A DSP-Based π/4-DQPSK Modem

A Thesis

submitted to the College of Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in the Department of Electrical Engineering

University of Saskatchewan

Saskatoon

by

Wentao Li

Spring, 1999

© Copyright Wentao Li, 1999. All rights reserved
PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Degree of Master of Science from the University of Saskatchewan, the author agrees that the libraries of this University may make it freely available for inspection. The author further agrees that permission for copying of this thesis in any manner, in whole or in part for scholarly purposes may be granted by the professor who supervised this thesis work or, in his absence, by the Head of the Department or the Dean of the College of Graduate Studies and Research at the University of Saskatchewan. Any copying, publication, or use of this thesis, or parts thereof, for financial gain without the author’s written permission is strictly prohibited. Proper recognition shall be given to the author and the University of Saskatchewan in any scholarly use which may be made of any material in this thesis.

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

Head of the Department of Electrical Engineering,

57 Campus Drive,

University of Saskatchewan,

Saskatoon, Saskatchewan,

Canada S7N 5A9
ACKNOWLEDGEMENTS

I would like to express my appreciation and gratitude to my supervisor, Dr. J. Eric Salt, for his guidance, support, and patience during my graduate education. He has been an invaluable source of technical knowledge and has certainly helped inspire many of the ideas expressed in this thesis. I had the great benefits of the graduate level courses EE819 (Multirate Digital Signal Processing) and EE845 (Random Variables in Engineering Systems) taught by Dr. J. Eric Salt.

I would like to thank Mr. Garth Wells for his technical assistance. I would like to thank the management, and staff of Telecommunications Research Labs (TRLabs), Saskatoon, Canada, for providing a graduate scholarship and allowing me to use their facilities during this research.

I would like to thank the graduate students of TRLabs: Neil Gerein, Jason Dielschneider and Lyle Swanson, for suggesting better presentations.

I would like to thank my parents and sisters for providing the moral support and encouragement.

Finally, special thanks goes out to my wife, Hui, for her love, understanding and support, and to my son, Jack, now thirteen months, for his smile and cry. He had to play with his mother without his father’s time during many weekends and holidays. My wife and my son have provided the happy environment that was my most vital support in completing this thesis.
ABSTRACT

A modem method known as $\pi/4$ offset, differentially encoded quadrature phase shift keying ($\pi/4$-DQPSK) has been adopted for the United States and Japanese digital cellular time division multiple access (TDMA) systems and Personal Communications Systems (PCS). The rationale for such a choice is the high bit rate-bandwidth ratio and its applicability to noncoherent detection. Most current systems are implemented with analog hardware.

In the $\pi/4$-DQPSK modem described in this thesis, DSP solutions are employed to process not only the baseband signal but also the intermediate frequency (IF) signal. The DSP technologies used in this modem include (1) Digital Complex Sampling, (2) Polyphase Filters, (3) Canonic Signed Digit Multipliers for FIR Filter, (4) Non-Data-Aided Timing Parameter Estimation, (5) Multirate Signal Processing. With these DSP solutions, various problems in analog and baseband DSP approaches, such as dc offset voltages, dc voltage drifts, analog filter phase distortions, quadrature phase and gain imbalance, and amplifier and mixer nonlinearities, are eliminated and very precise and controllable performance can be achieved without sophisticated compensation techniques. Other advantages of the DSP solutions include the ability to easily program the hardware to accommodate different data rates, modulation formats and filter specifications. The DSP solution is also an efficient method to minimize power consumption, size and cost of the systems. The mathematical model, simulation results, hardware designs in Very High Speed Integrated Circuit Hardware Description Language (VHDL), and testing results are presented in this thesis.

The DSP-based $\pi/4$-DQPSK modem is designed and implemented on two Altera FLEX10K70 chips. The chip size is 4,428 logical cells (approximately 82k gates). The maximum bit rate is 5Mbit/sec. For an additive white Gaussian noise channel, the results of a bit error rate (BER) measurement indicate that the BER performance of the modem degrades about 1.5 dB from theory when 17th-order square root raised cosine matched
filters are used. The performance degradation due to a carrier frequency offset is also investigated in terms of the BER.
# TABLE OF CONTENTS

PERMISSION TO USE ................................................................. i

ACKNOWLEDGEMENTS ......................................................... ii

ABSTRACT ........................................................................ iii

TABLE OF CONTENTS .......................................................... v

LIST OF FIGURES .................................................................. ix

LIST OF TABLES .................................................................... xvi

LIST OF ABBREVIATIONS ..................................................... xvii

1. Introduction ........................................................................ 1

1.1 Background ....................................................................... 1

1.2 DSP Solutions ................................................................... 3

1.3 π/4-DQPSK Modems for U.S. and Japanese Digital Cellular and Personal Communications Systems Standards ........................................... 4

1.4 The Objectives of this Thesis ............................................. 5

1.5 Thesis Organization .......................................................... 5

2. Mathematics and Basic Digital Communication Functions ...... 6

2.1 Complex Envelope Representation ..................................... 6

2.2 System Clocks ................................................................... 7

2.3 Digital Complex Mixers, Digital Up Converters ................. 8
2.4 Digital Complex Sampling .................................................. 12

2.5 π/4-DQPSK Signal Description ........................................... 15

2.6 Intersymbol Interference, Matched Filters and Eye Patterns ............. 18

2.7 π/4-DQPSK Differential Encode and Decode ............................... 23

2.8 Sampling Rate Conversion .................................................. 27

2.8.1 Downsampling ............................................................. 27

2.8.2 Upsampling ............................................................... 29

3. Architecture and Model of a DSP-Based π/4-DQPSK Modem 33

3.1 Introduction ........................................................................ 33

3.2 Architecture and Model of a DSP-Based π/4-DQPSK Modem .............. 35

3.2.1 Modulator ..................................................................... 37

3.2.2 Demodulator ................................................................. 39

3.3 System-level Considerations .................................................. 44

4. Efficient FIR Filter Design ...................................................... 48

4.1 Introduction ........................................................................ 48

4.2 Linear Phase FIR Filter ......................................................... 48

4.3 Silicon Saving Techniques and Pipelining ................................ 50

4.3.1 Canonic Signed Digit Multiplier ....................................... 50

4.3.2 Truncation .................................................................... 51

4.3.3 Pipelining ..................................................................... 51

4.4 Implementing the Square Root Raised Cosine Filter ....................... 52

4.4.1 Square Root Raised Cosine Filter Design in Matlab ................. 52

4.4.2 Square Root Raised Cosine Filter Architecture in FLEX10K ...... 55
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.3</td>
<td>Performance of the Designed Filter</td>
<td>57</td>
</tr>
<tr>
<td>4.5</td>
<td>Half-band Filters</td>
<td>59</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Half-Band Filter Characteristics</td>
<td>59</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Lagrange Half-Band Filter</td>
<td>60</td>
</tr>
<tr>
<td>4.6</td>
<td>Decimator and Interpolator with Polyphase Filters</td>
<td>65</td>
</tr>
<tr>
<td>4.6.1</td>
<td>Aliasing in the Transition Band</td>
<td>65</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Decimator with Polyphase Filters</td>
<td>67</td>
</tr>
<tr>
<td>4.6.3</td>
<td>Interpolator with Polyphase Filters</td>
<td>69</td>
</tr>
<tr>
<td>4.6.4</td>
<td>Power Consumption Estimation and Performance</td>
<td>71</td>
</tr>
<tr>
<td>4.7</td>
<td>Summary</td>
<td>72</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>73</td>
</tr>
<tr>
<td>5.2</td>
<td>Squaring Timing Recovery</td>
<td>75</td>
</tr>
<tr>
<td>5.3</td>
<td>Timing Recovery in a DSP-based Modem</td>
<td>76</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Description of NDA Timing Parameter Estimation Algorithm</td>
<td>76</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Simulation of the NDA Timing Parameter Estimation Algorithm</td>
<td>79</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Quality of the NDA Timing Parameter Estimation Algorithm</td>
<td>83</td>
</tr>
<tr>
<td>5.4</td>
<td>Implementation of the Algorithm in FLEX10K</td>
<td>87</td>
</tr>
<tr>
<td>5.5</td>
<td>Summary</td>
<td>100</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>101</td>
</tr>
<tr>
<td>6.2</td>
<td>Chip Size</td>
<td>101</td>
</tr>
<tr>
<td>6.3</td>
<td>Modulator Performance</td>
<td>102</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1.1 Wireless Communications -- the Overall System. .......................... 2

2.1 System Clocks of the Modem...................................................... 8

2.2 Digital Complex Mixer............................................................ 9

2.3 Digital Mixer Realization without any Multipliers....................... 10

2.4 Simulation Waveforms for a Mixer In1(n)x( cos2πn/4)................. 11

2.5 Digital Up Converter.............................................................. 11

2.6 Spectrum of Complex Envelope Signal x(t)................................... 12

2.7 Digital Complex Sampler........................................................ 13

2.8 Spectrum of the Signal x(n).................................................... 14

2.9 Spectrum of the Signal M_l(n)................................................ 14

2.10 Unfiltered π/4-DQPSK Signal Constellation............................ 17

2.11 Raised Cosine Filter Characteristics for r = 0, 0.35, 1. a) Impulse

    Responses. b) Magnitude Frequency Responses............................. 20

2.12 Eye Pattern Measurement...................................................... 22
2.13 Effect of Truncation on the Eye Diagram ........................................... 23

2.14 A Block Diagram of the π/4-DQPSK Differential Encode ...................... 24

2.15 Block Diagram of Differential Detection ............................................. 25

2.16 Downsampler ..................................................................................... 28

2.17 A Possible Realization of a Downsampler M ......................................... 28

2.18 Upsampler ......................................................................................... 29

2.19 Spectrum after Upsampling by 2 ......................................................... 30

2.20 An Upsampler with L=4. a) Logic Diagram of Upsampling by a Factor of 4.

b) Simulation Waveform of Upsampling by a Factor of 4 ............................ 31

3.1 Block Diagram of π/4-DQPSK Modulator Architecture ......................... 34

3.2 Schemes of Differential Detection of π/4-DQPSK Signal. a) Block

Diagram of the IF Band Differential π/4-DQPSK Demodulator. b) Block

Diagram of the Baseband Differential π/4-DQPSK Demodulator ............. 35

3.3 DSP-Based π/4-DQPSK Modem. a) Modulator. b) Demodulator ............ 36

3.4 Simulation Result with 25-tap SRRC Filters ........................................ 46

3.5 Simulation Result with 17-tap SRRC Filters ........................................ 47

4.1 Description of a FIR Filter ................................................................... 48

4.2 First Direct Form of a Transversal Filter ............................................. 49

4.3 Transversal Structure for Symmetrical FIR Filter with an Odd Number
4.4 Multiplication of $x$ with 75 Based on CSD

4.5 Multiplication of $x$ with 75 Based on CSD with MNA


4.7 A SRRC Filter Logic Block Diagram

4.8 Implementing a Multiply by 71


4.10 Tolerance Specification of a Lowpass Filter

4.11 Frequency Response of the Lagrange Half-band Filter for Different Parameters

4.12 Lagrange Half-Band Filter with Parameter ($i=3, N=11$) and Lowpass
Filtering................................................................. 63

4.13 Logic Block Diagram of the Lagrange Half-Band Filter with \((i=3,N=11)\). 64

4.14 Design, Simulation and Test for the Lagrange Half-Band Filter with

Parameter \(i=3, N=11\)................................................... 65

4.15 Decimator................................................................. 66

4.16 Aliasing in the Transition Band.......................................... 67

4.17 Memory-Saving Decimating by a Factor of 2 with Polyphase Filters..... 68

4.18 Block Diagram of a Decimator with Polyphase Filters.................. 69

4.19 Interpolator................................................................. 69

4.20 Memory-Saving Interpolating by a Factor of 2 with Polyphase Filters..... 70

4.21 Block Diagram of an Interpolator with Polyphase Filters............... 71

5.1 Timing Recovery Methods. a) Analog Timing Recovery. b) Hybrid

Timing Recovery. c) Digital Timing Recovery. d) DSP-Based

Timing Recovery.......................................................... 74

5.2 Squaring Timing Recovery................................................ 75

5.3 Block Diagram of a DSP-Based Timing Recovery............................ 76

5.4 Observation Interval..................................................... 78
5.5 Waveform and Spectrum of the Signal $x(m, \varepsilon) = |x_{im}(m, \varepsilon)| + |x_{om}(m, \varepsilon)|$

Over 64 Intervals .................................................................................................................................................. 80

5.6 Eye-Pattern Diagrams and 4-Point DFT over 64 Intervals with $\varepsilon = 0.5$

and the Absolute-Value Rectifier. a) $I$ Channel Eye-Pattern Diagram.

b) Magnitude of 4-Point DFT over 64 Intervals. c) $Q$ Channel Eye-Pattern Diagram. d) Compass Plot for First Component $\mathcal{SC}_1$ ........................................................................................................... 81

5.7 Eye-Pattern Diagrams and 4-Point DFT over 64 Intervals with $\varepsilon = 0.75$

and the Absolute-Value Rectifier. a) $I$ Channel Eye-Pattern Diagram.

b) Magnitude of 4-Point DFT over 64 Intervals. c) $Q$ Channel Eye-Pattern Diagram. d) Compass Plot for First Component $\mathcal{SC}_1$ ........................................................................................................... 81

5.8 Eye-Pattern Diagrams and 4-Point DFT over 64 Intervals with $\varepsilon = 0$ and

the Absolute-Value Rectifier. a) $I$ Channel Eye-Pattern Diagram.

b) Magnitude of 4-Point DFT over 64 Intervals. c) $Q$ Channel Eye-Pattern Diagram. d) Compass Plot for First Component $\mathcal{SC}_1$ ........................................................................................................... 82

5.9 Eye-Pattern Diagrams and 4-Point DFT over 64 Intervals with $\varepsilon = 0.25$

and the Absolute-Value Rectifier. a) $I$ Channel Eye-Pattern Diagram.
b) Magnitude of 4-Point DFT over 64 Intervals. c) Q Channel Eye-Pattern Diagram. d) Compass Plot for First Component $SC_i$. 82

5.10 Eye-Pattern Diagrams and 4-Point DFT over 64 Intervals with $\varepsilon = 0.25$ and the Square-Law Rectifier. a) I Channel Eye-Pattern Diagram.

b) Magnitude of 4-Point DFT over 64 Intervals. c) Q Channel Eye-Pattern Diagram. d) Compass Plot for First Component $SC_i$. 83

5.11 Computation of the First Order Coefficient of 4-Point Discrete Fourier Transform of $x(i,\varepsilon)$. 89

5.12 Averaging Operator. 89

5.13 Simulation Waveforms for the Averaging Operator. 91

5.14 Distribution of Events $A,B,C,D$ in Complex Plan $SC_i$. 92

5.15 Decision Circuit for Changing the Sampling Time. 95

5.16 Data Extractor. 96

5.17 State Diagram Corresponding to the Phase Shifter. 96

5.18 Simulation Waveform for the Phase Shifter. 100

6.1 Performance Measurement of the Modulator. 102

6.2 Power Spectrum of $\pi/4$-DQPSK Modulator Output Measured by HP89410A. 103

6.3 $\pi/4$-DQPSK Constellation Measured by HP89410A. 104

6.4 I Channel Eye-Pattern Diagram. 105
6.5 Q Channel Eye-Pattern Diagram ........................................... 105
6.6 Performance Measurement of the Demodulator ...................... 107
6.7 BER versus $E_b/N_0$ for the Additive White Gaussian Noise Channel .......... 108
6.8 Eye-pattern diagrams at the output of the Demodulator’s Matched Filters

with no Carrier Frequency Offset and the Additive White Gaussian Noise $E_b/N_0 = 15$ dB ................................................................. 109

6.9 BER versus $E_b/N_0$ as a Function of Carrier Frequency Offset for the Additive White Gaussian Noise Channel ........................................... 110

6.10 Eye-Pattern Diagrams at the Output of the Demodulator’s Matched Filters

with the Carrier Frequency Offset = 5Hz and the Additive White Gaussian Noise $E_b/N_0 = 15$ dB ................................................................. 110
LIST OF TABLES

2.1 Data-Phase-Change Correspondence for π/4-DQPSK.......................... 16

2.2 Two 8x8-bit ROMs : I_k_ROM and Q_k_ROM........................................... 24

2.3 Relationship between the Present address and the Input Symbol and the Previous Address................................................................. 24

4.1 Coefficients of a SRRC Filter................................................................. 54

4.2 Performance of SRRC Filter..................................................................... 58

4.3 Coefficients of Half-Band Filter with Parameter i=3, N=11.................... 62

4.4 Performance of the Lagrange half-band Filter with Parameter i=3, N=11.. 64

4.5 Comparison of the Relative Performance for Several Implementations...... 71

5.1 The Actions Taken in Response to an Event............................................ 93

5.2 Relationship of the Phase Offset and SC_i.............................................. 94

6.1 Chip Size of the DSP-Based π/4-DQPSK Modem................................. 102
LIST OF ABBREVIATIONS

BER  Bit Error Rate
CPLD Complex Programmable Logic Devices
CSD Canonic Signed Digit
DA Data-Aided
dB decibel
DD Decision-Directed
DFT Discrete Fourier Transform
$\pi/4$-DQPSK $\pi/4$ offset, Differentially Encoded Quadrature Phase Shift Keying
DSP Digital Signal Processing
EAB Embedded Array Block
EDIF Electronic Design Interchange Format
$E_b/N_0$ Energy per bit/Noise power spectral density ratio
FFT Fast Fourier Transform
FIFO First In First Out
FIR Finite Impulse Response
$I$ In-phase component
ICs Integrated Circuits
IF Intermediate Frequency
ISI Intersymbol Interference
kBd kilo Baud
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCs</td>
<td>Logic Cells</td>
</tr>
<tr>
<td>MNA</td>
<td>Minimum Number of Add</td>
</tr>
<tr>
<td>MTSO</td>
<td>Mobile Telephone Switching Office</td>
</tr>
<tr>
<td>msd</td>
<td>Mean Square Difference</td>
</tr>
<tr>
<td>MSPS</td>
<td>Million Samples Per Second</td>
</tr>
<tr>
<td>NDA</td>
<td>Non-Data-Aided</td>
</tr>
<tr>
<td>PCS</td>
<td>Personal Communications Systems</td>
</tr>
<tr>
<td>PLL</td>
<td>Phase Locked Loop</td>
</tr>
<tr>
<td>PRBS</td>
<td>Pseudorandom Binary Sequence</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Switched Telephone Network</td>
</tr>
<tr>
<td>$Q$</td>
<td>Quadrature-phase</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>SRRC</td>
<td>Square Root Raised Cosine</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>VCO</td>
<td>Voltage Controlled Oscillator</td>
</tr>
<tr>
<td>VHDL</td>
<td>Very High Speed Integrated Circuit Hardware Description Language</td>
</tr>
<tr>
<td>VSA</td>
<td>Vector Signal Analyzer</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Background

Historically all telephone communications were accomplished by a wired line network. All telephones were connected to the network with a pair of wires. Recently, a wireless network has emerged where the link between the network and telephone is wireless. This wireless technology has been employed and recognized as a step forward.

... a new era of human communications where wireless technologies become information skyways, a new avenue to send ideas and masses of information to remote locations in ways most of us would never have imagined. Wireless hand-held computers and phones will deliver the world to our fingertips, wherever we may be, with speed and flexibility.

William Clinton
President of the United States of America

Figure 1.1 shows an overall wireless communications systems. Communications today is on the move. In cars, office buildings, manufacturing plants, shopping malls -- wherever people go, wireless telephones and other communications systems go with them. Base stations are at the heart of wireless communications systems. All base station cell sites connect to the Mobile Telephone Switching Office (MTSO). The MTSO in turn interfaces to the Public Switched Telephone Network (PSTN) by connecting to a
Central Office. Control of all cell sites, all subscriber records, statistics, and billing is maintained at the MTSO. In the future, smaller "Microstations" will bring wireless services into offices, factories, schools, shopping malls -- wherever there is a need for mobile communications.

Figure 1.1 Wireless communications -- the overall systems

Driving this new mobility in communications is high-speed digital technology, and in the forefront of this technology is Digital Signal Processing (DSP). DSP is a prime enabler for digital communications and is confronting the specific needs of wireless systems for higher levels of integration and greater performance with low power consumption. DSP technology will also continue to be the key baseband technology on which digital wireless systems are built. DSP solutions will grow in importance as the partition between the radio and the baseband subsystem gradually moves to drive more and more functionality into the signal processor.
1.2 DSP Solutions

DSP algorithms can be implemented in software or hardware. In software implementation, a DSP processor such as TMS320C40 is used. In hardware implementation, the Application-Specific Integrated Circuits (ASICs) or the Field-Programmable-Gate-Arrays (FPGAs) are used. Designers are often forced to find a compromise between flexibility and performance. DSP processors are relatively inexpensive and very flexible, but they have severe throughput limitations. ASICs offer improved throughput, but are relatively inflexible. ASICs also have a disadvantage since they require long lead times for development and production. FPGAs such as Altera FLEX10k devices provide nearly the throughput of custom ASICs while maintaining full flexibility of a DSP processor. Because FPGAs can be reprogrammed to reconfigure the entire DSP implementation.

Altera FLEX10K devices can fully implement DSP solutions for a wide variety of the complex DSP digital communications functions including numerically controlled oscillators, complex mixers, frequency and timing recovery, linear feedback shift registers, FIR filters, and FFT functions. These functions can be integrated to complete system-level solutions.

Altera also offers powerful development tools to complete the design solution. Altera's MAX+PLUS II® software provides an integrated, easy-to-use development tool that supports all Altera device families. The MAX+PLUS II software is compatible with industry-standard EDA tools and provides direct VHDL, Verilog HDL, and EDIF interfaces.
1.3 \( \pi/4 \)-DQPSK Modems for U.S. and Japanese Digital Cellular and PCS Standards

Spectral efficiency is one of the most important requirements for PCS and digital cellular systems. The \( \pi/4 \)-DQPSK has been adopted for the United States and Japanese digital cellular TDMA systems and PCS. To meet the spectral efficiency requirement, the \( \pi/4 \)-DQPSK modem transmits two bits per symbol, providing twice the bandwidth efficiency of binary phase shift keying. The requirement of spectral efficiency can also be met by conventional filtered four phase modem schemes such as QPSK and Offset Keyed QPSK (OKQPSK). QPSK is used in many commercial communication systems and works well in systems where linear amplifiers are used. However, linear amplifiers are not power-efficient, making them undesirables for portable devices with limited battery power. Another major problem of QPSK is that a filtered QPSK waveform does not have a constant envelope. The 180-degree phase transitions will cause the envelope of the filtered signal to go through zero. If a high efficiency amplifier with non-linearity is employed in the transmitter, which is usually the case in mobile communications systems, QPSK signal will cause spectral spreading and interference into adjacent radio channels. OKQPSK has less variation on the envelope which reduces this spectral spreading and the resulting interference. However, OKQPSK requires coherent detection that means carrier recovery in the demodulator. The \( \pi/4 \)-DQPSK has an advantage over OKQPSK as it can be demodulated without a coherent reference (the carrier frequency and phase) at demodulator. Noncoherent detection offers a further advantage of cost-efficiency and power-efficiency in the radio frequency (RF) gain stages.
1.4 The Objectives of this Thesis

The objectives of this thesis are the following:

(1) To design and implement a DSP-based $\pi/4$ DQPSK modem on FPGAs.

(2) To determine the chip size in logical cells (LCs) and/or gates required to implement a DSP-based $\pi/4$ DQPSK modem on Altera FLEX10K70 chips.

(3) To determine the implementation loss associated with the DSP-based solution.

1.5 Thesis Organization

In addition to this introductory chapter, this thesis contains seven chapters. In Chapter 2, mathematical tools and some digital communication functions, which are employed in a DSP-based $\pi/4$ DQPSK modem, will be developed. In Chapter 3, the architecture and model of a DSP-based $\pi/4$-DQPSK modem will be described and the system-level design will be considered. In Chapter 4, the structure of matched filters, half-band filters and polyphase filters will be derived. Timing recovery is one of the most critical functions in the DSP-based the $\pi/4$ DQPSK modem. The timing recovery algorithm and realization will be presented in Chapter 5. In Chapter 6, the performance results of the DSP-based $\pi/4$-DQPSK modem are shown. Finally, the conclusions are presented in Chapter 7.
2. Mathematics and Basic Digital Communication Functions

This chapter serves as a short introduction to some mathematical tools that are used to describe digital signals. Some basic digital communication functions, which are employed in a DSP-based π/4-DQPSK modem, are also developed.

2.1 Complex Envelope Representation

In communication systems, information is often conveyed by means of a bandpass signal, resulting from modulating a sinusoidal carrier. Such a signal can be viewed as a sinusoid whose amplitude and phase are fluctuating with time.

In the continuous-time case, any bandpass signal $s(t)$ with carrier frequency $\omega_c = 2\pi f_c$ can be represented by the unique complex signal $s_c(t)$, where $s_c(t)$ is such that

$$s(t) = \text{Re}[s_c(t)e^{j\omega_c t}]$$

(2.1)

$s_c(t)$ is called the complex envelope of the modulated signal. Furthermore, two other equivalent representations are

$$s(t) = |s_c(t)|\cos(\omega_c t + \theta(t))$$

$$= I(t)\cos\omega_c t - Q(t)\sin\omega_c t,$$

(2.2)

where the complex signal $s_c(t)$ is defined in terms of the real signals $I(t)$ and $Q(t)$ by
\[ s_e(t) = I(t) + jQ(t) = |s_e(t)|e^{j\theta(t)} \]
\[ |s_e(t)| = \sqrt{I^2(t) + Q^2(t)} \]
\[ \theta(t) = \tan^{-1}\left( \frac{I(t)}{Q(t)} \right) . \]

Equation (2.3)

\[ I(t) \] is called the In-phase component and \[ Q(t) \] is called the Quadrature-phase component.

The characterization of continuous-time signals given above can be easily carried over to discrete-time signals. Such signals are usually obtained by uniformly sampling a continuous-time signal at a sufficiently high rate. The following equations are obtained

\[ s_e(n) = I(n) + jQ(n) = |s_e(n)|e^{j\theta(n)} \]
\[ |s_e(n)| = \sqrt{I^2(n) + Q^2(n)} \]
\[ \theta(n) = \tan^{-1}\left( \frac{I(n)}{Q(n)} \right) \]
\[ s(n) = \text{Re}\left[ s_e(n)e^{j2\pi Fc n/F_s} \right] = \text{Re}\left[ s_e(n)(\cos\frac{2\pi Fc n}{F_s} + j\sin\frac{2\pi Fc n}{F_s}) \right] , \]

where the sampling rate \( F_s \) in Hz is high enough to satisfy \( F_s > 2F_c + B_c \) where \( F_c \) is the intermediate carrier frequency in Hz and \( B_c \) is the bandwidth of \( s_e(t) \) in Hz.

2.2 System Clocks

Reliable clocking is critical to the successful operation of any digital design, regardless of whether it is constructed with discrete logic or programmable logic. A poorly designed clock configuration may lead to erratic behavior. Altera recommends using a global clock wherever possible in the FLEX10K devices. Global clocks are the simplest and most predictable clocks for a project. With a truly synchronous system, a single master Clock, driven by an input pin, clocks every flipflop in the project. But it is not possible to employ a single master Clock to drive all flipflops in a multirate DSP system. The clocks shown in Figure 2.1 are used in this project. All clocks are derived
from clk_Fs. Altera FLEX10K70 provides two global Clock pins. The clock signal clk_Fs with the frequency Fs and the clock signal clk_Fc with the frequency Fc are selected as the global clocks since many of the subsystems depend on these clocks.

![Diagram of Altera FLEX10K70 global clock pins](image)

 clk_FsM is the system clock of the modulator.
 clk_FsD is the system clock of the demodulator.

**Figure 2.1 System Clocks of the Modem**

The frequency of clk_Fs is Fs = 10 MHz, the frequency of clk_Fs/2 is 5 MHz, the frequency of clk_Fc is Fc=2.5 MHz and the frequency of clk_Fd is Fd = 625 kHz, where Fd is the symbol frequency.

### 2.3 Digital Complex Mixers, Digital Up Converter

Digital complex mixers are used to obtain frequency translation of the input signal. An ideal digital mixer is a mathematical multiplication of the input data by a sinusoid.

The digital complex mixer illustrated in Figure 2.2 consists of two multipliers that multiply the input by sinusoids of the same frequency but with a difference in phase of $\frac{\pi}{2}$ radians.
Defining $N$ as the ratio of sampling rate to carrier frequency i.e. $N = \frac{F_s}{F_c}$, then for the special case $N = 4$, the two sinusoids are given by

$$\cos\frac{2\pi}{4} n = \begin{cases} 
0 & \text{for } n = 1, 3, 5, 7, \ldots \\
(-1)^n & \text{for } n = 0, 2, 4, 6, \ldots 
\end{cases}$$

$$-\sin\frac{2\pi}{4} n = \begin{cases} 
0 & \text{for } n = 0, 2, 4, 6, \ldots \\
(-1)^{n+1} & \text{for } n = 1, 3, 5, 7, \ldots 
\end{cases}$$

Since the sequence for the sinusoids are only three valued, multipliers can be realized as data selector and two’s complementer as shown in Figure 2.3. In this figure the bold lines are buses that represent the binary numbers. The logic functions shown in Figure 2.3 operate on each bit in the bus.

The input sequence of 8-bit values, where $In_1(n) = \{\ldots Data0, Data1, Data2, Data3, Data4, Data5, Data6, Data7\ldots\}$ multiplied by $\cos2\pi n/4$, gives the progression:

$$In_1(n) = \{\ldots Data0, Data1, Data2, Data3, Data4, Data5, Data6, Data7\ldots\}$$

$$\cos2\pi n/4 = \{\ldots 1, 0, -1, 0, 1, 0, -1, 0 \ldots\}$$

$$out_1(n) = In_1(n) \times (\cos2\pi n/4) = \{\ldots Data0, 0, -Data2, 0, Data4, 0, -Data6, 0 \ldots\}$$
Figure 2. 3 Digital Mixer Realization without Any Multipliers

After In1(n) is clocked into the register reg1, the waveform becomes Q1 as shown in Figure 2.4. The circuit that follows reg1 involves two legs: one entering in Data_a and the other entering in Data_b. Data_a alternates between Q1 and 0. After clk_Fc delays a period of clk_Fs, Fcd is obtained. When Fcd is high, subsequently Data_a is forced to zero. Data_b alternates between 0 and the two's complement of Q1. The bottom leg is forced to zero when Fcd is low. The multiplexer selects Data_a for two samples (one of which is forced to zero) and then Data_b for two samples (one of which is forced to zero). The output B1 of BusMux2, which is synchronized to the falling edge of clk_Fs, produces the sequence:

\[
\text{Out1}(n) = \begin{cases} 
\text{In1}(n), & n = 0, 4, 8, \\
0, & n = 1, 5, 9, \\
\text{In1}(n), & n = 2, 6, 10, \\
0, & n = 3, 7, 11,.
\end{cases}
\]
The operation of the mixer \((\text{In2}(n) \times (-\sin 2\pi n/4))\) is similar to above.

\[
\begin{align*}
\text{clk}_F &= \begin{array}{cccccccc}
\text{Data0} & \text{Data1} & \text{Data2} & \text{Data3} & \text{Data4} & \text{Data5} & \text{Data6} & \text{Data7} \\
\text{Q1} & & & & & & & \\
\text{clk}_F & & & & & & & \\
\text{Fcd} & & & & & & & \\
\text{D}_1 & & & & & & & \\
\text{D}_2 & & & & & & & \\
-\text{D}_2 & & & & & & & \\
\text{D}_3 & & & & & & & \\
\text{D}_4 & & & & & & & \\
\text{B1} & & & & & & & \\
\text{Out1} & & & & & & & 
\end{array}
\end{align*}
\]

Figure 2.4 Simulation Waveforms for a Mixer \(\text{In1}(n) \times (\cos 2\pi n/4)\)

A digital up converter shown in Figure 2.5 consists of a digital complex mixer followed by an adder. Assuming that the input signals \(v_f(n)\) and \(v_q(n)\) are signals with bandwidth less than \(\omega_s\), the output \(v(n)\) of the digital up converter is

\[
v(n) = v_f(n)\cos \frac{2\pi n}{N} - v_q(n)\sin \frac{2\pi n}{N}
\]

\[v(n) = v_f(n)\cos \frac{2\pi n}{N} - v_q(n)\sin \frac{2\pi n}{N}, \quad (2.5)\]

which for bandlimited \(v_f(n)\) and \(v_q(n)\) is the equation for a bandpass signal.

Figure 2.5 Digital up Converter
2.4 Digital Complex Sampling

Digital complex sampling is a technique that directly samples the intermediate frequency (IF) signal to produce the baseband I and Q signals of the complex envelope. By using this technique the analog/digital interface can be moved closer to the antenna.

If the IF local oscillator signal has a frequency difference of $\Delta f$ relative to the received carrier signal, the phase drifts by $\Delta \theta = 2\pi \Delta f T$ between the $k$th and $(k+1)$th symbol, where $k$ is the symbol index. In this thesis, it is assumed that $|\Delta \theta| << \frac{\pi}{4}$ and $\Delta \theta$ is a constant $\Delta \theta_k$ between the $k$th and $(k+1)$th symbol.

Considering an IF signal $x(t)$ with bandwidth $2B$ Hz or $4\pi B$ radians/second that has the form

$$x(t) = x_r(t) \cos(\omega_c t + \Delta \theta_k) - x_q(t) \sin(\omega_c t + \Delta \theta_k); \quad kT \leq t < (k+1)T,$$

where $x_r(t)$ is the in-phase component, $x_q(t)$ is the quadrature-phase component, and $\omega_c = 2\pi F_c$ is the carrier frequency, Figure 2.6 shows a spectrum that qualifies for the signal $x(t)$.

![Figure 2.6 Spectrum of Complex Envelope Signal $x(t)$](image)

A digital complex sampler is illustrated in Figure 2.7. The sampling frequency of the A/D is chosen to be more than twice the RF bandwidth to exceed the Nyquist rate, i.e. $F_s > 2*2B = 4B$. The sampling frequency of the A/D is set at a frequency that makes its
output at bandpass signal centered at $\frac{\pi}{2}$ radians/sample. (The symbol is centered at $\frac{\pi}{2}$)

the absence of frequency offset error referred to earlier as $\Delta f$.) That is the carrier frequency and sampling rate are related by

$$F_c = mF_s \pm \frac{F_s}{4} \quad \text{for some integer } m.$$  \quad (2.7)

The sampling rate for the complex sampler in Figure 2.7 uses $m = 0$ and a positive $\frac{F_s}{4}$ so that $F_s = 4F_c$.

![Figure 2.7 Digital Complex Sampler](image)

The equation for the output of the A/D is

$$x(t) \bigg|_{t = \frac{n}{F_s}} = x(n) = x_I(n)\cos\left(\frac{2\pi}{4} n + \Delta \theta_k\right) - x_Q(n)\sin\left(\frac{2\pi}{4} n + \Delta \theta_k\right).$$ \quad (2.8)

The spectrum of signal $x(n)$ is shown in Figure 2.8. It is a bandpass signal with center frequency $\omega_0 = \frac{\pi}{2}$ radians/sample.
The outputs of the digital complex mixer are given by

\[ M_f(n) = x(n)\cos(\frac{\pi}{4} n) \]
\[ = \frac{1}{2} x_i(n)\cos(\Delta \theta_k) + \frac{1}{2} x_i(n)\cos(\pi n + \Delta \theta_k) - \frac{1}{2} x_q(n)\sin(\Delta \theta_k) - \frac{1}{2} x_q(n)\sin(\pi n + \Delta \theta_k) \]  \hspace{1cm} (2.9)

and

\[ M_q(n) = -x(n)\sin(\frac{\pi}{4} n) \]
\[ = \frac{1}{2} x_i(n)\sin(\Delta \theta_k) - \frac{1}{2} x_i(n)\sin(\pi n + \Delta \theta_k) + \frac{1}{2} x_q(n)\cos(\Delta \theta_k) - \frac{1}{2} x_q(n)\cos(\pi n + \Delta \theta_k) \]  \hspace{1cm} (2.10)

These outputs consist of the lowpass and the high highpass components as illustrated in Figure 2.9.

The lowpass FIR filters shown in Figure 2.7 remove the highpass components.

Assuming these lowpass filters are ideal, the outputs of the complex sampler are given by
\[ R_{x_1}(n) = \frac{1}{2} x_1(n) \cos \Delta \theta_k - \frac{1}{2} x_0(n) \sin \Delta \theta_k \] (2.11)

and

\[ R_{x_2}(n) = \frac{1}{2} x_1(n) \sin \Delta \theta_k + \frac{1}{2} x_0(n) \cos \Delta \theta_k \] (2.12)

It is clear from these equations that the phase offset determines the amount of crosstalk between the I and Q channels. This crosstalk is inherently removed with the differential detection scheme used in \(\pi/4\)-DQPSK demodulation. This will be discussed in more detail later.

Digital complex sampling is very attractive for implementation. It avoids the costly analog mixers and ensures the two oscillators have phase difference of exactly \(\pi/2\). Having the quadrature oscillators is crucial for optimum detection. It is difficult to achieve true quadrature component with analog circuits.

### 2.5 \(\pi/4\)-DQPSK Signal Description

The transmitted signal in \(\pi/4\)-DQPSK has the form

\[ x(t) = \cos(\omega_c t + \Phi(t)) \] (2.13)

where the \(\Phi(t)\) is the phase term that carries the information. The phase term \(\Phi(t)\) is constant over a symbol period \(T\) therefore

\[ x(t) = \cos(\omega_c t + \Phi_k) \quad \text{for } kT \leq t \leq (k+1)T. \] (2.14)

Using trigonometric identities the transmitted signal can be represented as
\[ x(t) = \cos \Phi_k \cos \omega_c t - \sin \Phi_k \sin \omega_c t \]
\[ = I_k \cos \omega_c t - Q_k \sin \omega_c t, \quad (2.15) \]

where \( I_k = \cos \Phi_k \) and \( Q_k = \sin \Phi_k \), are the amplitudes of the inphase and quadrature components of the \( k \)th symbol.

In \( \pi/4 \)-DQPSK modulation, information is transmitted as changes of phase. The phase angle \( \Phi_k \) for the \( k \)th symbol can be expressed as

\[ \Phi_k = \Phi_{k-1} + \Delta \Phi_k, \quad (2.16) \]

where \( \Phi_{k-1} \) is the phase for the \((k-1)\)th symbol and \( \Delta \Phi_k \) is the phase change.

The \( \pi/4 \)-DQPSK modulation carries 2 bits of information per symbol. The data-phase-change correspondence for the two bits of information is shown in Table 2.1.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Gray Code</th>
<th>( \Delta \Phi_k )</th>
<th>Sign bit of ( \sin \Delta \Phi_k )</th>
<th>Sign bit of ( \cos \Delta \Phi_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0</td>
<td>( \pi/4 )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0 1</td>
<td>( 3\pi/4 )</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1 1</td>
<td>( -3\pi/4 )</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1 0</td>
<td>( -\pi/4 )</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Using trigonometric identities or \( I_k = \cos \Phi_k \) and \( Q_k = \sin \Phi_k \), these quantities can be expressed as

\[ I_k = I_{k-1} \cos \Delta \Phi_k - Q_{k-1} \sin \Delta \Phi_k \quad (2.17) \]

and

\[ Q_k = I_{k-1} \sin \Delta \Phi_k + Q_{k-1} \cos \Delta \Phi_k. \quad (2.18) \]

The relationship between \( \Delta \Phi_k \) and the input symbol is given in Table 2.1. The table shows that the sign bits of \( \sin \Delta \Phi_k \) and \( \cos \Delta \Phi_k \) form a Gray code of the symbol value. Equation (2.17) and (2.18) shows that both \( I_k \) and \( Q_k \) are functions of the previous values.
$I_{k-1}$ and $Q_{k-1}$ and the sine and cosine of the $\Delta \Phi_k$ which is defined by the input symbol. If the reference phase at symbol 0 is 0 radians, then $I_k$ and $Q_k$ can take the amplitudes of $\pm 1$, 0 and $\pm \frac{\sqrt{2}}{2}$. If $k$ is odd number, only two levels are possible. They are $\frac{\sqrt{2}}{2}$. If $k$ is even number, three levels are possible. They are $-1,0,$ or $1$.

For the initial state $I_0 = 1$, $Q_0 = 0$ and $\Phi_0 = 0$. If the symbol sent is “3”; then at $k = 1$, $\Delta \Phi_k = -\pi / 4$. From equation (2.17) and (2.18), $I_1 = \frac{\sqrt{2}}{2}$, $Q_1 = -\frac{\sqrt{2}}{2}$. This pattern continues, allowing a total of eight points $(I_k, Q_k)$ for $\pi/4$-DQPSK signal. Figure 2.10 illustrates the $\pi/4$-DQPSK signal constellation.

![Unfiltered $\pi/4$-DQPSK Signal Constellation](image)
2.6 Intersymbol Interference, Matched Filters and Eye Patterns

The phase transitions for π/4-DQPSK constellation shown in Figure 2.10 are instantaneous. These instantaneous transitions result in a large amount of spectral splatter, which requires an unreasonably large channel bandwidth. To limit the spectrum to a reasonable bandwidth, baseband filtering is employed.

Since the baseband filter spreads the symbols in time, it can cause significant intersymbol interference (ISI) in the receiver if it is not done properly. Nyquist’s First Method for elimination of ISI is to use a special overall transfer function \( H_e(f) \), which includes the transmitted and received filters.

If this special function \( H_e(f) \) is a rectangle function defined by

\[
H_e(f) = \frac{1}{R_s} \prod \left( \frac{f}{R_s} \right) = \begin{cases} 
1, & \text{if } |f| \leq \frac{1}{2T}, \\
0, & \text{otherwise}
\end{cases}
\]  

(2.19)

where \( R_s \) is the symbol rate and \( T = \frac{1}{R_s} \), there will be no ISI. The impulse response of \( H_e(f) \) is

\[
h_e(t) = \frac{\sin(\pi R_s t)}{\pi R_s t}.
\]  

(2.20)

Unfortunately, this is the optimum filtering to produce a minimum bandwidth system and is not realizable. \( H_e(f) \) is difficult to approximate because of the steep skirts in the filter transfer function at \( f = 1/2T \). Additionally, the decay in the lobes of the time-domain response is very slow.

The frequency response does not need to be ideal lowpass to eliminate ISI. The requirement for zero ISI is that \( H_e(f) \) be linear phase and satisfy
One filter that satisfies this condition is the raised cosine filter. Its frequency response is given by

\[
H_e(f) = \begin{cases} 
1 & \text{if } 0 \leq f \leq \frac{1-r}{2T} \\
\frac{1}{2} \left[ 1 + \cos \left( \frac{n\pi}{T} \left( f - \frac{1-r}{2T} \right) \right) \right] & \text{if } \frac{1-r}{2T} \leq f \leq \frac{1+r}{2T} \\
0 & \text{if } \frac{1+r}{2T} \leq f 
\end{cases}
\]  

(2.22)

where \( r \) is the rolloff factor.

The 6dB bandwidth of the system is \( R_s/2 \), however the absolute bandwidth of the system is

\[
B = \frac{1+r}{2} R_s .
\]

(2.23)

The symbol rate that can be supported by the system is:

\[
R_s = \frac{2B}{1+r}.
\]

(2.24)

The time response of this transfer function is

\[
h_e(t) = \frac{\sin \left( \frac{T}{T} \right) \cos \left( \frac{n\pi T}{T} \right)}{\frac{t}{T} \left( 1 - 4 \frac{r^2 T^2}{T^2} \right)}.
\]

(2.25)

Figure 2.11 shows the time domain impulse response and frequency responses of the raised cosine filter for \( r = 0, 0.35, 1 \) and \( T = 1 \). This figure shows that the impulse response become more compact as the absolute bandwidth is increased.
A filter with a response that is the square root of equation (2.22) is known as a "square root raised cosine filter with roll-off factor r."

Since optimum detection is accomplished with matched filters, a raised cosine channel characteristic can be implemented with two square root raised cosine (SRRC) filters: one for the transmitting side that determines the transmitted symbol waveform and one for the receiving side that provides matched filter detection.

Since

\[
\sqrt{\frac{1}{2} \left[ 1 + \cos\left( \frac{\pi T}{r} \left( f - \frac{1-r}{2T} \right) \right) \right]} = \cos\left( \frac{\pi T}{2r} \left( f - \frac{1-r}{2T} \right) \right),
\]

(2.26)
the frequency response of a SRRC filter is given by:

\[
H_T(f) = H_R(f) = \begin{cases} 
1 & \text{if } 0 \leq f \leq \frac{1-r}{2T} \\
\cos\left(\frac{\pi T}{2r} \left( f - \frac{1-r}{2T} \right) \right) & \text{if } \frac{1-r}{2T} \leq f \leq \frac{1+r}{2T} \\
0 & \text{if } \frac{1+r}{2T} \leq f 
\end{cases}
\]  

(2.27)

The subscripts \( T \) and \( R \) are used to distinguish between the filter in the transmitter and the one in the receiver.

The impulse response of the SRRC filter is given by:

\[
h_T(t) = h_R(t) = \begin{cases} 
1 - r + \frac{4r}{\pi} & \text{if } t = 0 \\
\frac{r}{\sqrt{2}} \left\{ \left[ 1 + \frac{2}{\pi} \right] \sin \left( \frac{\pi}{4r} \right) + \left[ 1 - \frac{2}{\pi} \right] \cos \left( \frac{\pi}{4r} \right) \right\} & \text{if } t = \pm \frac{T}{4r} \\
\frac{\sin \left[ \pi (1-r) \frac{t}{T} \right] + 4r \frac{t}{T} \cos \left[ \pi (1+r) \frac{t}{T} \right]}{\pi \frac{t}{T} \left[ 1 - \left( \frac{4r \frac{t}{T} \pi}{T} \right)^2 \right]} & \text{otherwise}
\end{cases}
\]

(2.28)

Eye patterns are often employed in the qualitative evaluation of receiver performance. These patterns may be obtained using the BPSK system shown in Figure 2.12. The output of the matched filter in the receiver is fed to the vertical input of an oscilloscope and the symbol clock is fed to the external trigger of the oscilloscope. The transmitted digital signal is recovered by sampling the received analog signal and then making a threshold decision. In the optimal case, the decision point or sampling point is the point where the eye is most open.
The effects of the truncation of filter coefficients and the effects of truncating the filter response can be illustrated by way of example. In this example there is no noise in the channel and a rolloff factor of $r = 0.35$ is used. The structure for the simulation is shown in Figure 2.12. The simulation results are illustrated in Figure 2.13. The sample points are at $t=1$ and $t=2$. In Figure 2.13a) and 2.13b) the filters have 17 coefficients. In Figure 2.13c) and 2.13d) the filters have 25 coefficients. In Figure 2.13a) and 2.13c) the filter coefficients are floating point numbers with the mantissa being 4 digits. In Figure 2.13b) and 2.13d) the filter coefficients are fixed-point 8-bit numbers. The wordlengths of the input and output are 8 bits and wordlength of the internal registers is 16 bits. From this simulation it is shown that the truncation of filter coefficients and impulse response length both introduce ISI. In order to minimize the ISI, the higher order matched filter should be used on the demodulator.
2.7 $\pi/4$-DQPSK Differential Encode and Decode

The input changes the carrier phase to produce the constellation shown in Figure 2.10. If the carrier is at one of the four points in the constellation denoted by $\Theta$, then it shifts to one of the four points denoted by $\otimes$ for the next symbol duration, and vice versa. The carrier always shifts its phase by $m\pi/4$ where $m$ is $\pm 1$ or $\pm 3$, which mean the phase cannot remain the same for two successive symbols. A pair $(I_k, Q_k)$, which represents the $k$th $\pi/4$-DQPSK symbol, has 8 possible states. The value for $I_k$ and $Q_k$ may be precomputed and stored in two 8x8-bit ROMs ($I_k$-ROM and $Q_k$-ROM, see Table 2.2). Two ROMs have the same address and the present address is determined by the input symbol and the previous address (see Table 2.3).
Table 2.2 Two 8x8-bit ROMs: \( I_{k}\_ROM \) and \( Q_{k}\_ROM \)

<table>
<thead>
<tr>
<th></th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>48H</td>
</tr>
<tr>
<td>001</td>
<td>66H</td>
</tr>
<tr>
<td>010</td>
<td>48H</td>
</tr>
<tr>
<td>011</td>
<td>9AH</td>
</tr>
<tr>
<td>100</td>
<td>B8H</td>
</tr>
<tr>
<td>101</td>
<td>B8H</td>
</tr>
<tr>
<td>110</td>
<td>48H</td>
</tr>
<tr>
<td>111</td>
<td>66H</td>
</tr>
</tbody>
</table>

LEGEND

\[ \frac{x}{2} \rightarrow 48H; \quad -\frac{x}{2} \rightarrow B8H; \quad 1 \rightarrow 66H; \quad -1 \rightarrow 9AH; \]

Table 2.3 Relationship Between the Present Address and the Input Symbol and the Previous Address

<table>
<thead>
<tr>
<th>Input Symbol</th>
<th>Previous Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>000</td>
</tr>
<tr>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>01</td>
<td>011</td>
</tr>
<tr>
<td>10</td>
<td>111</td>
</tr>
<tr>
<td>11</td>
<td>101</td>
</tr>
</tbody>
</table>

A block diagram of the \( \pi/4\)-DQPSK differential encoder is shown in Figure 2.14.

Figure 2.14 A Block Diagram of the \( \pi/4\)-DQPSK Differential Encoder

The S/P block converts the serial data stream TxData into di-bit symbols. These symbols are translated to Gray code so that the codewords representing the adjacent phase differ by one bit. The Address_ROM is a 32x3-bit ROM. Its addresses \( A4 \) and \( A3 \) are from the Gray encoder. After the 3-bit data outputs of Address_ROM are delayed by
a symbol period, they are fed back to the addresses A2~A0. At the same time, these 3-bit
data outputs provide the addresses of \( I_k_{\text{ROM}} \) and \( Q_k_{\text{ROM}} \) (both of them are 8x8-bit
ROM). The \( I_k_{\text{ROM}} \) and \( Q_k_{\text{ROM}} \) produce 8-bit values for \( I_k \) and \( Q_k \) that produce the
proper \( \Delta \Phi_k \).

A block diagram of the differential detection circuit found in the receiver is shown in
Figure 2.15.

![Block Diagram of Differential Detection](image)

**Figure 2.15 Block Diagram of Differential Detection**

Since the receiver does not use a coherent reference, the reference used in the receiver
will be offset from the reference phase used in the transmitter. The received values of \( IR_k \)
and \( QR_k \) are then given by

\[
IR_k = \cos(\Phi_k + \Delta \theta_k) \tag{2.29}
\]

and

\[
QR_k = \sin(\Phi_k + \Delta \theta_k). \tag{2.30}
\]

where \( \Delta \theta_k \) was defined in the section 2.4. The points in the constellation shown in
Figure 2.10 can be expressed using complex notation with \( I \) being the real axis and \( Q \)
being imaginary axis. The complex number representing \( IR_k \) and \( QR_k \) is given by

25
\[ s_R(k) = IR_k + jQR_k = e^{j(\Phi_k + \Delta \theta_k)}. \] (2.31)

Taking the complex conjugate of \( s_R(k) \) gives

\[ s^*_R(k) = IR_k - jQR_k = e^{-j(\Phi_k + \Delta \theta_k)}. \] (2.32)

The circuit shown in Figure 2.15 executes a complex multiplication

\[
D(k) = s_R(k)s^*_R(k-1)
= e^{j(\Phi_k + \Delta \theta_k)}e^{-j(\Phi_{k-1} + \Delta \theta_{k-1})}
= e^{j(\Delta \Phi_{k-1} + \Delta \theta_{k-1})}.
\] (2.33)

Since \( |\Delta \theta_k - \Delta \theta_{k-1}| = |\Delta \theta| \ll \frac{\pi}{4} \) (assumption made in the section 2.4) and \( (\Phi_k - \Phi_{k-1}) \) is a multiple of \( \frac{\pi}{4} \), \( |\Delta \theta| \) is much less than the step size for phase change and the following equation is obtained

\[
\Phi_k - \Phi_{k-1} + (\Delta \Phi_k - \Delta \Phi_{k-1}) \approx \Phi_k - \Phi_{k-1}.
\] (2.34)

Equation (2.34) shows that a carrier frequency offset, which causes \( \Delta \theta_k - \Delta \theta_{k-1} \neq 0 \), will introduce error. This is why carrier frequency offset causes the bit error rate (BER) degradation.

Substituting equation (2.34) into equation (2.33) gives

\[
D(k) = e^{j\Delta \Phi_k}
= \cos \Delta \Phi_k + j \sin \Delta \Phi_k,
\] (2.35)

where \( D(k) = D_1(k) + jD_Q(k) \) with \( D_1(k) \) and \( D_Q(k) \) as shown in Figure 2.15. Thus, \( D_1(k) = \cos \Delta \Phi_k \) and \( D_Q(k) = \sin \Delta \Phi_k \).

Equation (2.33) and Equation (2.35) show that the result of the delay and complex multiplication is a function of the difference between \( \Phi_k \) and \( \Phi_{k-1} \), which are the phase angles of the \( k \)th and \((k-1)\)th symbols respectively.
From the Table 2.1 it is seen that the sign bits of $D_i(k)$ and $D_q(k)$ are the Gray codewords of the transmitted information symbol.

### 2.8 Sampling Rate Conversion

The algorithms used in the DSP-based modem incorporate sampling rate conversion. The sampling rate will be decreased at one point in the system and then increased later. If the sampling rate is considerably greater than twice the bandwidth of a signal, decreasing the sampling rate of the signal can be useful. The reduction of the sampling rate is called decimation. Decimation consists of an anti-aliasing filtering followed by downsampling.

If a baseband signal is to modulate a digital carrier signal, the resulting passband signal has twice the bandwidth and therefore requires twice the sampling rate. Increasing the sampling rate of a signal is called interpolation. Interpolation consists of upsampling, followed by an anti-imaging filtering.

Systems that employ multiple sampling rates in the processing of digital signals are called multirate digital signal processing (DSP) systems. In multirate DSP systems, different sample rates are used within a system to achieve the most efficient computation at each stage.

#### 2.8.1 Downsampling

The signal flow representation of a downsampler is shown in Figure 2.16.
The sampling rate of a signal $x(n)$ is reduced by a factor $M$ by taking only every $M$th value of the signal. The output of the downsampler of Figure 2.16 is given by

$$y_i(m) = x(m \cdot M + \lambda), \quad \lambda = 0, 1, 2, ... M - 1,$$

where the constant $\lambda$ is the phase offset.

The spectrum of $y_i(m)$ is given by

$$Y_i(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} X(e^{j(\omega - 2\pi k)/M}) e^{-j2\pi k M/M}.$$  \hspace{1cm} (2.37)

Equation (2.37) indicates that the phase offset $\lambda$ shows up in the spectrum as complex exponential factors with a magnitude of 1. If the spectrum $X(e^{j\omega})$ is not band-limited to the interval $[0, \pi/M]$, there is overlapping of the sum terms in Equation (2.37). This aliasing can’t be removed by a filter used after downsampling. Therefore an anti-aliasing filter must be used before downsampling to limit the bandwidth of $x(n)$ to at most $\pi/M$.

Figure 2.17 shows a possible realization of a downsampler $M$. 

**Figure 2.17 A Possible Realization of a Downsampler M**
The frequency of the clock signal clk is $f$ and the sampling rate of the signal $x(n)$ is also $f$. The register $reg1$ is used to synchronize $x(n)$ to the signal clk. The counter operates at the input clock frequency $f$ divided by $M$. The frequency of the $reg2$ clock is $f/M$ and shifted by a period of clk. The register $reg2$ takes only every $M$th value of the signal $x(n)$, so the sampling rate of $y(m)$ is $f/M$.

### 2.8.2 Upsampling

Upsampling is accomplished by inserting zeros between samples. The signal flow representation of a upsampler is shown in Figure 2.18.

![Figure 2.18 Upsampler](image)

The sampling rate of $x(n)$ is $L$ times that of $y(m)$. The output of the upsampler is given by

$$x(n) = \begin{cases} 
  y(n/L) & \text{for } n = mL, m \text{ integer}, \\
  0 & \text{otherwise}.
\end{cases} \quad (2.38)$$

The spectrum of $x(n)$ can be found from its $Z$ transform. The $z$-transform of $x(n)$ is given by

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}. \quad (2.39)$$

Since $x(n)$ is zeros for $n \neq mL$, where $m$ is an integer, we obtain
\[ X(z) = \sum_{m=-\infty}^{\infty} x(mL)z^{-mL} \]
\[ = \sum_{m=-\infty}^{\infty} y(m)(z^L)^{-m} \]
\[ = Y(z^L) \quad (2.40) \]

Setting \( z = e^{j\omega} \), the equation (2.40) will become

\[ X(e^{j\omega}) = Y(e^{jL\omega}). \quad (2.41) \]

As an example, the spectrum after upsampling with \( L=2 \) is shown in Figure 2.19.

\[ \text{Figure 2.19 Spectrum after Upsampling by 2} \]

The spectrum of \( x(n) \) is just that of \( y(m) \) scaled and repeated. The spectrum of \( x(n) \) first repeats itself at \( \omega = \frac{2\pi}{L} \). The repeated spectrum is called an image. Going from the upsampled signal to the ideally interpolated signal can be interpreted in the frequency domain as the removal of the image spectrum. To do this, a low-pass filter after upsampling should be employed. This filter is called an anti-imaging filter and should have a cutoff frequency \( \omega_c = \frac{\pi}{L} \) with a sampling rate \( \omega = 2\pi \). The anti-imaging filter
must have a gain of $L$ to keep the spectrum of the filter’s output as large as the original signal $y(m)$.

A diagram for an upsampler with $L=4$ is shown in Figure 2.20a). The component $C$ is a 2-bit counter and its outputs are decoded by a gate $\text{And1}$. The output of $\text{And1}$ is asserted during a corresponding clock period of $\text{clk}_F$. The output of $\text{And1}$ may contain “glitches” on state transitions where one or two bits change, even though the outputs of $C$ are glitch free and $\text{And1}$ does not have any static hazards. To “clean up” the glitches, the output of $\text{And1}$ is connected to the register $\text{reg2}$ that samples the stable output of $\text{And1}$ on the next clock tick of $\text{clk}_F$. In the four clock periods of $\text{clk}_F$, the output $\text{upclr}$ of $\text{reg2}$ is high for a clock period of $\text{clk}_F$.

![a) Logic Diagram of Upsampling by a Factor of 4](image)

![b) Simulation Waveform of Upsampling by a Factor of 4](image)

Figure 2.20 An Upsampler with $L=4$
The signal clk_Fd is from a counter that operates at input clock clk_Fc divided by 4 (see section 2.2). The input signal \( y(m) \) with the sampling rate \( F_d \) is synchronized to the positive-going edge of clk_Fd by a register reg1. When upclr is low, data Q1 from reg1 is forced to zero by a gate And2. So, in a Fd period, the output of And2 keeps the value of data Q1 for a Fc period, then it is set to zeros for three Fc periods. The register reg3 is employed to synchronize the output of And2 to the positive-going edge of clk_Fc. The sampling rate of \( x(n) \) is increased by 4 by placing 3 equally spaced zeros between each pair of samples of \( y(m) \). The simulation waveforms for upsampling by 4 in MAX+PLUS II are shown in Figure 2.20 b).
3. Architecture and Model of a DSP-Based π/4-DQPSK Modem

3.1 Introduction

A modulation-demodulation (Modem) is traditionally implemented with analog circuitry. Unfortunately, it becomes difficult and expensive to implement modems with complex modulation formats and high data rates using analog hardware because the overall performance becomes very sensitive to various problems, such as dc offset voltages, dc voltage drifts, analog filter phase distortions, quadrature phase and gain imbalance, and amplifier and mixer nonlinearities[3]. However, with DSP implementations, virtually all of these distortions are eliminated and very precise and controllable performance can be achieved without sophisticated compensation techniques. Other advantages of DSP implementations include the ability to easily program the hardware to accommodate different data rates, modulation formats and filter specifications.

A conceptual implementation block diagram of π/4-DQPSK modulator architecture is shown in Figure 3.1. The binary data are first converted to 2–bit data in parallel. The data are then differentially encoded to a symbol $(t_k, q_k)$. In baseband, the signals are filtered by SRRC filters. The bandlimited signals are used to quadrature modulate a carrier.
Figure 3.1 Block Diagram of π/4-DQPSK Modulator Architecture

Figure 3.2 shows two schemes of differential detection of π/4-DQPSK signal. The differential detector is a nonlinear operation. For the differential detector to be ISI-free, the matched ISI-free SRRC filtering must be performed before this nonlinear operation [4]. The advantage of the conventional IF band differential detection scheme shown in Figure 3.1a) is that no local oscillator (LO) is needed. A SRRC bandpass filter is employed to match the transmitted signals. Figure 3.1b) shows the baseband differential detection. Two SRRC lowpass filters are employed to match the transmitted signals. The LO is assumed to have the same frequency as the unmodulated carrier but with a constant phase difference [5]. This scheme provides alternative efficient and low-power solutions. In schemes of the conventional IF band differential detection, the SRRC BPF is required to have a SRRC amplitude and linear phase response. It is impractical to design and implement a narrow-band with such a stringent specification [6]. One solution is to employ the raised cosine filtering in the modulator. If the raised cosine filtering is performed in the modulator, a maximum-flat narrow-band filter can be used as the demodulator BPF. In this case, the noise bandwidth is wider than the minimum Nyquist bandwidth. In scheme of the baseband differential detection, the design challenge lies on the design of the LO. The frequency offset between the LO and the unmodulated carrier causes BER degradation [1][5]. These three schemes of differential
detection of π/4-DQPSK signals are proven to be equivalent in BER performance [4] [6].

a) Block Diagram of the IF Band Differential π/4-DQPSK Demodulator

b) Block Diagram of the Baseband Differential π/4-DQPSK Demodulator

Figure 3.2 Schemes of Differential Detection of π/4-DQPSK Signal

In this chapter, the architecture and model of a DSP-based π/4-DQPSK modem will be described and the system-level design will be also considered.

3.2 Architecture and Model of a DSP-Based π/4-DQPSK Modem

The A/D or/and D/A boundary between the analog processing and digital processing traditionally has been at baseband. In current architectures for DSP-based π/4-DQPSK modems, the A/D and D/A converters operate on the IF signal. This increase in the frequency of operation is due the increased speeds of ICs and the development of multirate DSP technologies. Figure 3.3 illustrates the architecture of a DSP-based π/4-DQPSK modem.
Figure 3.3 DSP-Based $\pi/4$-DQPSK Modem

Legend
HB: Half-band filter
SRRRC: Square Root Raised Cosine
TxData: the Transmitted Serial Data
TxClock: the Synchronous Clock of TxData
RxData: the Recovered Serial Data
RxClock: the Recovered bit Clock
DataClock: the Recovered Symbol Clock

THE SHADOWY FUNCTIONS ARE OUT OF THIS THESIS.
3.2.1 Modulator

The structure of the $\pi/4$-DQPSK modulator is shown in Figure 3.3a). The S/P block converts the serial data stream into 2-bit data. These data bits are differentially encoded and mapped into a pair $(I_k, Q_k)$, which represents a $\pi/4$-DQPSK symbol. While the differential encoding simplifies demodulation, it often results in the loss of a pair of symbols when one symbol is severely corrupted with noise. In a Rayleigh fading channel this translates to approximately a 3-dB loss in $E_b/N_0$ relative to coherent $\pi/4$-QPSK [1].

Before the pair $(I_k, Q_k)$ is filtered, its sampling rate must be increased in order that Nyquist's Sampling Theorem is not violated by the modulation process. After upsampling by a factor of 4, the pair $(I_k, Q_k)$ is shaped using two independent square root raised cosine (SRRC) filters. The filters generate the real and imaginary parts of the complex bandlimited baseband signal:

$$u_r(m) = \sum_{k=-\infty}^{\infty} I_k h_r(mT/4 - kT)$$

and

$$u_q(m) = \sum_{k=-\infty}^{\infty} Q_k h_r(mT/4 - kT),$$

where $T$ is the symbol interval, $u_r(m)$ and $u_q(m)$ are sampled 4 times per symbol, and $h_r(mT/4)$ is obtained by sampling $h_r(t)$ at points $mT/4$. The subscript $r$ in $h_r(t)$ is used to indicate the filter is located in the modulator (Transmitter). The spectral shape of the frequency response of $h_r(t)$ is a square root raised cosine. $h_r(t)$ is often referenced as a pulse shaping filter (for more details see section 2.6).
After SRRC filtering, one bit of information is modulated with four periods of a sinusoidal carrier and the ratio of sampling frequency to carrier frequency is 4. So, the sampling rate at the inputs of the digital complex mixer is 16 samples/symbol. The SRRC filter output must be increased 4 times to satisfy the requirement of sampling rate at the inputs of the digital complex mixer. The realization of an upsampling of 4 is accomplished with dyadic cascading. There are two stages with each stage upsampling by a factor of 2. A half-band filter is employed as an anti-imaging filter. After upsampling by a factor of 4 and assuming this upsampler is followed with an ideal anti-imaging filter, equation (3.1) and equation (3.2) become:

\[ v_I(n) = \sum_{k=-\infty}^{\infty} I_k h_r(nT/16 - kT) \]  \hspace{1cm} (3.3)

and

\[ v_Q(n) = \sum_{k=-\infty}^{\infty} Q_k h_r(nT/16 - kT). \]  \hspace{1cm} (3.4)

The digital complex mixer generates the quadrature intermediate frequency (IF) signal \( v(n) \):

\[ v(n) = v_I(n)\cos(2\pi \frac{n}{4}) - v_Q(n)\sin(2\pi \frac{n}{4}). \]  \hspace{1cm} (3.5)

This signal is converted to an analog signal using an 8-bit D/A converter followed by a reconstruction filter. The analog signal can be expressed by

\[ v(t) = v_I(t)\cos(2\pi Fc t) - v_Q(t)\sin(2\pi Fc t), \]  \hspace{1cm} (3.6)

where \( Fc \) is the intermediate carrier frequency,

\[ v_I(t) = \sum_{k=-\infty}^{\infty} I_k h_r(t) \]  \hspace{1cm} (3.7)

and
The analog IF signal \( \nu(t) \) is translated to the appropriate RF channel for transmission.

3.2.2 Demodulator

The demodulator with the baseband differential detection shown in Figure 3.3b) performs the inverse function of the modulator. The RF receiver provides the tuning, amplification and mixing necessary to produce the IF signal \( x(t) \). The carrier frequency recovery is an essential task of the demodulator. The local oscillator that down converts the radio signal to an IF signal must be controlled to get an accurate IF carrier frequency. The reason is that digital timing and intermediate carrier frequency offset estimation algorithms work properly only in the presence of a small residual frequency offset [7]. The carrier frequency offset estimation algorithm is out of the scope of this thesis. In this thesis it is assumed that the RF receiver produces an IF signal with a nearly coherent frequency. If the local oscillator signal has a frequency difference of \( \Delta f \) relative to the received carrier signal, the phase drifts \( \Delta \theta = 2\pi \Delta f T \) from symbol to symbol. This phase drift causes BER degradation [1][5].

The IF signal produced by the receiver can be expressed as:

\[
x(t, \epsilon) = x_i(t, \epsilon) \cos(2\pi F_c t + \Delta \theta_i) - x_q(t, \epsilon) \sin(2\pi F_c t + \Delta \theta_q) + n(t),
\]

where \( \Delta \theta_k = 2\pi \Delta f T \) as defined in section 2.4, \( \epsilon \) is unknown but known to be a slowly changing parameter, \( n(t) \) is assumed as zeros-mean white Gaussian noise, while \( x_i(t, \epsilon) \) and \( x_q(t, \epsilon) \) may be expressed as:
\[ x_i(t, \varepsilon) = \sum_{i=-\infty}^{\infty} I_i h_i(t - iT - \varepsilon T) \]  \hspace{1cm} (3.10) \\

and \\

\[ x_q(t, \varepsilon) = \sum_{i=-\infty}^{\infty} Q_i h_i(t - iT - \varepsilon T) . \]  \hspace{1cm} (3.11) \\

The digital complex sampler generates digital baseband \( R_x(n) \) and \( R_x(n) \) with a sampling rate of 16 samples per symbol. According to the discussion in section 2.4, these samples may be expressed as:

\[ R_x(n, \varepsilon) = \frac{1}{2} x_i(n, \varepsilon) \cos \Delta \theta_i - \frac{1}{2} x_q(n, \varepsilon) \sin \Delta \theta_i + n_i(n) \]  \hspace{1cm} (3.12) \\

and \\

\[ R_x(n, \varepsilon) = \frac{1}{2} x_i(n, \varepsilon) \sin \Delta \theta_i + \frac{1}{2} x_q(n, \varepsilon) \cos \Delta \theta_i + n_q(n) , \]  \hspace{1cm} (3.13) \\

where \\

\[ x_i(n, \varepsilon) = \sum_{i=-\infty}^{\infty} I_i h_i(nT/16 - iT - \varepsilon T) \]  \hspace{1cm} (3.14) \\

and \\

\[ x_q(n, \varepsilon) = \sum_{i=-\infty}^{\infty} Q_i h_i(nT/16 - iT - \varepsilon T) . \]  \hspace{1cm} (3.15) \\

In order to reduce the size of the SRRC filters in the demodulator, the sampling rate of signal \( R_x(n) \) and \( R_x(n) \) is reduced to 4 samples per symbol prior to being filtered by the SRRC filters. The downsampling is accomplished with dyadic cascading two stages, each downsampling by a factor of 2. The bandwidth of the wanted signal component at the output of the digital complex sampler must be limited to less than \( \pi/2 \) since this signal is downsampled. A half-band filter is employed as an anti-aliasing filter.
Assuming the sampling rate conversion is perfect, then
\[ x_{id}(m, \varepsilon) = \frac{1}{2} x_i(4m, \varepsilon) \cos \Delta \theta + \frac{1}{2} x_q(4m, \varepsilon) \sin \Delta \theta + n_i(m) \quad (3.16) \]
and
\[ x_{iq}(m, \varepsilon) = \frac{1}{2} x_i(4m, \varepsilon) \sin \Delta \theta + \frac{1}{2} x_q(4m, \varepsilon) \cos \Delta \theta + n_q(m), \quad (3.17) \]

where \(n_i(m)\) and \(n_q(m)\) are noise components at the outputs of the downsamplers.

\(x_i(4m, \varepsilon)\) and \(x_q(4m, \varepsilon)\) are given by
\[ x_i(4m, \varepsilon) = \sum_{i=-\infty}^{\infty} I_i h_r(mT/4 - iT - \varepsilon T) \quad (3.18) \]
and
\[ x_q(4m, \varepsilon) = \sum_{i=-\infty}^{\infty} Q_i h_r(mT/4 - iT - \varepsilon T). \quad (3.19) \]

Then \(x_{id}(m, \varepsilon)\) and \(x_{iq}(m, \varepsilon)\) are filtered by the matched filters, which have a gain of 2.

The inphase and quadrature matched filters have the same responses \(h_r(nT/4)\) and produce:
\[ x_{im}(m, \varepsilon) = 2x_{id}(m, \varepsilon) * h_r(mT/4) \]
\[ = 2 \sum_{i=-\infty}^{\infty} x_{id}(i, \varepsilon) h_r(mT/4 - iT/4) + n_i(m) \quad (3.20) \]
\[ = 2 \sum_{i=-\infty}^{\infty} x_{id}(i, \varepsilon) h_r(iT/4 - mT/4) + n_i(m) \]
and
\[ x_{qm}(m, \varepsilon) = 2 \sum_{i=-\infty}^{\infty} x_{iq}(i, \varepsilon) h_r(iT/4 - mT/4) + n_q(m), \quad (3.21) \]

where \(n_i(m)\) and \(n_q(m)\) are the noise components at the outputs of the inphase and quadrature matched filters.
Substituting equation (3.16) into equation (3.20) and equation (3.17) into equation (3.21) gives

\[ x_{\text{in}}(m, \varepsilon) = \sum_{i=-\infty}^{\infty} x_i(i, \varepsilon) h_{\tau}(iT/4 - mT/4) \cos \Delta \theta_k \]
\[ - \sum_{i=-\infty}^{\infty} x_i(i, \varepsilon) h_{\delta}(iT/4 - mT/4) \sin \Delta \theta_k + n_i(m) \]  
(3.22)

and

\[ x_{\text{on}}(m, \varepsilon) = \sum_{i=-\infty}^{\infty} x_i(i, \varepsilon) h_{\tau}(iT/4 - mT/4) \sin \Delta \theta_k \]
\[ + \sum_{i=-\infty}^{\infty} x_i(i, \varepsilon) h_{\delta}(iT/4 - mT/4) \cos \Delta \theta_k + n_i(m). \]  
(3.23)

Substituting equation (3.14) into equation (3.22) and equation (3.15) into equation (3.23) gives

\[ x_{\text{in}}(m, \varepsilon) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I_{ij} h_{\tau}(iT/4 - jT - \varepsilon T) h_{\delta}(iT/4 - mT/4) \cos \Delta \theta_k \]
\[ - \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} Q_{ij} h_{\tau}(iT/4 - jT - \varepsilon T) h_{\delta}(iT/4 - mT/4) \sin \Delta \theta_k + n_i(m) \]  
(3.24)

\[ = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I_{ij} h_{\tau}(iT/4 - jT - \varepsilon T) h_{\delta}(iT/4 - mT/4) \cos \Delta \theta_k \]
\[ - \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} Q_{ij} h_{\tau}(iT/4 - jT - \varepsilon T) h_{\delta}(iT/4 - mT/4) \sin \Delta \theta_k + n_i(m) \]

and

\[ x_{\text{on}}(m, \varepsilon) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I_{ij} h_{\tau}(iT/4 - jT - \varepsilon T) h_{\delta}(iT/4 - mT/4) \sin \Delta \theta_k \]
\[ + \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} Q_{ij} h_{\tau}(iT/4 - jT - \varepsilon T) h_{\delta}(iT/4 - mT/4) \cos \Delta \theta_k + n_i(m). \]  
(3.25)

Letting

\[ p(mT/4 - jT - \varepsilon T) = \sum_{i=-\infty}^{\infty} h_{\tau}(iT/4 - jT - \varepsilon T) h_{\delta}(iT/4 - mT/4) \]  
(3.26)
and changing the variable \( i \) to variable \( j \) through the substitution 
\[
jT/4 = iT/4 - IT - \varepsilon T,
\]
and changing the variable \( m \) to variable \( w \) through the substitution 
\[
wT/4 = mT/4 - IT - \varepsilon T,
\]
equation (3.26) becomes
\[
p(wT/4) = \sum_{j=-\infty}^{\infty} h_r(jT/4) h_r(jT/4 - wT/4)
= \sum_{j=-\infty}^{\infty} h_r(jT/4) h_r(wT/4 - jT/4)
= h_r(wT/4) * h_r(wT/4)
\]
(3.27)

According to the discussion in section 2.6, \( p(wT/4) \) is a raised cosine function, which is real and symmetric.

Substituting equation (3.26) into equation (3.24) and equation (3.25) yields
\[
x_{im}(m, \varepsilon) = \sum_{l=-\infty}^{\infty} I_l p(mT/4 - IT - \varepsilon T) \cos \Delta \theta_k
- \sum_{l=-\infty}^{\infty} Q_l p(mT/4 - IT - \varepsilon T) \sin \Delta \theta_k + n_i(m)
\]
(3.28)

and
\[
x_{Qm}(m, \varepsilon) = \sum_{l=-\infty}^{\infty} I_l p(mT/4 - IT - \varepsilon T) \sin \Delta \theta_k
+ \sum_{l=-\infty}^{\infty} Q_l p(mT/4 - IT - \varepsilon T) \cos \Delta \theta_k + n_q(m).
\]
(3.29)

The timing recovery function extracts the timing information from the outputs of the matched filters. It provides the clock signals for the two downsamplers so that they are able to choose the optimal sample point from the output of each matched filter. If the matched filters are ideal and timing recovery is perfect, there will be no ISI. The outputs of the downsamplers are the recovered \( I \) and \( Q \) signal which are given by:
\[ I_k(k) = I_k \cos \Delta \theta_k - Q_k \sin \Delta \theta_k \]  
(3.30)

and

\[ Q_k(k) = I_k \sin \Delta \theta_k + Q_k \cos \Delta \theta_k \]  
(3.31)

where \( I_k \) and \( Q_k \) are the in-phase and quadrature components of the signal corresponding to the \( k \)th symbol.

Next, \( I_R(k) \) and \( Q_R(k) \) are differentially detected by a delay-and-multiply process. The phase difference between consecutive symbols is derived by performing a complex multiplication of the quadrature values of the current symbol and the conjugate of the previous symbol as explained in section 2.7.

The data formatter formats the recovered data. The data is hard-limited by using the sign of \( D_I(k) \) and \( D_Q(k) \). The output, \( \text{RxData} \), is a serial data stream, which is synchronized to the clock, \( \text{RxClock} \).

### 3.3 System-Level Considerations

The D/A converter and A/D converter are important components of the system. They can limit the overall system performance. An unavoidable side effect of the D/A converter is the introduction of \( \sin(x)/x \) amplitude distortion into the transmit spectrum. Thus, a \( x/\sin(x) \) compensation filter is required to equalize the \( \sin(x)/x \) frequency response roll-off of the D/A converter.

As shown in Figure 3.3b), the A/D converter is placed at the IF front-end of the demodulator where the data rate is \( 16/T \). For example, a symbol rate of 625 kBd, the A/D converter must operate at 10 MHz.
The simulations [3] show that the ISI SNR will be 12 dB down from 43.7 dB when a pair of 8-bit D/A and A/D converters decreases to the 6 bits. In this thesis, a pair of 8-bit D/A and A/D converters was used.

In the demodulator the SRRC filters determine the bandwidth of a π/4-DQPSK signal and control the ISI. According to equation (2.23), the bandwidth of the π/4-DQPSK signal for a symbol rate of 625 kBd is

$$B_s = 2B = R_s(1 + r) = 625 \times (1 + 0.35) = 843.75 \text{kHz}.$$  

The spectral efficiency is

$$\eta = \frac{\text{Bit Rate}}{\text{Bandwidth}} = \frac{625 \times 2}{843.75} = 1.48 \text{ bits/Hz}$$

The tolerable levels of ISI within the overall modem are an important factor in determining the input, internal registers and output wordlengths of the SRRC filters as well as the D/A and A/D wordlengths.

In section 2.6 the relationship between the ISI and SRRC filters was discussed. In that simulation the matched filters were cascaded with each other for the baseband signal and didn’t incorporate sampling rate conversion, modulation and demodulation. Due to the finite stopband attenuation in practical matched filter designs, the sampling rate conversion, modulation, and demodulation will introduce additional ISI.

The following simulations are used to simulate the effects of the length of the impulse response of the SRRC filter on the ISI. They also incorporate the effects of the sampling rate conversion and the quadrature modulation and demodulation. In the simulations, the modulator and demodulator are connected back-to-back, thereby simulating an ideal distortionless channel and also perfect carrier and clock recovery. For these simulations, 8 bits were used for the output wordlength of the modulator and for the input wordlength.
of the demodulator and for the SRRC filters, the input and output wordlengths are 8 bits while the wordlength of the internal registers is 16 bits.

The wordlengths of the inputs, outputs and internal registers of the other filters (the half-band filters) are the same as the SRRC filters.

Figure 3.4a) and Figure 3.5a) show the power spectrum at the output of the digital up converter in the modulator while Figure 3.4b) and Figure 3.5b) show the eye pattern at the I channel output of the matched filter in the demodulator. In Figure 3.4) the impulse length for the SRRC filter is 25. The ISI SNR is about 25 dB. Figure 3.5 shows the result of a same simulation with the impulse length for the SRRC filter shortened to 17. As a result the ISI SNR degrades from 25 dB to 18 dB. These results clearly indicate that very good ISI performance can be achieved without imposing unreasonable requirements on the larger impulse length of the SRRC filters.

![Figure 3.4 Simulation Result with 25-tap SRRC Filters](image)

Figure 3.4 Simulation Result with 25-tap SRRC Filters
In this thesis, the modulator and demodulator will be designed using 17-tap SRRC filters with a wordlength of 8 bits for input and output and a wordlength of 16 bits for the internal registers.

Figure 3.5 Simulation Result with 17-tap SRRC Filters
4. Efficient FIR Filter Design

4.1 Introduction

In the following chapter, the structures of the SRRC filter, halfband filter and interpolation/decimation with polyphase filter are derived, and the design details are also presented. Some techniques to reduce the implementation cost (or the number of logic cells used) are described.

4.2 Linear Phase FIR Filter

The FIR filter which has an impulse response sequence \( h(0), h(1), \ldots, h(N-1) \) has \( N \) coefficients and is said to be an \( N \) th order filter. The transfer function for this filter is:

\[
H(z) = h(0) + h(1)z^{-1} + \cdots + h(N-1)z^{-(N-1)}
\]  

(4.1)

The output of the system shown in Figure 4.1 is described by the equation

\[
y(n) = \sum_{k=0}^{N-1} h(k)s(n-k)
\]  

(4.2)

![Figure 4.1 Description of a FIR Filter](image)

The signal flow graph for the filter given by equation (4.2) is shown in Figure 4.2.

This structure is called a direct form of a FIR filter.
When the impulse response of a FIR filter satisfies \( h(n) = h(N-1-n) \), the filter is a symmetrical FIR filter and has a linear phase response. This symmetry allows the symmetric coefficients to be added together before they are multiplied by the coefficients. Taking advantage of the symmetry lowers the number of multipliers from \( N \) to \( N/2 \) if \( N \) is even and from \( N \) to \( (N+1)/2 \) for \( N \) odd. Since multipliers are expensive in terms of the number of logic cells they require, reducing the number of multiplier reduces the circuitry required to implement the filter. Figure 4.3 shows the transversal structure for a symmetric FIR filter with an odd number \( N \) of coefficients.
4.3 Silicon Saving Techniques and Pipelining

4.3.1 Canonic Signed Digit Multiplier

To save silicon size (or logic cells), the multiplication of a signal with a fixed coefficient is accomplished with shifters and adders or subtractors (the cost of a subtractor is roughly the same as an adder). In the simple shift and add strategy adders are required where the binary digits are one. They are not required where the binary digit is a zero. The number of adders can be reduced by using a method based on the canonic signed digit (CSD) expression of the multiplicand [8].

The number of adders or subtractors used by this realization structure is not necessarily the minimum if the multiplicand is larger than 44. For example, 75 can be expressed as $75 = 2^6 + 2^3 + 2 + 1$. Multiplication of $x$ by 75 based on CSD can be done with three shifters and three adders as shown in Figure 4.4. A shifter accomplishes the multiplications of $2^6, 2^3$ and 2. The shifter incurs no cost since it is a hardwired.

![Figure 4.4 Multiplication of $x$ with 75 Based on CSD](image)

If the number 75 is represented as $75 = (2^2 + 1)(2^4 - 1)$, then a multiplication by 75 is performed by the architecture shown in Figure 4.5. Multiplication of $x$ by 75 can be done with just two shifters and two adders. A method to find the canonic signed digit form with the minimum number of add or subtract (MNA) operations was developed by D.Li [9].
4.3.2 Truncation

The precision of a number depends on the number of bits used to represent it. Since the input signal is already corrupted with at least some noise, it is not necessary to represent the input with infinite precision. The number of bits used to represent the input should be high enough so that the SNR is not dropped below a critical threshold. Thus the A/D converter must be of reasonable precision, which is 8-bit for the modem implemented in this thesis.

The precision of representation in registers inside the modem changes. Multiplication by an 8-bit constant gives a 16-bit result. Addition of two 8-bit numbers gives a 9 bit result. It is easy to see that as the signal is processed the precision of representation grows. Since the added precision is unnecessary and very costly, the precision is reduced after multipliers resulting in errors being generated. The precision will be reduced by truncation, that is, the dropping of the least significant bits.

4.3.3 Pipelining

It is called pipelining to insert flip-flops between the complex blocks of combinatorial logic. The longest delay path from the output of any register to the input of the register it feeds must be less than the register clock speed. If the delay path is too long, pipelining
should be employed. It usually has little impact on the size and cost of the design, but at the same time it lowers the propagation delay between registers and increases the speed of FIR filter. The FLEX architecture has a flipflop in each logic cell. Therefore, an adder and a register require only one logical cell per bit.

4.4 Implementing the Square Root Raised Cosine Filter
4.4.1 Square Root Raised Cosine Filter Design in Matlab

A filter is required to limit the bandwidth of the signal before transmission. The filter must have a special frequency response, otherwise it will cause ISI. The special shape of the filter used for this response is a square root raised cosine function. Its characteristics have been discussed in Section 2.6.

Matlab includes a function to help with the design of the square root raised cosine (SRRC) filters. The function “RCOSFIR(R, N_T, RATE, T, FILTER_TYPE)” produces the impulse response of a raised cosine filter or SRRC filter. N_T indicates the length of the response in symbols. RATE is the sampling rate normalized to the symbol rate. T is the symbol interval. The order of the FIR filter is length of the filter in symbols times RATE. The type of filter, i.e. square root raised cosine, is specified in Filter_Type. The parameters used in the SRRC filter for this thesis are

\[ R=0.35, \ N_T=[-2, 2] \text{ (width=4 symbols)}, \ RATE=4, \ T=1, \ \text{FILTER\_TYPE} = '\text{sqrt}' \].

The parameter R was chosen to be 0.35 since this is the value used in the filters of cellular telephones. The parameter N_T was found experimentally to be long enough to give good response. The RATE is normalized to 4.
Figure 4.6 illustrates the impulse response and frequency response for the SRRC filter with the parameters given above. The frequency response is shown twice: one with a linear scale and once with a logarithmic scale. The stop-band attenuation is more than about 25 dB below the pass-band.

The coefficients obtained from Matlab are floating point numbers with the mantissa being 4 digits. The coefficient must be scaled so that the largest coefficient is 128(for 8-
bit resolution). After the coefficients are scaled, they are rounded to the nearest integer. The integer values are then converted to the form of CSD with MNA. The floating point, scaled integer and the form of CSD with MNA are shown in Table 4.1.

### Table 4.1 Coefficients of a SRRC Filter

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Floating Point</th>
<th>Integer</th>
<th>CSD with MNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h(0)=h(16)$</td>
<td>0.0286</td>
<td>7</td>
<td>$7=2^3-1$</td>
</tr>
<tr>
<td>$h(1)=h(15)$</td>
<td>-0.0110</td>
<td>-3</td>
<td>$-3=-(2+1)$</td>
</tr>
<tr>
<td>$h(2)=h(14)$</td>
<td>-0.0676</td>
<td>-16</td>
<td>$-16=-2^4$</td>
</tr>
<tr>
<td>$h(3)=h(13)$</td>
<td>-0.0943</td>
<td>-22</td>
<td>$-22=-2(2^3+2+1)$</td>
</tr>
<tr>
<td>$h(4)=h(12)$</td>
<td>-0.0423</td>
<td>-10</td>
<td>$-10=-2(2^2+1)$</td>
</tr>
<tr>
<td>$h(5)=h(11)$</td>
<td>0.1034</td>
<td>24</td>
<td>$24=2^3(2+1)$</td>
</tr>
<tr>
<td>$h(6)=h(10)$</td>
<td>0.3039</td>
<td>71</td>
<td>$71=2^6+2^3-1$</td>
</tr>
<tr>
<td>$h(7)=h(9)$</td>
<td>0.4786</td>
<td>112</td>
<td>$112=2^4(2^3-1)$</td>
</tr>
<tr>
<td>$h(8)$</td>
<td>0.5478</td>
<td>128</td>
<td>$128=2^7$</td>
</tr>
</tbody>
</table>

Rounding the floating-point number to obtain the integer values in effect change the impulse response and therefore will change the frequency response. The normalized mean square difference ($msd$) in the two responses of $h(n)$ and $h_1(n)$ is given by

\[
\begin{align*}
\text{msd} &= \frac{1}{2\pi} \int_{2\pi} |H(e^{j\omega}) - H_1(e^{j\omega})|^2 d\omega = \sum_{n=-\infty}^{\infty} (h(n) - h_1(n))^2.
\end{align*}
\]

According to this equation, the normalized mean square difference in the two responses is given by $msd = 1.47 \times 10^{-5}$.
4.4.2 Square Root Raised Cosine Filter Architecture in FLEX10K

Figure 4.7 illustrates a 17-tap SRRC filter architecture. This filter has sixteen 8-bit registers arranged in a shifter configuration. The output of each register is called a tap. In this symmetrical filter each tap weight has an equal value counterpoint. The registers corresponding to taps of equal value are summed before the weighting i.e. multiplying. All the products are summed. Then the 8-bit output is obtained by truncating the summed result.

Xin[7..0] 8 bits

Figure 4.7 A SRRC Filter Logic Block Diagram

Inputs of the SRRC filter are 8 bits and internal wordlengths are 16 bits (It was discussed in section 3.3 how to determine the wordlengths of the filter). Truncation is
used going into three locations (See Figure 4.7). After multiplication by the coefficients the result is truncated through the omission of the LSB. The results of sum13, sum14 and sum16 are also truncated.

Since the maximum clock frequency of the modem is 40MHz, it was not necessary to place registers after every adder (this technique is referred to as pipelining).

Figure 4.8 shows how the multiplication by $h(6)$, which is 71, is implemented using CSD with MNA. Recall from Table 4.7 that $71=2^6 + 2^3 - 1$. The coefficient multiplier is replaced with two adders and two shifters.

\[ \text{Figure 4. 8 Implementing a Multiply by 71} \]

A VHDL design file to implement a multiply by the coefficient 71 is shown below
This function costs 33 LCs while it costs 48 LCs to realize a multiply by 71 with the conventional multiplier.

4.4.3 Performance of the Designed Filter

The size of the filter implemented in the EPF10k70-4 is 440 Logic Cells (LCs). The filter length is 17 taps, the input is 8 bits and the filter coefficient uses 8-bit resolution.
The internal resolution of the filter is 16 bits and the maximum operating frequency is 30 MSPS. These values are summarized in Table 4.2.

**Table 4.2 Performance of a SRRC Filter**

<table>
<thead>
<tr>
<th>Taps</th>
<th>Precision/Bits</th>
<th>Size</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input Coef. Internal Output LCs Gates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>8 8 16 8 440 8,090</td>
<td>30 MSPS</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4.9 Frequency Responses of Realized a SRRC Filter**
The realized filter was simulated in MAX+PLUSII. The frequency response obtained from this simulation shown in Figure 4.9b) is compared to the one obtained from the design in Matlab with floating point coefficients shown in Figure 4.9a). The sampling rate is 10MHz and the input is a sine wave sweeping from 100,000 to 5 MHz. Figure 4.9c) was taken from the output of a HP4195A Network/Spectrum Analyzer.

4.5 Half-band Filters

4.5.1 Half-band Filter Characteristics

There is a class of lowpass filters called half-band filters where every second coefficient is zero. Half-band filters are based on the symmetrical FIR filter designs discussed in section 4.2. They have the property that approximately half of the filter coefficients are exactly zeros. The number of multipliers needed in implementing a half-band filter is approximately half of that needed for an ordinary symmetrical FIR filter.

Figure 4.10 illustrates the tolerance specification of a lowpass filter.

\[
\begin{align*}
\Delta H(\omega) & \quad \text{Frequency response of the filter} \\
1 & \quad \text{Ideal response} \\
\delta_p & \quad \text{Passband ripple} \\
\delta_s & \quad \text{Stopband attenuation}
\end{align*}
\]

Figure 4.10 Tolerance Specification of a Lowpass Filter

In a special case that
the filters are called half-band filters. A half-band filter always has \( N = 4i - 1 \) coefficients, where \( i \) is a positive integer. Clearly \( N \) is always odd. Every second coefficient from the center coefficient outward is zero with the exception of the center coefficient which is always equal to 1/2.

### 4.5.2 Lagrange Half-Band Filter

The Lagrange half-band filter is a special filter with a monotonic decreasing passband and stopband. The coefficients of Lagrange half-band filter are determined by the formula [2]:

\[
\begin{align*}
N &= 4i - 1, \quad i \in \text{integer} \\
\h_i(2n) &= \frac{(-1)^{i-1}}{(i-n)!(i-1+n)!(2n-1)}, \quad 0 \leq n < (N+1)/2 \\
\h_i(2n-1) &= 0, \quad 1 \leq n < (N+1)/2 \text{ and } n \neq (N-1)/4 \\
\h_i((N-1)/4) &= 0.5
\end{align*}
\] (4.5)

Frequency responses of the Lagrange half-band filters with parameter from \((i=2,N=7)\) to \((i=5,N=19)\) are shown in Figure 4.11.
It is readily seen that as \( i \) increases, the Lagrange half-band filters begin to approximate the ideal lowpass filters. Passbands and stopbands are monotonic (no ripples). The filters have a maximally flat passband at the expense of an increase in the width of the transition band. Some of the advantages of filters with maximally flat passbands over their equiripple counterparts are discussed in [10] and [11]. For example, assume that a signal \( s(n) \) is band-limited to \( \omega_p \), but immersed in noise, and it is desired that the noise be removed with as little signal distortion as possible. Assume that most of the signal energy is concentrated at lower frequencies (around \( \omega = 0 \)) even though \( \omega_p \) may not be small. It is then desirable to have a lowpass filter that is very flat around \( \omega = 0 \), rather than a filter having equiripple error over the entire passband range (from 0 to \( \omega_p \)). As another example, the filters in the DSP-based \( \pi/4 \)-DQPSK modem are in cascade. If these filters have equiripple passbands, the passband error clearly accumulates as the
signal travels through the entire system. Under such a situation, a filter with a very flat passband (at least in the region where the signal is expected to have most of its energy) is, therefore, preferred. In addition to these advantages, it is generally easier to equalize the amplitude distortion introduced by filters with maximally flat passbands [12].

If the filter with parameter \((i=3,N=11)\) is scaled so that the largest coefficient is 128, then the integer coefficients and CSD-MNA representations of the coefficients are listed in Table 4.3.

<table>
<thead>
<tr>
<th>(h(0)=h(10))</th>
<th>Floating Point</th>
<th>Integer</th>
<th>CSD with MNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0059</td>
<td>2</td>
<td>2=2</td>
<td></td>
</tr>
<tr>
<td>(h(2)=h(8))</td>
<td>0.0488</td>
<td>-13</td>
<td>-13=((-2^3 + 2^2 + 1))</td>
</tr>
<tr>
<td>(h(4)=h(6))</td>
<td>0.2930</td>
<td>75</td>
<td>75=(2^2+1)(2^4-1)</td>
</tr>
<tr>
<td>(h(5))</td>
<td>0.5</td>
<td>128</td>
<td>128=2^7</td>
</tr>
</tbody>
</table>

The Lagrange half-band filter with parameter \((i=3,N=11)\) is used in the demodulator. D.C. gain of this filter is 2 or 6dB. Its frequency response is illustrated in Figure 4.12b). The signal \(M_I\) is from the \(I\) channel digital mixer in the demodulator shown in Figure 3.1b). Its power spectrum is illustrated in Figure 4.12a). The desired signal bandwidth is within the maximally flat passband of the lowpass filter. The lowpass filter removes the double frequency terms at around \(\omega=\pi\) as illustrated in Figure 4.12c).
The architecture of the Lagrange half-band filter with parameter $(i=3,N=11)$ is shown in Figure 4.13. It is similar to that of the SRRC filter architecture discussed in Section 4.4.2.
The performance in FLEX10k70-4 is illustrated in Table 4.4.

<table>
<thead>
<tr>
<th>Taps</th>
<th>Precision/Bits</th>
<th>Size</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input Coef. Internal Output LCs Gates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>

The results of simulation in Matlab, simulation in MAX+PLUS II and the circuit testing using HP4195A Network/Spectrum Analyzer are shown in Figure 4.14. The stopband has a rejection of -40dB for frequencies more than 0.8π radians/samples.
4.6 Decimator and Interpolator with Polyphase Filters

4.6.1 Aliasing in the transition band

Aliasing is permitted in the transition band if a half-band filter is employed as an anti-aliasing or anti-imaging filter for the Power-of-2 decimator or interpolator [13].

Figure 4. 14 Design, Simulation and Test for the Lagrange Half-Band Filter with Parameter $i=3, N=11$
In the Section 2.9.1, decimation has already been defined. Figure 4.15 illustrates an anti-aliasing filter $h(n)$ followed by a downsampler.

Assume that the anti-aliasing filter $h(n)$ is a Lagrange half-band filter with $N$ coefficients and $N=4i-1$, $i \in \text{Integer}$. It is assumed that the bandwidth of $s(n)$ is $S_b$ and the wanted message bandwidth is $M_s$. The spectrum of $s(n)$ is shown in Figure 4.16a). The frequency response of the Lagrange half-band filter with the passband $\omega_p$ is shown in Figure 4.16b). The symmetry properties of the Lagrange half-band filter impose certain restrictions on the use as an anti-aliasing or anti-imaging filter. The restrictions are that the message bandwidth $M_s$ of the input signal should be less than $\omega_p$. After decimating or interpolating, the signal with bandwidth between $\omega_p$ and $2\pi-\omega_p$ (transition band of half-band filter) is aliased. After filtering, the spectrum of the signal $x(n)$ which is sufficient band-limiting is obtained and shown in Figure 4.16c). Figure 4.16d) and 4.16e) illustrate the spectrum of output signal $y(n)$ in different frequency space ($\omega_0 = 2\omega$). The spectrum of output signal $y(n)$ within bandwidth $M_s$ is not stained although the spectrum outside bandwidth $M_s$ is aliased. The same thing will happen in interpolation.
4.6.2 Decimator with Polyphase Filters

The output $y(n)$ shown in Figure 4.15 is given by
According to the equation derived by N.J. Fliege [2], the above becomes

\[
y(m) = \sum_{r=\infty}^{m} h(2r)s_0(m-r) + h(\frac{N-1}{2})s_1(m - \frac{N-1}{4})
\]

\[
y(m) = h(2m)s_0(m) + h(\frac{N-1}{2})s_1(m - \frac{N-1}{4})
\]  

(4.7)

where \( s_1(r) \) can be obtained by shifting \( s(n) \) in steps of 1, then downsampling \( s(n) \) by 2 without a phase offset, and \( s_0(r) \) can be obtained directly by downsampling \( s(n) \) by 2 without a phase offset.

If the subfilter \( h(2m) \) is implemented in the direct form of Figure 4.2, the delay chain of \( s_1(m) \) can be combined together with \( s_0(m) \) in a single one as shown in Figure 4.17.

**Figure 4.17 Memory-Saving Decimating by a Factor of 2 with Polyphase Filters**

The Lagrange half-band filter with parameter \( i=2, N=7 \) is employed as an anti-aliasing filter \( h(n) \). According to the structure shown in Figure 4.17, the decimator with the Lagrange half-band polyphase filter is illustrated in Figure 4.18. The \( \text{reg3} \) and \( \text{reg4} \) realize the downsampling by 2. The circuits of part 2 are running at a rate 2-times slower...
than the circuits of part 1. The data rate is reduced by half after this filter. The designs of the coefficient multipliers are similar to those in the SRRC filter.

![Block Diagram of a Decimator with Polyphase Filters](image)

**Figure 4. 18 Block Diagram of a Decimator with Polyphase Filters**

### 4.6.3 Interpolator with Polyphase Filters

Interpolators are the dual of decimators. According to the signal flow graph reversal theorem, the respective signal flow graphs can be derived from each other. The inputs and outputs are swapped, downsamplers and upsamplers are interchanged and the directions of all signals are reversed. So, the interpolator architecture is of a form similar to the decimator architecture discussed in section 4.6.2.

An interpolator consists of an upsampler followed by an anti-imaging filter $h(n)$ as shown in Figure 4.19, where $h(n)$ is a half-band filter with $N$ coefficients.

![Interpolator](image)

**Figure 4. 19 Interpolator**
Upsampling produces the interim signal \( x(n) \) from the input signal \( y(m) \). The output signal \( s(n) \) is obtained by filtering \( x(n) \) with a half-band filter \( h(n) \)

\[
s(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)
\]  

(4.8)

According to the structure of a memory-saving polyphase interpolator derived by N.J. Fliege [2], an interpolator with a factor of 2 upsampler is shown in Figure 4.20.

![Figure 4.20 Memory-Saving Interpolating by a Factor of 2 with Polyphase Filters](image)

The Lagrange half-band filter with parameter \((i=3,N=11)\) is employed as an anti-imaging filter \( h(n) \). The architecture of an interpolator with the Lagrange half-band polyphase filter is illustrated in Figure 4.21. The operating frequency of these subfilters is only half of the output data rate. A commutator in the output is used to represent the combining of the signals \( s_A(m) \) into the output signal \( s(n) \).
4.6.4 Power Consumption Estimation and Performance

In a decimator with polyphase filters (DPF), the subfilters are running at the output rate rather than the input rate, saving roughly 50% of the power consumed by a conventional decimator by 2 (anti-alasing filter directly followed by downsampling by 2). In an interpolator with polyphase filters (IPF), the subfilters are running at the input rate rather than the output rate, saving roughly 50% of the power consumed by a conventional interpolator by 2 (upsampling by 2 directly followed by anti-imaging filter). Table 4.5 lists the performance and size of DPF and IPF.

<table>
<thead>
<tr>
<th>Filter Name and Taps</th>
<th>Precision/Bits</th>
<th>Size</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input</td>
<td>Coef.</td>
<td>Internal</td>
</tr>
<tr>
<td>DPF: 11 taps</td>
<td>8</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>IPF: 7 taps</td>
<td>8</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>
4.7 Summary

In a DSP-based modem design the cost in terms of a silicon footprint (number of logic cells required) is the main parameter to be minimized. This can be accomplished by using halfband filters, canonic signed digit multipliers and wordlength truncation. While not as important, the power consumption of the device is also something that should be minimized. The power consumption can be reduced by using polyphase filters and rate conversion and performing the filtering at the lower rate. The decimator/interpolator with polyphase filters minimize the power consumption and the implementation cost while maintaining the system performance parameters such as data throughput rate. These efficient filter designs satisfy the requirements of both filtering and sampling rate conversion in the DSP-based $\pi/4$-DQPSK modem.
5. Timing Recovery

5.1 Introduction

Timing recovery is one of the most critical demodulator functions in a DSP-based π/4-DQPSK modem. A timing parameter estimator must perform timing recovery in a DSP-based modem if sampling is not synchronized to the data symbol timing. Normally, an algorithm, which can extract the timing information from the sampled data stream, is required.

Timing recovery algorithms can be classified into two main categories according to the data dependency [13]:

Class 1: Decision-Directed (DD) or Data-Aided (DA)

Class 2: Non-data-Aided (NDA)

When the detected data sequence is known and is used in the estimation algorithm as if it were the true sequence, then the estimation is DD or DA. Otherwise, the estimation is NDA.

In analog-implemented modems, NDA timing recovery is usually performed by either a feedback loop that adjusts the phase of a local oscillator, or by a feedback arrangement that regenerates a timing signal from the incoming signal (see Figure 5.1a) [14]. The signal that is used to derive timing information is thus usually a continuous signal in both time and amplitude.
A modification of this continuous signal solution is to first sample the analog signal and then use a digital algorithm to control the sampling instant [15][16][17]. Such a solution is illustrated in Figure 5.1b) and is called hybrid timing recovery. The digital algorithm is used to control the analog VCO circuit.

Figure 5.1 Timing Recovery Methods

A completely digital timing recovery circuit is shown in Figure 5.1c). In this system
the sampling rate in the receiver is not synchronized to the sampling rate in the transmitter. The timing information is derived from the samples of the baseband signal. Gardner and Erup [18] have developed a timing adjustment algorithm, which performs interpolation.

A second DSP-based timing recovery system is illustrated in Figure 5.1d). The inputs to the estimator are the output of the $I$ and $Q$ SRRC matched filters. The output of the estimator is used to control the phase offset of the downsamplers. The inherent problem of the DSP-based timing recovery method is that the signal sampling is not synchronized to the symbol timing; the two rates are incommensurate and the sample times never coincide exactly with desired strobe times. Recognition of incommensurability (or the estimation of the unknown timing parameter) is vital to understanding the DSP-based timing recovery problem.

5.2 Squaring Timing Recovery

![Figure 5.2 Squaring Timing Recovery](image)

The squaring synchronizer is a spectral line generating synchronizer. The block diagram of the squaring synchronizer is shown in Figure 5.2. A spectral line at the data clock frequency (or at a multiple of this frequency) is extracted with a narrow-band filter or reconstructed with a PLL [13][14]. Since such lines are not ordinarily encountered in bandwidth-efficient systems, some nonlinear processing of the signal is used to generate
such lines. The most common are the square-law rectifier and the absolute-value rectifier. References [19] gives results for timing extractors equipped with the square-law rectifier or with the absolute-value rectifier. It appears that the absolute-value rectifier hold a substantial advantage over the square-law rectifier in the performance and the performance. Unfortunately, clock circuits implemented with the absolute-value rectifier are more difficult to model mathematically than the square-law rectifier.

5.3 Timing Recovery in a DSP-Based π/4-DQPSK Modem

5.3.1 Description of the NDA Timing Parameter Estimation Algorithm

A timing recovery circuit in a DSP-based π/4-DQPSK modem is shown in Figure 5.3.

![Figure 5.3 Block Diagram of a DSP-Based Timing Recovery](image)

The outputs from the SRRC matched filters (assuming the matched filters are ideal) have been derived in the section 3.22. Equation (3.28) and (3.29) are rewritten as:
\[
x_{\text{im}}(m, \varepsilon) = \sum_{i=-\infty}^{\infty} I_i p(mT/4 - iT - \varepsilon T) \cos \Delta \theta_i - \sum_{i=-\infty}^{\infty} Q_i p(mT/4 - iT - \varepsilon T) \sin \Delta \theta_i + n_i(m) \tag{5.1}
\]

and

\[
x_{\text{qm}}(m, \varepsilon) = \sum_{i=-\infty}^{\infty} I_i p(mT/4 - iT - \varepsilon T) \sin \Delta \theta_i + \sum_{i=-\infty}^{\infty} Q_i p(mT/4 - iT - \varepsilon T) \cos \Delta \theta_i + n_q(m), \tag{5.2}
\]

where \( p(\bullet) \) is a raised cosine function, \( \Delta \theta \) is defined in section 2.4, \( \varepsilon \in [0, 1] \) expresses the offset between the proper decision point and the point where the decision is currently being made, \( n_i(m) \) and \( n_q(m) \) are zero-mean white Gaussian noise components at the outputs of the inphase and quadrature matched filters. The parameter \( \varepsilon \) is unknown but known to be a slowly varying time delay.

The timing parameter estimation algorithm will extract the timing information \( \varepsilon \) from the outputs of the SRRC matched filters.

To simplify equation (5.1) and (5.2), the abbreviations

\[
s_{\text{im}}(m, \varepsilon) = \sum_{i=-\infty}^{\infty} I_i p(mT/4 - iT - \varepsilon T) \cos \Delta \theta_i - \sum_{i=-\infty}^{\infty} Q_i p(mT/4 - iT - \varepsilon T) \sin \Delta \theta_i \tag{5.3}
\]

and

\[
s_{\text{qm}}(m, \varepsilon) = \sum_{i=-\infty}^{\infty} I_i p(mT/4 - iT - \varepsilon T) \sin \Delta \theta_i + \sum_{i=-\infty}^{\infty} Q_i p(mT/4 - iT - \varepsilon T) \cos \Delta \theta_i \tag{5.4}
\]

are used.

Equation (5.1) and (5.2) become:
\[ x_{in}(m, \varepsilon) = s_{in}(m, \varepsilon) + n_i(m) \quad (5.5) \]

\[ x_{om}(m, \varepsilon) = s_{om}(m, \varepsilon) + n_o(m) \quad (5.6) \]

They are band-limited to \( B_o = \frac{(1+\alpha)}{2T} \). In this study the matched filter roll-off factor \( \alpha \) is set at 0.35. Since squaring the signal (it is called the absolute-value rectifier if squaring is replaced by the absolute-value operating) doubles the bandwidth \( B_o \), the bandwidth of

\[ x(m, \varepsilon) = [x_{in}(m, \varepsilon)]^2 + [x_{om}(m, \varepsilon)]^2 \quad (5.7) \]

is limited to twice this value \( B_x = 2B_o \).

If the sampling rate is chosen to be 4 samples per symbol, then the sampling period \( T_s \) is \( T/4 \). In this case the squaring operation does not cause aliasing, since the ratio of sampling rate to signal bandwidth is greater than 2, i.e.

\[ \frac{1/T}{B_x} = \frac{1/T}{2B_o} = \frac{4/T}{2(1+\alpha)/2T} = \frac{4}{(1+\alpha)} = \frac{4}{1.35} = 2.96 > 2. \]

The estimation algorithm will use \( L \) symbols and \( L \) will be an even number. Since each symbol is sampled 4 times, there are \( 4L \) samples in the estimation range (see Figure 5.4).

**Figure 5.4 Observation Interval**

The \( 4L \) samples have been organized into \( L \), 4-sample observation intervals. After synchronization, these intervals will correspond to a symbol time. The 4-point discrete
Fourier transform (DFT) is performed on each of the $L$ intervals, and the first order coefficient of the 4–point DFT in the $i$th observation interval is denoted $C_{i,j}$.

The sum of the $C_{i,j}$ for the $L$ intervals is given by

$$SC_i = \sum_{j=0}^{L-1} C_{i,j} = \sum_{j=0}^{L-1} \left[ \sum_{m=0}^{L-1} x(m + 4i, \varepsilon)e^{-j\frac{2\pi}{4}(m+4j)} \right] .$$

(5.8)

where $SC_i$ is the sum. The phase in units of radians/sample can be estimated by

$$\hat{\varepsilon} = -\frac{1}{2\pi} \arg(SC_i) .$$

(5.9)

The quality of this estimation is derived in the section 5.3.3.

5.3.2 Simulation of the NDA Timing Parameter Estimation Algorithm

For the simulation of the NDA timing parameter estimation algorithm the modulator and demodulator are connected back to back. The impulse length for the SRRC filters is 17 and perfect carrier recovery is assumed. True jitter-free timing recovery is not possible since the estimation is based on samples which have a limited sampling rate (the sampling rate is chosen to be 4 samples per symbol at the outputs of the matched filters) and have an arbitrary offset from the zeros.

Figure 5.5 illustrates the waveform and the spectrum of the signal $x(m, \varepsilon)=|x_{in}(m, \varepsilon)|+|x_{om}(m, \varepsilon)|$ over 64 intervals. In Figure 5.5 b) the DC component is set to be zero. From this figure it can be seen that a line spectrum at a symbol rate is contained in the spectrum of the signal $x(m, \varepsilon)=|x_{in}(m, \varepsilon)|+|x_{om}(m, \varepsilon)|$ over 64 intervals. This simulation result shows it is possible to extract a timing information at the
symbol rate with the absolute-value rectifier. The following simulation will show how to extract a timing information.

\[
x(m, \varepsilon) = |x_{im}(m, \varepsilon)| + |x_{qm}(m, \varepsilon)| \text{ over 64 Intervals}
\]

Figure 5.5 Waveform and Spectrum of the Signal \( x(m, \varepsilon) = |x_{im}(m, \varepsilon)| + |x_{qm}(m, \varepsilon)| \) over 64 Intervals

Eye-Pattern diagrams and 4-point DFTs over 64 intervals with the absolute-value rectifier and different values of \( \varepsilon \) are shown in Figure 5.6 to Figure 5.9. The DC component is set to be zero. The decision point is determined by \( \varepsilon \). From these figures it is seen that each different value of \( \varepsilon \) matches a different vector \( SC_i \). From these simulation results it is also shown that the magnitude of \( SC_i \) is a non-zero number. The difference between Figure 5.9 and Figure 5.10 is that the different rectifier is used. The results (Figure 5.9d and Figure 5.10d) show that the angle of the vector \( SC_i \) is almost the same. The implementation of the absolute-value rectifier is much cheaper than the square-law rectifier.
a) I Channel Eye-Pattern Diagram

b) Magnitude of 4-point DFT over 64 Intervals

Figure 5. 6 Eye-Pattern Diagrams and 4-Point DFT over 64 Intervals with $\varepsilon=0.5$ and the Absolute-Value Rectifier

c) Q Channel Eye-Pattern Diagram

d) Compass Plot for First Component $SC_j$

Figure 5. 7 Eye-Pattern Diagrams and 4-Point DFT over 64 Intervals with $\varepsilon=0.75$ and the Absolute-Value Rectifier
Figure 5. 8 Eye-Pattern Diagrams and 4-Point DFT over 64 Intervals with $\varepsilon = 0$ and the Absolute-Value Rectifier

Figure 5. 9 Eye-Pattern Diagrams and 4-Point DFT over 64 Intervals with $\varepsilon = 0.25$ and the Absolute-Value Rectifier
5.3.3 Quality of the NDA Timing Parameter Estimation Algorithm

In this section it is proven mathematically that equation (5.9) is an unbiased estimation of $\varepsilon$. The variance of $\hat{\varepsilon}$ will not be derived. It is clear from general principle of statistics that increasing the number of observation interval will reduce the variance. A reasonable value for $L$ will have to be found experimentally.

It is assumed that $I_k$ and $Q_k$ are i.i.d (independent, identically, distributed) data and the matched filters in the receiver are linear phase filters (real and symmetric impulse response).

The input to the DFT block is
\[ x(m, \varepsilon) = [x_{im}(m, \varepsilon)]^2 + [x_{om}(m, \varepsilon)]^2 \]
\[ = [s_{im}(m, \varepsilon) + n_{i}(m)]^2 + [s_{om}(m, \varepsilon) + n_{o}(m)]^2. \quad (5.10) \]

Then the expectation of \( x(m, \varepsilon) \) given \( \varepsilon \) is
\[ E[x(m, \varepsilon) | \varepsilon] = E[s_{im}(m, \varepsilon) + n_{i}(m) | \varepsilon]^2 + E[s_{om}(m, \varepsilon) + n_{o}(m) | \varepsilon]^2]. \quad (5.11) \]

Since the noise and symbols are independent of each other, and \( E[n_{i}(n)] = E[n_{o}(n)] = 0 \), the cross terms of the binomial equation will vanish. The remaining terms will be:
\[
E[x(m, \varepsilon) | \varepsilon] = E[s_{im}(m, \varepsilon) | \varepsilon] + E[s_{om}(m, \varepsilon) | \varepsilon] + E[n_{i}(m)] + E[n_{o}(m)]
\]
\[ = E \left[ \sum_{l=-\infty}^{\infty} I_{l} p(mT/4 - lT - \varepsilon T) \cos \Delta \theta_{i} - \sum_{l=-\infty}^{\infty} Q_{l} p(mT/4 - lT - \varepsilon T) \sin \Delta \theta_{i} \right] | \varepsilon \]
\[ + E \left[ \sum_{l=-\infty}^{\infty} I_{l} p(mT/4 - lT - \varepsilon T) \sin \Delta \theta_{i} + \sum_{l=-\infty}^{\infty} Q_{l} p(mT/4 - lT - \varepsilon T) \cos \Delta \theta_{i} \right] | \varepsilon \]
\[ + E[n_{i}(m)] + E[n_{o}(m)]. \quad (5.12) \]

Since \( I_{i} \) and \( I_{o} \), \( Q_{i} \) and \( Q_{o} \), are independent of \( \varepsilon \), \( E[I_{i}, I_{i} | \varepsilon] = E[I_{i}, I_{i}] \) and \( E[Q_{i}, Q_{i} | \varepsilon] = E[Q_{i}, Q_{i}] \) are obtained. The equation above becomes
\[
E[x(m, \varepsilon) | \varepsilon] = \sum_{l=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} E[I_{i}, I_{i}] p(mT/N - lT - \varepsilon T) p(mT/N - iT - \varepsilon T) + E[n_{i}(m)]
\]
\[ + \sum_{l=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} E[Q_{i}, Q_{i}] p(nT/N - lT - \varepsilon T) p(nT/N - iT - \varepsilon T) + E[n_{o}(m)]. \quad (5.13) \]

Since independently distributed symbols are of mean power equal to 1, equation (5.13) becomes
where $\delta^2$ is noise power and it is assumed that the noise power is a known constant.

This equation shows that $E[x(m, \varepsilon)|\varepsilon]$ is a function of the variable $(mT/4-iT-\varepsilon T)$. Then $E[x(m, \varepsilon)|\varepsilon]$ can be represented by the function $q(mT/4-\varepsilon T) = E[x(m, \varepsilon)|\varepsilon]$. The function $q(*)$ is given by

$$q(mT/4-iT-\varepsilon T) = E[x(m, \varepsilon)|\varepsilon] = 2 \sum_{i=-\infty}^{\infty} [p(mT/4-iT-\varepsilon T)]^2 + 2\delta^2.$$  \hspace{1cm} (5.15)

It is pointed out that the function $q(*)$ is periodic with period $T$ (there are 4 samples in a $T$ period). This is a consequence of $q(mT/4-\varepsilon T) = q((m+4)T/4-\varepsilon T)$, which is proven below:

$$q((m+4)T/4-\varepsilon T) = 2 \sum_{i=-\infty}^{\infty} [p((m+4)T/4-iT-\varepsilon T)]^2 + 2\delta^2$$

$$= 2 \sum_{i=-\infty}^{\infty} [p(mT/4-(i-1)T-\varepsilon T)]^2 + 2\delta^2$$

$$= 2 \sum_{i=-\infty}^{\infty} [p(mT/4-iT-\varepsilon T)]^2 + 2\delta^2$$

$$= q(mT/4-\varepsilon T).$$  \hspace{1cm} (5.16)

If $\varepsilon = 0$, equation (5.15) becomes:

$$q(mT/4) = 2 \sum_{i=-\infty}^{\infty} [p(mT/4-iT)]^2 + 2\delta^2.$$  \hspace{1cm} (5.17)

For negative $m$, the above equation becomes:
\[ q(-mT/4) = 2\sum_{i=-\infty}^{\infty} [p(-mT/4 - iT)]^2 + 2\delta^2 \]
\[ = 2\sum_{i=-\infty}^{\infty} [p(-(mT/4 + iT))]^2 + 2\delta^2 . \]

In section 3.3.2, it is pointed out that \( p(mT/4) \) is a symmetric function:
\[ p(-mT/4) = p(mT/4). \]
Equation (5.18) will become:
\[ q(-mT/4) = 2E[p(mT/4 + iT)] + 2\delta^2 \]
\[ = 2\sum_{k=0}^{\infty} [p(mT/4 - kT)]^2 + 2\delta^2 \]
\[ = q(mT/4) . \]

So, \( q(mT/4) \) is a \( T \) periodic and symmetric function. The following relationship is obtained:
\[ q((4L - i)T/4) = q(LT - iT/4) \]
\[ = q(-iT/4) \]
\[ = q(iT/4) \]
where \( L \) was defined in section 5.3.1.

According to equation (5.8), the expectation of \( SC_i \) is given by:
\[ E[SC_i|x] = \sum_{m=0}^{4L-1} E[x(m, \varepsilon)|x] e^{-j2\pi m} \]
\[ = \sum_{m=0}^{4L-1} q(mT/4 - \varepsilon T)e^{-j2\pi m} \]
\[ = \left[ \sum_{m=0}^{4L-1} q((m - 4\varepsilon)T/4)e^{-j2\pi (m-4\varepsilon)} \right] e^{-j2\pi \varepsilon} . \]
Since \( q(mT/4) \) is periodic with period of 4, equation (5.22) becomes:
$$E[SC_1|e] = \left[ \sum_{m=0}^{4L-1} q(mT/4) e^{-j\frac{2\pi m}{4}} \right] e^{-j2\pi e}.$$ (5.23)

Since \(q(kT/4)\) is a symmetrical function, the imaginary part of its DFT is zero. Therefore

$$\left[ \sum_{m=0}^{4L-1} q(mT/4) e^{-j\frac{2\pi m}{4}} \right]$$

is real and the argument of \(E[SC_1|e]\) is \(-2\pi e\).

According to equation (5.9), the expectation of \(\hat{e}\) is given by:

$$E[\hat{e}] = -\frac{1}{2\pi} \arg(E[SC_1|e])$$

$$= -\frac{1}{2\pi} (-2\pi e)$$

$$= e,$$ (5.24)

where \(0 \leq e < 1\).

Thus \(\hat{e}\) is an unbiased estimate of the data clock phase \(e\) over the observation interval \(L\). In the continuous-time filter and square timing recovery, the timing is determined by detecting the zeros of the timing wave. Therefore, true jitter-free timing recovery is possible if the timing wave exhibits only amplitude jitter, but no phase jitter. In our case, however, the estimation is based on samples that have an arbitrary offset from the zeros and thus exhibit random amplitude fluctuations. Therefore, only asymptotically jitter-free recovery can be obtained [13].

5.4 Implementation of the Algorithm in a FLEX10K

A particularly simple realization is obtained for four samples per symbol from the matched filter's output, \(N = T/ T_s = 4\). The variance of the timing estimation was found experimentally for an observation interval of 64 symbol times, and for \(E_b/N_0 = 25\)dB this variance was \(1 \times 10^{-5}\) (symbol period)\(^2\).
For $N=\frac{T}{T_s}=4$, a multiplication-free realization of the estimator can be obtained.

This is shown below. The experimental component, i.e. $SC_1$, can be calculated from the expression

$$SC_1 = \sum_{i=0}^{L-1} C_{1,i} = \sum_{i=0}^{L-1} \left[ \sum_{n=0}^{3} x(4i + n, \epsilon)e^{-j\frac{2\pi n}{4}} \right]$$

$$= \sum_{i=0}^{L-1} \left[ \sum_{n=0}^{3} x(4i + n, \epsilon)(-j)^n \right]. \tag{5.25}$$

Splitting the real and imaginary part yields

$$\text{Re}[SC_1] = \sum_{i=0}^{L-1} x(4i, \epsilon) - x(4i + 2, \epsilon) \tag{5.26}$$

and

$$\text{Im}[SC_1] = \sum_{i=0}^{L-1} -x(4i + 1, \epsilon) + x(4i + 3, \epsilon) \tag{5.27}.$$
Figure 5.11 Computation of the First Order Coefficient of the 4-Point Discrete Fourier Transform of $x(i,e)$

The structure of an averaging operator is shown in Figure 5.12. The accumulator performs the sum of 64 6-bit numbers and should be reset by a signal $Aclr$ once every 64 input data clocks. The feedback path from a register $reg_1$ to the other $Add$ input provides the accumulator action:

$$Sum(n+1) = Sum(n) + Din(n) \quad 0 \leq n \leq 63 \quad (5.28)$$

Figure 5.12 Averaging Operator
The Add output is widened by 6 extension bits, so that at least $2^6 = 64$ repetitive operations can be performed without overflow. The least significant six bits of the outputs of reg1 are truncated and connected to register reg2. The frequency of DataClock is 64 times that of Aclk which is the clock for reg2. The VHDL description of the averaging operator is shown below:

```
-- function: averaging operator for 64 6-bit signed numbers
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;

ENTITY ACC IS
  PORT(
    DataClock : IN  STD_LOGIC;
    Din : IN  SIGNED(5 DOWNTO 0) ;
    Dout : OUT  SIGNED(5 DOWNTO 0);
    Aclk : OUT  STD_LOGIC
  );
END ACC;

ARCHITECTURE a OF ACC IS

BEGIN
  PROCESS (DataClock)
  VARIABLE cnt : INTEGER RANGE 0 TO 63;
  VARIABLE sum : SIGNED(11 DOWNTO 0);
  BEGIN
    IF (DataClock' EVENT AND DataClock = '1') THEN
      sum :=sum + Din;
      IF cnt /= 63 THEN
        cnt :=cnt+1;
        Aclk <='0';
      ELSE
        cnt :=0;
        Aclk <='1';
      END IF;
      Dout <=sum(11 DOWNTO 6);
      sum(11 DOWNTO 0) :='000000000000';
    END IF;
  END IF;
END PROCESS;
END a;
```
The simulation waveforms from MAX+PLUS II are presented in Figure 5.13. The input data is set to the constant number 10 and it is synchronized to DataClock. The pulse width of Ack is a DataClock period.

![Figure 5.13 Simulation Waveforms for the Averaging Operator](image)

The next step is to obtain $\hat{\varepsilon}$ from the argument of $SC_I$. Recall that $\hat{\varepsilon}$ is the error in sampling the method as a function of a symbol period. In this thesis, the matched filter sampling rate is four times the symbol rate. There are four samples per symbol. The downsampler samples one sample from four samples of matched filter outputs. The time at the downsampler must correspond to one of the four sampling times of the matched filter. Therefore, before corrective action is taken to shift the sampling time of the downsampler, $\hat{\varepsilon}$ must be quantized. Then, depending on the value of $\hat{\varepsilon}$, the sampling times for the downsamplers will be either not shifted or shifted by 1, 2, or 3 system samples which corresponds to 0.25, 0.5, or 0.75 symbols.

The quantization regions are shown in Figure 5.13. If the phase of $SC_I$ is in region A, then the sampling time is within 1/8 of a symbol and therefore can not be improved by shifting. If the phase of $SC_I$ is in region B, then the sampling downsampler clock should be delayed by one system sample (0.25 symbols). If the phase of $SC_I$ is in region C, then the matched filter sampling should be delayed by two system samples (0.5 symbols). And if the phase of $SC_I$ is in region D, then the matched filter sampling should be...
delayed by three system samples (0.75 symbols), which is equivalent to advancing it by one sample (0.25 symbols).

![Diagram of SC Plan](image)

**Figure 5.14 Distribution of Events A, B, C, D in Complex Plan SC<sub>i</sub>**

To minimize the algorithm against the occasional noise estimate, the shift adjustment is restricted to a delay of one system sample or an advance of one system sample. This means that when the phase of SC<sub>i</sub> falls into region C, the delay cannot be two sample points as called for. Instead region C must be divided into two regions C<sub>oo</sub> and C<sub>o1</sub> as shown in Figure 5.13. If the phase of SC<sub>i</sub> is in region C<sub>oo</sub>, then the action taken is the same as if the phase of SC<sub>i</sub> is in region B. The action taken is to delay the matched filter sampling clock by 0.25 symbols. If the phase of SC<sub>i</sub> is in region C<sub>o1</sub>, then the action taken is to advance the matched filter sampling clock by 0.25 symbols, which is the same action taken for a phase in region D.

From the geometry in Figure 5.13, the phase of SC<sub>i</sub> can be categorized to region A, B, C<sub>oo</sub>, C<sub>o1</sub> or D using the real and imaginary parts of SC<sub>i</sub> with the following equations
A = \{ \Re [SC_i] \geq 0, \Im [SC_i] \leq 0 \} \cap \{ |\Re [SC_i]| \geq |\Im [SC_i]| \} \\
- \cup \{ \Re [SC_i] > 0, \Im [SC_i] > 0 \} \cap \{ |\Re [SC_i]| > |\Im [SC_i]| \} , \tag{5.29}

B = \{ \Re [SC_i] \geq 0, \Im [SC_i] \geq 0 \} \cap \{ |\Re [SC_i]| \leq |\Im [SC_i]| \} \\
\cup \{ \Re [SC_i] < 0, \Im [SC_i] > 0 \} \cap \{ |\Re [SC_i]| < |\Im [SC_i]| \} , \tag{5.30}

C_{oo} = \{ \Re [SC_i] \leq 0, \Im [SC_i] \geq 0 \} \cap \{ |\Re [SC_i]| \geq |\Im [SC_i]| \} \\
C_{oi} = \{ \Re [SC_i] < 0, \Im [SC_i] < 0 \} \cap \{ |\Re [SC_i]| > |\Im [SC_i]| \} , \tag{5.31}

D = \{ \Re [SC_i] \leq 0, \Im [SC_i] \leq 0 \} \cap \{ |\Re [SC_i]| \leq |\Im [SC_i]| \} \\
\cup \{ \Re [SC_i] > 0, \Im [SC_i] < 0 \} \cap \{ |\Re [SC_i]| < |\Im [SC_i]| \} . \tag{5.32}

The actions taken in response to an event are shown in Table 5.1.

<table>
<thead>
<tr>
<th>EVENT</th>
<th>SIGNAL</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>Shorten</td>
<td>ACTION</td>
</tr>
<tr>
<td>A</td>
<td>Set to 0</td>
<td>Set to 0</td>
</tr>
<tr>
<td>B or C_{oo}</td>
<td>Set to 1</td>
<td>Set to 0</td>
</tr>
<tr>
<td>C_{oi} or D</td>
<td>Set to 0</td>
<td>Set to 1</td>
</tr>
</tbody>
</table>

The A, B, C_{oo}, C_{oi} or D can be determined using the sign bits of the three functions \Re [SC_i], \Im [SC_i], and |\Re [SC_i]| - |\Im [SC_i]|. This is illustrated in Table 5.2, where
\[ X = \text{sign bit of Re}[SC_1], \quad Y = \text{sign bit of Im}[SC_1], \quad \text{and} \]
\[ Z = \text{sign bit of } [\text{Re}[SC_1] - \text{Im}[SC_1]] \]

Table 5.2 The Relationship of the Phase Offset and SC₁.

<table>
<thead>
<tr>
<th>( X )</th>
<th>( Y )</th>
<th>( Z )</th>
<th>Event</th>
<th>( \text{Step} )</th>
<th>( \text{Length} )</th>
<th>( \text{Shorten} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( A )</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>( B )</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>( A )</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( D )</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( C_{50} )</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>( B )</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>( C_{21} )</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( D )</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

According to Table 5.1, the Boolean expressions for the outputs \( \text{Length} \) and \( \text{Shorten} \) are written in terms of \( X, Y \) and \( Z \) as follows:

\[ \text{Length} = \overline{X} \& \overline{Y} \& Z \# X \& \overline{Y} \& \overline{Z} \# X \& Y \& Z \]  \hspace{1cm} (5.33)

and

\[ \text{Shorten} = \overline{X} \& Y \& Z \# X \& Y \& \overline{Z} \# X \& Y \& Z \]  \hspace{1cm} (5.34)

where \& is the logical “AND” and \# is the logical “OR”.

The circuit that makes the decision to advance, delay or leave the sampling time is shown in Figure 5.15. The signals \( L \) and \( S \) are employed to control the phase shifter. It is desired that the phase shifter changes the phase of \text{DataClock} signal only once in one \text{Aclk} period (64 \text{DataClock} periods). Two and-gates are used to limit the impulse width of \text{Length} and \text{Shorten} to one \text{DataClock} period. Two and-gates outputs may contain glitches on state transitions. To sample the stable states of two and-gates outputs, \text{DataClock} is inverted as the clock signals of registers reg4 and reg5.
The circuit in Figure 5.16 shows the matched filter sampling circuit. The L and S signal are outputs from Figure 5.15 and used to control the phase of the sampling clock. The output pair \((I_R(k), Q_R(k))\) from the two downsamplers will be used as the inputs of differential detector (see Figure 2.18). Notice that \(x(m, \varepsilon)\) generated by the circuit is given by \(x(m, \varepsilon) = |x_{m}(m, \varepsilon)| + |x_{o}(m, \varepsilon)|\) while the analysis assumed \(x(m, \varepsilon)\) was given by \(x(m, \varepsilon) = x_{m}(m, \varepsilon)^2 + x_{o}(m, \varepsilon)^2\). This change was made to save complexity. Experimentally it was found that the performance of the circuit was actually the same for absolute value circuit. This was also the finding of [19].
Figure 5.16 Data Extractor

The phase shifter can be designed using the state machine illustrated in Figure 5.17.

Figure 5.17 State Diagram Corresponding to the Phase Shifter
The state diagram has one circle for each state, and an arrow for each transition. This state machine has ten states. The letter inside each circle is a state name. Each arrow leaving a given state points to the next state for the given conditions. The symbol "<=" represents the machine output value to the left side of the symbol. The VHDL design file that implements a ten-state state machine in the Figure 5.17 is shown below.

```vhdl
-- function: phase shifter. It is a ten-state machine.
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;

ENTITY pshifter IS
    PORT(
        clk_Fs : IN STD_LOGIC;
        S : IN STD_LOGIC;
        L : IN STD_LOGIC;
        Adjust : OUT STD_LOGIC;
        DataClock : OUT STD_LOGIC;
        BitClock : OUT STD_LOGIC;
    );
END pshifter;

ARCHITECTURE a OF pshifter IS
    TYPE STATE_TYPE IS (s0, s1, s2, s3, s4, s5, s6, s7, s8, s9);
    SIGNAL state : STATE_TYPE;
    SIGNAL D_record : STD_LOGIC; -- use to record third Bit of state
    SIGNAL B_record : STD_LOGIC; -- use to record second Bit of state
    SIGNAL clk_2Fc : STD_LOGIC;
    SIGNAL SL : STD_LOGIC_VECTOR(1 DOWNTO 0);

    BEGIN
        -- A 1 bit counter
        PROCESS (clk_Fs)
        VARIABLE cnt : INTEGER RANGE 0 TO 1;
        BEGIN
            IF (clk_Fs'EVENT AND clk_Fs = '1') THEN
                IF cnt=1 THEN
                    clk_2Fc <= '1';
                    cnt :=0;
                ELSE
                    cnt := cnt+1;
                    clk_2Fc <= '0';
                END IF;
            END IF;
        END PROCESS;
       ...
    END ARCHITECTURE a;
```
PROCESS (D_record)
BEGIN
  IF (D_record'EVENT AND D_record = '1') THEN
    SL(1) <= S;
    SL(0) <= L;
  END IF;
END PROCESS;

PROCESS (clk_2Fc)
BEGIN
  IF (clk_2Fc'EVENT AND clk_2Fc = '1') THEN
    CASE state IS
      WHEN s0=>
        IF SL="01" THEN
          state <= s9;
          D_record <= '0';
          B_record <= '0';
          Adjust <= '1';
        ELSE
          state <= s7;
          D_record <= '1';
          B_record <= '1';
          Adjust <= '0';
        END IF;
      WHEN s1=>
        state <= s0;
        D_record <= '0';
        B_record <= '0';
        Adjust <= '0';
      WHEN s2=>
        state <= s1;
        D_record <= '0';
        B_record <= '0';
        Adjust <= '0';
      WHEN s3=>
        IF SL="10" THEN
          state <= s0;
          D_record <= '0';
          B_record <= '0';
          Adjust <= '1';
        ELSE
          state <= s2;
          D_record <= '0';
          B_record <= '1';
          Adjust <= '0';
        END IF;
      WHEN s4=>
        state <= s3;
        D_record <= '0';
        B_record <= '1';
        Adjust <= '0';
      WHEN s5=>
        state <= s4;
        D_record <= '1';
        B_record <= '0';
    END CASE;
  END IF;
END PROCESS;
The signal \( D_{\text{record}} \) is used to store the third bit of the current state of the state machine. It corresponds to the recovered data clock signal \( \text{DataClock} \). The signal \( B_{\text{record}} \) is used to store the second bit of the current state of the state machine. It corresponds to the recovered data clock signal \( \text{BitClock} \). The first process statement is sensitive to the \( D_{\text{record}} \) or \( \text{DataClock} \). This statement is employed to synchronize the inputs \( S \) and \( L \) to the rising edge of \( \text{DataClock} \). The second process statement is sensitive to the \( \text{clk}_2\text{Fc} \) signal. The state machine includes this Process Statement that is activated on every positive edge of the \( \text{clk}_2\text{Fc} \) control signal. The signal \( \text{state} \) stores the current state of the state machine. The declaration of the type \text{STATE_TYPE} defines the state from \( s_0 \) to \( s_9 \) for \text{pshifter}. At start-up, the state machine is initialized to the first state in the TYPE Declaration. Otherwise, the CASE Statement defines the transitions between the states, i.e., determines
which state to enter on the next rising edge of clk_2Fc. The simulation waveforms from MAX+PLUS II are presented in Figure 5.18.

![Simulation Waveform for the Phase Shifter](image)

**Figure 5.18 Simulation Waveform for the Phase Shifter**

### 5.5 Summary

In this chapter, a DSP-based timing recovery algorithm has been derived. The periodic function \( x(m, \varepsilon) = [x_{in}(m, \varepsilon)]^2 + [x_{qn}(m, \varepsilon)]^2 \), where \( x_{in}(m, \varepsilon) \) and \( x_{qn}(m, \varepsilon) \) are outputs of the inphase and quadrature matched filters, is used for the search of the unknown parameter \( \varepsilon \). The argument of expectation of fundament component of the DFT of \( x(m, \varepsilon) \) is proportional to \( \hat{\varepsilon} \). The effects of noise are reduced by using \( L \) symbol times (i.e. \( L \) periods of \( x(m, \varepsilon) \)) to calculate the fundament DFT component. The estimate is given by

\[
\hat{\varepsilon} = -\frac{1}{2\pi} \arg \left( \sum_{i=0}^{L-1} \sum_{m=0}^{J} x(m + 4i, \varepsilon) e^{-j\frac{2\pi}{L}(m+4i)} \right).
\]

Since this algorithm provides a direct estimation of \( \varepsilon \), it does not have the "local maximum" problem associated with a hill-climber algorithm. A multiplication-free implementation in FLEX10K has been achieved.
6. Performance Measurement and Results

6.1 Introduction

In this chapter, the performance results of the DSP-based π/4-DQPSK modem are presented. The performance measure has two components: chip size and bit error rate (BER) vs. $E_b/N_0$. The two measures are investigated separately. Information relating to chip size is given in section 6.2 while the performance is described in section 6.3 and 6.4. Modulator performance is the subject of section 6.3 while BER performance of the demodulator is the subject of section 6.4.

The simulation results in Altera MAX+PLUSII show that the maximum bit rate of the modem is 5Mbit/sec (the maximum system clock is 40MHz). In following testing, the bit rate is 1.25 Mbit/sec since the maximum system clock of the FPGA testing board at the TRLabs is 10MHz.

6.2 Chip Size

The circuits for the digital filters and decision algorithms discussed in earlier chapters were integrated into two complete circuits: one for the modulator and the other for the demodulator. Each was implemented in separate FLEX10K70 chip. The FLEX10K70 has approximately the capacity of a 70,000-gate ASIC. The details of the chip utilization are shown in Table 6.1.
Table 6.1 Chip Size of the DSP-based $\pi/4$-DQPSK Modem

<table>
<thead>
<tr>
<th>Chip</th>
<th>Device</th>
<th>Input Pins</th>
<th>Output Pins</th>
<th>Memory</th>
<th>Chip Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Bits</td>
<td>Utilized</td>
<td>LCs</td>
<td>Gates</td>
</tr>
<tr>
<td>Modulator</td>
<td>EPF10k70-4</td>
<td>3</td>
<td>30</td>
<td>224</td>
<td>1 %</td>
</tr>
<tr>
<td>Demodulator</td>
<td>EPF10k70-4</td>
<td>10</td>
<td>33</td>
<td>16</td>
<td>0 %</td>
</tr>
</tbody>
</table>

6.3 Modulator Performance

The quality of a modulator can be assessed from an eye-pattern diagram, power spectrum and constellation diagram. These diagrams and graphs are generated from the structure shown in Figure 6.1. The Digital Transmission Analyzer generates the pseudorandom binary sequence (PRBS) generator pattern to be used for the modulator. The PRBS pattern is synchronized to an external clock input signal, which is provided by the output signal $\text{TxClock}$ of the Modulator. The pattern length is $2^{15}$-1. The $\pi/4$-DQPSK discrete signal from the modulator is converted to the analog IF signal by an 8-bit D/A converter followed by a reconstruction filter. The analog IF signal is converted to the CH1 input of the HP89410A, DC-10 MHz Vector Signal Analyzer (VSA).

![Figure 6.1 Performance Measurement of the Modulator](image)

102
VSA is an instrument that allows modulation analysis of modern digital
communication signals. The instrument is similar to that of an oscilloscope, however;
the VSA allows the additional analysis of both the in-phase and quadrature-phase
components of a signal. The analyzer provides many different ways of viewing
demodulated data. The VSA works in digital demodulation mode and has built-in linear
phase matched filters with orders up to 401. Four different measurement data and data
formats can be selected: spectrum, trajectory, constellation and eye-pattern diagram. The
spectrum of a digitally modulated carrier before demodulation is shown in Figure 6.2.
The $\pi/4$-DQPSK signal trajectory is shown in Figure 6.3. The polar IQ constellation is
shown in Figure 6.4, and the eye-pattern diagrams of the $I$ and $Q$ channels are shown in
Figure 6.5 and Figure 6.6.

![Figure 6.2 Power Spectrum of $\pi/4$-DQPSK Modulator Output Measured by HP89410A](image-url)

Figure 6.2 Power Spectrum of $\pi/4$-DQPSK Modulator Output Measured by HP89410A
The power spectrum plotted in Figure 6.2 shows that the sidelobes are approximately 30 dB down from the main lobe. The bit rate is 1.25Mbit/s. The shape of this spectrum is very close to the simulation result illustrated in Chapter 3 Figure 3.5).

The constellation is shown in Figure 6.3. The transitions between points in the constellation are not instantaneous because of the SSRC filter. This relativity shows transitions have different trajectories that depend on the transmitted data history. The constellation diagram illustrated in Figure 6.3 is what it should be. The constellation points are at angles of $0, \frac{\pi}{4}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}$ and $\frac{7\pi}{4}$.

Figure 6.3 $\pi/4$-DQPSK Constellation Measured by HP89410A
Figure 6. 4 I Channel Eye-Pattern Diagram

Figure 6. 5 Q Channel Eye-Pattern Diagram
The Eye-pattern diagrams are shown in Figure 6.4 and Figure 6.5. They have much larger eye opening than the ones shown in Chapter 3 Figure 3.5. The reason for this is the matched filters used in the HP-89410A are much longer than the 17th order filter used in chapter 3. The higher order filters reduce ISI and make the decision points sharper.

6.4 Demodulator Performance

The BER performance of the system with the modulator/demodulator in tandem was found with the structure shown in Figure 6.6. The eye-pattern diagrams inside the demodulator can provide insight into the amount of ISI that results from truncating the impulse response of the matched filter. The internal digital outputs of the matched filters are fed to two external D/A converters for the sole purpose of displaying the eye-pattern diagram. In this structure the carrier estimation circuit of the demodulator is not used. A 10MHz oscillator provides the system clock signal for the modulator. At the same time it also provides the external reference signal for the HP 3324A Synthesized Function/Sweep Generator. The HP 3324A provides the system clock signal for the demodulator. The HP 3764 is used to measure the BER between the generated PRBS pattern (the pattern length is $2^{15}-1$) and the received pattern. It is assumed that the channel introduces white Gaussian noise. The white Gaussian noise is generated by a HP 33120A, 15 MHz Function/Arbitrary Waveform Generator.
In the first test, the system clock $\text{clk}_{\text{FsD}}$ in the demodulator is phase locked to the system clock $\text{clk}_{\text{FsM}}$ in the modulator. The HP 3324A is used to introduce random phase error between 0 and 360° in the two 10MHz clocks. The output signal of HP 3324A will be locked onto the external reference input. The phase of the output signal, which provides the system clock signal for the demodulator is changed during testing. This test structure models a properly functionary IF frequency control circuit.

Figure 6.7 illustrates the theoretical and testing result of BER versus $E_b/N_0$ for the additive white Gaussian noise channel with no carrier frequency offset. The theoretical
performance is given in [1]. In Figure 6.7 it is observed that the BER performance of the modulator/demodulator pair, which employs 17-order square root raised cosine matched filters, degrades about 1.5 dB from theory.

![Graph of BER versus Eb/N0](image)

**Figure 6.7 BER versus \( E_b/N_0 \) for the Additive White Gaussian Noise Channel**

The eye-pattern diagrams at the output of the demodulator's matched filters are illustrated in Figure 6.8 with no Carrier Frequency Offset and the Additive White Gaussian Noise \( E_b/N_0 = 15 \) dB. They have smaller eye opening than the eye-pattern diagrams shown in Figure 5.5 and Figure 5.6. The main reason is that the impulse response of SRRC filter in Hp 89410A is much longer than the one employed in the demodulator.
Figure 6. 8 Eye-pattern diagrams at the output of the Demodulator's Matched Filters with no Carrier Frequency Offset and the Additive White Gaussian Noise $E_b/N_0 = 15$ dB

In the second test, the frequency offset is set between the system clock clk_FsD in the demodulator and the system clock clk_FsM in the modulator. Figure 6.9 illustrates the BER degradation resulting from carrier frequency offset. The timing recovery circuit is sensitive to the carrier frequency offset. For larger frequency offset, the carrier frequency offset estimation circuit should be employed to improve the performance of the BER[1].
Figure 6.9 BER versus $E_b/N_0$ as a Function of Carrier Frequency Offset for the Additive White Gaussian Noise Channel

The eye-pattern diagrams at the output of the demodulator’s matched filters are illustrated in Figure 6.10 with the carrier frequency offset and the Additive White Gaussian Noise $E_b/N_0 = 15$ dB. Since the frequency offset exists, there will be crosstalk between the I channel and Q channel. This is the reason that three eyes show up in Figure 6.10. The crosstalk will be removed by the differential detector.
Figure 6. 10 Eye-Pattern Diagrams at the Output of the Demodulator's Matched Filters with the Carrier Frequency Offset = 5Hz and the Additive White Gaussian Noise $E_b/N_0 = 15$ dB
7. Conclusions and Future Work

The objectives of the research work reported in this thesis were the following:

(1) To design and implement a DSP-based $\pi/4$-DQPSK modem on FPGAs

(2) To determine the chip size in LCs and/or gates required to implement a DSP-based $\pi/4$ DQPSK modem on Altera FLEX10K70 chips.

(3) To determine the implementation loss associated with the DSP-based solution.

7.1 Conclusions

A high bit rate DSP-based $\pi/4$ DQPSK modem was designed and implemented on two Altera FLEX10K70 chips. The algorithm used in the modem can accommodate bit rate up to 5 Mbit/sec. The modulator algorithm requires 2,056 LCs (approximately 38k gates) while the demodulator requires 2,372 LCs (approximately 44k gates). Simulation results show that the ISI can be further reduced by increasing the order of the square root raised cosine filter. Since higher order filters cost more, there is a cost/performance tradeoff that has to be optimized on an application by application basis.

The IF signal (as opposed to a baseband signal) was sampled using IF digital sampling technique. By using this technique the A/D interface can be moved closer to the antenna. Since the sampling rate is four times the carrier frequency, a multiplier-free digital complex mixer is obtained. This technique has two advantages:
• It reduces the effect of analog imperfections such as quadrature phase and gain imbalance.

• It reduces the number of gates required thereby reducing the power consumption and cost.

Multirate DSP operations such as filtering and frequency translation often change the minimum sample rate required to represent a signal without aliasing. In multirate DSP systems, different sample rates are used within a system to achieve the most efficient computation at each stage. Multirate signal processing techniques can be used to reduce the power consumption and the cost.

Implementing the decimator/interpolator with polyphase filters minimizes the power consumption and the implementation cost while maintaining system performance. These efficient filter designs satisfy the requirements of both filtering and sampling rate conversion in the DSP-based π/4-DQPSK modem.

The DSP-based timing recovery algorithm provides a direct estimation of timing information. It does not have the "local maximum" problem associated with a hill-climber algorithm. This algorithm can be implemented on the FPGA without any additional analog circuits. The timing recovery circuit worked well in the absence of IF frequency offset. The timing recovery circuit was responsible for some ISI due to coarse sampling.

For an additive white Gaussian noise channel, the result of BER measurement indicates that the BER performance of the modem degrades about 1.5 dB from theory when 17th order square root raised cosine filters were used.
7.2 Future Work

From the work reported in this thesis it is seen that a higher order matched filters would provide significant improvement in the BER performance. The use of the higher order matched filters, on the other hand, will make the modulator and demodulator more costly. A trade-off study between cost and performance would be of value. An alternate approach would be to look at reducing the ISI caused by the short filter with a decision feedback equalizer.

The timing recovery circuit did not work well if there is a carrier frequency offset between the modulator and the demodulator. This was due to the poor resolution in sampling the matched filter. More work could be done to improve the resolution or investigate other clock recovery circuits.
References


115


