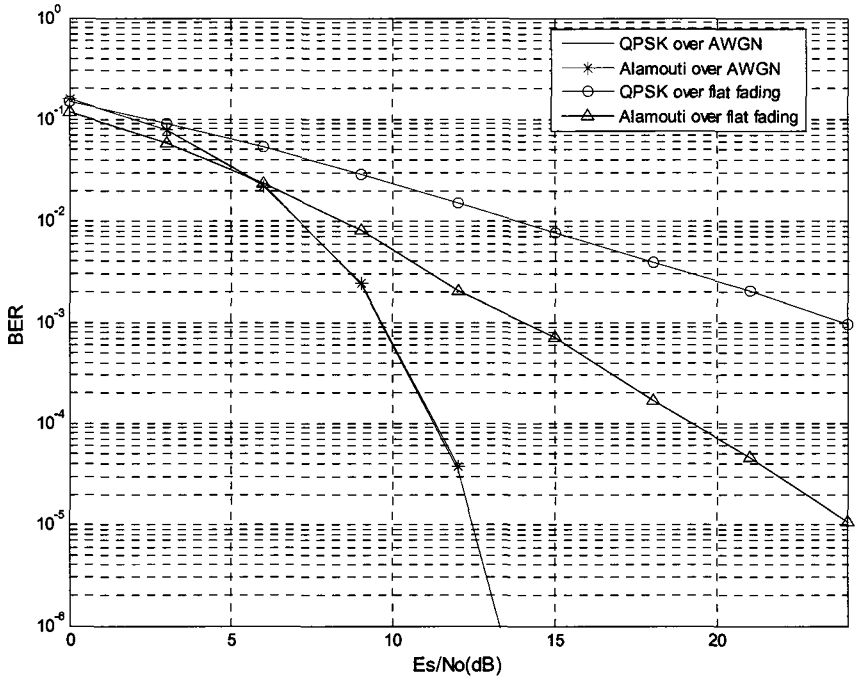
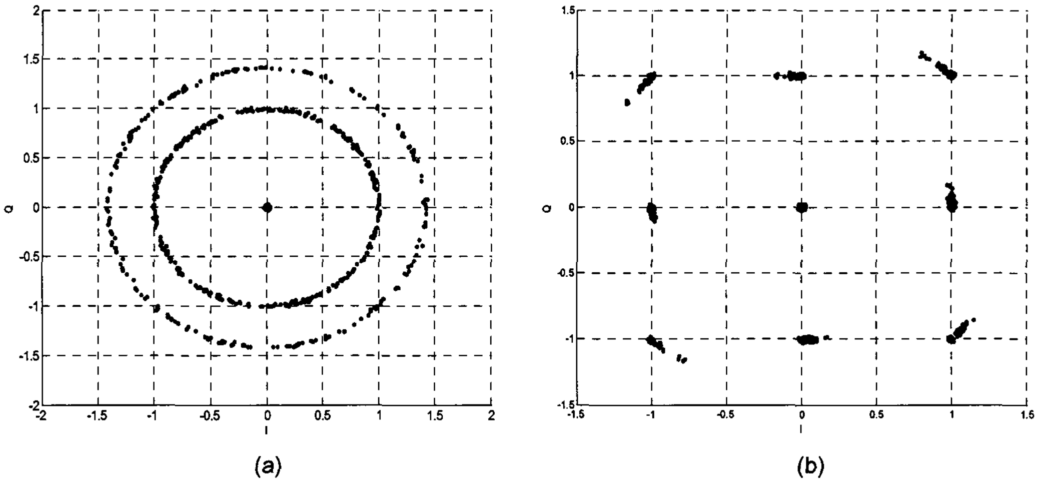


Figure 4-11 BER performance of Alamouti scheme over AWGN and flat fading channel



(a)

(b)

Figure 4-12 Signal constellation before (a) and after (b) carrier recovery

Next, a normalized frequency offset of 0.001 Hz is introduced to the transmitted signal, and CORDIC based carrier recovery is performed in the receiver. Figure 4-12 shows the signal constellation before and after carrier recovery in the absence of channel distortion. As we can see, the frequency offset rotates the constellation. After carrier recovery is performed, received signal is de-rotated and locked to a stable status, and 9 points can be clearly seen on the plot.

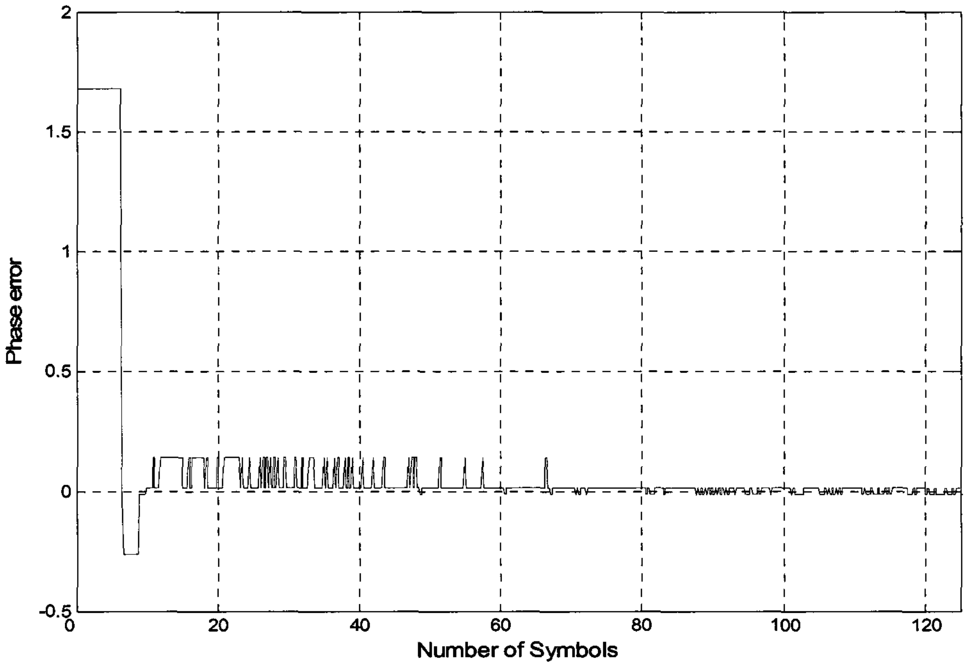


Figure 4-13 Phase error of MIMO carrier recovery loop @ 0.001 Hz frequency offset

Figure 4-13 shows the phase error signal at the output of CORDIC phase error detector. As we can see, the acquisition time for this carrier recovery loop is about 70 symbols. After that, the phase error converges to zero, and NCO can then provide a nearly constant signal to adjust the phase of received signal.

Finally, the BER vs E_s/N_0 curve is given for measuring the BER performance of CORDIC based carrier recovery in an AWGN channel. As shown in Figure 4-14, the performance is severely affected by SNR. When SNR is less than 5 dB, the loop can not stay in lock due to strong noise, and BER is between 0.2 and 0.4. As SNR increases, BER approaches phase matching condition. When SNR is higher than 10 dB, the carrier recovery loop has less than 0.2 dB performance degradation compared with ideal condition.

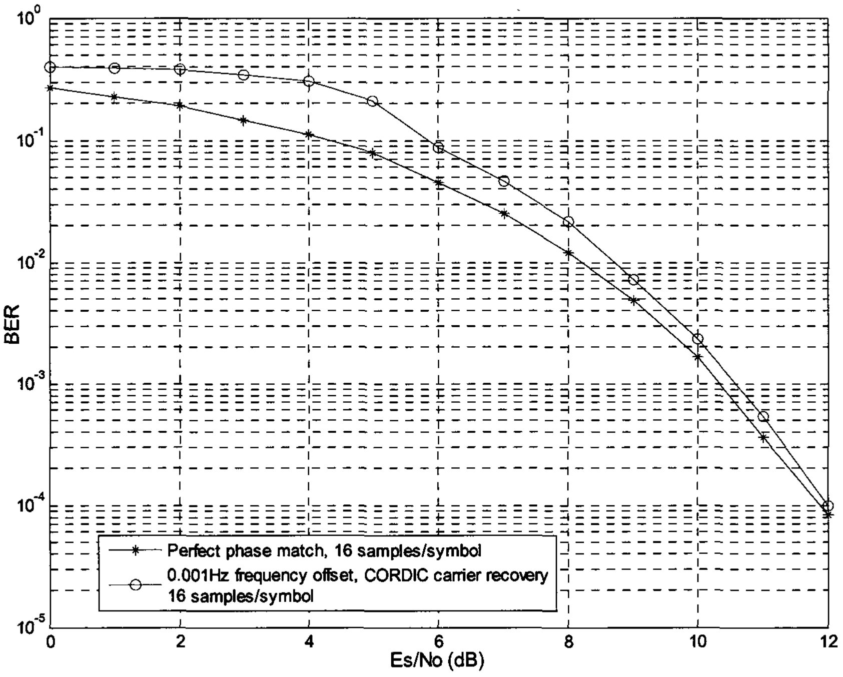


Figure 4-14 BER performance of CORDIC based carrier recovery

Chapter 5

Hardware Description and Test Results

In chapter 2 & 3, we talked about the design and implementation issues of an SDR based SISO system. After modeling using System Generator, and synthesizing and implementing the design using ISE software, a bitstream performing low-level execution of high-level abstraction is generated, and then downloaded to target FPGA. We can thus realize the design on hardware and test its performance. In this chapter, the experimental platform and other test equipment are first introduced. The test results of our SISO system design are then given and analyzed.

5.1 Introduction of Test Equipment

5.1.1 XtremeDSP Board

The testbed on which we implement the design is Nallatech XtremeDSP board. It is a development platform providing FPGA technology, as well as high performance DACs and ADCs. The board mainly contains three Xilinx's FPGAs, which are [49]

- A Virtex-4 user FPGA used as the main part of the design.
- A Virtex-2 clock FPGA for clock management. It controls and routes the clock to different components on the board.
- A Spartan-II interface FPGA which is responsible for interface control between FPGAs, or board and PC through USB/PCI interface.

The XtremeDSP board provides two ADCs with resolution of 14 bits and maximum sampling frequency of 105 MHz, and two 14-bit DACs with sampling frequency of 160 MHz.

The DAC also features [49]:

- A half band symmetric interpolation filter with oversampling factor of 2.

  It can be configured for low pass or high pass response. With low pass response, the interpolation filter works at 2 times the input data rate to DAC, and significantly reduce the magnitude of the first spectrum image of original signal. In our application, the first image of the up-converted signal is moved from 75 MHz to 175 MHz by interpolation filtering. The images centered at multiple of sampling rate because of sampling theory will be further attenuated by the DAC's $\sin(x)/x$ response.
- A PLL clock multiplier that can provides synchronous clock for edge trigger latches, interpolation filter and DAC.

On the other hand, the ADC also features:

- An anti-aliasing filter with cutoff frequency of 58 MHz, which can successfully exact high frequency component for our application.

Furthermore, 4 clock resources are available for the board, which are:

- An on board fixed oscillator of 105 MHz.
- A programmable oscillator working up to 120 MHz.
- A user-provided oscillator socket.
- An external clock input via MCX connector or some user pins.

In addition, the board has some other important features:

- PCI/USB interface used for communication between FPGAs and host PC
- JTAG chains used for hardware test and configuration
- Two banks of ZBT-SRAM used for data storage. Each bank has 16 MB capacity with working frequency of 133 MHz.

Besides the sufficient hardware resources and configuration flexibility, the XtremeDSP platform provides easy and fast transition from algorithm simulation to hardware verification when combined with System Generator. This platform is therefore ideal for signal processing application such as SDR, HDTV, and so on. It is also suitable for our SISO system implementation.

5.1.2 FB100A BER Tester

The BER performance of the design is measured by Aeroflex FB100A BER test system [51]. The BER tester can generate user defined word pattern and pseudo random binary sequence (PRBS) with selectable length up to $2^{23}-1$. The operating bit rate ranges from DC to 100 Mbps. Both internal clock and external clock are available for bit generation. In our application, an external clock of 12.5 MHz is required to generate a PRBS with rate of 12.5 Mbps. During each transmission block, the machine adds a certain number of preambles to the binary sequence. These preambles are monitored in the received sequence. If they are correctly detected, the tester is considered in synchronization. If not, the machine is considered out of sync.

Aside from ability to calculate BER, the tester also functions as an AWGN channel emulator, which can add bandpass white Gaussian noise to carrier wave from 5 MHz to 2.4 GHz. The noise power can be controlled by the desired carrier to noise ratio, and carrier signal power is tracked and measured by the machine. During BER calculation, the tester can generate plot of BER vs SNR automatically. In addition, the machine supports testing with various signal modes, such as continuous, TDMA, and burst mode. Based on properties introduced above, FB100A can satisfy our application requirement, and ease the test process.

The other test equipment includes digital oscilloscope and spectrum analyzer.

5.2 Hardware Setup and Connection

After introducing the hardware properties, we now explain how these equipments are correctly setup and connected to accomplish the implementation and testing. Figure 5-1 illustrates the hardware connection and signal routing path. First of all, we use the programmable oscillator as the clock resource, and set master clock as 100 MHz for XtremeDSP board. This choice also makes it convenient to transfer the design to other clock frequency. In order to generate a binary data with rate of 12.5 Mbps which is synchronous with master clock, the 100 MHz is divided by 8 to obtain a synchronous 12.5 MHz clock. This is done by Digital Clock Manager (DCM) of Virtex-4 chip [48]. DCM provides a wide range of powerful clock management features. It also performs on-board clock de-skewing and clock forwarding in our application. The DCM contains a delay-locked loop (DLL), and clock distribution delays can be completely eliminated by de-skewing the DCM's output clock with respect to the input clock. In addition, DCM provides precise frequency synthesis, which can realize synchronous frequency division and multiplication. It also allows coarse and fine phase shifting compared with input clock [48]. In our design, the generated 12.5 MHz clock is forwarded to a double data rate (DDR) register, and then taken out of the board from a user pin to drive the PRBS generator (BER tester). The propagation delay of master clock signal distributed throughout the whole FPGA is also eliminated due to the on board de-skew feature of DCM.

Figure 5.1 illustrates the connection between FPGA and BER tester, along with the numbered signal routing path. The PRBS length is chosen as $2^{23}-1$ with distribution sufficiently close to the ideal uniform distribution. The generated bits are fed into the XtremeDSP board through a user pin (Figure 5-1-1), and the baseband signal processing and digital up-conversion are performed using the FPGAs. After going through DAC, an analog signal of 25 MHz is generated (Figure 5-1-2). It is then connected with the AWGN channel emulator (BER tester) via 75 ohm port. The bandpass noise with bandwidth of approximately 8.37 MHz centered at 25

MHz is added to carrier with bandwidth of 8.735 MHz. The carrier bandwidth is equal to signal bandwidth calculated in chapter 2, and the noise bandwidth is chosen from the available values provided by the machine so that it is as close to the signal bandwidth as possible. The measured carrier signal power is -6.2 dBm. The output of channel emulator (Figure 5-1-3) goes through the ADC and FPGAs to perform the down-conversion and demodulation. Finally, the detected bits are taken out of the FPGA board via some user pin (Figure 5-1-4), and fed into the tester to accomplish the BER computation.

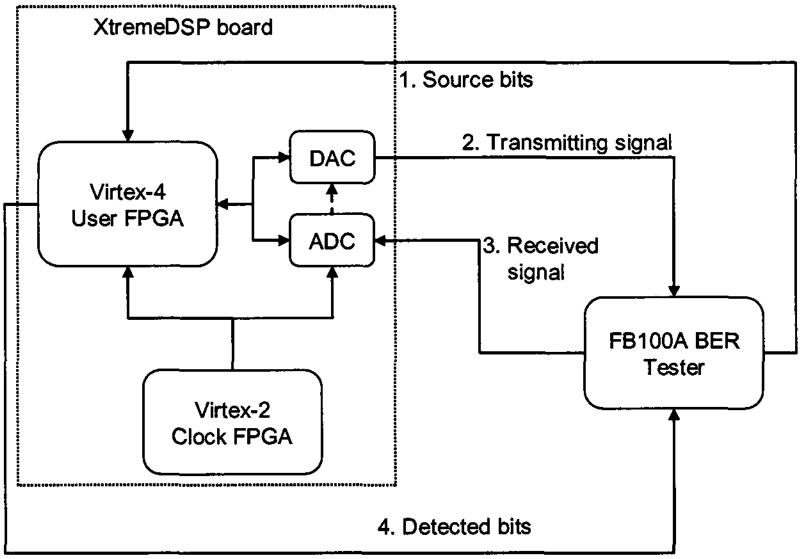


Figure 5-1 Hardware connection and signal routing path

5.3 Hardware Test Results

After setting and connecting all the hardware platform and test equipment correctly, we can measure the system performance using oscilloscope, spectrum analyzer, and BER tester. The following signal observation is based on a SISO system with IF of 25 MHz, and carrier recovery is performed in the receiver side. The BER performance and hardware utilization are given with different test scenarios.

5.3.1 Signal Observation in Time and Frequency Domain

Figure 5-2 shows the time domain observation of transmitted signal at 25 MHz on the oscilloscope. As seen in the figure, the signal phase changes at the symbol transition due to the use of QPSK modulation scheme. If the time axis is enlarged, a sinusoid signal which represents the carrier will be shown on the oscilloscope.

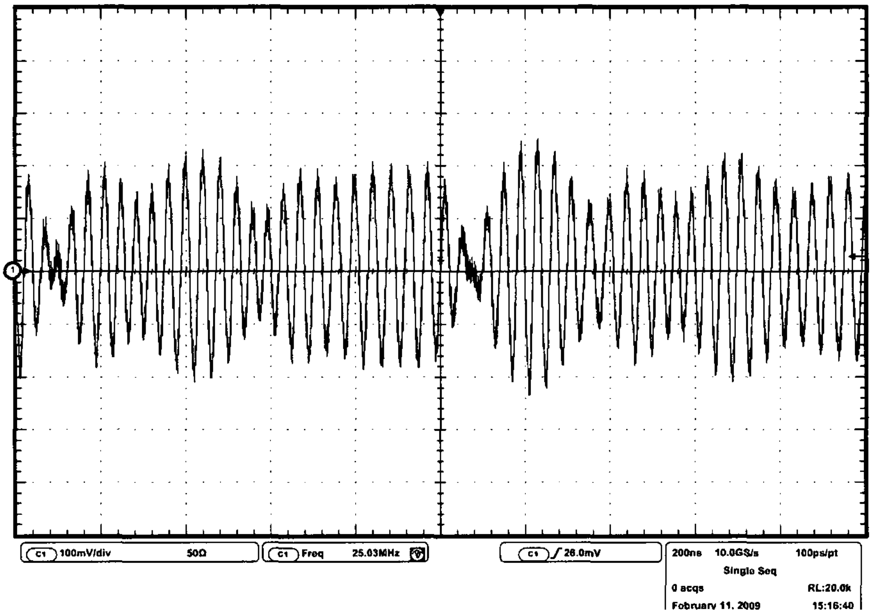


Figure 5-2 Modulated QPSK signal in time domain @ 25 MHz

The signal can also be observed from frequency domain. Figure 5-3 shows the spectrum of signal centered at 25 MHz. As seen in the figure, the signal bandwidth is around 8.8 MHz, and the out-of-band signal is attenuated by approximately 60 dB. The signal spectrum with stop frequency of 200 MHz is shown in Figure 5-4. As discussed before, the fist spectrum image is moved from 75 MHz to 175 MHz, which is realized by the internal 2x interpolation filter with sampling frequency of 200 MHz. The images centered at multiple of sampling rate of 100 MHz are because of sampling theory. Their magnitude is further attenuated by DAC's $\sin(x)/x$ response. The out-of-band signal can be successfully removed by an internal bandpass filter in the

BER tester, a low pass filter preceding the ADC with cut-off frequency of 58 MHz, and implementing proper digital filter on FPGA in the receiver side.

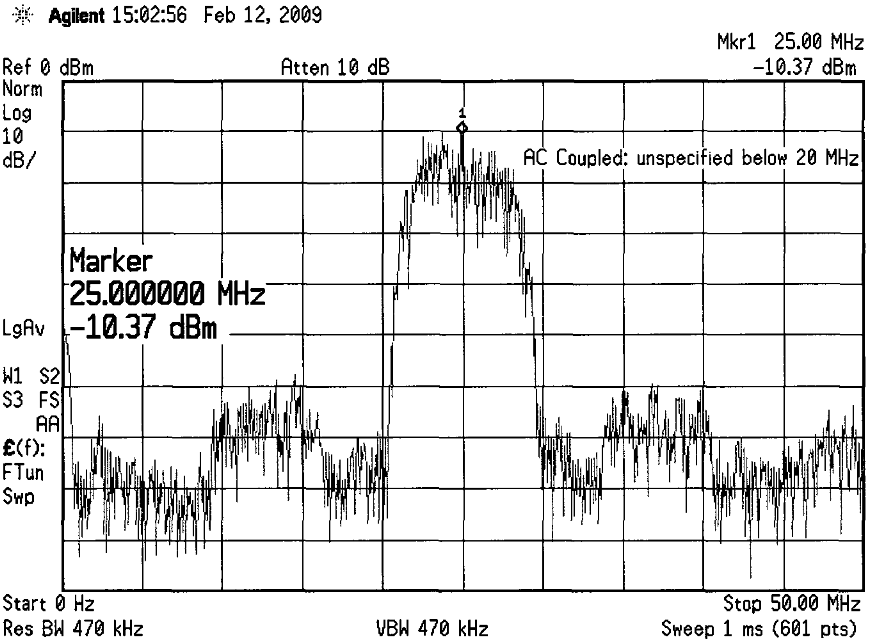


Figure 5-3 Signal spectrum centered @ 25 MHz

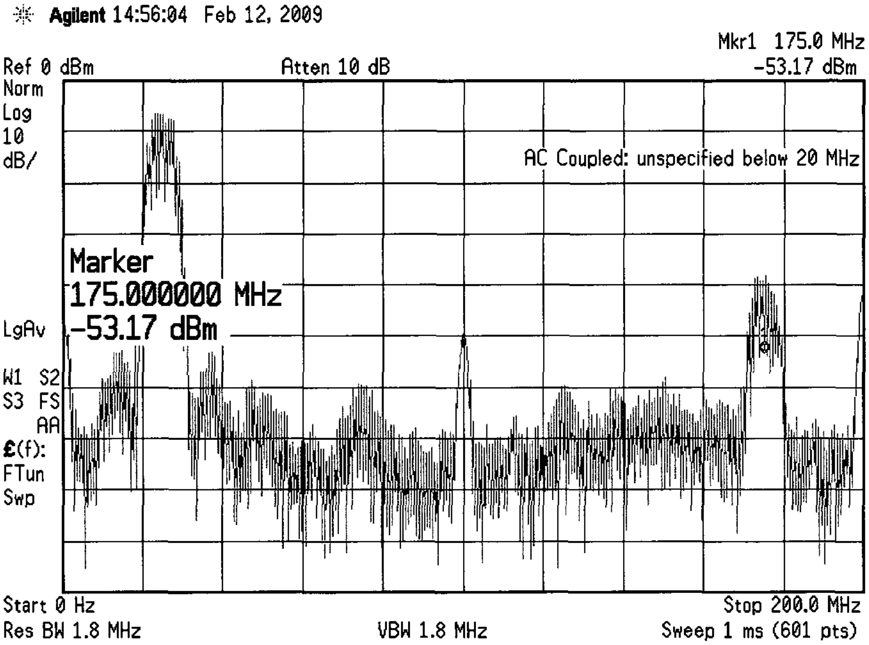


Figure 5-4 Spectrum showing the first image @ 175 MHz

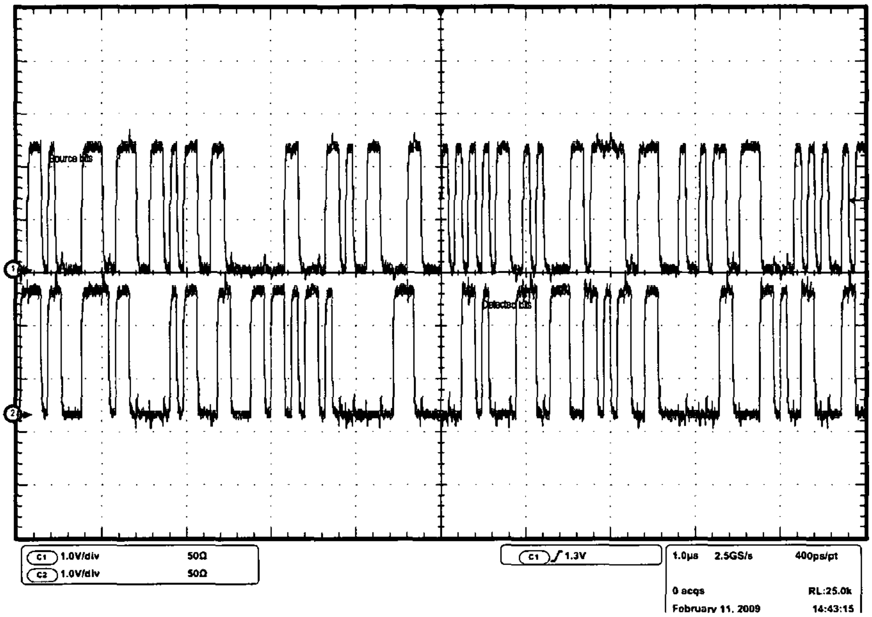


Figure 5-5 Source bits (upper) and detected bits (lower)

The upper plot and lower plot on Figure 5-5 are binary source data from BER tester and final detected bits from FPGA, respectively. A comparison can be made between the upper left part and lower right part. As seen in the figure, the detected bits are exactly the same as source bits, only with about 5 ms delay due to the hardware routing and path delay.

5.3.2 Signal Observation Using Constellation Plot

The X-Y mode of oscilloscope gives the signal constellation when I and Q value are input to the oscilloscope. When the transmitted signal experiences a frequency offset of 0.001 MHz, the received signal constellation rotates as shown in Figure 5-6(a). After carrier recovery is performed in the receiver, the demodulated signal is taken out of the board to oscilloscope. As shown in Figure 5-6(b), the constellation is de-rotated and locked to a stable state, and four QPSK signal points are clearly seen in the plot. These results agree with the previous simulation results.

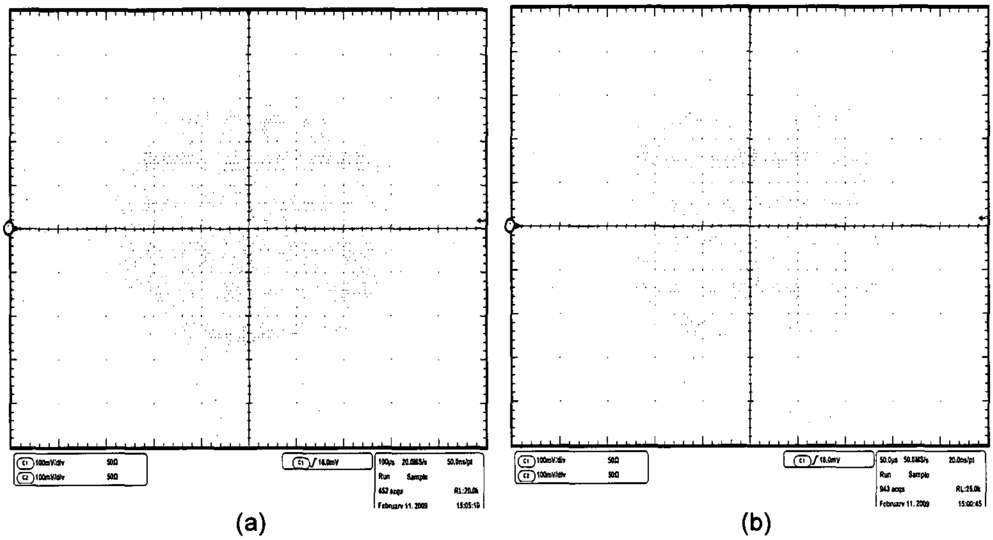


Figure 5-6 Signal constellation before (a) and after (b) carrier recovery

5.3.3 BER Performance

The BER calculation is performed using BER tester. The BER vs E_b/N_0 curves are given with five different test points (E_b/N_0 values). A total of 7.5×10^9 bits were sent for every test point. Figure 5-7 compares the BER performance between software simulation and hardware test of a SISO system, where signal experiences frequency offset of 0.001 MHz and carrier recovery is performed in the receiver. As seen in the figure, the hardware test shows approximately 1.2 dB implementation loss compared with simulation result. Since a number of factors affect the performance of a real hardware implementation rather than software simulation, such as quantization noise, inaccuracy of AWGN bandwidth, self noise of equipment and so on, this 1.2 dB loss is considered as acceptable in a practical design. The BER comparison of a SISO system with both carrier and timing recovery is shown in Figure 5-8, and a 25% symbol timing offset is manually set between transmitter and receiver. As seen in the figure, when roll-off factor $\beta = 0.35$, the ML detector has better performance than Gardner's detector. This also agrees with the previous simulation results. The implementation loss is still 1.2 dB in this scenario.

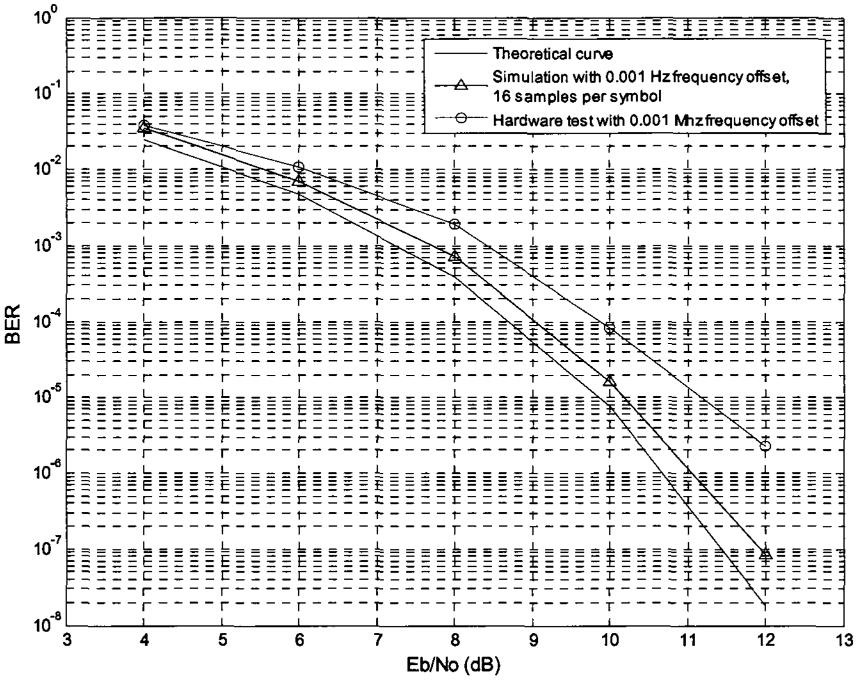


Figure 5-7 BER comparison between software simulation and hardware test (Carrier recovery)

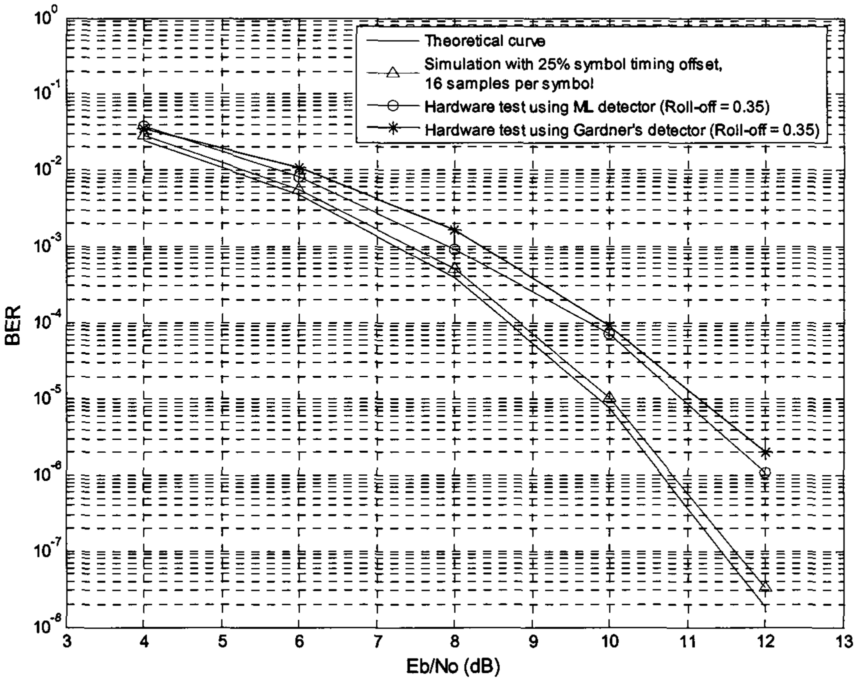
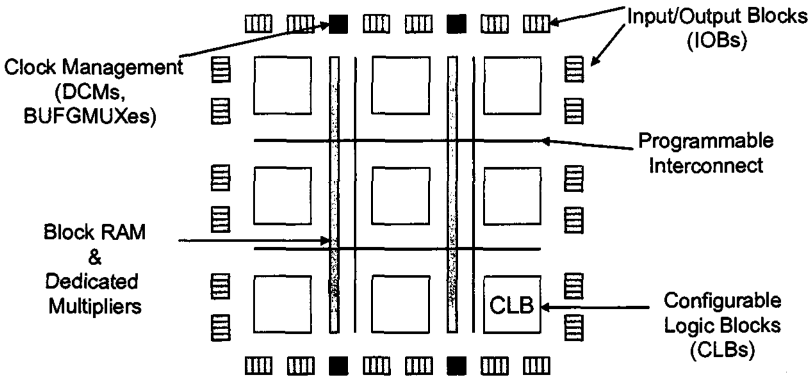


Figure 5-8 BER comparison between software simulation and hardware test (Carrier and timing recovery)

5.3.4 Hardware Utilization

Before examining the hardware utilization of our design, we first review the architecture of Virtex-4 FPGA, and introduce a powerful embedded block, DSP48 slice.



Input/Output Blocks (IOBs)
Clock Management (DCMs, BUFGMUXes)
Programmable Interconnect
Block RAM & Dedicated Multipliers
CLB
Configurable Logic Blocks (CLBs)

Figure 5-9 Virtex-4 FPGA overview

Figure 5-9 illustrates the architecture of Virtex-4 FPGA. The device is organized as an array of logic blocks and programmable routing channels used to provide the connectivity between the logic blocks, I/O pins and other resources. The basic components of Virtex-4 and other Xilinx series FPGAs include configurable logic block (CLBs), input/output blocks (IOBs), programmable interconnect, Block RAM and other resources such as DCMs, embedded multipliers and so on. CLBs are the main logic resource for implementing sequential and combinatorial circuits. Each CLB has four slices, where slice is the elementary programmable logic block in Xilinx FPGAs. Furthermore, each slice includes [48]:

- Two 4-input LUTs used as logic function generators.
- Two dedicated user-controlled multiplexers for combinational logic.
- Dedicated arithmetic logic that can realize XOR, multiplication and addition.
- Two 1-bit registers that can be configured as flip-flops or latches.

The Virtex-4 XC4VSX35 chip provided along Nallatech XtremeDSP board has overall 96×40 CLB arrays. Therefore, 15360 slices, 30720 LUTs and 30720 flip-flops are available [52].

The DSP48, or XtremeDSP slice is an element of Virtex-4 chip that facilitates DSP integration in FPGA. Each DSP48 combines an 18-bit by 18-bit signed multiplier with a 48-bit adder. A programmable multiplexer is followed by the multiplier to select the adder's inputs. Numbers of DSP functions can be realized by DSP48 instead of use of general FPGA fabric, such as multiplication, multiplexer, digital filtering, complex arithmetic and so on. DSP48 slice is a unique hard coded IP (Intellectual Property) embedded in each Virtex-4 device. It delivers high performance, low power consumption, and efficient device utilization [50]. On the Virtex-4 XC4VSX35 chip, overall 192 DSP48 slices are available. In our design, all the multipliers are built using DSP48 slices.

Table 5-1 shows the hardware utilization, power consumption and timing report of our SISO system designs with three scenarios. The report is generated from ISE tool after synthesis and place-and-route of the design. When timing is assumed to be perfect, and carrier recovery is performed to recover a 0.001 MHz frequency offset, as system A shown in the table, only 14% of slices and 29% of embedded multipliers are consumed for the system. In addition, total power consumption including static and dynamic power is 948mw. Timing report shows that the maximum clock frequency for system A is 126.541 MHz.

It is clear that, as symbol timing recovery is incorporated into the design, the hardware consumption increases significantly. When Gardner's timing error detector is used (system B), 29% of slices and 63% of DSP48 slices are consumed. On the other hand, when ML detector is used, 41% of slices and 94 % of DSP48 slices are consumed. The increment of resource usage is mainly due to the use of 1024-tap interpolation filter with parallel polyphase structure and other components in timing recovery design. As we mentioned before, a resource-efficient MAC structure exits. However, the processing speed of MAC depends on the filter length. Since the master clock of the XtremeDSP board is limited to 120 MHz, the MAC is not suitable for implementation of this 1024-tap filter. In addition, by comparing system B and C, it is evident that using Gardner's detector saves more than 10% of slices and 30% of multipliers. This is because only one interpolation filter is used in Gardner's algorithm instead of two in ML detector. However, a trade off has to be made between performance and hardware utilization. As we discussed in chapter 3, if bandwidth is not strictly limited, Gardner algorithm is preferred due to its hardware-efficient and good performance with roll off factor larger than 0.5. We also notice that since all the multipliers are realized by DSP48 slices, the burthen of general FPGA fabric is released. To save DSP48 consumption, we can also use the general fabric to realize the multipliers. The total power consumption for system using Gardner's detector is 976mw, while it is 1002mw when ML detector is used.

System	Slice (15360)		Flip Flop (30720)		4 input LUT (30720)		DSP48 (192)		Maximum clock frequency	Total power (static+dynamic) 1.2V-2.5V
	used	%	used	%	used	%	used	%		
A	2221	14%	2373	7%	2622	8%	56	29%	126.541MHz	948mw
B	4503	29%	3911	12%	4510	14%	122	63%	109.603MHz	976mw
C	6355	41%	5446	17%	7041	22%	182	94%	109.603MHz	1002mw

A: SISO + Carrier recovery
B: SISO + Carrier/Timing recovery (Gardner's detector)
C: SISO + Carrier/Timing recovery (ML detector)

Device: Virtex-4 XC4VSX35-FF668-10

Table 5-1 Resource consumption and timing report

5.3.5 Work Station Overview

Figure 5-10 and 5-11 show our workstation in Wireless Design Lab. It mainly contains a host PC, an XtremeDSP board and a BER tester. The hardware setup has been discussed in the previous section of this chapter.

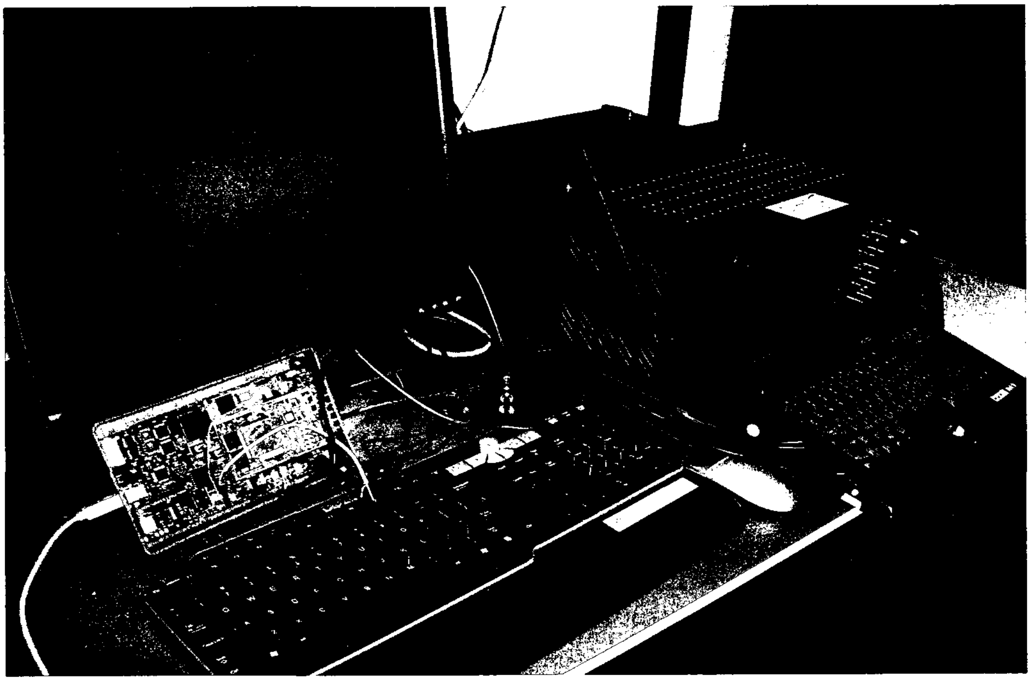
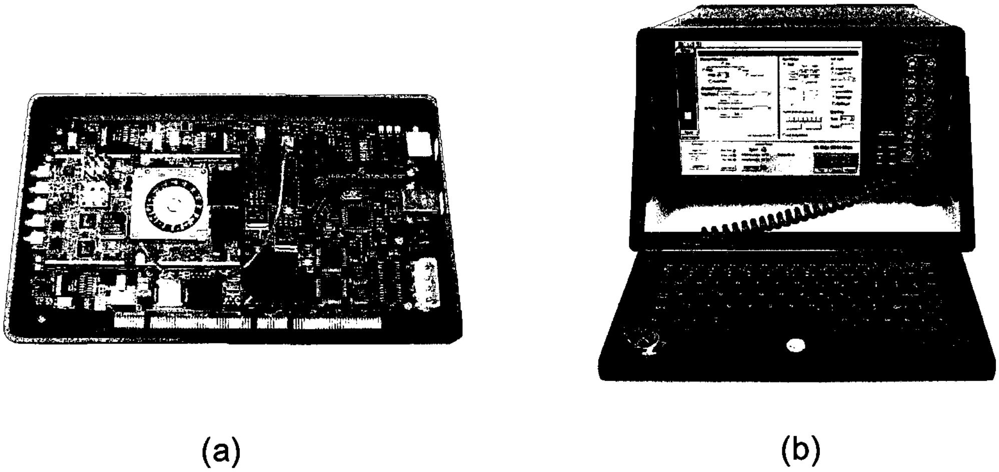


Figure 5-10 Workstation Overview



(a)

(b)

Figure 5-11 XtremeDSP board (a) and BER tester (b)

Chapter 6

Conclusion and Future Work

6.1 Conclusion and Summary of the Thesis

In this thesis, an SDR based SISO system using QPSK modulation scheme is implemented on FPGA. This system produces signal with an IF of 25 MHz and throughput of 12.5 Mbps. One carrier recovery and two symbol timing recovery algorithms (Gardner and ML) are investigated and implemented. A 2x1 MIMO system using Alamouti scheme and CORDIC based carrier recovery is designed as well. The SDR based SISO system can be easily incorporated to the MIMO design. Throughout this thesis, detailed design information of major components such as baseband signal processing, synchronization, and digital up/down conversion is presented along with both computer simulation results and real hardware performance. The comparisons of different algorithms and component structures also provide information of choosing the suitable algorithm or structure according to specific implementation considerations and system requirement. The thesis is summarized as below.

In chapter 2, we explained the design and implementation of a SISO system with QPSK modulation. The background and detailed design process are provided for each component. In addition, the parallel polyphase structure for interpolation and decimation filter is proposed. Compared with the MAC method, this structure is suitable for hardware with limited processing speed but requiring high data rate.

The synchronization issues for SISO system are discussed in chapter 3. One carrier recovery algorithm and two symbol timing recovery algorithms are investigated and designed. Simulation results show that BER performance of proposed CR loop has 0.3 dB degradation compared with ideal situation in an AWGN channel. On the other hand, in STR loop design, ML detector performs better than Gardner's detector when the roll-off factor β is small. As β increases, the acquisition time of Gardner based design becomes shorter, and the BER performance approaches ML based design.

In chapter 4, we discussed the design and implementation of a 2x1 MIMO system using Alamouti scheme. The carrier recovery design using extended CORDIC algorithm is emphasized. The simulation results show that the acquisition time of this CORDIC based design is 70 symbols, and has less than 0.2 dB performance degradation in an AWGN channel in large SNR.

In chapter 5, the hardware test results of the SISO system design are presented. It shows that the system with carrier and timing recovery implemented in the receiver has 1.2 dB implementation losses. Furthermore, use of ML detector is hardware-cost, but provides better performance than Gardner's detector when roll-off factor is small. However, if bandwidth is not strictly limited, Gardner's detector is preferred due to its resources-efficiency and good performance when roll-off factor is large.

6.2 Future Work

This thesis provides detailed information of FPGA based system design. It can be used as a base for the ongoing and future projects conducted in Wireless Design Lab. The following is recommended for future work:

- Most of the basic components in a practical system such as baseband signal processing, synchronization, digital up and down conversion are successfully implemented in this thesis. As a future work, a more complex and practical system design is to be designed and implemented. The tasks include channel coding, space time coding, equalization, channel estimation, power control and so forth.

- Due to the lack of a MIMO channel emulator, the MIMO system can not be tested at this stage. However, this equipment will be available soon, and more complex MIMO system design can be tested. As a future work, by using the powerful SignalMaster Quad platform which consists of 4 high speed DSPs, 2 Virtex-4 FPGAs, totally 16 ADCs and DACs, 8 Gbps inter-FPGA channel and other impressive features, a MIMO system with up to 4 transmitters and 4 receivers can be implemented. Furthermore, with the help of the dual band RF transceiver, the test of MIMO system in the real physical channel is also possible and desired.
- FPGA is becoming a preferred choice for digital system implementation. The latest Virtex-6 devices use up to 759K logic cells with 6-LUT (64-bit ROM) technology, and supports up to 2016 XtremeDSP Slices with 25-bit by 18-bit multipliers. Mixed-mode clock managers are also provided [53]. Due to the state-of-art hardware and software resources available in the lab, more advanced signal processing techniques and various wireless protocols could be implemented and tested. The recent techniques attracting both academia and industry include 3GPP LTE (Long term evolution), Wimax and combination of MIMO and OFDM, and so forth.

Bibliography

[1] L. Pucker, "Finding MIMO: A Proposed Model for Incorporating Multiple Input, Multiple Output Technology Into Software Defined Radios", *Proceedings of the SDR Forum SDR '05 Technical Conference*, Nov. 2005

[2] SDR Forum, www.sdrforum.org

[3] C. Dick, "Design and implementation of high-performance FPGA signal processing datapaths for software-defined radios" *VMEbus Systems*, August 2001

[4] M. Cummings, S. Haruyama, "FPGA in the Software Radio," *IEEE Communications Magazine*, vol. 37, no. 2, pp. 108-112, Feb. 1999

[5] C. Dick and F. Harris, "FPGA QAM Demodulator Design," *12th international Conference on Field Programmable Logic and Applications*, Montpellier, France, Sept., 2002.

[6] F. Cardells et al., "Design of a DVB-S receiver in FPGA", *2003 IEEE Workshop on Signal Processing Systems* (SiPS 2003), Seoul Korea, Aug. 2003.

[7] F. Cardells et al, "Efficient FPGA-based QPSK Demodulation Loops: Application to the DVB standard", *12th international Conference on Field Programmable Logic and Applications*, Montpellier, France, Sept, 2002.

[8] H. Tarn et al, "Designing Efficient Wireless Digital Up and Down Converters Leveraging CORE Generator and System Generator," *Xilinx Application Note,* XAPP1018, Xilinx. Inc., Oct. 2007

[9] S. Creaney and I. Kostarnov, "Designing Efficient Digital Up and Down Converters for Narrowband Systems," *Xilinx Application notes,* XAPP1113, Xilinx Inc. Nov. 2008

[10] Rob Pelt, “Wideband Modem Design Using FPGAs,” in *Proceeding of the SDR 04 Technical Conference and Product Exposition,* 2004.

[11] S. W. Cox, J. A. Seely, “Rapid FPGA Modem Design Techniques For SDRs Using Altera DSP Builder,” *Global signal processing conference & expos for the industry,* 2004

[12] P. Murphy, F. Lou, and J. Patrick Frantz, “A hardware testbed for the implementation and evaluation of MIMO algorithms,” in *Proceedings of the 2003 Conference On Mobile and Wireless Communications Networks*, Oct. 2003.

[13] A. van Zelst and T.C.W. Schenk, "Implementation of a MIMO OFDM-based wireless LAN system," *IEEE Trans. Signal Processing*, vol.52, no.2, pp.483–494, Feb. 2004.

[14] W. Xiang, P. Richardson, B. Walkenhorst, X. Wang, and T. Pratt, “A High-Speed Four-Transmitter Four-Receiver MIMO OFDM Testbed: Experimental Results and Analyses,” *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–10, 2006.

[15] K. Bialkowski, P. Uthansakul, M. Bialkowski, and A. Postula, “Design of MIMO Testbed with an FPGA Board for Fast Signal Processing,” *2006 Auswireless Conference*, 2006.

[16] S. Caban, C. Mehlfhrer, R. Langwieser, A. L. Scholtz, and M. Rupp, “Vienna MIMO testbed,” *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1451–1458, 2006.

[17] K. Bialkowski et al, “2x2 MIMO testbed for dual 2.4GHz/5GHz band,” *International Conference on Electromagnetics in Advanced Applications ICEAA*, Italy, 2007

[18] D. Bates et al, “A 4×4 FPGA-based wireless testbed for LTE applications,” *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications* (PIMRC 2008) Sept. 2008.

[19] Xilinx System Generator for DSP, v 9.2 User Guide, Xilinx Inc., 2007

[20] B. Sklar, *Digital Communications: Fundamentals and Applications,* Second Edition, Prentice Hall, PTR, 2001

[21] T, S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, NewJersey, 1996.

[22] J.G. Proakis, *Digital Communications*, Fourth Edition, New York: McGraw Hill, 2001

[23] K. Gentile, "The Care and Feeding of Digital, Pulse-Shaping Filters," *RF Design magazine,* pp. 50-61, April 2002.

[24] Fliege, N. J. *Multirate Digital Signal Processing: Multirate Systems, Filter Banks, Wavelets*, West Sussex, England: John Wiley & Sons, 1994.

[25] DDS v5.0 data sheet, *Xilinx Product Specification*, ds246, April, 2005

[26] C. Dick, *Multirate Signal Processing,* Prentice Hall PTR, 2001

[27] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing.* New York: Wiley, 1998.

[28] U. Mengali and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers.* New York: Plenum Press, 1997.

[29] L. Litwin, "Matched Filtering and Timing Recovery in Digital Receivers," *RF Design magazine*, pp. 32-48, Sept. 2001

[30] C. Dick, F. Harris, and M. Rice, "FPGA Implementation of Carrier Synchronization for QAM Receivers," *Journal of VLSI Signal Processing*, Kluwer, vol. 36, issue 1, pp. 57-71, Jan. 2004.

[31] F. M. Gardner, "A BPSK/QPSK timing-error detector for sampled receivers," *IEEE Trans. Comm.*, vol. COM-34, no. 5, pp. 423–429, May. 1986.

[32] F. M. Gardner, "Interpolation in Digital Modems – Part I: Fundamentals," *IEEE Trans. Comm.,* vol. COM-41, pp. 501-507, Mar. 1993.

[33] F. M. Gardner, L. Eurp, and R. A. Harris, "Interpolation in Digital Modems – Part II: Implementation and Performance," *IEEE Transaction on Communications,* vol. COM-41, pp. 998-1008, Jun. 1993.

[34] C. Dick, F. Harris, and M. Rice, "Synchronization in software radios – carrier and timing recovery using FPGAs," in *Proceedings of 2000 IEEE Symposium on Field-Programmable Custom Computing Machines*, pp. 195–204, April 2000.

[35] F. Harris and M. Rice, "Multirate digital filters for symbol timing synchronization in software defined radios," *IEEE Journal on Select Areas in Communications*, vol.19, pp. 2346–2357, Dec. 2001.

[36] Hwang and J. Ballagh, "Building Custom FIR Filters Using System Generator," *12th international Conference on Field Programmable Logic and Applications*, Montpellier, France, Sept, 2002.

[37] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Select Areas in Communications*, vol. 16, pp. 1451–1458, Oct. 1998.

[38] B. Vucetic and J. Yuan, *Space-Time Coding*, John Wiley, 2003.

[39] D. Gesbert, M. Shafi, D. Shiu, P. Smith, " From theory to practice: An overview of space-time coded MIMO wireless systems ", *IEEE Journal on Selected Areas on Communications* (JSAC). special issue on MIMO systems, April 2003

[40] F. Harris, C. Dick, and M. Rice, "Digital Receivers and Transmitters Using Polyphase Filter Banks for Wireless Communications, " *IEEE Transactions on Microwave Theory and Techniques*, vol 51, no. 4, pp. 1395 - 1412, April 2003.

[41] C. W. Farrow. "A Continuously Variable Digital Delay Element," In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS '88)*, pp. 2641–2645, June 1988.

[42] M. Henker and G. Fettweis, "Combined Filter for Sample Rate Conversion, Matched Filtering, and Symbol Synchronization in Software Radio Terminals," *Proceedings of the European Wireless*, pp. 61-66, Sept. 2000.

[43] J. Valls and et al., "The use of CORDIC in software defined radios: A tutorial," *IEEE Communications Magazine*, vol. 44, no. 9, pp. 46–50, Oct. 2006.

[44] Y.H. Hu, "CORDIC-Based VLSI Architectures for Digital Signal Processing," *IEEE Signal Processing Magazine*, pp. 16-35, July 1992.

[45] A. shoari, M. kamarei, and A. Radmand, "Implementation of Costas loop using CORDIC algorithm for software radio applications," in *Proceedings of IEEE Communications*, Vol. 152, No. 1, pp. 113-118, 2005.

[46] D. R. Stephens, *Phase-Locked Loops for Wireless Communications – Digital, Analog and Optical Implementations*, second edition, Kluwer Academic Publishers, 2002.

[47] V. Tarokh, H. Jafarkhani, and A.R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Transactions on Information Theory*, vol. 45, pp. 1456–1467, July 1999.

[48] Virtex-4 FPGAs User Guide, v2.6, Xilinx Inc., Dec. 2008

[49] XtremeDSP Development Kit-IV User Guide, Issue 2, Xilinx Inc., 2005.

[50] XtremeDSP for Virtex-4 FPGAs User Guide, Xilinx Inc., 2005.

[51] FB100A BER Test System User Manual, FastBit Tech, Aeroflex Test Solutions, 2004.

[52] Virtex-4 Family Overview, Xilinx Inc., 2004

[53] Virtex-6 Family Overview, Xilinx Inc., 2009