THE UNIVERSITY OF CALGARY

Low Complexity IF-Sampling Receiver for DS/SS

by

Wai William Wong

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN

PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER

ENGINEERING

CALGARY, ALBERTA

October, 1999

Abstract

In modern communication systems, the IF-sampling technique is becoming widely used because it allows more signal processing to be performed in the digital domain than in the analog domain by bringing the analog-to-digital convertor (ADC) closer to the antenna. The drawback of this technique is that the inphase (I) and quadrature (Q) samples are obtained at different instances in time which can cause serious amplitude imbalances between the I and Q samples. To solve this problem, additional hardware and firmware must be added to the receiver. Therefore, the complexity of existing IF-sampling receivers is still quite high. C.Y. Hung has developed a low complexity IF-sampling system [30] using the lowest sampling rate, 2 samples/symbol, which shows promising performance. However, this system uses a pair of variable gains, one for I and one for Q, to compensate for the amplitude imbalance problem between the I and Q samples. To implement the pair of variable gains using discrete components still involves a certain degree of difficulty. This thesis proposes a modification of the system in [30] which reduces the complexity of the receiver by eliminating the pair of variable gains and achieves a better performance. The thesis also includes an implementation solution using purely discrete components to show the simplicity of the receiver.

Acknowledgment

I would like to thank Dr. M. Fattouche and Dr. L.E. Turner, my co-supervisors, for their support and guidance in the successful completion of this project. I would also like to thank T. Mathews and P.J.W. Graumann, graduate students at the University of Calgary, Electrical and Computer Engineering Department, for their technical supports. Also I would like to thank both TRLabs and the University of Calgary for their financial supports in the form of scholarship and teaching assistantships. Finally, I would like to express my gratitude to my parent for providing me the opportunity to study in Canada and for giving me the required financial and spiritual supports.

TABLE OF CONTENTS

Abstract iii

Acknowledgment iv

Table of Contents v

List of Tables viii

List of Figures ix

List of Symbols xiv

Chatper 1 Introduction 1

Chatper 2 Fundamental Theory of the Proposed IF-sampling Receiver 6

2.1 π/4-Shifted-DQPSK 6

2.1.1 Modulation Scheme for the π/4-Shifted-DQPSK 8

2.1.2 Detection Algorithm for the π/4-Shifted-DQPSK 10

2.1.3 Limitations of π/4-Shifted-DQPSK 13

2.2 DS/SS for the Proposed IF-sampling Receiver 13

2.2.1 Overview of ISI 14

2.2.2 Suppressing ISI using a DS/SS system 18

2.2.3 Barker Codes 23

2.3 $F_s/4$-Downconversion 26

2.3.1 Time-Domain Analysis 28

2.3.2 Frequency-Domain Analysis 31

2.4 Nyquist Sampling Theorem 38
2.4.1 Lowpass Sampling Theorem 39
2.4.2 Bandpass Sampling Theorem 41
2.5 Nyquist Pulse for Minimum ISI 44
2.6 Summary 51

Chapter 3 Previous Research for the DS/SS IF-sampling System 53
3.1 The DS/SS Transmitter 54
3.2 The Channel 58
3.3 The DS/SS IF-sampling Receiver 61
3.4 The Detector 71
3.5 Shaping Pulse Design 71
3.6 The Performance of the DS/SS IF-sampling receiver 77
3.6.1 Despreading 78
3.6.2 The Effect of Bandpass White Noise on the DS/SS IF-sampling Receiver 85
3.6.3 Simulation Setup and Results 91
3.7 Consideration for System Implementation 97
3.8 Summary 98

Chapter 4 The Proposed DS/SS IF-sampling System 99
4.1 Removing G_I and G_Q 99

4.2 Performance Analysis for the Bandpass White Gaussian Channel 103
4.2.1 Despreading 105
4.2.2 The Effect of Bandpass White Noise in the Proposed Receiver . 108
4.2.3 Simulation Setup and Results 112
4.3 Summary 116

Chapter 5 Implementation of the Proposed DS/SS IF-sampling Receiver 117
5.1 The Proposed Receiver 118
5.2 The Despreader 118
5.2.1 The Despreading Unit 119
5.2.1.1 The Matched Filter 123
5.2.1.2 The Power Detector 129
5.2.2 The Synchronization Controller 131
5.3 The π/4-shifted-DQPSK Decoder 137
5.4 Summary 152

Chapter 6 Conclusion and Future Work 153
Appendix A Implementation Tool 156
Appendix B Extra Diagrams 160
References 166

LIST OF TABLES

2.1.1.1 Table of differential phases according to Grey coding 8

2.1.1.2 Table of $I_d[n]$ and $Q_d[n]$ versus ϕ_n 10

2.2.3.1 All possible matched filter outputs for $\pi/4$-shifted DQPSK symbols 25

3.6.1.1 Table of the tap weights for $0 \le t_d \le 0.25T_c$ 81

3.6.1.2 Table of the tap weights for $-0.25 \le t_d \le 0$ 83

4.1.1 Amplitude variations of the time-shifted duobinary pulse $g(t)$ 102

5.3.1 The Booth's coefficients 141

LIST OF FIGURES

1.1 The block diagram of the super-heterodyne receiver 1

1.2 The block diagram of the conventional IF-sampling receiver 3

1.3 The block diagram of the proposed IF-sampling receiver 4

2.1.1 Signal constellation at the output of the modulator 7

2.1.1.1 Signal constellation at the output of the modulator 9

2.1.2.1 Differential detector for $\pi/4$-shifted-DQPSK 11

2.2.1.1(a) Spectrum of $p(t)$ 17

2.2.1.1(b) Time-domain function $p(t)$ (sinc function) 17

2.2.1.1(c) Received baseband signal 17

2.2.2.1 The spreading process 19

2.2.2.2 The despreading process 19

2.2.3.1 The autocorrelation of the Barker code 23

2.2.3.2 The matched filter for Barker correlation 24

2.3.1 The block diagram of an ideal channel 27

2.3.1.1 Graphical representation on the effect of sampling the cosine and sine carriers with $F_s = 4f_{IF}$ 29

2.3.1.2 The signal processes in the $F_s/4$-downconversion 30

2.3.2.1 The frequency spectrum of $x(t)$ (BW is the lowpass bandwidth) 31

2.3.2.2 The Nyquist sampler 31

2.3.2.3 The frequency spectrum of $x_s(t)$ 33

2.3.2.4(a) Spectrum of $r_{IF}(t)$ ($BW = W$ is the bandpass bandwidth) 35

2.3.2.4(b) Spectrum of $I'(t)$ 35

2.3.2.4(c) Spectrum of $I(t)$ 35

2.3.2.4(d) Spectrum of $I_d[n]$ 35

2.3.2.5(a) Spectrum of $r_{IF}[n]$ 36

2.3.2.5(b) Spectrum of $r'_{IF}[n]$ 36

2.3.2.5(c) Spectrum of $I_d[n]$ 36

2.4.1.1 Frequency spectrum of a lowpass signal 39

2.4.1.2 The spectrum of the sampled lowpass signal ($f_s < 2f_H$) 40

2.4.2.1 The spectrum of $y(t)$ ($BW = W$ is the bandpass bandwidth) 41

2.5.1 The spectrum of $p_2(t)$ (with 100% roll-off) 46

2.5.2 The raised cosine pulse $p_2(t)$ (with 100% roll-off) 46

2.5.3 The baseband received signal $r(t)$ ($r[n]$ is the output of the sampler in the presence of sampling error) 47

2.5.4 The spectrum of $r[n]$ ($BW = \frac{1}{2T_p}$ is the lowpass bandwidth) 47

2.5.5 The baseband received signal $r'(t)$ ($r'[n]$ is the output of the sampler in the presence of sampling error) 48

2.5.6 The spectrum of $r'[n]$ ($BW = \frac{1}{T_p}$ is the lowpass bandwidth) 48

2.5.7 The block diagram to form $p_{dm}(t)$ 49

2.5.8 The spectrum of $p_{dm}(t)$ 50

2.5.9 The duobinary pulse $p_{dun}(t)$ 50

3.1 Block diagram of the bandpass communication system model 53

3.1.1 Transmitter for DS/SS IF-sampling system ($M = 11$ and $p[n]$ is the 11-chip Barker sequence) 55

3.2.1 Power density spectrum of the bandpass white noise, $n(t)$ 59

3.2.2(a) Power density spectrum of $x(t)$ and $y(t)$ 61

3.2.2(b) Autocorrelation functions of $x(t)$ and $y(t)$ 61

3.3.1 Architecture of the IF-sampling receiver for DS/SS system 62

3.3.2(a) Hypothetical RF spectrum with 100% excess bandwidth 65

3.3.2(b) Spectrum of the IF signal at the input of the sampler ($f_{IF} = \frac{F_s}{4}$) 65

3.5.1(a) Time domain zero ISI and I/Q amplitude balance criterions 75

3.5.1(b) Frequency domain zero ISI and I/Q amplitude balance criterions 75

3.5.2 Magnitude response of the shaping pulse $g(t)$ with $W = 1/T_c$ 76

3.5.3 The impulse response of the shaping pulse $g(t)$ with $W = 1/T_c$ 77

3.6.2.1 Relationship between $\phi_{xx}(\tau)$, $\phi_{yy}(\tau)$, $\phi_{z_i z_i}[k]$, and $\phi_{z_q z_q}[k]$ for 100% excess bandwidth duobinary pulse 88

3.6.2.2(a) Power density spectra of $x(t)$ and $y(t)$ 89

3.6.2.2(b) Power density spectra of $z_i[m]$ and $z_q[m]$ 89

3.6.3.1 Simulated performance of the DS/SS IF-sampling system 96

3.7.1 IF-stage of the DS/SS IF-sampling receiver 97

4.1.1 Impulse response of the time-shifted duobinary pulse $g(t)$ 101

4.1.2 Frequency response of the time-shifted duobinary pulse $g(t)$ 102

4.2.1 Architecture of the proposed DS/SS IF-sampling receiver 104

4.2.2(a) ISI in the received inphase chip sequence 106

4.2.2(b) ISI in the received quadrature chip sequence 106

4.2.2.1 Relationship between $\phi_{xx}(\tau)$, $\phi_{yy}(\tau)$, $\phi_{z_i z_i}[k]$, and $\phi_{z_q z_q}[k]$ for the time-shifted duobinary pulse 110

4.2.2.2(a) Power density spectra of $x(t)$ and $y(t)$ 111

4.2.2.2(b) Power density spectra of $z_i[m]$ and $z_q[m]$ 111

4.2.3.1 Simulated performance of the proposed DS/SS IF-sampling system 115

5.1 The block diagram of the proposed receiver 117

5.2.1 The block diagram of the despreader 119

5.2.1.1 The block diagram of the despreading unit 120

5.2.1.2 Timing diagram of the despreading unit 122

5.2.1.1.1 The block diagram of the matched filter 126

5.2.1.1.2 The structure of the adder/subtracter module and the register 127

5.2.1.2.1 The block of the power detector 130

5.2.2.1 The block diagram of the synchronization controller 132

5.2.2.2 A sample output magnitude of the despreading unit 133

5.2.2.3 The timing diagram of the synchronization controller 136

5.3.1 The structure of the $\pi/4$-shifted DQPSK decoder 143

5.3.2 The block diagram of the cycle controller 144

5.3.3 Block diagram of the cycling adder/subtracter module 145

5.3.4 Detail structure of *Block*1 146

5.3.5 The structure of *Booth Logic* 147

5.3.6 Detail structure of *Block*2 148

5.3.7 Detail structure of *Block*3 150

5.3.8 The timing diagram of the $\pi/4$-shifted DQPSK decoder 151

A.1 The block diagram of the 4-bit ripple-carry adder module 157

A.2 The .mcr file for the 4-bit ripple-carry adder module 158

A.3 The .sim file for the 4-bit ripple-carry adder module 158

A.4 The .tem file for the 4-bit ripple-carry adder module 159

B.1 The state diagram of the counter controller 161

B.2 The state diagram of the timer controller 162

B.3 The block diagram of the counter controller 163

B.4 The block diagram of the timer controller 163

B.5 Detail block diagram of a 5-bit counter unit 164

B.6 The state diagram of the cycle controller 165

LIST OF SYMBOLS

ADC — Analog-to-digital Convertor

I — Inphase

Q — Quadrature

RF — Radio Frequency

IF — Intermediate Frequency

LO — Local Oscillator

LNA — Low Noise Amplifier

DSP — Digital Signal Processor

FPGA — Field Programmable Gate Array

DQPSK — Differential Quadrature Phase Shift Keying

DS/SS — Direct Sequence Spread Spectrum

ISI — Intersymbol Interference

SNR — Signal-to-noise Ratio

DS — Direct Sequence

PSK — Phase Shift Keying

FH — Frequency Hopping

FSK — Frequency Shift Keying

DC — Direct Current

GPS — Global Positioning System

BPF — Bandpass Filter

LPF — Lowpass Filter

DTFT — Discrete-time Fourier Transform

BER — Bit-error-rate

dB — Decibel

Chapter 1 Introduction

Most digital communication systems use the traditional super-heterodyne receiver. The simplest super-heterodyne receiver, shown in Figure 1.1, consists of three main stages: (1) the radio frequency (RF) stage, (2) the intermediate frequency (IF) stage, and (3) the baseband stage.

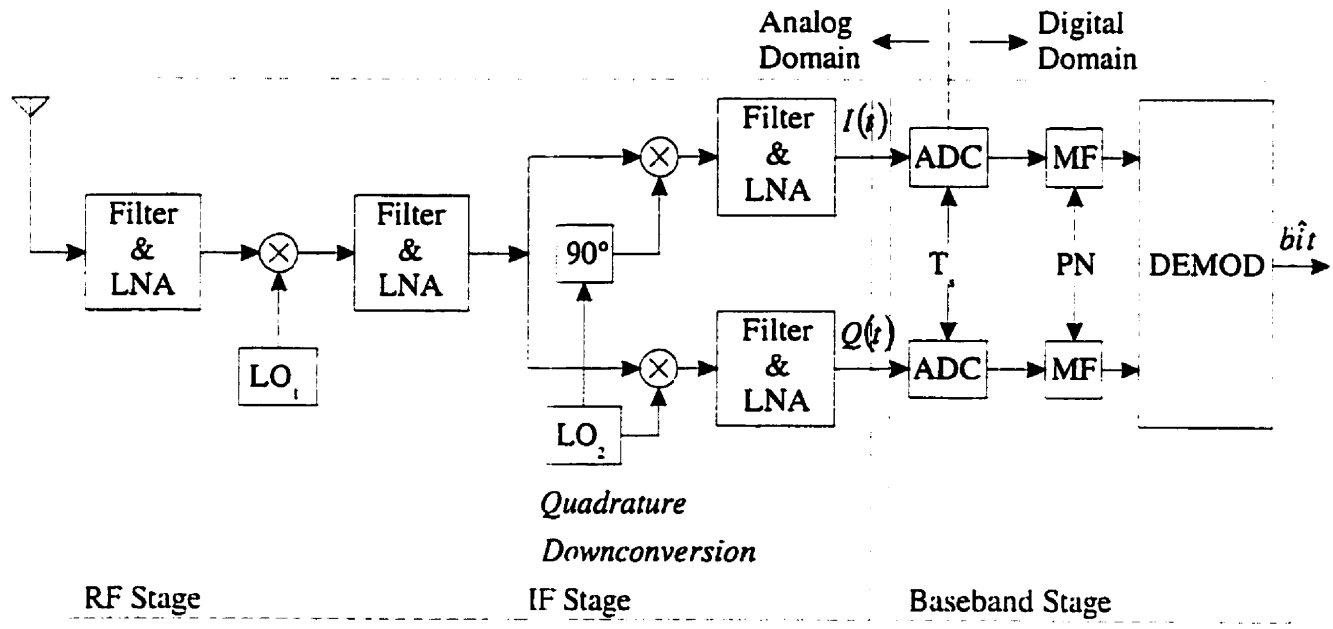


Figure 1.1 The block diagram of the super-heterodyne receiver.

A more complicated super-heterodyne receiver will have additional IF stages to provide better sensitivity and better selectivity as required for the individual receiver. The problem of the super-heterodyne receiver is that it contains many analog components such as local oscillators (LO), mixers, filters, and low noise amplifiers (LNA). As shown in Figure 1.1, the components on the left-hand side of the analog-to-digital converter

(ADC) are analog and the components on the right-hand side of the ADC are digital. As shown, more than half of the receiver is formed by analog components. The disadvantage of analog components is that they are costly to manufacture and are not reliable in the long run due to their aging property. In high-volume applications, it is almost impossible to make the same kind of components perfectly align with each other even with automated tools. This problem leads to the situation where the receiver coming back from the manufacturer does not work the same way as the prototype from the laboratory. Consequently, extra time and effort are needed for the assembly and testing phases. Therefore, a solution to this problem is to eliminate the analog components inside the receiver as much as possible. In addition, there are two analog signal paths after the quadrature downconverter in the super-heterodyne receiver. Since the analog components on one path cannot align to the corresponding components on the other path perfectly, it introduces phase and amplitude imbalances between inphase (I) and quadrature (Q) symbols [1][2] which can cause serious performance degradation in the receiver. This is another well-known problem of the super-heterodyne receiver which is due to the use of analog components.

Removing analog components inside a receiver has attracted considerable research in the recent past [2][3][4]. A common solution is to move the ADC closer to the antenna in order to digitize the analog signal in an earlier stage of the receiver. Once the signal is in the digital domain, the entire signal processing can be achieved by digital components such as digital signal processor (DSP) and field programmable gate array (FPGA); these components give persistent performance which does not vary with their age. In addition, the functionality of digital components is commonly driven by software

which greatly enhances the flexibility to change their behavior. However, sampling right after the antenna is not realistic since some amount of band-select and filtering must be preformed prior to the ADC in order to minimize the adjacent channel interference. As a result, the perfect place for the ADC is at the output of the RF stage which forms the well known conventional IF-sampling receiver as shown in Figure 1.2.

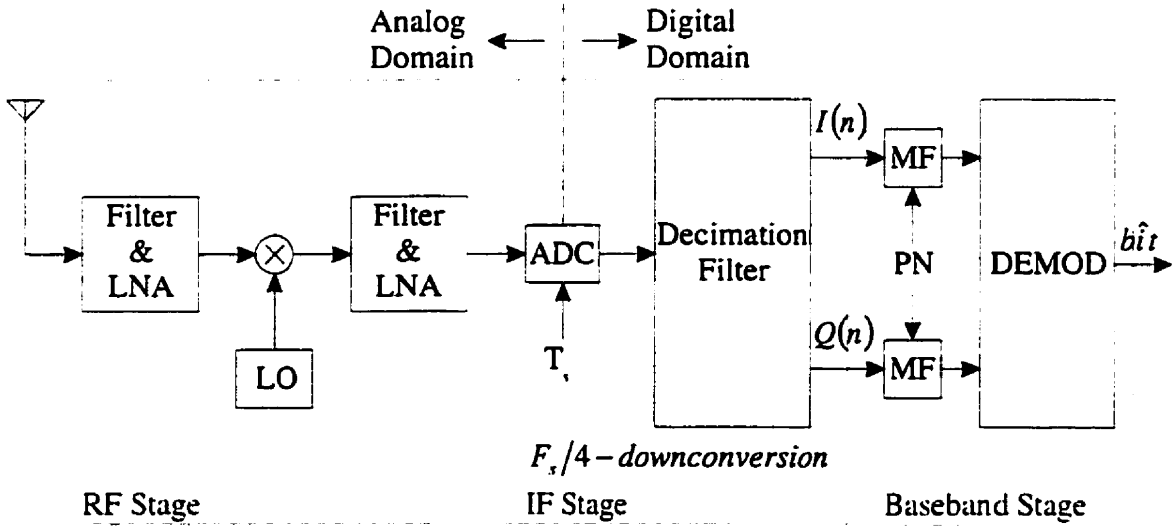


Figure 1.2 The block diagram of the conventional IF-sampling receiver.

Note that $F_s/4$-downconversion is the technique used to replace quadrature downconversion and it is discussed in the next chapter. The architecture of the conventional IF-sampling receiver solves two major problems presented in the super-heterodyne receiver. Firstly, it eliminates all analog components (LO, mixer, filter, and LNA) in the IF stage which reduces the cost for production and improves the reliability of the IF stage. Secondly, it removes the phase and gain imbalance problem in the super-heterodyne receiver by using only one ADC to digitize the analog signal. As a result, this architecture has been adopted in many studies [3][4][5] and there are, in fact, commercial

chips [6][7] available using the same architecture. However, the drawback of the IF-sampling receiver is that the I and the Q samples are no longer being sampled at the same time instance[3][8]. This creates a sampling timing misalignment problem between the I and the Q samples. Consequently, existing IF-sampling receivers have to use complicated algorithms and circuits [3][9][10] tc remove the sampling timing misalignment problem. Thus the complexity of existing IF-sampling receivers is still quite high.

The main objective of this research is to design and implement a low complexity IF-sampling system which does not require any complicated algorithms to handle the sampling timing misalignment problem and achieves an acceptable performance. The architecture of the proposed IF-sampling receiver in Figure 1.3 is even simpler than the conventional IF receiver.

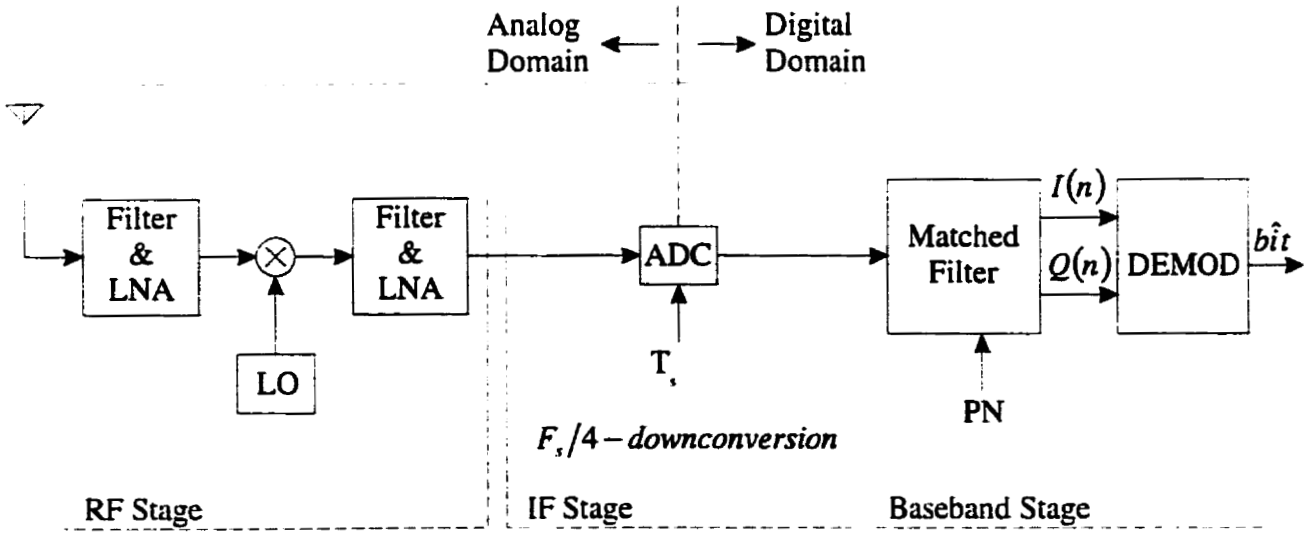


Figure 1.3 The block diagram of the proposed IF-sampling receiver.

This thesis is divided into four sections. The first section (Chapter 2) reviews five fundamental algorithms in an IF-sampling system. The second section discusses key issues with a previous study on low complexity IF-sampling systems. The second section is covered in Chapter 3. The third section deals with the solution to the problems in Chapter 3 and is covered in Chapter 4. The last section focuses on the implementation of the proposed IF-sampling receiver and is covered in Chapter 5. Finally, a summary of this thesis and potential future work is covered in Chapter 6.

Chapter 2 Fundamental Theory of the Proposed IF-sampling Receiver

Generally, a stationary wireless communication channel has two major characteristics, namely amplitude distortion and phase distortion. To compensate for these distortions without introducing a huge complexity to the receiver structure, $F_s/4$-downconversion can be used along with $\pi/4$-shifted Differential Quadrature Phase Shift Keying (DQPSK), and Direct Sequence Spread Spectrum (DS/SS). The $\pi/4$-shifted DQPSK and the DS/SS are used to suppress the unwanted channel characteristics. The $F_s/4$-downconversion is a method which allows a direct analog to digital conversion on the IF signal. Hence, it reduces the receiver complexity. The main objective of this chapter is to review these three algorithms. In addition, the Nyquist sampling theorem and the Nyquist pulse for minimum Intersymbol Interference (ISI) are also discussed in this chapter in order to support the $F_s/4$-downconversion.

2.1 π/4-Shifted-DQPSK

The $\pi/4$-shifted-DQPSK is a digital modulation scheme that is widely used in digital cellular systems [11]-[15]. It is a bandwidth efficient modulation scheme that has bandwidth efficiency ranging from 1 bit/sec/Hz (100% excess bandwidth signaling pulse) to 2 bit/sec/Hz (0% excess bandwidth signaling pulse). Notice that the bandwidth efficiency is calculated by using the bandpass bandwidth. Since $\pi/4$-shifted-DQPSK is a

type of phase modulation scheme, it is insensitive to channel attenuation. In addition, this modulation scheme can be demodulated using a differential detector which removes any constant phase error at the receiver. As a result, it allows the use of a non-coherent detector so that the receiver cost can be greatly reduced. The advantages of the $\pi/4$-shifted-DQPSK can simplify the structure of an IF-sampling receiver. Consequently, it is adopted throughout the thesis. This section will look at the modulation scheme, the detection algorithm and the limitations of the $\pi/4$-shifted-DQPSK.

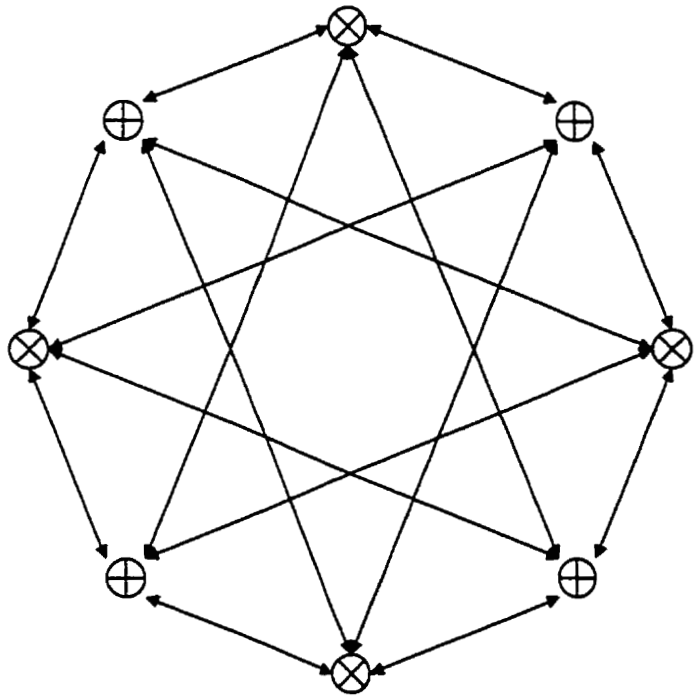


Figure 2.1.1 Signal constellation at the output of the modulator.

2.1.1 Modulation Scheme for the π/4-Shifted-DQPSK

The signal constellation at the output of the modulator for the $\pi/4$-shifted-DQPSK is shown in Figure 2.1.1. The signal constellation is defined as all the possible absolute phases at the output of the modulator. In the $\pi/4$-shifted-DQPSK modulator, a pair of information bits (dibit) is mapped onto differentially encoded signal phases, $\Delta\phi_n$, using Gray code as illustrated in Table 2.1.1.1.

Most Significant Bit U_{MSB}	Least Significant Bit U_{LST}	Current Differential Phase $\Delta\phi_n$
0	0	$\pi/4$
1	0	$3\pi/4$
1	1	$-3\pi/4$
0	1	$-\pi/4$

Table 2.1.1.1 Table of differential phases according to Gray coding.

Consequently, the signal constellation at the output of the demodulator only contains the four differential phases in Table 2.1.1.1 as shown in Figure 2.1.1.1. Gray coding [16, pp. 201] ensures that there is only one bit change across adjacent dibits. When there is a demodulation error, the output phase is most likely to shift to the neighboring dibits in the output signal constellation.

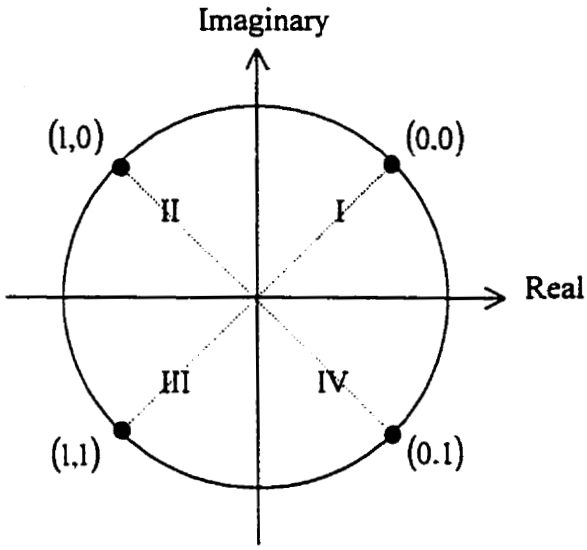


Figure 2.1.1.1 Signal constellation at the output of demodulator.

Hence, Gray coding can minimize the output bit error probability. An absolute phase is obtained as

$$\phi_n = \Delta\phi_n + \phi_{n-1} \qquad (2.1.1.1)$$

where ϕ_{n-1} is the previous absolute phase and it can be initially any phase in the signal constellation. The absolute phase is usually represented by a complex symbol, that is,

$$S[n] = I_d[n] + jQ_d[n] = e^{j\phi_n} \qquad (2.1.1.2)$$

where

$$I_d[n] = \cos(\phi_n) \equiv \text{I symbol}$$

$$Q_d[n] = \sin(\phi_n) \equiv \text{Q symbol}$$

All the possible values of $I_d[n]$ and $Q_d[n]$ for different absolute phases are summarized in Table 2.1.1.2.

Absolute Phase ϕ_n	Inphase Symbol $I_d[n]$	Quadrature Symbol $Q_d[n]$
0	1	0
$\pi/4$	0.7071	0.7071
$\pi/2$	0	1
$3\pi/4$	-0.7071	0.7071
π	-1	0
$-3\pi/4$	-0.7071	-0.7071
$-\pi/2$	0	-1
$-\pi/4$	0.7071	-0.7071

Table 2.1.1.2 Table of $I_d[n]$ and $Q_d[n]$ versus ϕ_n.

2.1.2 Detection Algorithm for the π/4-Shifted-DQPSK

At the receiver end, a complex symbol, $S'[n]$, is formed by combining an output sample from the I channel with the corresponding output sample from the Q channel. This means that the output complex symbol is given as

$$S'[n] = I'_d[n] + jQ'[n] = e^{j\phi'_n} \qquad (2.1.2.1)$$

where

$I'_d[n] \equiv$ received I sample

$Q'_d[n] \equiv$ received Q sample

$\phi'_n \equiv$ received absolute phase

The received absolute phase consists of the transmitted absolute phase and the channel phase, that is,

$$\phi'_n = \phi_n + \phi_c \qquad (2.1.2.2)$$

where ϕ_c is the constant phase error introduced by the communication channel. To retrieve the information dibit, a differential (i.e. non-coherent) detector is used. The differential detector, shown in Figure 2.1.2.1, consists of a delay, a complex conjugate operator, a multiplier and two decision devices.

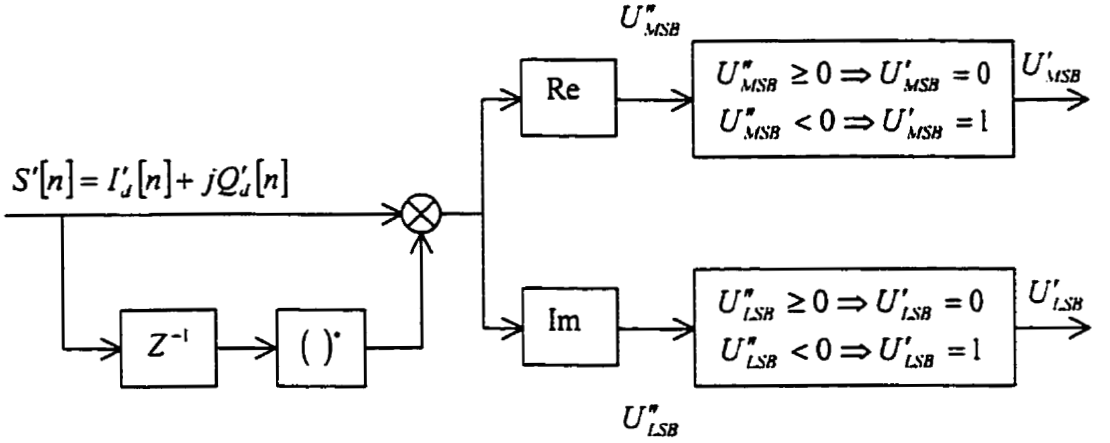


Figure 2.1.2.1 Differential detector for $\pi/4$-shifted-DQPSK.

Mathematically, the differential detection is achieved by multiplying the current complex symbol to the complex conjugate of the previous complex symbol [17, pp. 274], that is.

$$\begin{aligned} S'[n]\cdot S'^{*}[n-1] &= (I'_d[n]+jQ'_d[n])\cdot(I'_d[n-1]-jQ'_d[n-1]) \\ &= \left(e^{j\phi'_n}\right)\cdot\left(e^{-j\phi'_{n-1}}\right) \\ &= e^{j(\phi'_n-\phi'_{n-1})} \\ &= e^{j(\phi_n+\phi_c-\phi_{n-1}-\phi_c)} \\ &= e^{j(\Delta\phi_n+\phi_{n-1}-\phi_{n-1})} \\ &= e^{j\Delta\phi_n} \\ &= \cos(\Delta\phi_n)+j\sin(\Delta\phi_n) \end{aligned} \qquad (2.1.2.3)$$

where $(\ )^*$ is the complex conjugate operator. As a result, the information dibit can be retrieved from the output complex symbol of the differential detector, and the phase error caused by the non-ideal channel is successfully removed. To retrieve the most significant bit, the decision rule is

$$U'_{MSB}=\begin{cases}0, & \mathrm{Re}\left(S'[n]\cdot S'^{*}[n-1]\right)\geq 0\\ 1, & \mathrm{Re}\left(S'[n]\cdot S'^{*}[n-1]\right)<0\end{cases} \qquad (2.1.2.4)$$

Similarly, the decision rule for retrieving the least significant bit is

$$U'_{LSB}=\begin{cases}0, & \mathrm{Im}\left(S'[n]\cdot S'^{*}[n-1]\right)\geq 0\\ 1, & \mathrm{Im}\left(S'[n]\cdot S'^{*}[n-1]\right)<0\end{cases} \qquad (2.1.2.5)$$

Notice that the decision device only cares about the sign of the real part and imaginary part of the output complex symbol from the differential detector. This means that it is not necessary to calculate the information dibit by using the trigonometric function even though the information dibit is stored inside the differential phase. This also saves a lot of hardware to implement a differential detector. Note that there are also some other types of differential detectors as shown in the literature [18][11].

2.1.3 Limitations of π/4-Shifted-DQPSK

So far $\pi/4$-shifted-DQPSK seems to be a perfect modulation scheme for reducing receiver complexity. However this come with a price. Firstly, differential detection has a 3 dB performance degradation in Signal-to-Noise Ratio (SNR) compared to coherent detection [17, pp. 274-278]. In addition, $\pi/4$-shifted-DQPSK requires the channel characteristics to remain stationary over symbol periods in order for the channel phase to remain constant between two consecutive symbols. Hence one must weigh the 3 dB loss against the reduction in implementation complexity and must be sure that the physical channel is suitable for $\pi/4$-shifted-DQPSK before making such a selection.

A common problem over the wireless problem is ISI which is caused by the amplitude and phase distortion of the channel. The following section proposes to use DS/SS to combat ISI.

2.2 DS/SS for the Proposed IF-sampling Receiver

The main characteristic of a spread-spectrum signal is that its bandwidth is much greater than the symbol rate [17, pp. 695-752][19]. With the extra bandwidth, spread-spectrum signals can be used for suppressing jamming, in-band interference and ISI. There are two types of spread-spectrum signals. The first type is called direct-sequence

(DS) and is commonly used in conjunction with Phase Shift Keying (PSK). The other type is called Frequency Hopping (FH) which is commonly used in conjunction with Frequency Shift Key (FSK). Since DS/SS is used for suppressing ISI in the proposed IF-sampling receiver, this section starts with a brief overview of ISI, then it shows that ISI is suppressed through the spreading and despreading process in a DS/SS system. Finally, it provides a discussion on the Barker code which is the PN sequence used in the proposed DS/SS modulation scheme.

2.2.1 Overview of ISI

Generally, ISI is caused by the amplitude and phase distortion that exist on some propagation channels. In theory, a distortionless channel must have a flat amplitude response. In reality, every practical communication channel has a limited bandwidth. To be able to transmit a signal faithfully over a real channel, the signal bandwidth must be smaller than or equal to the channel bandwidth [20, pp. 190-191]. Consider a baseband channel, $h(t)$, with a lowpass bandwidth, W. Let $p(t)$ be a baseband signal that has the maximum allowable bandwidth for faithful transmission over $h(t)$. Therefore the bandwidth of $p(t)$ is $\leq W$. Assume the frequency spectrum of $p(t)$ [Figure 2.2.1.1(a)] is rectangular with amplitude $\frac{1}{2W}$, which means that

$$P(f) = \Im[p(t)] = \begin{cases} \frac{1}{2W}, & |f| \le W \\ 0, & otherwise \end{cases} \qquad (2.2.1.1)$$

where $\Im[\ ]$ is the continue-time Fourier transform operator. Hence $p(t)$ [Figure 2.2.1.1(b)] is obtained by taking the inverse Fourier transform of $P(f)$ and the result is

$$p(t) = \Im^{-1}[P(f)] = \frac{\sin(\pi t/T_p)}{\pi t/T_p} \qquad (2.2.1.2)$$

where $T_p = \frac{1}{2W}$ is the symbol duration. The conclusion drawn from this is that the tails of a band-limited signaling pulse are longer than their symbol interval (i.e. $-T_p/2 \leftrightarrow T_p/2$). In the presence of sampling error, these long tails become ISI. If $p(t)$ is used to carry symbol information, then the received baseband signal is

$$\begin{aligned} r(t) &= s(t) \\ &= \sum_{k=-\infty}^{\infty} a_k \cdot p(t - t_d - kT_p) \end{aligned} \qquad (2.2.1.3)$$

where

$s(t) \equiv$ transmitted signal

$a_k \equiv$ transmitted symbols (assuming a_k starts at $k = 0$ and $a_k = 0$ for $k < 0$)

$t_d \equiv$ sampling error

$k \equiv$ symbol index

assuming that the channel is noise free with a unity gain. After the sampling process, the discrete version of $r(t)$ is

$$
\begin{aligned}
r[n] &= r(t)\Big|_{t=nT_p} = r(nT_p) \\
&= \sum_{k=-\infty}^{\infty} a_k \cdot p(nT_p - t_d - kT_p) \\
&= a_n \cdot p(-t_d) + \underbrace{\sum_{k \neq n} a_k \cdot p(nT_p - t_d - kT_p)}_{ISI}
\end{aligned}
\qquad (2.2.1.4)
$$

where $n = \{0,1,2,...,\infty\}$ is the time index. If $t_d = 0$, then $r[n]$ becomes a_n because $p(0) = 1$ and $p[(n-k)T_p] = 0$ as shown in Figure 2.2.1.1(b). If $t_d \neq 0$, then each $r[n]$ is a combination of the current and neighboring symbols and this is the type of interference called ISI. Figure 2.2.1.1(c) demonstrates the effect of the sampling error. Without sampling error, the sampling points are $\{s_0, s_1, ...\}$. At these sampling points, each sample only contains the information of the current symbol. However, the sampling points shift to right hand side by $0.5T_p$ (i.e. $t_d = 0.5T_p$) in the presence of sampling error and the resulting sampling points are $\{s_0', s_1', ...\}$. At these incorrect sampling points, every sample is corrupted by ISI. Notice that the_sampling error is caused by the misalignment between the transmitter and receiver clock.

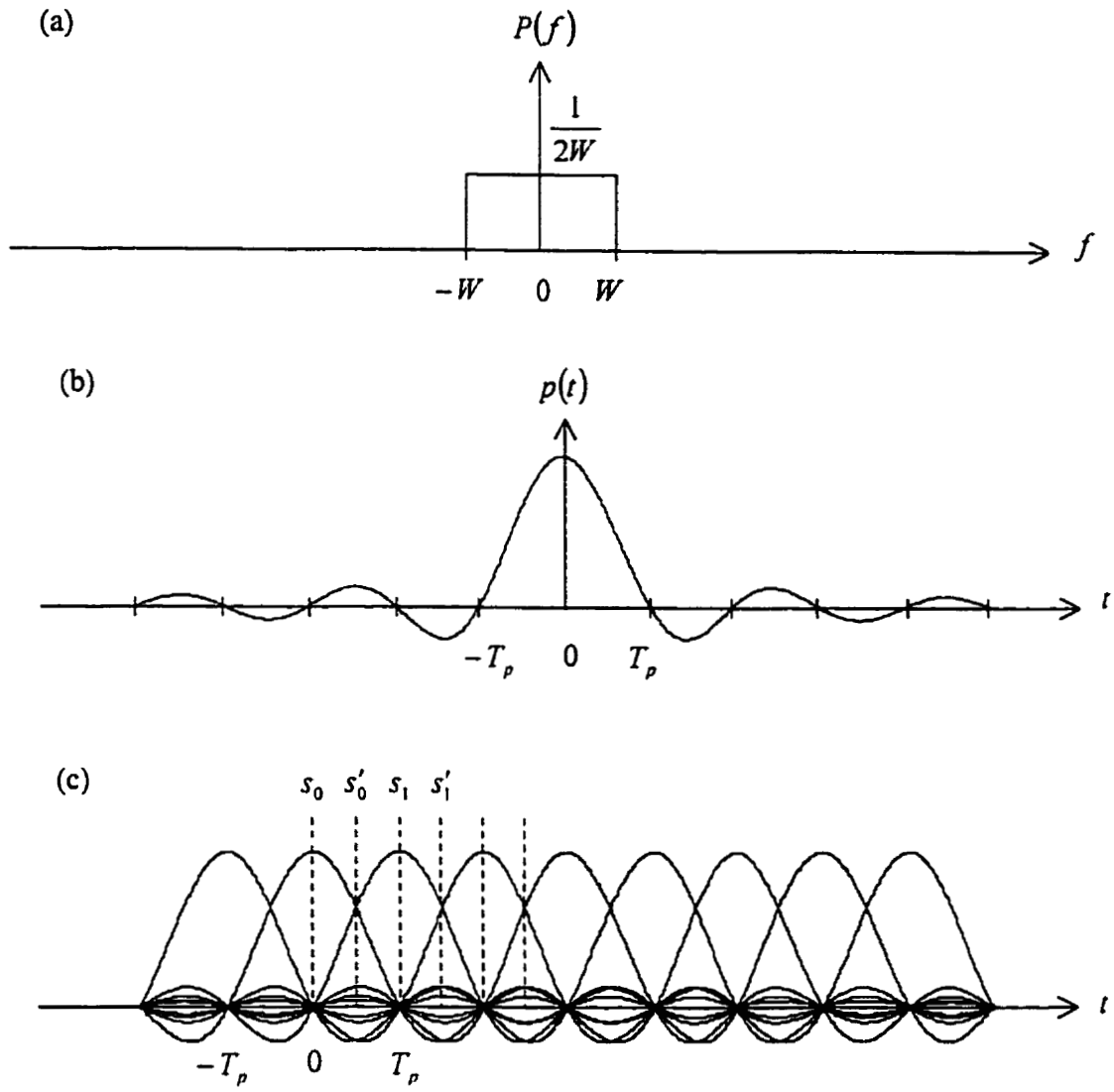


Figure 2.2.1.1 (a) Spectrum of $p(t)$.

(b) Time-domain function $p(t)$ (sinc function).

(c) Received baseband signal.

2.2.2 Suppressing ISI using a DS/SS system

In a DS/SS system, there are two additional processes in the transceiver compared to a conventional system. The two processes are called spreading and despreading. To realize the concept of spreading and despreading, the following discussion assumes that the communication channel is noise free and the channel attenuation is zero.

Spreading is performed inside the transmitter in which each transmitted symbol is multiplied by a PN sequence to form an output sequence [21]. Each element in the output sequence is called a chip. Mathematically, the spreading process can be expressed as

$$b_n = \sum_{l=-\infty}^{\infty} a_l \cdot c[n - lM] \qquad (2.2.2.1)$$

where

$b_n \equiv$ the output sequence

$a_l \equiv$ the transmitted symbol

$c[n] \equiv$ the PN sequence and $c[n] \neq 0$ if $lM \leq n < lM + M - 1$

$M \equiv$ the PN sequence length

$l \equiv$ the symbol index

$n \equiv$ the chip index

This equation corresponds to two functional blocks in a transmitter, namely, up-sampler and multiplier, as shown in Figure 2.2.2.1.

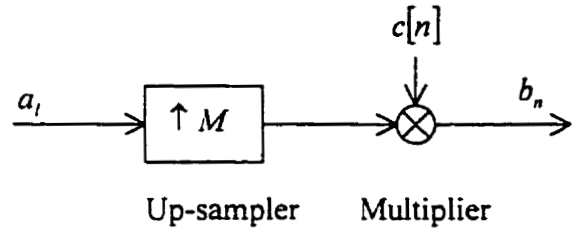


Figure 2.2.2.1 The spreading process.

Since each transmitted symbol becomes a sequence of chips, the data rate is increased by a factor of M after the spreading process. Consequently, a DS/SS system requires M times the bandwidth of a conventional system.

On the other hand, the despreading process is performed by the receiver in which a symbol is reconstructed from the received chips through the matched filtering and the decimation processes as shown in Figure 2.2.2.2.

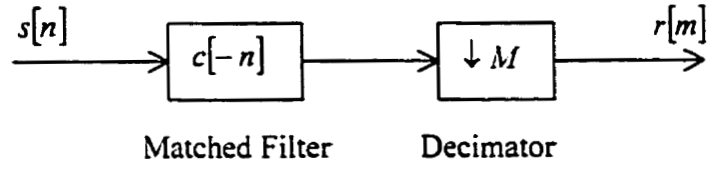


Figure 2.2.2.2 The despreading process.

Mathematically, the despreading process can be represented by

$$\begin{aligned} r[m] &= \sum_{n=-\infty}^{\infty} s[n] \cdot c[n - mM] \\ &= \sum_{n=-\infty}^{\infty} b_n \cdot c[n - mM] \end{aligned} \qquad (2.2.2.2)$$

where $s[n]$ is the transmitted signal sample assuming no pulse shaping. By substituting b_n from Equation 2.2.2.1 into Equation 2.2.2.2, the reconstructed symbol is

$$r[m] = \sum_{n=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} a_l \cdot c[n - lM] \cdot c[n - mM] \qquad (2.2.2.3)$$

Since $c[n]$ only exists from $n = lM$ to $n = lM + M - 1$, the product $c[n - lM] \cdot c[n - mM]$ is non-zero only when $l = m$. Therefore Equation 2.2.2.3 becomes

$$\begin{aligned} r[m] &= \sum_{n=-\infty}^{\infty} a_m \cdot c[n - mM] \cdot c[n - mM] \\ &= a_m \cdot \sum_{n=-\infty}^{\infty} c[n - mM] \cdot c[n - mM] \end{aligned} \qquad (2.2.2.4)$$

The term $\sum_{n=-\infty}^{\infty} c[n - mM] \cdot c[n - mM]$ in Equation 2.2.2.4 is a special case in the autocorrelation of $c[n]$. The autocorrelation of a PN sequence is defined as

$$\begin{aligned} \phi_c[m] &= \sum_{n=-\infty}^{\infty} c[n] \cdot c[n - m] \\ &= \begin{cases} M, & m = 0 \\ x, & otherwise \end{cases} \end{aligned} \qquad (2.2.2.5)$$

where

$$|x| \le x_{max} \text{ and } x_{max} = \begin{cases} M - 1. & real \quad PN \quad sequence \\ 0. & perfect \quad PN \quad sequence \end{cases}$$

As a result, Equation 2.2.2.4 and 2.2.2.5 implies that $\sum_{n=-\infty}^{\infty} c[n-mM]\cdot c[n-mM]=M$ and the output of the despreading process becomes

$$r[m]=Ma_m \tag{2.2.2.6}$$

where M is also called the processing gain of the PN sequence. This shows that the despreading process cancels the effect of the spreading process. Hence there is no change to the communication system after adding a spreading and a despreading process.

In order to show that spreading and despreading can suppress ISI, it is convenient to use the perfect PN sequence to start with. Generally, the transmitted signal can be represented by Equation 2.2.1.3. After spreading, a_k in Equation 2.2.1.3 is replaced by b_n in Equation 2.2.2.1 so that the transmitted signal becomes

$$s(t)=\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} a_l\cdot c[k-lM]\cdot p(t-t_d-kT_c) \tag{2.2.2.7}$$

where $T_c=\frac{1}{M}T_p$ is the chip duration. Therefore the sampled transmitted signal is

$$\begin{aligned} s[n] &= s(t)\big|_{t=nT_c}=s(nT_c)\\ &=\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} a_l\cdot c[k-lM]\cdot p(nT_c-t_d-kT_c)\\ &=\sum_{l=-\infty}^{\infty} a_l\cdot c[n-lM]\cdot p(-t_d)\\ &+\sum_{k\neq n}\sum_{l=-\infty}^{\infty} a_l\cdot c[k-lM]\cdot p(nT_c-t_d-kT_c) \end{aligned} \tag{2.2.2.8}$$

Let $i=n-k$, then Equation 2.2.2.8 becomes

$$\begin{aligned} s[n] &=\sum_{l=-\infty}^{\infty} a_l\cdot c[n-lM]\cdot p(-t_d)\\ &+\sum_{i=0}\sum_{l=-\infty}^{\infty} a_l\cdot c[n-i-lM]\cdot p(iT_c-t_d)\Rightarrow ISI \end{aligned} \tag{2.2.2.9}$$

Substituting Equation 2.2.2.9 into Equation 2.2.2.2 and applying the autocorrelation property of the PN sequence (Equation 2.2.2.5), the received symbol is

$$\begin{aligned} r[m] &= \sum_{n=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} a_l \cdot c[n-lM]\cdot p(-t_d)\cdot c[n-mM] \\ &+ \sum_{n=-\infty}^{\infty}\sum_{i\neq 0}\sum_{l=-\infty}^{\infty} a_l \cdot c[n-i-lM]\cdot p(iT_c - t_d)\cdot c[n-mM] \\ &= a_m \cdot p(-t_d)\cdot \sum_{n=-\infty}^{\infty} c[n-mM]\cdot c[n-mM] \\ &+ a_m \cdot p(iT_c - t_d)\cdot \sum_{n=-\infty}^{\infty}\sum_{i\neq 0} c[n-i-mM]\cdot c[n-mM] \\ &= Ma_m \cdot p(-t_d) \\ &+ xa_m \sum_{i=1}^{M-1} p(iT_c - t_d) \Rightarrow ISI \end{aligned} \qquad (2.2.2.10)$$

The ISI term, $xa_m \sum_{i=1}^{M-1} p(iT_c - t_d)$, is zero since $x = 0$ for the perfect PN sequence. Hence $r[m] = Ma_m \cdot p(-t_d)$ and the ISI is completely removed. If a non-perfect PN sequence is used then x is not equal to zero. However, if $\max(|x|) << M$, the ISI term in Equation 2.2.2.10 will be significantly reduced.

It is very important to realize that the autocorrelation of the PN sequence plays a significant role in suppressing the ISI. In Equation 2.2.2.10, the tails from the neighboring symbols are attenuated by x and the desired symbol is amplified by M. As a result, a good PN sequence (i.e. with a large $M/\max(|x|)$ ratio) can enhance the system performance.

2.2.3 Barker Codes

Barker codes are widely used in the existing digital communication systems for synchronization and signal detection. This is because any Barker code has an autocorrelation function which approximates that of a perfect PN sequence,

$$\phi_{Barker}[m] = \begin{cases} M, & m = 0 \\ -1, & otherwise \end{cases} \qquad (2.2.3.1)$$

Equation 2.2.3.1 can also be obtained from [22]. The ratio $M/\max(|x|)$ for a Barker code is $M/1 = M$ which is the upper limit for a real PN sequence. This autocorrelation property is also shown in Figure 2.2.3.1

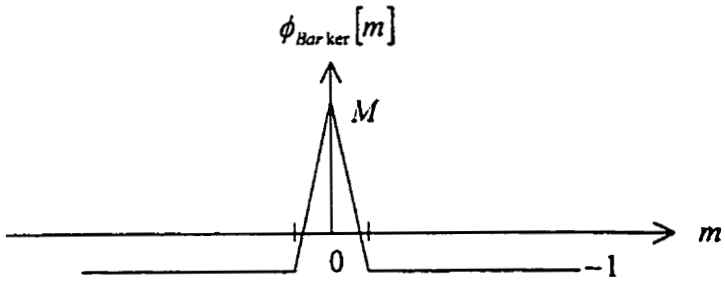


Figure 2.2.3.1 The autocorrelation of the Barker code.

The most commonly used Barker code has a sequence length of 11 (i.e. $M = 11$) and is given as:

$$\{b_0, b_1, \ldots, b_{M-1}\} = \{+1,-1,+1,+1,-1,+1,+1,+1,-1,-1,-1\} \qquad (2.2.3.2)$$

where the left most chip is transmitted first. Throughout the thesis, we use this Barker code for the proposed IF-sampling receiver. According to the $\pi/4$-shifted DQPSK signal constellation in Figure 2.1.1.1, the amplitudes of I and Q samples are limited to $\{\pm 1, \pm 0.7071, 0\}$. In both I and Q channels, the only possible combinations of two consecutive samples, (a_i, a_{i+1}), are $p_1 = (1,0.7)$, $p_2 = (1,-0.7)$, $p_3 = (0.7,0)$, $p_4 = (0.7,1)$, $p_5 = (0.7,-1)$, $p_6 = (0,-0.7)$, $p_7 = (0,0.7)$, $p_8 = (-0.7,-1)$, $p_9 = (-0.7,0)$, $p_{10} = (-0.7,1)$, $p_{11} = (-1,-0.7)$, and $p_{12} = (-1,0.7)$. When a matched filter is used to correlate the Barker code as shown in Figure 2.2.3.2, all possible output values from the matched filter, $y[m]$, are listed in Table 2.2.3.1.

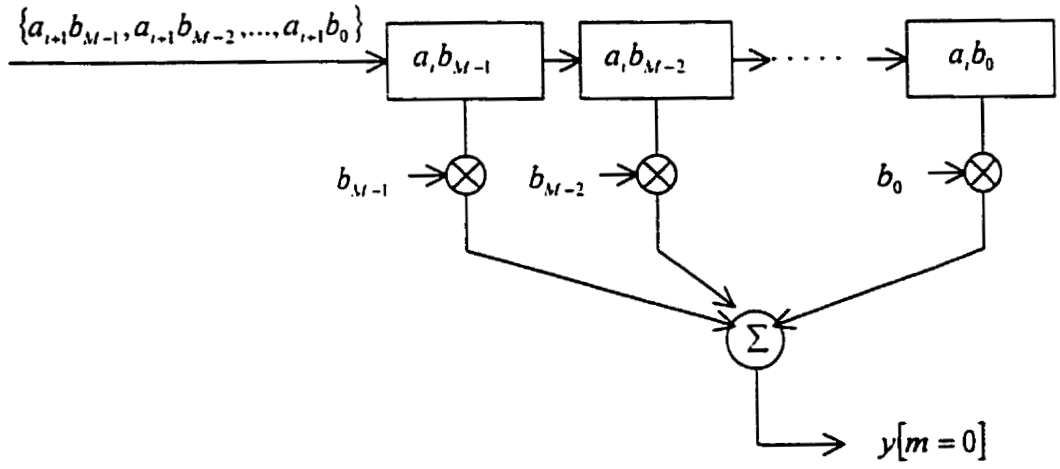


Figure 2.2.3.2 The matched filter for Barker correlation.

m	$y[m]$											
	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
0	11	11	7.7	7.7	7.7	0	0	-7.7	-7.7	-7.7	-11	-11
1	-0.7	0.7	0	-1	1	0.7	-0.7	1	0	-1	0.7	-0.7
2	-1	-1	-0.7	-0.7	-0.7	0	0	0.7	0.7	0.7	1	1
3	-0.7	0.7	0	-1	1	0.7	-0.7	1	0	-1	0.7	-0.7
4	-1	-1	-0.7	-0.7	-0.7	0	0	0.7	0.7	0.7	1	1
5	-0.7	0.7	0	-1	1	0.7	-0.7	1	0	-1	0.7	-0.7
6	-1	-1	-0.7	-0.7	-0.7	0	0	0.7	0.7	0.7	1	1
7	-0.7	0.7	0	-1	1	0.7	-0.7	1	0	-1	0.7	-0.7
8	-1	-1	-0.7	-0.7	-0.7	0	0	0.7	0.7	0.7	1	1
9	-0.7	0.7	0	-1	1	0.7	-0.7	1	0	-1	0.7	-0.7
10	-1	-1	-0.7	-0.7	-0.7	0	0	0.7	0.7	0.7	1	1

Table 2.2.3.1 All possible matched filter outputs for $\pi/4$-shifted DQPSK symbols.

The values in Table 2.2.3.1 are obtained according to the following example.

Assuming the current and the next symbols come into the matched filter corresponding to the constellation point p_1. Where the current symbol is $a_i = 1$. which corresponds to the I symbol. and the next symbol is $a_{i+1} = 0.7$. which corresponds to the Q symbol. the current chip sequence is

$$\{a_i b_0, a_i b_1, \ldots, a_i b_{M-1}\} = \{1, -1, 1, 1, -1, 1, 1, 1, -1, -1, -1\}$$

and the next chip sequence is

$$\{a_{i+1} b_0, a_{i+1} b_1, \ldots, a_{i+1} b_{M-1}\} = \{0.7, -0.7, 0.7, 0.7, -0.7, 0.7, 0.7, 0.7, -0.7, -0.7, -0.7\}$$

When all chips of the current symbol are shifted into the matched filter, the output of the matched filter is

$$y[m=0] = a_i b_0 b_0 + a_i b_1 b_1 + \ldots + a_i b_{M-1} b_{M-1}$$
$$= (1)(1) + (-1)(-1) + \ldots + (-1)(-1) = 11$$

In the next clock cycles, the output of the matched filter is

$$y[m=1] = a_i b_1 b_0 + a_i b_2 b_1 + \ldots + a_i b_{M-1} b_{M-2} + a_{i+1} b_0 b_{M-1}$$
$$= (1)(-1) + (-1)(1) + \ldots + (-1)(-1) + (0.7)(-1) = -0.7$$

and so on.

The values in Table 2.2.3.1 show that the matched filter output is limited to $\{\pm a_i M, \pm a_i, \pm a_{i+1}, 0\}$ if the Barker code is used for spreading.

The limitation of the Barker code is that the length of the longest Barker sequence is 13. In fact, it has been hypothesized that the only possible Barker sequence lengths are 1, 2, 3, 4, 5, 7, 11, and 13 [22][23, pp. 564]. Therefore the processing gain of the code is always less than or equal to 13 which may not be enough for some applications.

2.3 F_S/4-Downconversion

To perform IF-sampling, the sampling frequency, F_s, of the sampler must be proportional to the IF carrier frequency, f_{IF}. If $f_{IF} = (2n+1)F_s/4$, where $\{n \in I, 0 \le n \le \infty\}$, $F_s/4$-downconversion [4][5][24] is the term to describe the series of

processes that convert an analog IF signal to a digital baseband signal. This is the most common downconversion technique used in existing IF-sampling systems. There are two ways to choose F_s according to the Nyquist sampling theorem. One way is determined from the highest frequency component and the other is from the signal bandwidth. In this section, n is set to zero which means that the received IF signal is treated as a lowpass signal and F_s is chosen according to the highest frequency component. In order to show that IF-sampling can replace baseband sampling, it is necessary to show the conditions under which both sampling methods produce the same output signal. Verifying such conditions can be achieved either in the time-domain or in the frequency-domain. The time-domain analysis shows that the two methods produce the same I and Q output samples. Whereas, the frequency-domain analysis shows the output signals from the two methods have the same frequency components. To simplify the analysis, an ideal communication channel is used. The ideal channel, shown in Figure 2.3.1, is defined as a channel with a unity gain.

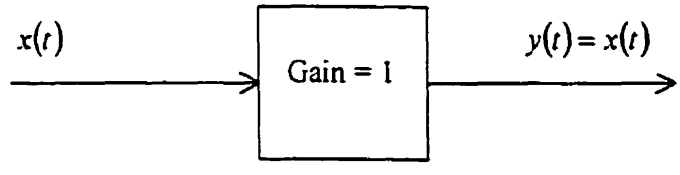


Figure 2.3.1 The block diagram of an ideal channel.

With this channel, the received IF signal at the receiver can be expressed as

$$r_{IF}(t) = I(t)\cdot\cos(2\pi f_{IF}t) - Q(t)\cdot\sin(2\pi f_{IF}t) \qquad (2.3.1)$$

where

$\cos(2\pi f_{IF}t) \equiv$ cosine carrier

$\sin(2\pi f_{IF}t) \equiv$ sine carrier

$I(t) \equiv$ I signal

$Q(t) \equiv$ Q signal

2.3.1 Time-Domain Analysis

$F_s/4$-downconversion consists of three processes, namely sampling, sign correction, and sorting. In the sampling process, $r_{IF}(t)$ is sampled by a sampler at a rate of $F_s = 4f_{IF}$. Therefore, the sampling period is

$$T_s = \frac{1}{F_s} = \frac{1}{4f_{IF}} \qquad (2.3.1.1)$$

and the output of the sampler is

$$r[n] = r_{IF}(t)\big|_{t=nT_s} \qquad (2.3.1.2)$$

where $n = \{0,1,2,3,\ldots,\infty\}$. By substituting Equation 2.3.1 and Equation 2.3.1.1 into Equation 2.3.1.2, the output of the sampler can be represented as

$$\begin{aligned} r_{IF}[n] &= I_d[n]\cdot\cos\left(n\frac{\pi}{2}\right) - Q_d[n]\cdot\sin\left(n\frac{\pi}{2}\right) \\ &= \{I_d[0], -Q_d[1], -I_d[2], Q_d[3], \ldots\} \end{aligned} \qquad (2.3.1.3)$$

where

$I_d[n] = I(t = nT_s) \equiv$ discrete-time I signal

$Q_d[n] = Q(t = nT_s) \equiv$ discrete-time Q signal

As a result, $r[n]$ consists of samples of $I(t)$ interleaved by samples of $Q(t)$ with sign inversion introduced on some of the samples. This special result is achieved through the orthogonality property between the sine and cosine carriers. At the sampling points, the sine and cosine carriers are periodic sequences, that is,

$$\cos(2\pi f_{IF} t)\Big|_{t=nT_s=\frac{n}{4f_{IF}}} = \cos\left(n\frac{\pi}{2}\right) = \{1,0,-1,0,.....\} \qquad (2.3.1.4)$$

and

$$\sin(2\pi f_{IF} t)\Big|_{t=nT_s=\frac{n}{4f_{IF}}} = \sin\left(n\frac{\pi}{2}\right) = \{0,1,0,-1,0,.....\} \qquad (2.3.1.5)$$

At any sampling point, only one of the two sequences contains a non-zero value. Therefore, these two sequences are orthogonal to each other at the sampling points. The orthogonality property is also shown graphically in Figure 2.3.1.1.

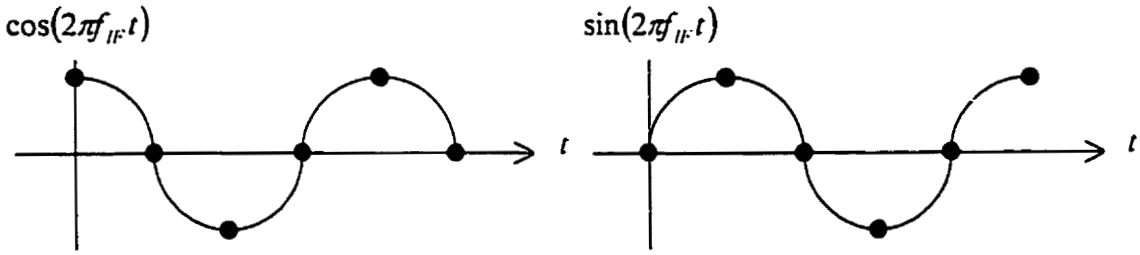


Figure 2.3.1.1 Graphical representation on the effect of sampling the cosine and sine carriers with $F_s = 4f_{IF}$.

This unique property ensures that each of the output samples from the sampler contains the information about either the inphase or the quadrature signal, but not about both.

The sign correction on $r[n]$ can be achieved simply by multiplying $r[n]$ with $\sqrt{2}\cos\left(n\frac{\pi}{2}+\frac{\pi}{4}\right)=\{1,-1,-1,1,1,-1,.....\}$. This sign correction sequence is unique for $f_{IF}=(4n+1)F_s/4$. Finally, $I_d[n]$ and $Q_d[n]$ are obtained through sorting (i.e. decimation). As a result, $F_s/4$-downconversion produces the same output samples for the I and Q channels as comparing to the baseband sampling. A summary of all the processes in the $F_s/4$-downconversion is shown in Figure 2.3.1.2.

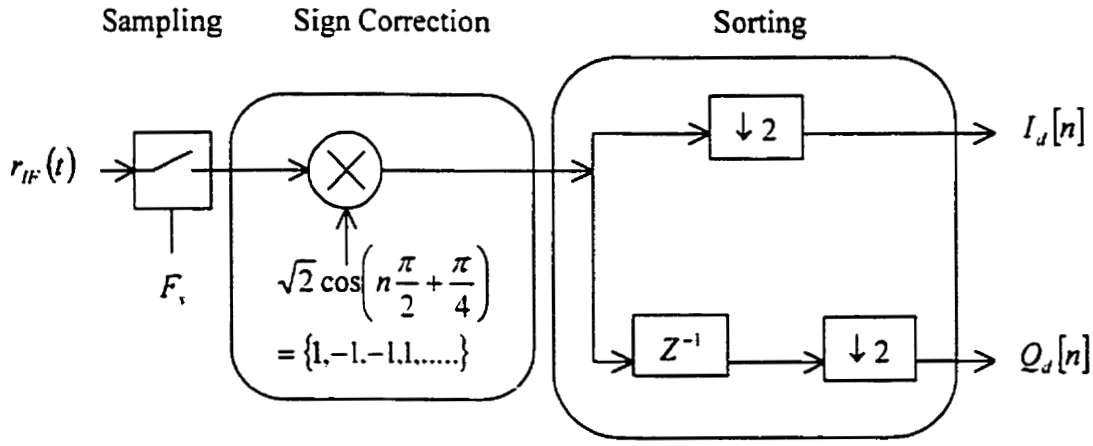


Figure 2.3.1.2 The signal processes in the $F_s/4$-downconversion.

The drawback of $F_s/4$-downconversion is that the I and Q samples are obtained at different time instants. This means that there is a timing misalignment between the I and Q samples [3][8] which, ideally, should be taken at the same instant in time. There are many ways to solve this timing misalignment problem. The approach that is used in this thesis is discussed in Chapter 3.

2.3.2 Frequency-Domain Analysis

To perform the frequency-domain analysis, the first step is to understand the sampling theorem [20, pp. 91]. Assuming there is a lowpass signal, $x(t)$, which has a bandwidth $W/2$ as shown in Figure 2.3.2.1.

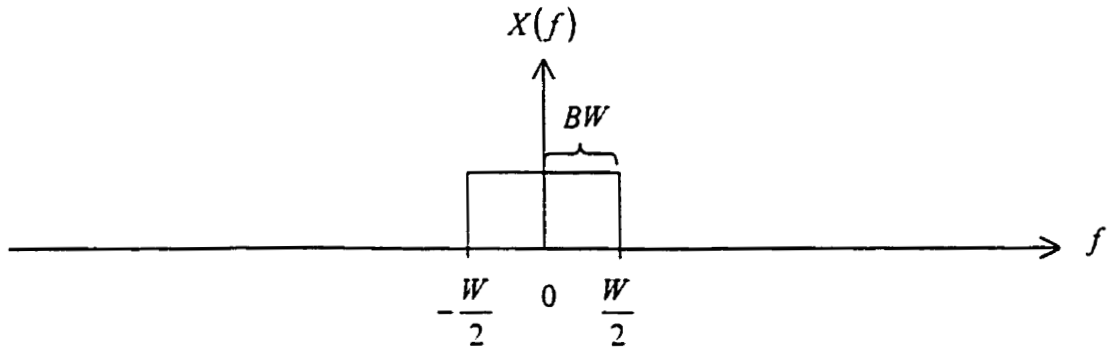


Figure 2.3.2.1 The frequency spectrum of $x(t)$ (BW is the lowpass bandwidth).

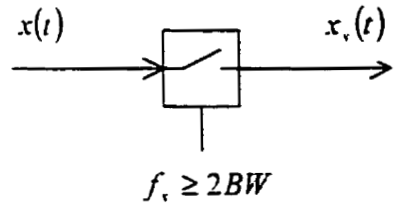


Figure 2.3.2.2 The Nyquist sampler.

In Figure 2.3.2.2, a Nyquist sampler is used to digitize the entire frequency band from direct current (DC) to $f_s/2$, where f_s is the sampling frequency of the sampler and must be greater than or equal to W according to the Nyquist sampling theorem. Then the output of the sampler in the time-domain is

$$\begin{aligned} x_s(t) &= x(t)\sum_{n=-\infty}^{\infty}\delta(t-nT_s) \\ &= \sum_{n=-\infty}^{\infty}x(nT_s)\cdot\delta(t-nT_s) \end{aligned} \qquad (2.3.2.1)$$

where

$T_s = 1/f_s \equiv$ sampling period, and

$\delta(t) \equiv$ continuous-time impulse function.

To view the frequency components of $x_s(t)$ requires $X_s(f)$, the continuous-time Fourier transform of $x_s(t)$,

$$\begin{aligned} X_s(f) &= \Im[x_s(t)] \\ &= \Im[x(t)]\cdot\Im\left[\sum_{n=-\infty}^{\infty}\delta(t-nT_s)\right] \end{aligned} \qquad (2.3.2.2)$$

Equation 2.3.2.2 can be simplified as

$$\begin{aligned} X_s(f) &= X(f)\cdot f_s\sum_{n=-\infty}^{\infty}\delta(f-nf_s) \\ &= f_s\sum_{n=-\infty}^{\infty}X(f-nf_s) \end{aligned} \qquad (2.3.2.3)$$

according to the sifting property of the impulse. Equation 2.3.2.3 states that $X_s(f)$ is $X(f)$ repeated at every nf_s. The repeated frequency components are called sampling images. This result is also shown graphically in Figure 2.3.2.3.

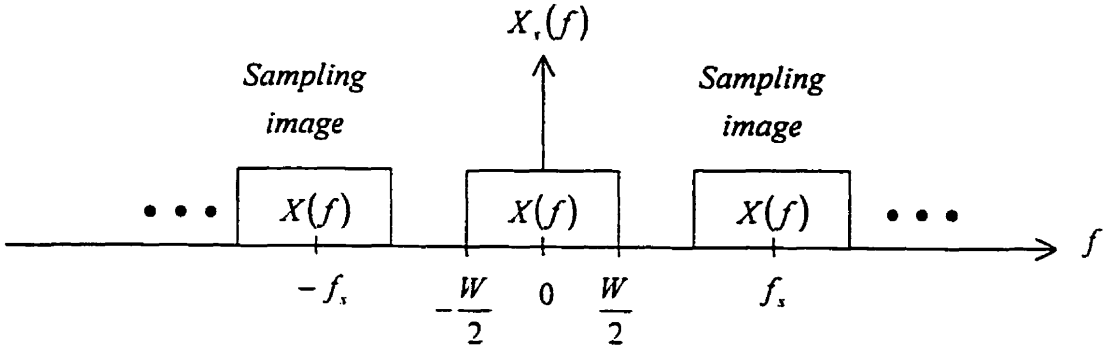


Figure 2.3.2.3 The frequency spectrum of $x_s(t)$.

To carry out our frequency domain analysis, we assume that both $I(t)$ and $Q(t)$ are lowpass signals with a bandwidth of $W/2$. This means that $r_{IF}(t)$ is a bandpass signal, which has a carrier frequency f_{IF} and a null-to-null bandpass bandwidth W. The frequency spectrum of $r_{IF}(t)$ is shown in Figure 2.3.2.4(a). In the baseband sampling (Figure 1.1), $r_{IF}(t)$ is first multiply by $2\cos(2\pi f_{IF}t)$ in the I channel to form an intermediate signal $I'(t)$, where $I'(t)$ can be expressed as

$$\begin{aligned} I'(t) &= r_{IF}(t)\cdot 2\cos(2\pi f_{IF}t) \\ &= [I(t)\cdot\cos(2\pi f_{IF}t) - Q(t)\cdot\sin(2\pi f_{IF}t)]\cdot 2\cos(2\pi f_{IF}t) \\ &= I(t)\cdot\cos(2\pi f_{IF}t)\cdot 2\cos(2\pi f_{IF}t) - Q(t)\cdot\sin(2\pi f_{IF}t)\cdot 2\cos(2\pi f_{IF}t) \\ &= I(t) + \underbrace{I(t)\cdot\cos(4\pi f_{IF}t) - Q(t)\cdot\sin(4\pi f_{IF}t)}_{\text{Double Frequency Terms}} \end{aligned} \qquad (2.3.2.4)$$

This process is called the mixing operation and the resulting signal has a frequency spectrum as shown in Figure 2.3.2.4(b). Then $I'(t)$ is filtered by a lowpass filter to remove the double frequency terms. Hence, the output lowpass signal is

$$I(t) = LPF[I'(t)] \tag{2.3.2.5}$$

where $LPF[\ ] \equiv$ lowpass filtering operator. The frequency spectrum of $I(t)$ is shown in Figure 2.3.2.4(c). Finally, $I(t)$ is sampled by the Nyquist sampler to form $I_d[n]$ and the spectrum of $I_d[n]$ is shown in Figure 2.3.2.4(d).

In $F_s/4$-downconversion. $r_{IF}(t)$ is treated as a lowpass signal and is sampled directly by the Nyquist sampler during the sampling process. The output of this process is $r_{IF}[n]$. which has a frequency spectrum shown in Figure 2.3.2.5(a) according to the sampling theorem. In the sign correction process, $r_{IF}[n]$ is multiply by $\sqrt{2}\cos\left(n\frac{\pi}{2}+\frac{\pi}{4}\right)$ to produce an intermediate signal

$$r'_{IF}[n] = r_{IF}[n]\cdot\sqrt{2}\cos\left(n\frac{\pi}{2}+\frac{\pi}{4}\right) \tag{2.3.2.6}$$

To examine the spectral content of $r'_{IF}[n]$, we calculate the discrete-time Fourier transform of $r'_{IF}[n]$:

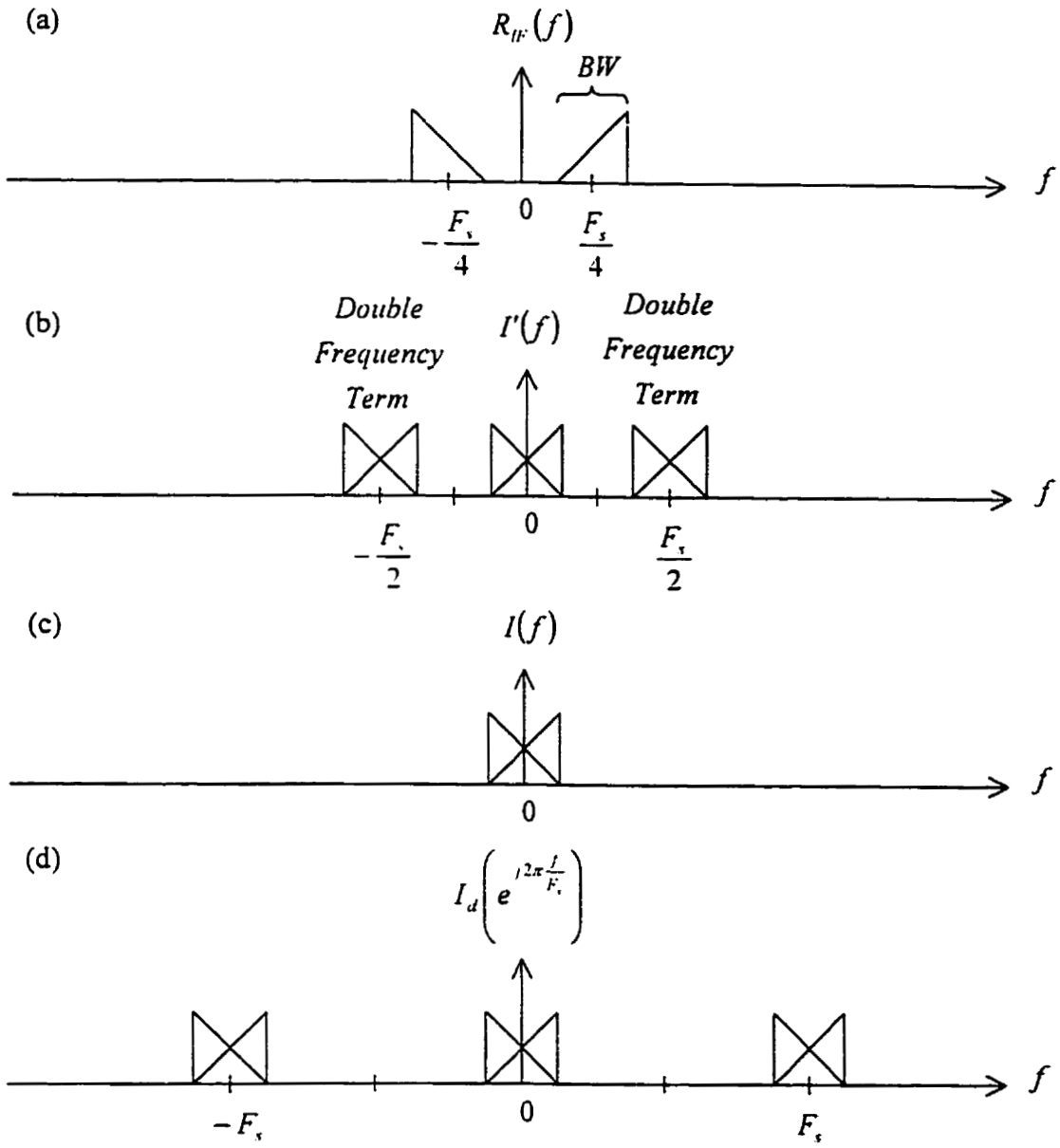


Figure 2.3.2.4 (a) Spectrum of $r_{IF}(t)$ ($BW = W$ is the bandpass bandwidth).

(b) Spectrum of $I'(t)$.

(c) Spectrum of $I(t)$.

(d) Spectrum of $I_d[n]$.

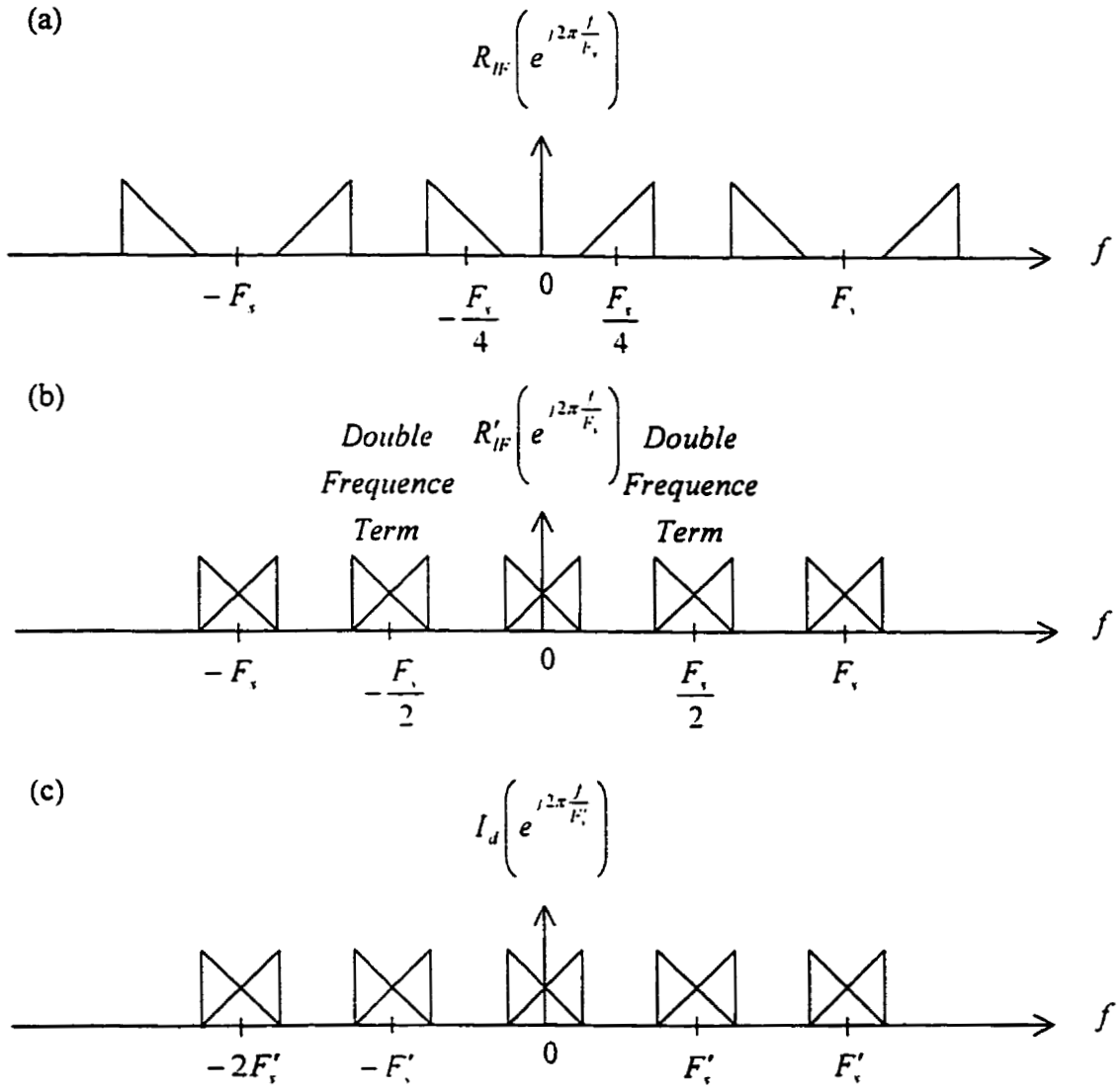


Figure 2.3.2.5 (a) Spectrum of $r_{IF}[n]$.

(b) Spectrum of $r'_{IF}[n]$.

(c) Spectrum of $I_d[n]$.

$$R'_{IF}\left(e^{j2\pi\frac{f}{F_s}}\right) = Z\{r_{IF}[n]\} \otimes Z\left\{\sqrt{2}\cos\left(n\frac{\pi}{2}+\frac{\pi}{4}\right)\right\}$$

$$= R_{IF}\left(e^{j2\pi\frac{f}{F_s}}\right) \otimes \left[\frac{\sqrt{2}}{2}\delta\left(f-\frac{F_s}{4}\right)\cdot e^{j\frac{\pi}{4}} + \frac{\sqrt{2}}{2}\delta\left(f+\frac{F_s}{4}\right)\cdot e^{-j\frac{\pi}{4}}\right]$$

$$= \frac{\sqrt{2}}{2} R_{IF}\left(e^{j\frac{2\pi}{F_s}\left(f-\frac{F_s}{4}\right)}\right)\cdot e^{j\frac{\pi}{4}}$$

$$+ \frac{\sqrt{2}}{2} R_{IF}\left(e^{j\frac{2\pi}{F_s}\left(f+\frac{F_s}{4}\right)}\right)\cdot e^{-j\frac{\pi}{4}} \qquad (2.3.2.7)$$

where

$Z\{\ \} \equiv$ discrete-time Fourier transform operator, and

$\otimes \equiv$ convolution operator.

With this equation, the graphical representation of $R'_{IF}\left(e^{j2\pi\frac{f}{F_s}}\right)$ is shown in Figure 2.3.2.5(b). The last process in the $F_s/4$-downconversion is sorting. This process performs a decimation-by-2 on $r'_{IF}[n]$. According to decimation theory [25, pp. 594], the output spectrum after a decimation-by-2 is

$$X_d\left(e^{j2\pi\frac{f}{f'_s}}\right) = X\left(e^{j2\pi\frac{f}{f_s}}\right) + X\left(e^{j2\pi\frac{f-0.5f_s}{f_s}}\right) \qquad (2.3.2.8)$$

where

$X\left(e^{j2\pi\frac{f}{f_s}}\right) \equiv$ original signal

$f'_s = \frac{1}{2} f_s$ for decimation-by-2

As a result, the spectrum of $I_d[n]$ is

$$I_d\left(e^{j2\pi\frac{f}{F_s}}\right) = R'_{IF}\left(e^{j2\pi\frac{f}{F_s}}\right) + R'_{IF}\left(e^{j2\pi\frac{f-0.5F_s}{F_s}}\right) \qquad (2.3.2.9)$$

and is plotted in Figure 2.3.2.5(c). Notice that $F_s/4$-downconversion does not require the lowpass filter which is used to remove the double frequency terms resulted from the mixing operation in the super-heterodyne receiver. This is because the double frequency terms coincide with the sampling images in $R'_{IF}\left(e^{j2\pi\frac{f}{F_s}}\right)$ and they are removed during the decimation process. Comparing the output I signals from the baseband-sampling and the $F_s/4$-downconversion, it is clear that they have the same spectral components as shown in Figure 2.3.2.4(d) and 2.3.2.5(c), respectively. Although the above analysis is only performed on the I channel, the same method can be used to analyze the Q channel and the result will be the same. Consequently, $F_s/4$-downconversion can replace the conventional baseband-sampling.

2.4 Nyquist Sampling Theorem

In the previous section, $r_{IF}(t)$ is treated as a lowpass signal during the sampling process in order to simplify the analysis. In this section, the Nyquist sampling theorem for both lowpass and bandpass signal is examined in detail.

2.4.1 Lowpass Sampling Theorem

To sample a lowpass signal. the Nyquist sampling theorem [26, pp. 86] states that f_s must be selected such that

$$f_s \geq 2f_H \tag{2.4.1.1}$$

where f_s is the sampling frequency and f_H is the highest frequency of interest in the signal to be sampled as shown in Figure 2.4.1.1.

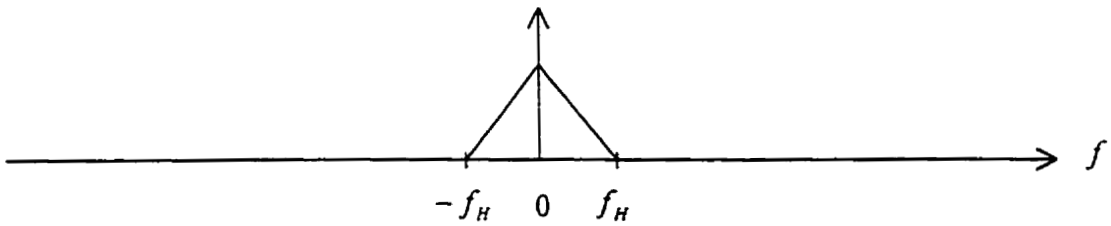


Figure 2.4.1.1 Frequency spectrum of a lowpass signal.

This relationship can guarantee that there is no aliasing. Aliasing is a term to describe the type of spectral distortion when the sampling images, created from the sampling process, overlap with each other as shown in Figure 2.4.1.2.

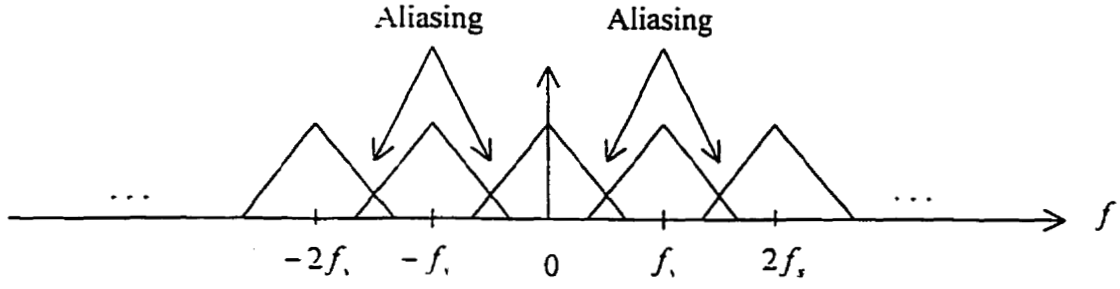


Figure 2.4.1.2 The spectrum of the sampled lowpass signal ($f_s < 2f_H$).

Bandpass signals can be sampled the same way as lowpass signals. This means that Equation 2.4.1.1 becomes

$$f_s \geq 2f'_H \qquad (2.4.1.2)$$

where $f'_H = f_H + f_c$ and f_c is the carrier frequency. In this way, all information of the signal from DC to f'_H is unambiguously identified. The disadvantage of this approach is that the sampling frequency required is too high since f'_H is much higher than f_H. For example, the IF carrier frequency and the baseband signal bandwidth for a Global Positioning System (GPS) signal [27] are 21.25 MHz and 1 MHz respectively. If this IF signal is sampled according to Equation 2.4.1.2, the sampling frequency required must be greater than or equal to 44.5 MHz. This means that it requires a state-of-the-art device to sample the IF signal. Even if one can find a sampling device operating at this frequency, it will increase the receiver cost drastically. Consequently, this approach is not affordable for a low cost receiver if the center frequency of the channel is very high.

2.4.2 Bandpass Sampling Theorem

For a bandpass signal with a finite bandwidth, there is another rule in the Nyquist sampling theorem to select the sampling frequency. This rule is known as the bandpass sampling theorem [8][28, pp. 321-337]. Let $y(t)$ be a bandpass signal, which has a carrier frequency f_c and a null-to-null bandpass bandwidth W as shown in Figure 2.4.2.1.

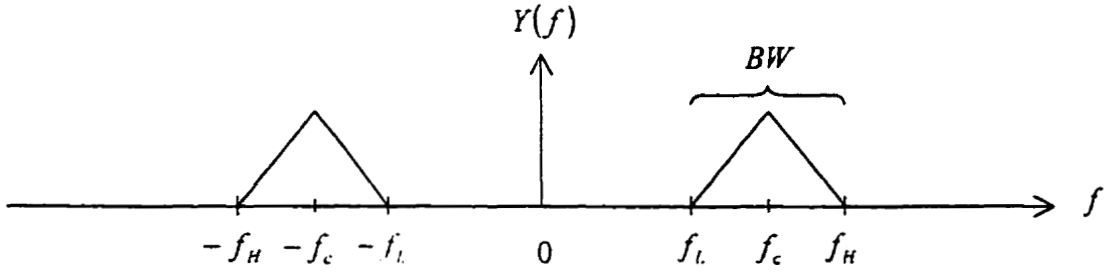


Figure 2.4.2.1 The spectrum of $y(t)$ ($BW = W$ is the bandpass bandwidth).

It is clear to see that the spectrum of $y(t)$ has an empty frequency slot from $-f_L$ to f_L. If the empty slot can accommodate some of the sampling images resulting from the sampling process, it may be possible to reduce the sampling rate. This is the fundamental concept behind the bandpass sampling theorem. Consequently, the bandpass sampling theorem states that the sampling rate of the sampler must satisfy the relationship

$$\frac{2f_H}{i} \le f_s \le \frac{2f_L}{i-1} \qquad (2.4.2.1)$$

$$2 \le i \le \frac{f_H}{f_H - f_L} \qquad (2.4.2.2)$$

where

$i \equiv$ an integer

$f_H \equiv$ the highest frequency component of interest in the signal to be sampled

$f_L \equiv$ the lowest frequency component of interest in the signal to be sampled

in order to sample $y(t)$ without in-band aliasing.

Equation 2.4.2.1 and Equation 2.4.2.2 raise three important points about the bandpass sampling theorem:

(a) Unlike the lowpass sampling theorem, the desired sampling rate, f_s, in the bandpass sampling theorem not only has a lower limit ($\frac{2f_H}{i}$) but also has an upper limit ($\frac{2f_L}{i-1}$). Therefore over-sampling may cause aliasing in this case.

(b) The bandpass sampling theorem is not guaranteed to work for all band-limited bandpass signals since i does not always exist. According to Equation 2.4.2.2, i exists only if

$$\left\lfloor \frac{f_H}{f_H - f_L} \right\rfloor \geq 2 \qquad (2.4.2.3)$$

where $\lfloor \ \rfloor$ is the floor operator. As a result, the bandpass sampling theorem can only apply to the bandpass signals which satisfy Equation 2.4.2.3.

(c) The maximum sampling rate can be obtained by substituting $i = 2$ into the upper limit in Equation 2.4.2.1 and the result is

$$f_{s,\max} = 2f_L \qquad (2.4.2.4)$$

The minimum sampling rate is obtained by substituting $i = \left\lfloor \frac{f_H}{f_H - f_L} \right\rfloor$ into the lower limit in Equation 2.4.2.1 and the result is

$$f_{s,\min} = \frac{2f_H}{\left\lfloor \frac{f_H}{f_H - f_L} \right\rfloor} \qquad (2.4.2.5)$$

If $\frac{f_H}{f_H - f_L}$ is an integer, Equation 2.4.2.5 can be further simplified to

$$\begin{aligned} f_{s,\min} &= 2(f_H - f_L) \\ &= 2W \end{aligned} \qquad (2.4.2.6)$$

where $W = f_H - f_L \equiv$ the bandpass bandwidth.

In a digital communication system, the bandpass signal often has a spectral content confined within $f_c \pm \frac{W}{2}$ (i.e. $f_H = f_c + \frac{W}{2}$ and $f_L = f_c - \frac{W}{2}$). Substituting this condition into Equation 2.4.2.3, a restriction on the carrier frequency is obtained as

$$\frac{f_c + \frac{W}{2}}{W} \geq 2$$

or

$$f_c \geq \frac{3}{2}W \qquad (2.4.2.7)$$

If Equation 2.4.2.6 is used to determine the minimum sampling frequency, then $\frac{f_c + \frac{W}{2}}{W}$ must be an integer. This modifies the restriction on the carrier frequency to

$$f_c = \left(\frac{3}{2} + k\right)W \qquad (2.4.2.8)$$

where $k = 0,1,2,....\infty$. As a result, the carrier frequency of a bandpass signal must satisfy Equation 2.4.2.8 in order to use the bandpass sampling theorem as stated in Equation 2.4.2.6.

2.5 Nyquist Pulse for Minimum ISI

In an analog signal, all frequency components inside its bandwidth carry information. Therefore any spectral distortion (like aliasing) will corrupt some of the information on the signal. In contrast, the information is modulated on top of the signaling pulse in a digital signal. Thus the information is preserved as long as the

modulated signaling pulse remains unchanged at the sampling point. This means that aliasing is allowed if it does not distort the output of the sampler.

For Example, two signaling pulses are used to transmit the information bits $\{1,-1,1\}$ in a digital Pulse Amplitude Modulation (PAM) communication system and the sampling rate f_s is fixed at $\frac{1}{T_p}$ (T_p is the symbol period). The first pulse $p_1(t)$ is the same as $p(t)$ from Equation 2.2.1.2 that has a rectangular lowpass (null-to-null) bandwidth of $\frac{1}{2T_p}$. The second pulse $p_2(t)$ is a raised cosine pulse [17, pp. 546] with 100% roll-off (i.e. $BW = \frac{1}{T_p}$). The spectrum of $p_2(t)$ (Figure 2.5.1) can be expressed as

$$P_2(f) = \begin{cases} T_p, & f = 0 \\ \frac{T_p}{2}\left[1 + \cos(\pi f T_p)\right], & |f| \le \frac{1}{T_p} \\ 0, & |f| > \frac{1}{T_p} \end{cases} \tag{2.5.1}$$

with impulse response [Figure 2.5.2]

$$p_2(t) = \frac{\sin(\pi t/T_p)}{\pi t/T_p} \cdot \frac{\cos(\pi t/T_p)}{1 - (4t^2)/T_p^2} \tag{2.5.2}$$

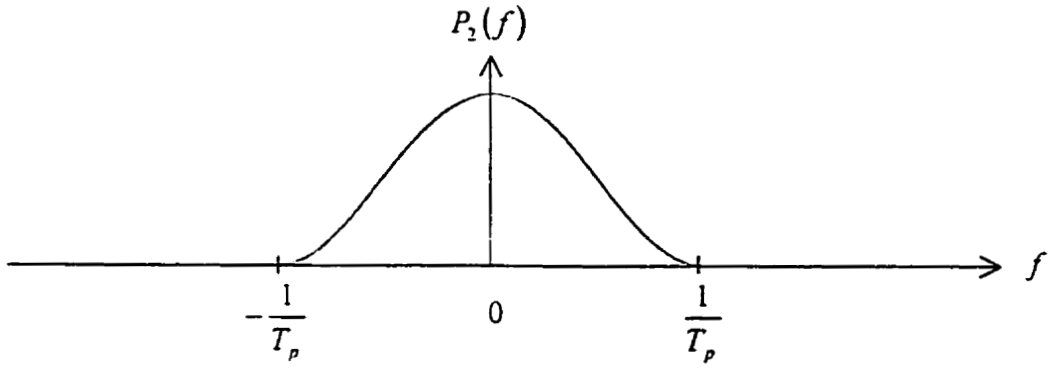


Figure 2.5.1 The spectrum of $p_2(t)$ (with 100% roll-off).

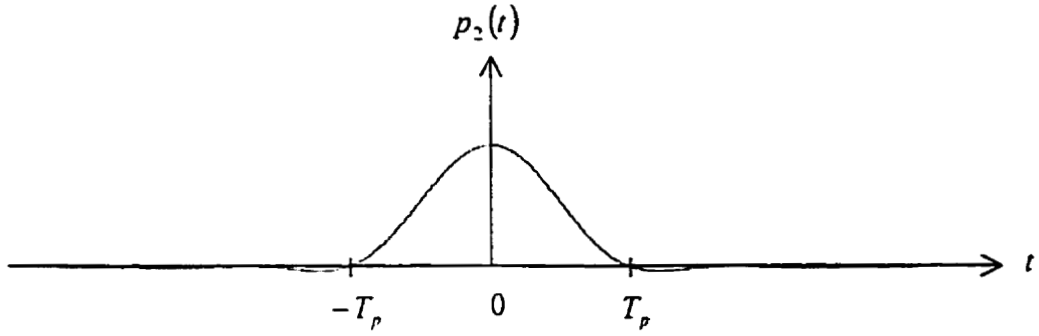


Figure 2.5.2 The raised cosine pulse $p_2(t)$ (with 100% roll-off).

Using $p_1(t)$ to transmit the information bits. the received baseband signal $r(t)$ (without channel distortion) is shown in Figure 2.5.3.

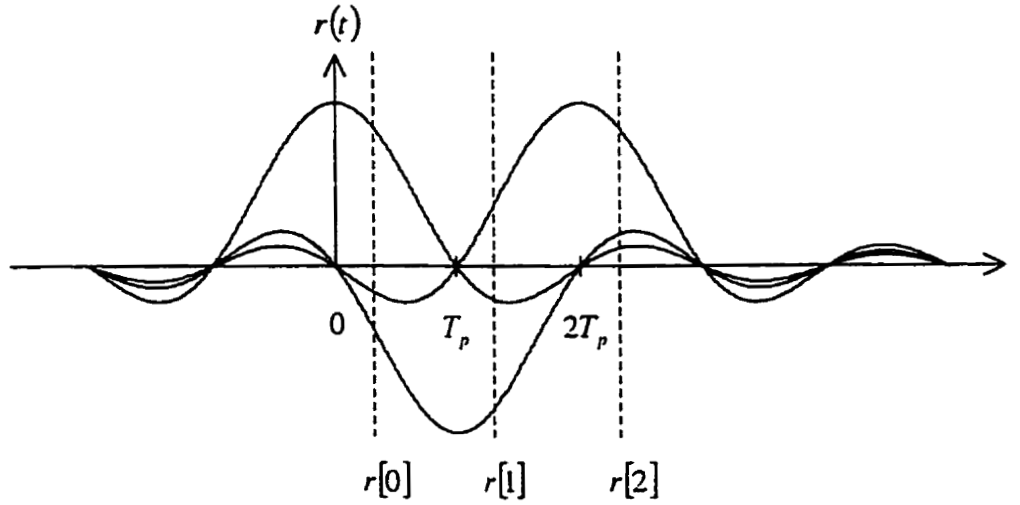


Figure 2.5.3 The baseband received signal $r(t)$ ($r[n]$ is the output of the sampler in the presence of sampling error).

Then the output of the sampler $r[n]$ has a frequency spectrum shown in Figure 2.5.4.

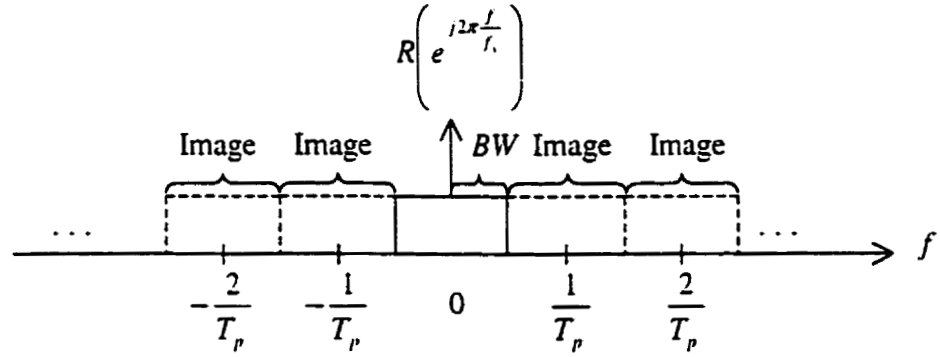


Figure 2.5.4 The spectrum of $r[n]$ ($BW = \frac{1}{2T_p}$ is the lowpass bandwidth).

Hence there is no aliasing in the spectrum of $r[n]$. Using $p_2(t)$ to transmit the information bits, the received baseband signal $r'(t)$ (without channel distortion) is shown in Figure 2.5.5.

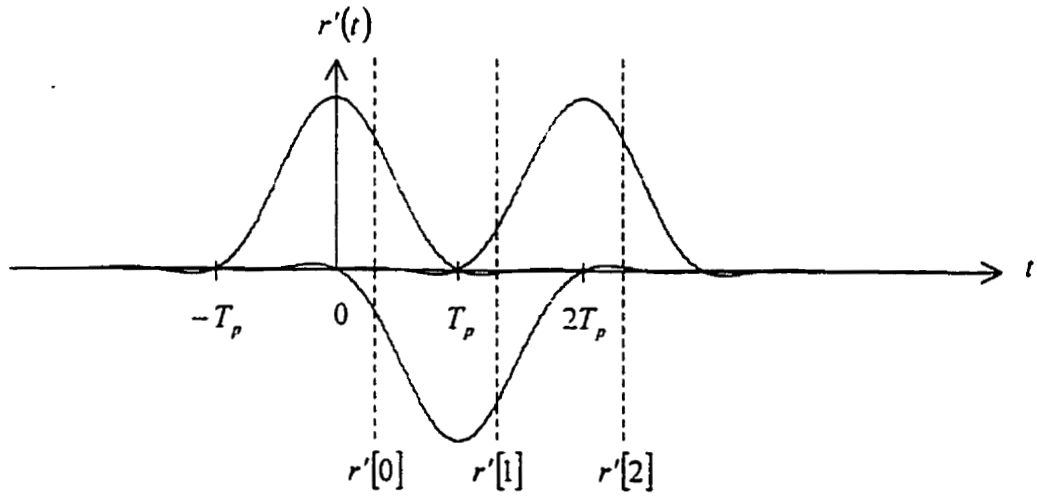


Figure 2.5.5 The baseband received signal $r'(t)$ ($r'[n]$ is the output of the sampler in the presence of sampling error).

In this case, the output of the sampler $r'[n]$ has a spectrum shown in Figure 2.5.6.

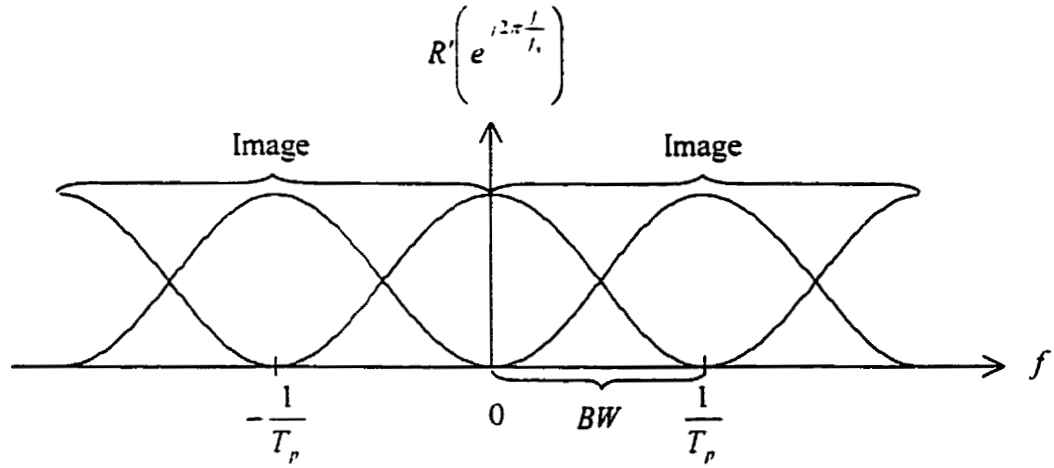


Figure 2.5.6 The spectrum of $r'[n]$ ($BW = \frac{1}{T_p}$ is the lowpass bandwidth).

It is clear to see that the sampling images overlap with each other in the spectrum of $r'[n]$. Comparing Figure 2.5.3 and Figure 2.5.5. the ISI in $r'[n]$ is much less than the ISI in $r[n]$ since the tail of $p_2(t)$ dies off much quicker then the tail of $p_1(t)$. Therefore the aliasing in the spectrum of $r'[n]$ does not necessarily harm the digital communication system since it can help improve the system performance by reducing the ISI as noticed by Nyquist [29, pp. 233-235].

There is another way to reduce the tail amplitude of a band-limited signaling pulse without increasing the bandwidth of the signaling pulse. Considering a pulse $p_{duo}(t)$ is formed according to Figure 2.5.7.

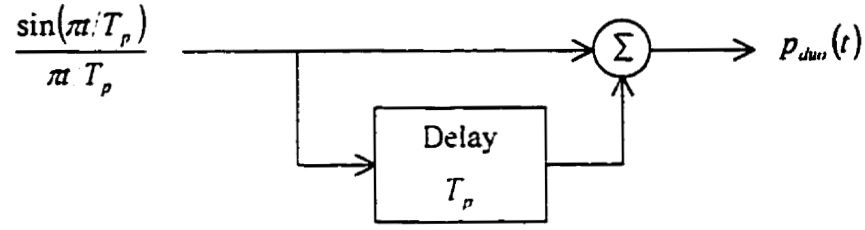


Figure 2.5.7 The block diagram to form $p_{duo}(t)$.

This pulse is called a duobinary pulse [20. pp. 198-200][29. pp. 237-243] and mathematically, its spectrum (Figure 2.5.8) can be expressed as

$$P_{duo}(f)=\begin{cases}2T_p\cdot\cos(\pi fT_p)\cdot e^{j\pi fT_p}, & |f|\le\dfrac{1}{2T_p}\\ 0, & |f|>\dfrac{1}{2T_p}\end{cases} \tag{2.5.3}$$

with impulse response [Figure 2.5.9]

$$\begin{aligned}p_{duo}(t)&=\frac{\sin(\pi t/T_p)}{\pi t/T_p}+\frac{\sin[\pi(t-T_p)/T_p]}{\pi(t-T_p)/T_p}\\&=\frac{T_p^2\cdot\sin(\pi t/T_p)}{\pi t(T_p-t)}\end{aligned} \tag{2.5.4}$$

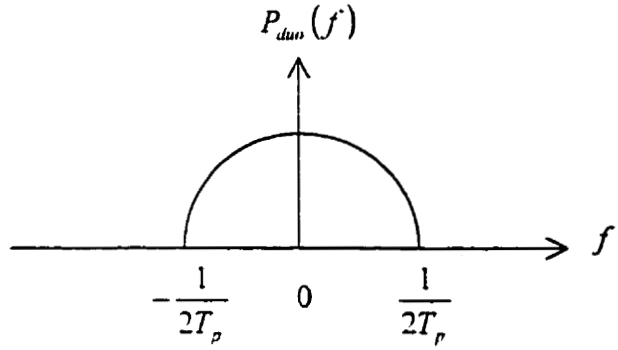


Figure 2.5.8 The spectrum of $p_{duo}(t)$.

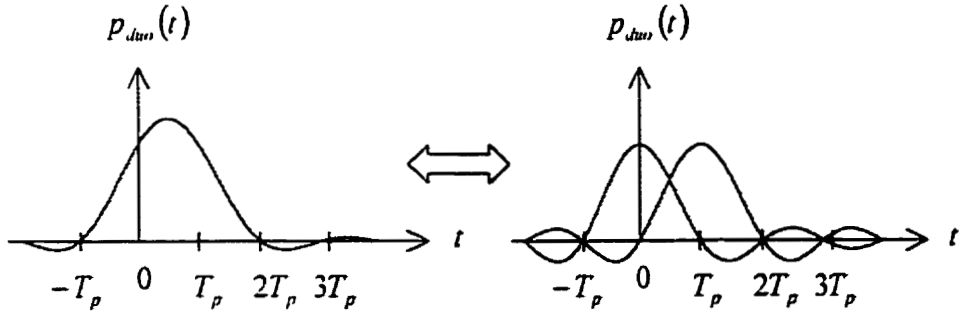


Figure 2.5.9 The duobinary pulse $p_{duo}(t)$.

Although the duobinary pulse has the same bandwidth as the sinc pulse shown in Figure 2.2.1.1, its tails are much smaller. This is because the positive side lobes of $\frac{\sin\left[\pi\left(t-T_p\right)/T_p\right]}{\pi\left(t-T_p\right)/T_p}$ suppress the negative side lobes of $\frac{\sin\left(\pi t/T_p\right)}{\pi t/T_p}$ and the negative side lobes of $\frac{\sin\left[\pi\left(t-T_p\right)/T_p\right]}{\pi\left(t-T_p\right)/T_p}$ suppress the positive side lobes of $\frac{\sin\left(\pi t/T_p\right)}{\pi t/T_p}$. As a result, the tails of the duobinary pulse become much smaller as demonstrated in Figure 2.5.9. The penalty for these smaller tails is that the main lobe is 1.5 times wider than the sinc pulse. This means that the ISI from the previous symbol becomes stronger.

2.6 Summary

This chapter provided a brief overview of $\pi/4$-shifted-DQPSK, DS/SS, $F_s/4$-downconversion, the Nyquist sampling theorem and the Nyquist pulse for minimum ISI.

The $\pi/4$-shifted-DQPSK is a modulation scheme, which attempts to protect the information signal from both the amplitude and phase distortion in a communication channel. To use this modulation scheme the communication channel must be stationary between two consecutive symbol periods and one must be prepared to pay a 3 dB penalty compared to coherently detected QPSK.

The DS/SS is a common technique for removing the ISI resulting from sampling error. The performance of the DS/SS depends on the autocorrelation property of the PN sequence. Since the 11-chip Barker code has a highly desirable autocorrelation function, it will be exclusively used throughout the thesis.

The $F_s/4$-downconversion retrieves the I and Q samples directly from the IF signal. Hence it reduces the receiver complexity. However, the price to pay for using $F_s/4$-downconversion is that it introduces timing misalignment between the I and Q samples.

The Nyquist sampling theorem ensures that there is no in-band spectral distortion after the sampling process. The Nyquist pulse for minimum ISI shows that a partial in-band distortion after the sampling process can minimize the ISI caused by the sampling error.

Chapter 3 Previous Research for the DS/SS IF-sampling System

This chapter begins with a detailed discussion on a DS/SS IF-sampling system in Kevin Hung's thesis [30]. This system consists of a DS/SS transmitter, an additive bandpass white Gaussian channel, a DS/SS IF-sampling receiver, and a $\pi/4$-shifted-DQPSK detector as demonstrated in Figure 3.1.

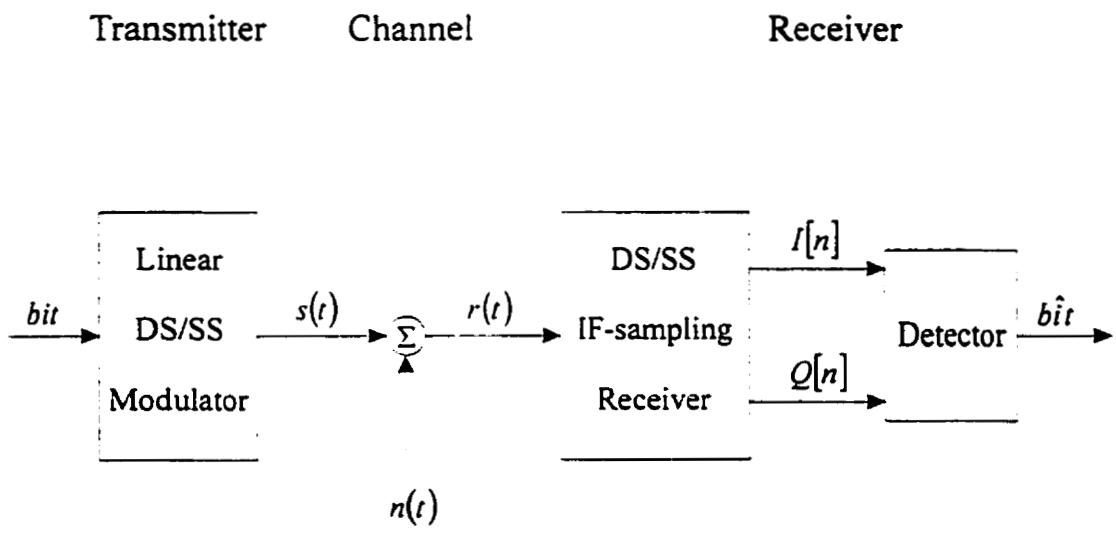


Figure 3.1 Block diagram of the bandpass communication system model.

The sampling rate of the IF-sampling receiver is kept at a minimum of 2 samples/chip in order to save the ADC cost. To handle the timing misalignment between the I and Q samples, this system uses a special shaping pulse in the transceiver and the design of this special shaping pulse is included in this chapter. The second objective of this chapter is to discuss the performance and the implementation of the system. To show the system performance, a discussion of ISI suppression, receiver noise analysis, and simulation results are presented in this chapter. In the discussion of system implementation, we investigate whether the system can be implemented with existing technology at low cost or not. Most of the material in this chapter is taken from Kevin's thesis [30].

3.1 The DS/SS Transmitter

Generally, a digital transmitter has two functions: converting a digital information signal into an analog baseband signal and modulating the baseband signal onto a carrier signal. The block diagram of the DS/SS transmitter is shown in Figure 3.1.1. This special transmitter achieves the functionality of a digital transmitter through four processes: bit mapping, spreading, pulse shaping, and frequency upconversion.

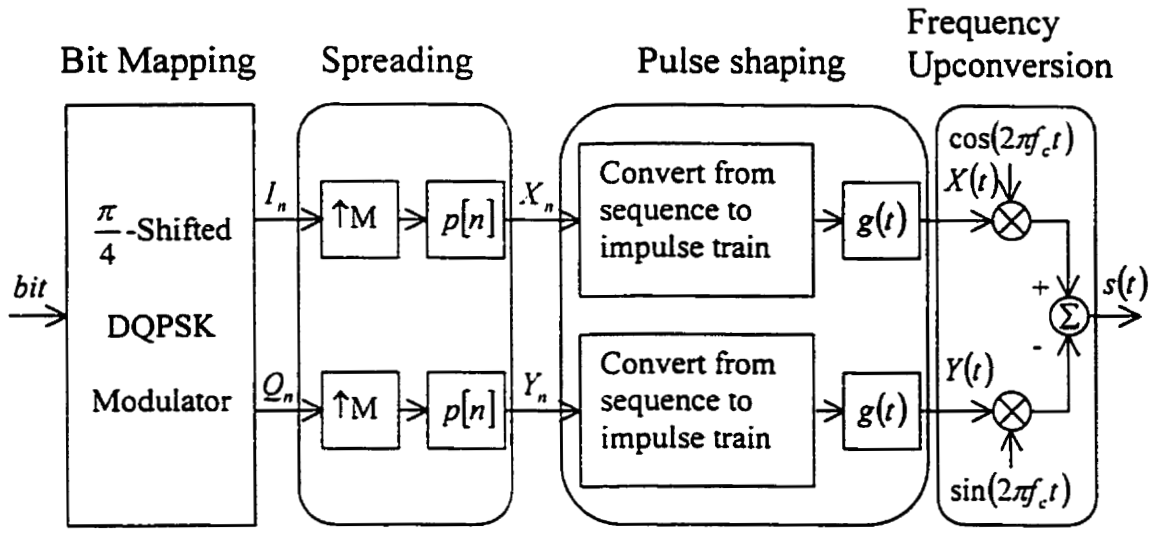


Figure 3.1.1 Transmitter for DS/SS IF-sampling system ($M = 11$ and $p[n]$ is the 11-chip Barker sequence).

In the bit mapping process, an information dibit is mapped to a complex symbol according to the $\pi/4$-shifted-DQPSK modulation scheme as discussed in Section 2.1.1. The result from Section 2.1 states that the output complex symbol is

$$I_n + jQ_n = e^{j\phi_n} \tag{3.1.1}$$

where

$I_n = \cos(\phi_n) \equiv$ inphase symbol at time index n

$Q_n = \sin(\phi_n) \equiv$ quadrature symbol at time index n

$\phi_n \equiv$ absolute phase at time index n

The spreading process is performed the same way as shown in Section 2.2.2. Therefore if the input I and Q symbols are I_n and Q_n, then the output I and Q sequences of the spreader are

$$X_n = \left\{ \sum_{k=-\infty}^{\infty} I_k \cdot \delta[n - kM] \right\} \otimes p[n] = \sum_{k=-\infty}^{\infty} I_k \cdot p[n - kM] \tag{3.1.2}$$

and

$$Y_n = \left\{ \sum_{k=-\infty}^{\infty} Q_k \cdot \delta[n - kM] \right\} \otimes p[n] = \sum_{k=-\infty}^{\infty} Q_k \cdot p[n - kM] \tag{3.1.3}$$

respectively.

The pulse shaping process is similar to the reconstruction process of a bandlimited signal from its samples in the Nyquist sampling theorem [26, pp. 87]. This process consists of two steps: (1) discrete sequence to analog impulse train conversion and (2) pulse shape filtering. Given that the input discrete sequences are X_n and Y_n, the output analog impulse trains from step (1) are

$$u_I(t) = \sum_{n=-\infty}^{\infty} X_n \cdot \delta[t - nT_c] \tag{3.1.4}$$

and

$$u_Q(t) = \sum_{n=-\infty}^{\infty} Y_n \cdot \delta[t - nT_c] \tag{3.1.5}$$

respectively where T_c is the chip period. In step (2), $u_I(t)$ and $u_Q(t)$ are passed to two pulse shaping filters having the same impulse response, $g(t)$. The output I and Q signals from step (2) are given as the convolution of $g(t)$ with $u_I(t)$ and $u_Q(t)$, that is

$$X(t) = u_I(t) \otimes g(t) = \sum_{n=-\infty}^{\infty} X_n \cdot g(t - nT_c) \tag{3.1.6}$$

$$Y(t) = u_Q(t) \otimes g(t) = \sum_{n=-\infty}^{\infty} Y_n \cdot g(t - nT_c) \tag{3.1.7}$$

In the frequency upconversion process, $X(t)$ and $Y(t)$ are frequency up-shifted by the cosine and sine carriers respectively to a carrier frequency f_c. Finally, the output of the transmitter is formed as

$$\begin{aligned} s(t) \quad &= \sum_{n=-\infty}^{\infty} X_n \cdot g(t - nT_c) \cdot \cos(2\pi f_c t) \\ &\quad - \sum_{n=-\infty}^{\infty} Y_n \cdot g(t - nT_c) \cdot \sin(2\pi f_c t) \end{aligned} \tag{3.1.8}$$

The bandpass signal $s(t)$ can also be represented by its complex-valued lowpass equivalent signal, $v(t)$, which can be expressed as

$$s(t) = \mathrm{Re}\left\{v(t) \cdot e^{j2\pi f_c t}\right\} \tag{3.1.9}$$

where $v(t) = \sum_{n=-\infty}^{\infty} (X_n + jY_n) \cdot g(t - nT_c)$. The relationship between $s(t)$ and $v(t)$ in frequency domain can be obtained by taking the Fourier transform of equation 3.1.9 and the result is

$$S(f) = \frac{1}{2}\left[V(f - f_c) + V^*(-f - f_c)\right] \tag{3.1.10}$$

where

$S(f) \equiv$ continuous-time Fourier transform of $s(t)$

$V(f) \equiv$ continuous-time Fourier transform of $v(t)$

Equation 3.1.10 states that the bandwidth of $s(t)$ is the same as the total bandwidth (including both positive and negative frequency spectra) of $v(t)$. The derivation of the power spectral density of $v(t)$ is given by [17, pp. 204] and the result indicates that if the encoded symbols, X_n and Y_n, are uncorrelated and have zero mean, that is

$$\phi_{XY}[m] = E\{X_n \cdot Y_{n-m}\} = 0 \qquad \text{for all } m \qquad (3.1.11)$$

$$E\{X_n\} = E\{Y_n\} = 0 \qquad \text{for all } n \qquad (3.1.12)$$

where $E\{\ \} \equiv$ the expectation operator, then the lowpass equivalent signal is a cyclostationary process with an average power density spectrum given by

$$\Phi_{vv}(f) = \frac{\sigma_i^2}{T_c} \cdot |G(f)|^2 \qquad (3.1.13)$$

where

$$\sigma_i^2 = \frac{1}{2} E\{(X_n + jY_n) \cdot (X_n + jY_n)^*\} \qquad (3.1.14)$$

Therefore, the power density spectrum of $v(t)$ has the same shape as the power density spectrum of the shaping pulse $g(t)$. Consequently, the total bandwidth of $v(t)$ is equal to the total bandwidth of $g(t)$. This also implies that the bandpass bandwidth of $s(t)$ is the same as the total bandwidth of $g(t)$ which dictates the required channel bandwidth.

3.2 The Channel

As described in Figure 3.1, the output signal of the transmitter, $s(t)$, is corrupted by an additive bandpass noise. Therefore, the input signal at the receiver can be expressed as

$$\begin{aligned} r(t) &= s(t)+n(t) \\ &= \sum_{n=-\infty}^{\infty} X_n \cdot g(t-nT_c)\cdot\cos(2\pi f_c t) \\ &\quad - \sum_{n=-\infty}^{\infty} Y_n \cdot g(t-nT_c)\cdot\sin(2\pi f_c t) \\ &\quad + n(t) \end{aligned} \tag{3.2.1}$$

where $n(t)$ is the additive bandpass noise. The noise is assumed to be a wide-sense stationary stochastic Gaussian process with zero mean, that is,

$$E\{n(t)\}=0 \qquad \text{for all time } t \tag{3.2.2}$$

In addition, the noise is also assumed to have a flat real-valued power density spectrum of $\frac{N_0}{2}$ in a frequency region with bandwidth of W Hz and centered at the carrier frequency. This means that the power density spectrum of $n(t)$ can be represented as

$$\Phi_{nn}(f)=\begin{cases}\frac{N_0}{2}, & |f-f_c|\le\frac{W}{2} \text{ or } |f+f_c|\le\frac{W}{2} \\ 0, & \textit{otherwise}\end{cases} \tag{3.2.3}$$

With this special power spectral density function, this random process is commonly known as bandpass white noise and it is shown graphically is Figure 3.2.1.

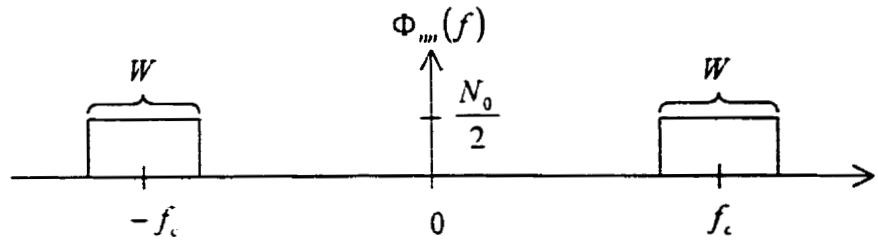


Figure 3.2.1 Power density spectrum of the bandpass white noise, $n(t)$.

Generally, any bandpass random process can be decomposed into two lowpass noise processes namely the I component $x(t)$ and the Q component $y(t)$. Thus, $n(t)$ can also be written as

$$n(t) = x(t) \cdot \cos(2\pi f_c t) - y(t) \cdot \sin(2\pi f_c t) \qquad (3.2.4)$$

where

$x(t) \equiv$ inphase component of $n(t)$

$y(t) \equiv$ quadrature component of $n(t)$

Since $n(t)$ is assumed to be a Gaussian process, it follows that both $x(t)$ and $y(t)$ are also zero-mean Gaussian random processes [17, pp. 162]. This implies that

$$E\{x(t)\} = E\{y(t)\} = 0 \qquad \text{for all time } t \qquad (3.2.5)$$

Since $n(t)$ is a wide sense stationary process, $x(t)$ and $y(t)$ are real-valued individually and jointly wide-sense stationary processes [17, pp. 159]. The autocorrelation functions of $x(t)$ and $y(t)$ are exactly the same and given as

$$\phi_{xx}(\tau) = E\{x(t) \cdot x(t-\tau)\} = N_0 \cdot \frac{\sin(\pi W \tau)}{\pi \tau} \qquad (3.2.6)$$

$$\phi_{yy}(\tau) = E\{y(t) \cdot y(t-\tau)\} = N_0 \cdot \frac{\sin(\pi W \tau)}{\pi \tau} \qquad (3.2.7)$$

It has been shown [17. pp. 159] that the two processes, $x(t)$ and $y(t)$, are also uncorrelated for all time shifts τ. therefore.

$$\phi_{xy}(\tau) = E\{x(t)\} \cdot E\{y(t-\tau)\} \qquad \text{for all time shifts } \tau \qquad (3.2.8)$$

This means that $x(t)$ and $y(t)$ are statistically independent since they are Gaussian processes. Substituting Equation 3.2.5 into Equation 3.2.8 yields

$$\phi_{xy}(\tau) = E\{x(t) \cdot y(t-\tau)\} = 0 \qquad \text{for all time shifts } \tau \qquad (3.2.9)$$

As a result, $x(t)$ and $y(t)$ are orthogonal to each other. The power density spectrum of $x(t)$ and $y(t)$ are obtained by taking the Fourier transform of $\phi_{xx}(\tau)$ and $\phi_{yy}(\tau)$. Hence

$$\Phi_{xx}(f)=\Phi_{yy}(f)=\begin{cases}N_0, & |f|<\frac{W}{2}\\ 0, & otherwise\end{cases} \tag{3.2.10}$$

The autocorrelation functions and the power spectral density functions of $x(t)$ and $y(t)$ are sketched in Figure 3.2.2.

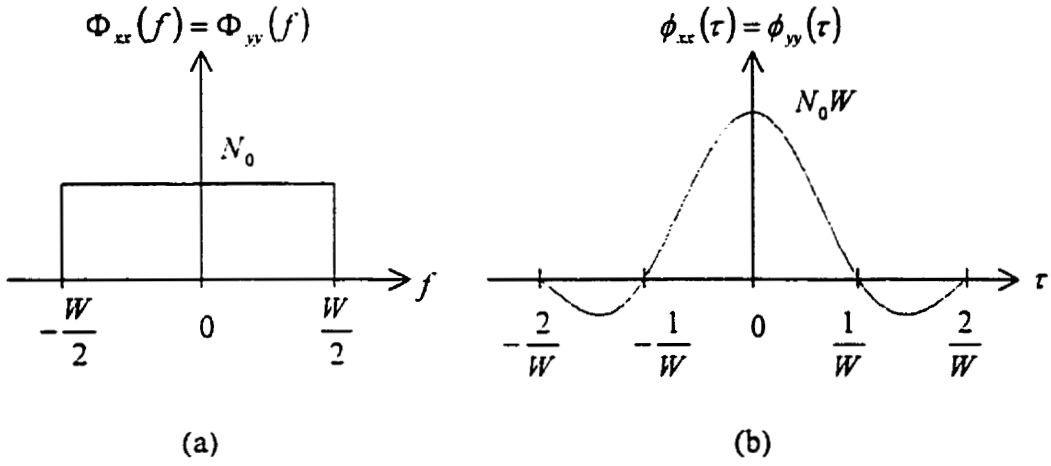


Figure 3.2.2 (a) Power density spectrum of $x(t)$ and $y(t)$.
(b) Autocorrelation functions of $x(t)$ and $y(t)$.

3.3 The DS/SS IF-sampling Receiver

The architecture of the DS/SS IF-sampling receiver is based on the concepts of PN-despreading, $F_s/4$-downconversion, and the Nyquist Pulse for minimum ISI.

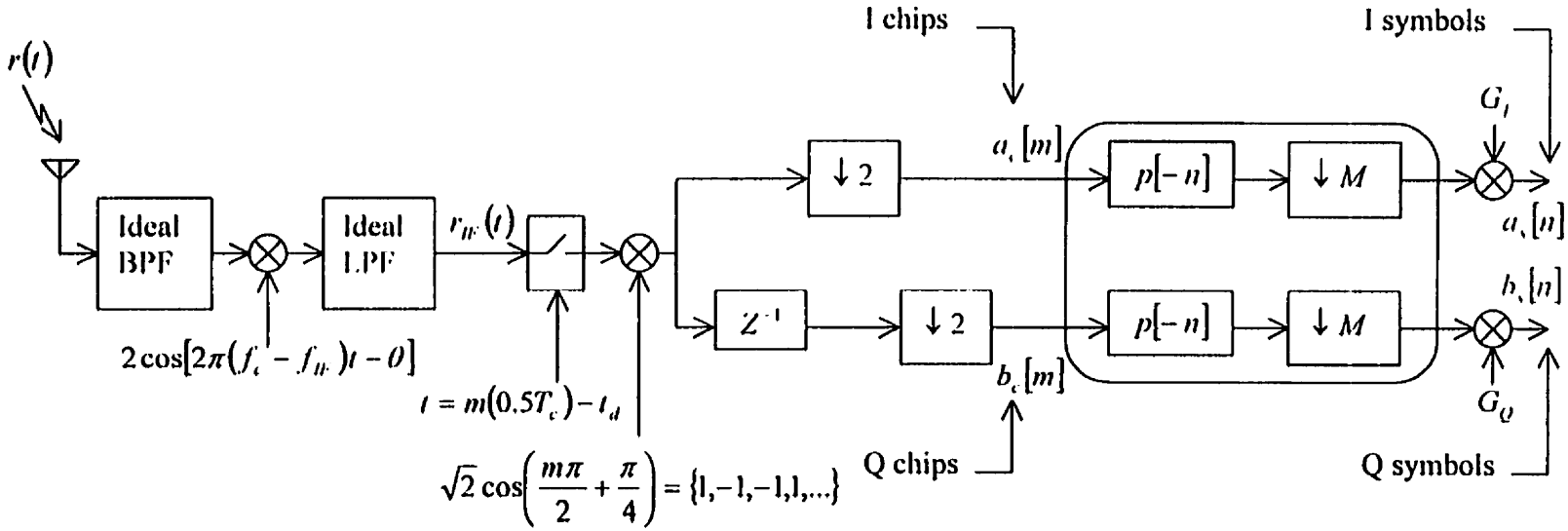


Figure 3.3.1 Architecture of the IF-sampling receiver for DS/SS system.

The receiver employs the minimum sampling rate of 2 samples/chip and a simple digital structure to recover the transmitted information symbols from the sampled IF signal. The structure of the DS/SS IF-sampling receiver is shown in Figure 3.3.1.

In Figure 3.3.1, the received RF signal, $r(t)$, is given as

$$\begin{aligned} r(t) &= \sum_{n=-\infty}^{\infty} X_n \cdot g(t - nT_c) \cdot \cos(2\pi f_c t) \\ &\quad - \sum_{n=-\infty}^{\infty} Y_n \cdot g(t - nT_c) \cdot \sin(2\pi f_c t) \\ &\quad + x(t) \cdot \cos(2\pi f_c t) \\ &\quad - y(t) \cdot \sin(2\pi f_c t) \end{aligned} \tag{3.3.1}$$

This RF signal is first filtered by an ideal bandpass filter (BPF) in order to remove any out-of-band interference. Then the filtered signal is mixed with a local oscillator, $2\cos[2\pi(f_c - f_{IF})t - \theta]$, where f_c is the RF carrier frequency and f_{IF} is the intermediate frequency. It is assumed that a perfect estimate of the carrier frequency is available in the receiver. However there exists an unknown phase offset between the received signal and the local oscillator. This phase offset is characterized by the constant phase θ at the output of the local oscillator. After mixing, the resulting signal is filtered by an ideal lowpass filter (LPF) to remove the double-frequency components resulting from mixing.

In order to perform $F_s/4$-downconversion, the intermediate frequency f_{IF} must satisfy the following condition:

$$f_{IF} = (2n+1)\cdot\frac{F_s}{4} \tag{3.3.2}$$

where $n = 0,1,2....$ Since a low intermediate frequency can ease the input bandwidth requirement of the sampler, the minimum intermediate frequency $F_s/4$ is used. Therefore, the sampling frequency, F_s, and the intermediate frequency f_{IF} can be expressed as

$$F_s = \frac{2}{T_c} \quad Hz \tag{3.3.3}$$

$$f_{IF} = \frac{F_s}{4} = \frac{1}{2T_c} \quad Hz \tag{3.3.4}$$

The bandwidth of the ideal LPF must be large enough to pass the IF signal without any distortion. Therefore the frequency response of the ideal LPF is given as

$$LPF(f) = \begin{cases} 1, & |f| \le \frac{1}{2Tc} + \frac{W}{2} \\ 0, & otherwise \end{cases} \tag{3.3.5}$$

where W is the bandwidth of the IF signal. Depending on the bandwidth and carrier frequency of the IF signal, spectral overlap can occur at both the input and output of the sampler. At the input of the sampler, spectral overlap occurs when $f_{IF} + \frac{W}{2} > \frac{F_s}{2}$. For example, the bandwidth of an IF signal is normally given as $W = 1/T_c$. If $W = 2/T_c$ (100% excess bandwidth) and $f_{IF} = F_s/4$, then spectral overlap will occur at the input of the sampler. This is illustrated in Figure 3.3.2.

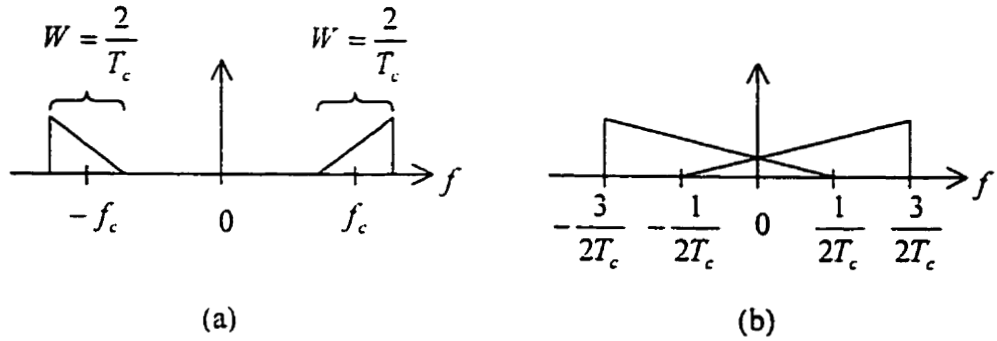


Figure 3.3.2 (a) Hypothetical RF spectrum with 100% excess bandwidth.
(b) Spectrum of the IF signal at the input of the sampler ($f_{IF} = \frac{F_s}{4}$).

To avoid spectral overlap, one could either choose a higher intermediate frequency or reduce the bandwidth of the IF signal. Spectral overlap can also happen during the sampling process as mentioned in Section 2.4. However, this type of spectral overlap only depends on the excess bandwidth of the IF signal. Unless the IF signal bandwidth is less than or equal to $1/T_s$, spectral overlap will always occurs during sampling. It is important to realize that the spectral overlap in the IF-sampling receiver will be constructive if the shaping pulse is chosen properly as shown in Section 2.5.

Whether spectral overlap occurs or not, the downconverted IF signal can always be represented as

$$
\begin{aligned}
r_{IF}(t) &= \sum_{n=-\infty}^{\infty} X_n \cdot g(t - nT_c) \cdot \cos(2\pi f_{IF} t + \theta) \\
&\quad - \sum_{n=-\infty}^{\infty} Y_n \cdot g(t - nT_c) \cdot \sin(2\pi f_{IF} t + \theta) \\
&\quad + x(t) \cdot \cos(2\pi f_{IF} t + \theta) \\
&\quad - y(t) \cdot \sin(2\pi f_{IF} t + \theta)
\end{aligned} \tag{3.3.6}
$$

This signal is then sampled by a sampler with a sampling rate of 2 samples/chip (i.e. $F_s = \frac{2}{T_c}$). The sampler is assumed to have an infinite resolution so that the quantization error can be ignored. However the sampling timing error still exists without a synchronization circuit in the receiver. Thus the actual sampling time is given as

$$t = m\frac{T_c}{2} - t_d \tag{3.3.7}$$

where t_d represents the sampling timing error and is always in the range from $-0.25T_c$ to $0.25T_c$. that is.

$$-0.25T_c \le t_d \le 0.25T_c. \tag{3.3.8}$$

A positive value of t_d corresponds to the sampling point off to the left; whereas a negative value of t_d corresponds to the sampling point off to the right. By substituting Equation 3.3.4 and 3.3.7 into Equation 3.3.6. the sampled IF sequence $r_{IF}[n]$ is given as

$$
\begin{aligned}
r_{IF}[m] &= \sum_{n=-\infty}^{\infty} X_n \cdot g\left(m\frac{T_c}{2} - t_d - nT_c\right) \cdot \cos\left(\frac{m\pi}{2} - \frac{\pi t_d}{T_c} + \theta\right) \\
&\quad - \sum_{n=-\infty}^{\infty} Y_n \cdot g\left(m\frac{T_c}{2} - t_d - nT_c\right) \cdot \sin\left(\frac{m\pi}{2} - \frac{\pi t_d}{T_c} + \theta\right) \\
&\quad + x\left(m\frac{T_c}{2} - t_d\right) \cdot \cos\left(\frac{m\pi}{2} - \frac{\pi t_d}{T_c} + \theta\right) \\
&\quad - y\left(m\frac{T_c}{2} - t_d\right) \cdot \sin\left(\frac{m\pi}{2} - \frac{\pi t_d}{T_c} + \theta\right)
\end{aligned} \tag{3.3.9}
$$

Without channel noise, sampling timing error, and phase offset, Equation 3.3.9 can be reduced to

$$r_{IF}[m] = \sum_{n=-\infty}^{\infty} X_n \cdot g\left(m\frac{T_c}{2} - nT_c\right) \cdot \cos\left(\frac{m\pi}{2}\right) - \sum_{n=-\infty}^{\infty} Y_n \cdot g\left(m\frac{T_c}{2} - nT_c\right) \cdot \sin\left(\frac{m\pi}{2}\right) \tag{3.3.10}$$

If the shaping pulse $g(t)$ is properly designed such that

$$\sum_{n=-\infty}^{\infty} g\left(m\frac{T_c}{2} - nT_c\right) = \begin{cases} 1, & n = m \\ 0, & n \neq m \end{cases} \tag{3.3.11}$$

then Equation 3.3.10 becomes

$$r_{IF}[m] = X_m \cdot \cos\left(\frac{m\pi}{2}\right) - Y_m \cdot \sin\left(\frac{m\pi}{2}\right) \tag{3.3.12}$$

Equation 3.3.12 represents the same concept as Equation 2.3.1.3 except that X_m and Y_m (in Equation 3.3.12) are chips, whereas $I_d[n]$ and $Q_d[n]$ (in Equation 2.3.1.3) are symbols. As a result, $r_{IF}[m]$ contains interleaved I and Q chips with sign inversions introduced on some of the chips. Hence $r_{IF}[m]$ can be rewritten as

$$r_{IF}[m] = \{X_0, -Y_0, -X_1, Y_1, X_2, -Y_2, \ldots\} \tag{3.3.13}$$

This result shows that this IF-sampling receiver can retrieve the I and Q chips from the analog IF signal directly which verifies the discussion in Section 2.3.1. The shaping pulse design for this IF-sampling receiver will be discussed later in this chapter.

To obtain the I chips $a_c[m]$, $r_{IF}[m]$ is multiplied by the sign conversion sequence $\sqrt{2}\cos\left(\frac{m\pi}{2}+\frac{\pi}{4}\right)$ and the resulting sequence is decimated by a factor of 2. Therefore the I chip can be expressed as

$$\begin{aligned} a_c[m] &= \sum_{n=-\infty}^{\infty} X_n \cdot g(mT_c - t_d - nT_c)\cdot\cos\left(-\frac{\pi t_d}{T_c}+\theta\right) \\ &\quad - \sum_{n=-\infty}^{\infty} Y_n \cdot g(mT_c - t_d - nT_c)\cdot\sin\left(-\frac{\pi t_d}{T_c}+\theta\right) \\ &\quad + x(mT_c - t_d)\cdot\cos\left(-\frac{\pi t_d}{T_c}+\theta\right) \\ &\quad - y(mT_c - t_d)\cdot\sin\left(-\frac{\pi t_d}{T_c}+\theta\right) \end{aligned} \qquad (3.3.14)$$

The Q chips $b_c[m]$ are obtained the same way except that the output sequence after multiplication is shifted to the left by one sample before decimation. Hence $b_c[m]$ is given as

$$\begin{aligned} b_c[m] &= \sum_{n=-\infty}^{\infty} X_n \cdot g(mT_c + 0.5T_c - t_d - nT_c)\cdot\sin\left(-\frac{\pi t_d}{T_c}+\theta\right) \\ &\quad + \sum_{n=-\infty}^{\infty} Y_n \cdot g(mT_c + 0.5T_c - t_d - nT_c)\cdot\cos\left(-\frac{\pi t_d}{T_c}+\theta\right) \\ &\quad + x(mT_c + 0.5T_c - t_d)\cdot\sin\left(-\frac{\pi t_d}{T_c}+\theta\right) \\ &\quad + y(mT_c + 0.5T_c - t_d)\cdot\cos\left(-\frac{\pi t_d}{T_c}+\theta\right) \end{aligned} \qquad (3.3.15)$$

In the following discussion, the noise terms in Equation 3.3.14 and 3.3.15 are ignored since they only affect the SNR of the system. The nature of the noise terms and the system SNR will be discussed later in the chapter. Without channel noise, Equations 3.3.14 and 3.3.15 become

$$a_c[m] = \sum_{n=-\infty}^{\infty} X_n \cdot g(mT_c - t_d - nT_c) \cdot \cos\left(-\frac{\pi t_d}{T_c} + \theta\right) - \sum_{n=-\infty}^{\infty} Y_n \cdot g(mT_c - t_d - nT_c) \cdot \sin\left(-\frac{\pi t_d}{T_c} + \theta\right) \quad (3.3.16)$$

$$b_c[m] = \sum_{n=-\infty}^{\infty} X_n \cdot g(mT_c + 0.5T_c - t_d - nT_c) \cdot \sin\left(-\frac{\pi t_d}{T_c} + \theta\right) + \sum_{n=-\infty}^{\infty} Y_n \cdot g(mT_c + 0.5T_c - t_d - nT_c) \cdot \cos\left(-\frac{\pi t_d}{T_c} + \theta\right) \quad (3.3.17)$$

Equations 3.3.16 and 3.3.17 show that both recovered I and Q chips, $a_c[m]$ and $b_c[m]$, contain the information about the transmitted I and Q chips. This phenomenon is called crosstalk. If the term $-\frac{\pi t_d}{T_c} + \theta$ is zero, then crosstalk is eliminated in both $a_c[m]$ and $b_c[m]$. Therefore crosstalk is introduced by the sampling timing error t_d and the phase offset θ. However this crosstalk can be removed by using the $\pi/4$-shifted-DQPSK detector as shown in Section 2.1.2. In order to apply the theory in Section 2.1.2, it requires a pair of I and Q symbols. Assuming the spreading and despreading are perfect, then the recovered I and Q symbols can be expressed as

$$a_s[m] = Ma_c[m] \quad (3.3.18)$$

$$b_s[m] = Mb_c[m] \quad (3.3.19)$$

as shown in Section 2.2.2. Also for simplification, let $I_m = \sum_{n=-\infty}^{\infty} X_n \cdot g(mT_c - t_d - nT_c)$ and $Q_m = \sum_{n=-\infty}^{\infty} Y_n \cdot g(mT_c + 0.5T_c - t_d - nT_c)$. Therefore a complex symbol at m is

$$
\begin{aligned}
S'[m] &= a_c[m] + jb_c[m] \\
&= M\left\{ I_m \cos\left(-\frac{\pi t_d}{T_c} + \theta \right) - Q_m \sin\left(-\frac{\pi t_d}{T_c} + \theta \right) \right\} \\
&+ jM\left\{ I_m \sin\left(-\frac{\pi t_d}{T_c} + \theta \right) + Q_m \cos\left(-\frac{\pi t_d}{T_c} + \theta \right) \right\} \\
&= M(I_m + jQ_m) \cdot e^{j\left(-\frac{\pi t_d}{T_c} + \theta \right)} = I'_m + jQ'_m = e^{j\phi'_m}
\end{aligned}
\tag{3.3.20}
$$

Equation 3.3.20 means the same as Equation 2.1.2.1 and the channel phase ϕ_c becomes $-\frac{\pi t_d}{T_c} + \theta$. Consequently, the $\pi/4$-shifted-DQPSK detector removes the phase error $-\frac{\pi t_d}{T_c} + \theta$ as shown in Section 2.1.2. In another word, any crosstalk caused by a constant phase error in a receiver can be ignored if differential demodulation is used. Without crosstalk, Equations 3.3.16 and 3.3.17 become

$$a_c[m] = \sum_{n=-\infty}^{\infty} X_n \cdot g(mT_c - t_d - nT_c) \tag{3.3.21}$$

$$b_c[m] = \sum_{n=-\infty}^{\infty} Y_n \cdot g(mT_c + 0.5T_c - t_d - nT_c) \tag{3.3.22}$$

Finally, $a_c[m]$ and $b_c[m]$ are despread by the despreader and the output I and Q symbols, $a_s[n]$ and $b_s[n]$, are formed by multiplying the outputs of the despreader by the gains, G_I and G_Q, respectively. Mathematically, these processes can be expressed as

$$a_s[n] = G_I \cdot \sum_{m=-\infty}^{\infty} a_c[m] \cdot p[m - nM] \tag{3.3.23}$$

$$b_s[n] = G_Q \cdot \sum_{m=-\infty}^{\infty} b_c[m] \cdot p[m - nM] \tag{3.3.24}$$

Note that G_I and G_Q are used to remove any amplitude imbalance between the I and Q channels. The details of the despreading process is discussed later in the chapter.

3.4 The Detector

After the IF-sampling receiver, the I and Q symbol corresponding to the same time index n form a complex symbol, that is,

$$S[n]=a_s[n]+jb_s[n] \tag{3.4.1}$$

This complex symbol is passed to the $\pi/4$-shifted-DQPSK detector as shown in Figure 3.1. The detector extracts the information dibit from the complex symbol according to Figure 2.1.2.1 and completes the operation of the DS/SS IF-sampling system.

3.5 Shaping Pulse Design

The purpose of this section is to design a suitable shaping pulse for the 2 samples/chip DS/SS IF-sampling system. A suitable shaping pulse should not introduce any ISI and be able to handle the timing misalignment problem shown in Section 2.3.1 when the sampling timing is perfect. The design procedures are similar to that of the Nyquist criterion [17, pp. 543] and the generalized Nyquist criterion [31]. During the design process, all distortions throughout the system are assumed to be zero. This means that the sampling timing error, phase offset, and the channel noise are assumed to be zero:

$$t_d=0,\ \theta=0 \tag{3.5.1}$$

$$x(t)=y(t)=0 \qquad \text{for all } t \tag{3.5.2}$$

In the previous section, the recovered I and Q chips, $a_c[m]$ and $b_c[m]$, are described by Equations 3.3.20 and 3.3.21 respectively. By removing the sampling timing error t_d, these two equations become

$$\begin{aligned} a_c[m] &= \sum_{n=-\infty}^{\infty} X_n \cdot g[(m-n)T_c] \\ &= X_m \cdot g(0) + \sum_{n \neq m} X_n \cdot g[(m-n)T_c] \end{aligned} \tag{3.5.3}$$

$$\begin{aligned} b_c[m] &= \sum_{n=-\infty}^{\infty} Y_n \cdot g[(m-n+0.5)T_c] \\ &= Y_m \cdot g(0.5T_c) + \sum_{n \neq m} Y_n \cdot g[(m-n+0.5)T_c] \end{aligned} \tag{3.5.4}$$

Equations 3.5.3 and 3.5.4 contain two potential problems if the shaping pulse $g(t)$ is not properly selected. Firstly, the terms, $\sum_{n \neq m} X_n \cdot g[(m-n)T_c]$ and $\sum_{n \neq m} Y_n \cdot g[(m-n+0.5)T_c]$, represent the ISI in the I and Q channels respectively. To minimize ISI, these two terms must be minimized. Secondly, X_m is scaled by $g(0)$ but Y_m is scaled by $g(0.5T_c)$. If $g(0) \neq g(0.5T_c)$, then the timing misalignment between the I and Q samples causes the I/Q amplitude imbalance as described in Section 2.3.1. To eliminate the ISI and the I/Q amplitude imbalance, the following two criterions must be satisfied:

$$g(t = nT_c) = \begin{cases} 1, & n = 0 \\ 0, & otherwise \end{cases} \tag{3.5.5}$$

$$g(t = nT_c + 0.5T_c) = \begin{cases} 1, & n = 0 \\ 0, & otherwise \end{cases} \tag{3.5.6}$$

Note that Equations 3.5.5 and 3.5.6 impose different restrictions on the same shaping pulse $g(t)$. In order to remove the ISI and the I/Q amplitude imbalance problem on both the I and the Q channels, the shaping pulse must be able to simultaneously satisfy both equations. Combining the two criterions, the overall restriction on the shaping pulse is given as

$$g[n]=g(t=0.5nT_c)=\begin{cases}1, & n=0,1\\ 0, & otherwise\end{cases} \tag{3.5.7}$$

Equation 3.5.7 is the time domain zero ISI and I/Q amplitude balance criterions for a coherent IF-sampling receiver with perfect sampling timing. Notice that Equation 3.5.7 takes the same form as the conventional Nyquist criterion except that the sampling rate on $g(t)$ is $2/T_c$ and two non-zero samples are required. This is mainly due to the fact that the I and Q chips are obtained at different time instants.

The restriction of the shaping pulse $g(t)$ can also be shown in frequency domain by taking the discrete-time Fourier transform (DTFT) of Equation 3.5.7. In general, the DTFT of $g[n]$ is denoted as $G(e^{j\omega})$ and the transform is defined as

$$G(e^{j\omega})=\sum_{n=-\infty}^{\infty} g[n]\cdot e^{-j\omega n} \tag{3.5.8}$$

Substituting $g[n]$ from Equation 3.5.7 into Equation3.5.8 yields

$$G(e^{j\omega})=1+e^{-j\omega}=\left(2\cdot\cos\frac{\omega}{2}\right)\cdot e^{-j\frac{\omega}{2}} \tag{3.5.9}$$

Applying the relationship between the normalized discrete-time frequency ω and the continuous-time frequency f: $\omega=2\pi\frac{f}{F_s}$, and the relationship between F_s and T_c: $F_s=\frac{2}{T_c}$, Equation 3.5.9 reduces to

$$G\left(e^{j2\pi\frac{f}{F_s}}\right)=\left(2\cdot\cos\frac{\pi fT_c}{2}\right)\cdot e^{-j\frac{\pi fT_c}{2}} \tag{3.5.10}$$

In general, the discrete-time Fourier transform, $G\left(e^{j2\pi\frac{f}{F_s}}\right)$, is related to the continuous-time Fourier transform, $G(f)$, by the sampling theorem as

$$G\left(e^{j2\pi\frac{f}{F_s}}\right)=F_s\cdot\sum_{k=-\infty}^{\infty}G(f-kF_s) \tag{3.5.11}$$

By combining Equation 3.5.10 and 3.5.11, the restriction on $G(f)$ for the IF-sampling receiver can be obtained as

$$\sum_{k=-\infty}^{\infty}G(f-kF_s)=T_c\cdot\cos\frac{\pi fT_c}{2}\cdot e^{-j\frac{\pi fT_c}{2}} \tag{3.5.12}$$

where $F_s=\frac{2}{T_c}Hz$. Equation 3.5.12 is the frequency domain representation of the zero ISI and I/Q amplitude balance criterions for a coherent IF-sampling receiver with perfect sampling timing. The criterions imply that in order to avoid ISI and I/Q amplitude imbalance on both I and Q channels, the periodic replication of $G(f)$ must have a magnitude response of

$$\left|\sum_{k=-\infty}^{\infty}G(f-kF_s)\right|=T_c\cdot\left|\cos\left(\frac{\pi fT_c}{2}\right)\right| \tag{3.5.13}$$

and a phase response of

$$\arg\left\{\sum_{k=-\infty}^{\infty}G(f-kF_s)\right\}=\begin{cases}-\frac{\pi fT_c}{2}, & \cos\frac{\pi fT_c}{2}>0, \quad |f|<\frac{1}{T_c}\\ -\frac{\pi fT_c}{2}+\pi, & \cos\frac{\pi fT_c}{2}<0, \quad |f|<\frac{1}{T_c}\end{cases} \tag{3.5.14}$$

Notice that, except at the points of phase discontinuity, the phase response is a linear function of frequency. In terms of the bandwidth requirement for the shaping pulse $g(t)$, Equation 3.5.13 implies that the minimum bandwidth is the frequency range of $G(f)$ and is equal to $1/T_c$. Both the time domain and frequency domain zero ISI and I/Q amplitude balance criterions are shown in Figure 3.5.1.

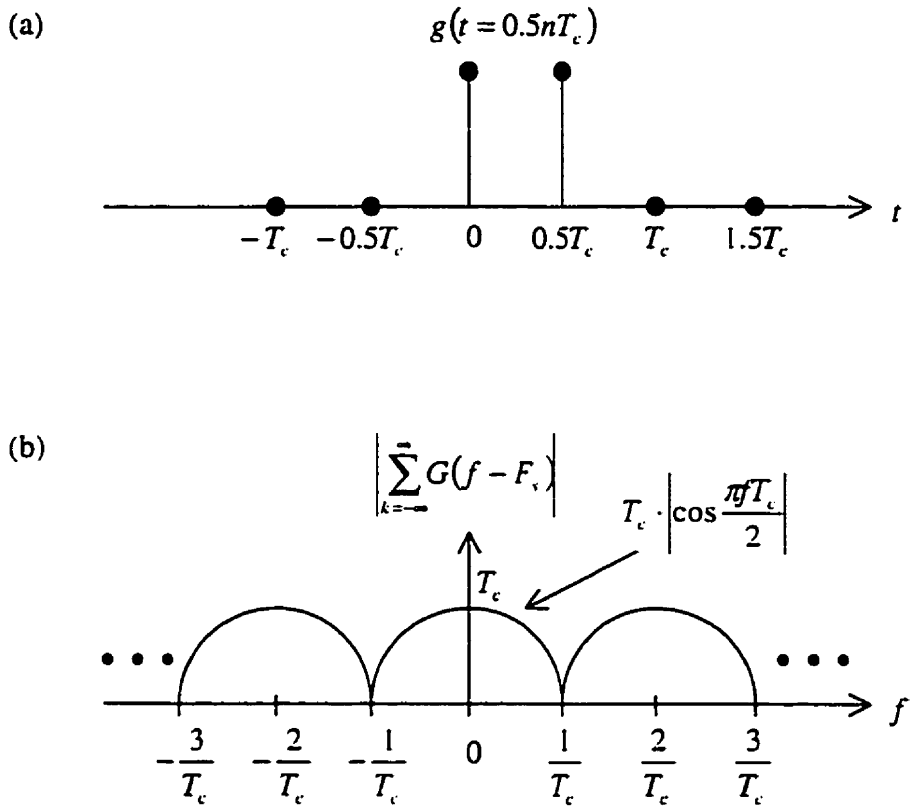


Figure 3.5.1 (a) Time domain zero ISI and I/Q amplitude balance criterions.
(b) Frequency domain zero ISI and I/Q amplitude balance criterions.
(Magnitude response only).

To keep the transmission bandwidth as low as possible, the bandwidth of the shaping pulse, W, is set to $2/T_c$. When $W = F_s = \frac{2}{T_c}$, the frequency spectrum of $g(t)$ is

$$G(f)=\begin{cases} T_c \cdot \cos\frac{\pi f T_c}{2} \cdot e^{-j\frac{\pi f T_c}{2}}, & |f| \le \frac{1}{T_c} \\ 0, & otherwise \end{cases} \qquad (3.5.15)$$

and the frequency aliasing is marginally avoided as shown in Figure 3.5.1(b). The magnitude response of $G(f)$ is shown in Figure 3.5.2.

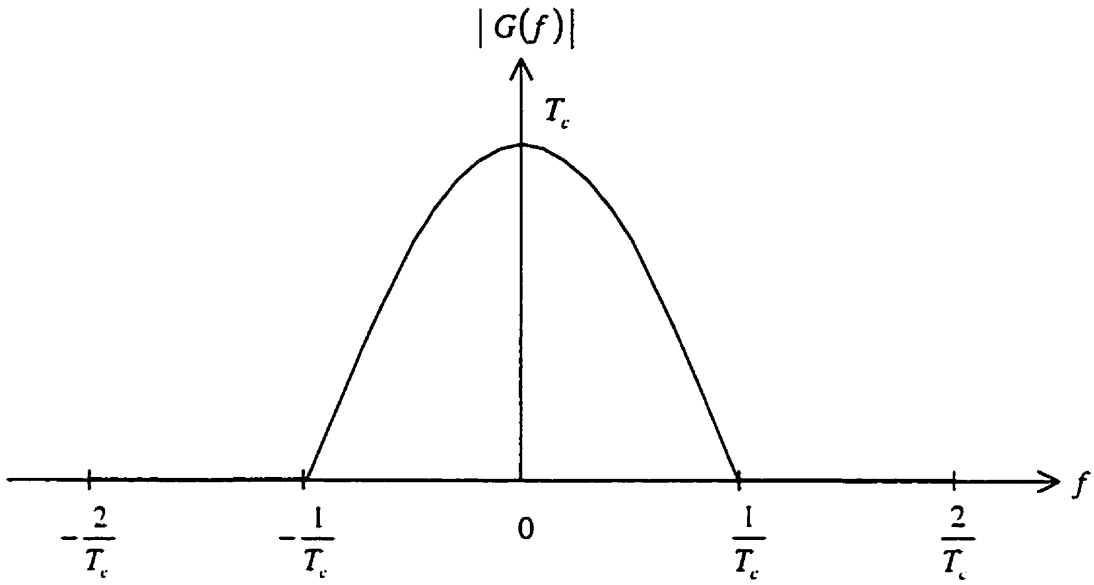


Figure 3.5.2 Magnitude response of the shaping pulse $g(t)$ with $W = 1/T_c$.

Finally, the impulse response of the shaping pulse, $g(t)$, can be obtained either from its samples, $g[n]$, by the Nyquist sampling theorem or by taking the inverse Fourier transform of $G(f)$. In either case, the result is

$$\begin{aligned} g(t) &= \frac{\sin(2\pi t/T_c)}{2\pi t/T_c} + \frac{\sin[2\pi(t-0.5T_c)/T_c]}{2\pi(t-0.5T_c)/T_c} \\ &= \frac{0.25T_c^2 \cdot \sin(2\pi t/T_c)}{\pi t(0.5T_c - t)} \end{aligned} \tag{3.5.16}$$

and $g(t)$ is plotted in Figure 3.5.3.

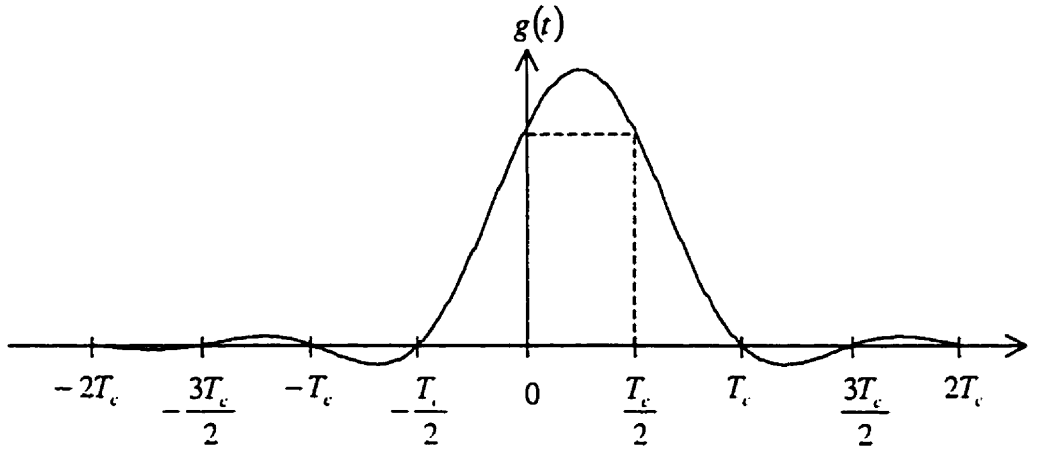


Figure 3.5.3 The impulse response of the shaping pulse $g(t)$ with $W = 1/T_c$.

Notice that this shaping pulse $g(t)$ is the duobinary pulse as described in Section 2.5 except that it has a 100% excess bandwidth. Thus this pulse can be viewed as a sum of two sinc pulses with a time offset of $0.5T_c$.

3.6 The Performance of the DS/SS IF-sampling receiver

The DS/SS IF-sampling receiver is designed for an analytical channel that is corrupted by ISI and bandpass white noise. This receiver removes ISI effectively during the despreading process which will be shown clearly in this section. The bandpass white noise in the I and Q channels is characterized by the last two terms in Equations 3.3.14 and 3.3.15. This section discusses the effect of the bandpass white noise in the IF-

sampling receiver. The performance of the receiver is demonstrated through simulation [30]. The simulation setup and results are also included in this section.

3.6.1 Despreading

To demonstrate the despreading process, crosstalk and noise in Equations 3.3.14 and 3.1.15 are ignored. Therefore the received I and Q chip sequences are described by Equations 3.3.21 and 3.3.22 as

$$a_c[m] = \sum_{n=-\infty}^{\infty} X_n \cdot g(mT_c - t_d - nT_c) \tag{3.6.1.1}$$

$$b_c[m] = \sum_{n=-\infty}^{\infty} Y_n \cdot g(mT_c + 0.5T_c - t_d - nT_c) \tag{3.6.1.2}$$

where X_n and Y_n are given by Equations 3.1.2 and 3.1.3 as

$$X_n = \sum_{k=-\infty}^{\infty} I_k \cdot p[n - kM] \tag{3.6.1.3}$$

$$Y_n = \sum_{k=-\infty}^{\infty} Q_k \cdot p[n - kM] \tag{3.6.1.4}$$

The shaping pulse $g(t)$ is the duobinary pulse with 100% excess bandwidth as described in the previous section. As mentioned previously, the duobinary pulse is the sum of two sinc pulses with a time offset of $0.5T_c$. According to Section 2.5, the tails of the two sinc pulses tend to cancel each other out. Therefore, the tails of the duobinary pulse can be assumed to be negligible. Consequently, the shaping pulse will be truncated in order to simplify the analysis. After the truncation the shaping pulse is given as

$$g(t)=\begin{cases}\dfrac{0.25T_c^2\cdot\sin(2\pi/T_c)}{\pi(0.5T_c-t)}, & -0.5T_c\le t\le T_c\\ 0, & otherwise\end{cases} \qquad (3.6.1.5)$$

In general, the recovered I and Q symbols can be represented by Equation 3.3.23 and 3.3.24,

$$a_r[n]=G_I\cdot\sum_{m=-\infty}^{\infty}a_c[m]\cdot p[m-nM] \qquad (3.6.1.6)$$

$$b_r[n]=G_Q\cdot\sum_{m=-\infty}^{\infty}b_c[m]\cdot p[m-nM] \qquad (3.6.1.7)$$

The values of G_I and G_Q depend on the timing error t_d and the results are presented in three cases: (1) $t_d=0$, (2) $0<t_d\le 0.25T_c$ and (3) $-0.25\le t_d<0$.

Case 1: $t_d=0$

If $t_d=0$ (i.e. no sampling timing error), the I and Q chip sequences become

$$a_c[m]=\sum_{n=-\infty}^{\infty}X_n\cdot g(mT_c-nT_c) \qquad (3.6.1.8)$$

$$b_c[m]=\sum_{n=-\infty}^{\infty}Y_n\cdot g(mT_c+0.5T_c-nT_c) \qquad (3.6.1.9)$$

In both Equations 3.6.1.8 and 3.6.1.9, the shaping pulses are non-zero if and only if $n=m$ according to Equation 3.6.1.5. Therefore Equations 3.6.1.8 and 3.6.1.9 can be simplified to

$$a_c[m]=X_m \qquad (3.6.1.10)$$

$$b_c[m]=Y_m \qquad (3.6.1.11)$$

Note that there is no ISI in either channel if the sampling timing is perfect. By substituting Equations 3.6.1.3 and 3.6.1.10 into Equation 3.6.1.6, the recovered I symbols can be expressed as

$$a_r[n] = G_I \cdot \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} I_k \cdot p[m - kM] \cdot p[m - nM] \qquad (3.6.1.12)$$

According to Section 2.2.2, Equation 3.6.1.12 can be simplified to

$$a_r[n] = G_I \cdot M \cdot I_n \qquad (3.6.1.13)$$

Using the same method, the recovered Q symbol is given as

$$b_r[n] = G_Q \cdot M \cdot Q_n \qquad (3.6.1.14)$$

Since the I and Q symbols have the same gain factor, G_I and G_Q should be unity in this case.

Case 2: $0 < t_d \leq 0.25T_c$

If $0 < t_d \leq 0.25T_c$, the I chip sequence (Equation 3.6.1.1) is non-zero when $n = m$ and $n = m - 1$. Therefore the I chip sequence is reduced to

$$a_c[m] = X_m \cdot g(-t_d) + X_{m-1} \cdot g(T_c - t_d) \qquad (3.6.1.15)$$

The Q chip sequence (Equation 3.6.1.2) is non-zero if and only if $n = m$. Therefore the Q chip sequence is reduced to

$$b_c[m] = Y_m \cdot g(0.5T_c - t_d) \qquad (3.6.1.16)$$

The values of $g(-t_d)$, $g(T_c - t_d)$ and $g(0.5T_c - t_d)$ are listed in Table 3.6.1.1.

t_d	$g(-t_d)$	$g(T_c - t_d)$	$g(0.5T_c - t_d)$
0	1.0	0	1.0
$0.10T_c$	0.78	0.13	1.17
$0.15T_c$	0.66	0.216	1.23
$0.20T_c$	0.54	0.315	1.26
$0.25T_c$	0.424	0.424	1.27

Table 3.6.1.1 Table of the tap weights for $0 \le t_d \le 0.25T_c$.

Equation 3.6.1.15 and 3.6.1.16 show that the presence of the sampling timing error ($0 < t_d \le 0.25T_c$) introduces an ISI term, $X_{m-1} \cdot g(T_c - t_d)$, in the I chip sequence which ISI term is caused by the previous chip. By substituting Equation 3.6.1.3 and 3.6.1.15 into Equation 3.6.6, the recovered I symbols can be expressed as

$$a_s[n] = G_I \cdot g(-t_d) \cdot \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} I_k \cdot p[m-kM] \cdot p[m-nM]$$
$$+ G_I \cdot g(0.5T_c - t_d) \cdot \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} I_k \cdot p[m-1-kM] \cdot p[m-nM] \quad (3.6.1.17)$$

Since Barker code is used in this system, the term $\sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} p[m-1-kM] \cdot p[m-nM]$ is always either $+1$ or -1. As a result, the recovered I symbols can be simplified to

$$a_s[n] = G_I \cdot g(-t_d) \cdot M \cdot I_n + G_I \cdot g(T_c - t_d) \cdot (\pm 1) \cdot I_n \quad (3.6.1.18)$$

Since $M >> 1$ and $g(-t_d) \ge g(T_c - t_d)$, the second term in Equation 3.6.1.18 is comparatively small and thus

$$a_s[n] = G_I \cdot g(-t_d) \cdot M \cdot I_n \quad (3.6.1.19)$$

In other words, the ISI term is removed during the despreading process. For the Q channel, the recovered symbols can be obtained by substituting Equations 3.6.1.4 and 3.6.1.16 into 3.6.1.7 resulting in

$$\begin{aligned} b_s[n] &= G_Q \cdot g(0.5T_c - t_d) \cdot \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} Q_k \cdot p[m - kM] \cdot p[m - nM] \\ &= G_Q \cdot g(0.5T_c - t_d) \cdot M \cdot Q_n \end{aligned} \tag{3.6.1.20}$$

In this case, the gain factors for the I and Q channels, G_I and G_Q, should be set to

$$G_I = \frac{1}{g(-t_d)} \tag{3.6.1.21}$$

and

$$G_Q = \frac{1}{g(0.5T_c - t_d)} \tag{3.6.1.22}$$

in order to remove the amplitude imbalance between the I and Q channels.

Case 3: $-0.25T_c \le t_d < 0$

If $-0.25T_c \le t_d < 0$, the I chip sequence (Equation 3.6.1.1) is non-zero if and only if $n = m$ and thus the I chip sequence is reduced to

$$a_c[m] = X_m \cdot g(-t_d) \tag{3.6.1.23}$$

However the Q chip sequence (Equation 3.6.1.2) is non-zero when either $n = m$ or $n = m + 1$. Therefore the Q chip sequence is reduced to

$$b_c[m] = Y_m \cdot g(0.5T_c - t_d) + Y_{m+1} \cdot g(-0.5T_c - t_d) \tag{3.6.1.24}$$

The values of $g(-t_d)$, $g(0.5T_c - t_d)$ and $g(-0.5T_c - t_d)$ are listed in Table 3.6.1.2.

t_d	$g(-t_d)$	$g(-0.5T_c - t_d)$	$g(0.5T_c - t_d)$
0	1.0	0	1.0
$-0.10T_c$	1.17	0.13	0.78
$-0.15T_c$	1.23	0.216	0.66
$-0.20T_c$	1.26	0.315	0.54
$-0.25T_c$	1.27	0.424	0.424

Table 3.6.1.2 Table of the tap weights for $-0.25 \le t_d \le 0$.

Equations 3.6.1.23 and 3.6.1.24 show that the presence of the sampling timing error ($-0.25 \le t_d < 0$) introduces an ISI term, $Y_{m+1} \cdot g(-0.5T_c - t_d)$, in the Q chip sequence and the ISI term is caused by the future chip. By substituting Equation 3.6.1.3 and 3.6.1.23 into Equation 3.6.6, the recovered I symbols can be expressed as

$$\begin{aligned} a_s[n] &= G_I \cdot g(-t_d) \cdot \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} I_k \cdot p[m-kM] \cdot p[m-nM] \\ &= G_I \cdot g(-t_d) \cdot M \cdot I_n \end{aligned} \qquad (3.6.1.25)$$

For the Q channel, the recovered symbols can be obtained by substituting Equations 3.6.1.4 and 3.6.1.24 into 3.6.1.7 and the result is

$$\begin{aligned} b_s[n] &= G_Q \cdot g(0.5T_c - t_d) \cdot \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} Q_k \cdot p[m-kM] \cdot p[m-nM] \\ &\quad + G_Q \cdot g(-0.5T_c - t_d) \cdot \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} Q_k \cdot p[m+1-kM] \cdot p[m-nM] \\ &= G_Q \cdot g(0.5T_c - t_d) \cdot M \cdot Q_n + G_Q \cdot g(-0.5T_c - t_d) \cdot (\pm 1) \cdot Q_n \end{aligned} \qquad (3.6.1.26)$$

Since $M >> 1$ and $g(0.5T_c - t_d) \geq g(-0.5T_c - t_d)$, the second term in Equation 3.6.1.26 is negligible and thus

$$b_x[n] \approx G_Q \cdot g(0.5T_c - t_d) \cdot M \cdot Q_n \qquad (3.6.1.27)$$

As a result, ISI is successfully removed from the Q channel. Finally G_I and G_Q, should be set to

$$G_I = \frac{1}{g(-t_d)} \qquad (3.6.1.28)$$

and

$$G_Q = \frac{1}{g(0.5T_c - t_d)} \qquad (3.6.1.29)$$

in order to remove the amplitude imbalance between the I and Q channels.

In the above discussion, it shows that the I and Q symbols experience different ISI (asymmetric ISI) in this system. The asymmetric ISI is caused by two kinds of impairments. The first impairment is the sampling timing error that is typical of all digital communication systems. The second impairment is the timing misalignment problem as mentioned in Section 2.3.1. This is a unique problem to this IF-sampling receiver. Consequently, the IF-sampling system needs to perform PN spreading/despreading in order to remove the asymmetric ISI.

3.6.2 The Effect of Bandpass White Noise on the DS/SS IF-sampling Receiver

This section studies the propagation of the bandpass white noise through the IF-sampling receiver. It is assumed that the bandpass white noise has the same bandwidth as the information signal. Any out-of-band noise is filtered out by the front-end bandpass filter shown in Figure 3.3.1. The received bandpass white noise in the I and Q channels can be extracted from Equations 3.3.14 and 3.3.15 as

$$\begin{aligned} z_i[m] &= x(mT_c - t_d)\cdot\cos\left(-\frac{\pi t_d}{T_c}+\theta\right) \\ &\quad - y(mT_c - t_d)\cdot\sin\left(-\frac{\pi t_d}{T_c}+\theta\right) \end{aligned} \tag{3.6.2.1}$$

and

$$\begin{aligned} z_q[m] &= x(mT_c + 0.5T_c - t_d)\cdot\sin\left(-\frac{\pi t_d}{T_c}+\theta\right) \\ &\quad + y(mT_c + 0.5T_c - t_d)\cdot\cos\left(-\frac{\pi t_d}{T_c}+\theta\right) \end{aligned} \tag{3.6.2.2}$$

respectively. To study the effect of these noise terms, their statistical properties namely mean, autocorrelation function, power density function, average power, cross-correlation function, and probability density function are investigated in this section. During the investigation, both t_d and θ are treated as deterministic quantities.

The statistical mean of $z_i[m]$ and $z_q[m]$ are given as

$$E\{z_i[m]\} = E\{x(mT_c - t_d)\} \cdot \cos\left(-\frac{\pi t_d}{T_c} + \theta\right) - E\{y(mT_c - t_d)\} \cdot \sin\left(-\frac{\pi t_d}{T_c} + \theta\right) \quad (3.6.2.3)$$

and

$$E\{z_q[m]\} = E\{x(mT_c + 0.5T_c - t_d)\} \cdot \sin\left(-\frac{\pi t_d}{T_c} + \theta\right) + E\{y(mT_c + 0.5T_c - t_d)\} \cdot \cos\left(-\frac{\pi t_d}{T_c} + \theta\right) \quad (3.6.2.4)$$

where $E\{\ \}$ is the expectation operator. Since both $x(t)$ and $y(t)$ have zero means (Equation 3.2.5), it follows that $z_i[m]$ and $z_q[m]$ also have zero means for all time, that is

$$E\{z_i[m]\} = E\{z_q[m]\} = 0 \qquad \text{for all } m \quad (3.6.2.5)$$

The autocorrelation of $z_i[m]$ and $z_q[m]$ are defined as

$$\phi_{z_i z_i}[m; m-k] = E\{z_i[m] \cdot z_i[m-k]\} \quad (3.6.2.6)$$

and

$$\phi_{z_q z_q}[m; m-k] = E\{z_q[m] \cdot z_q[m-k]\} \quad (3.6.2.7)$$

By substituting Equation 3.6.2.1 into Equation 3.6.2.6, the autocorrelation of $z_i[m]$ is expanded as

$$\begin{aligned}\phi_{z_i z_i}[m; m-k] = {} & E\left\{x(mT_c - t_d) \cdot x(mT_c - kT_c - t_d) \cdot \cos^2\left(-\frac{\pi t_d}{T_c} + \theta\right)\right\} \\ & - E\left\{x(mT_c - t_d) \cdot y(mT_c - kT_c - t_d) \cdot \cos\left(-\frac{\pi t_d}{T_c} + \theta\right) \cdot \sin\left(-\frac{\pi t_d}{T_c} + \theta\right)\right\} \\ & - E\left\{x(mT_c - kT_c - t_d) \cdot y(mT_c - t_d) \cdot \cos\left(-\frac{\pi t_d}{T_c} + \theta\right) \cdot \sin\left(-\frac{\pi t_d}{T_c} + \theta\right)\right\} \\ & + E\left\{y(mT_c - t_d) \cdot y(mT_c - kT_c - t_d) \cdot \sin^2\left(-\frac{\pi t_d}{T_c} + \theta\right)\right\} \end{aligned} \quad (3.6.2.8)$$

Since the cross-correlation between $x(t)$ and $y(t)$ is zero for all time shift (Equation 3.2.8), $\phi_{z_i z_q}[m;m-k]$ can be simplified as

$$\phi_{z_i z_i}[m;m-k] = \phi_{xx}(\tau = kT_c)\cdot\cos^2\left(-\frac{\pi_d}{T_c}+\theta\right) + \phi_{yy}(\tau = kT_c)\cdot\sin^2\left(-\frac{\pi_d}{T_c}+\theta\right) \qquad (3.6.2.9)$$

Also due to the fact that $\phi_{xx}(\tau)=\phi_{yy}(\tau)=N_0\cdot\frac{\sin(\pi W\tau)}{\pi\tau}$ (Equation 3.2.6 and 3.2.7), the final expression for $\phi_{z_i z_i}[m;m-k]$ is

$$\phi_{z_i z_i}[m;m-k] = \phi_{z_i z_i}[k]= N_0\cdot\frac{\sin(\pi WkT_c)}{\pi kT_c} \qquad (3.6.2.10)$$

The autocorrelation function of $z_q[m]$ can be obtained by repeating the above procedure for the autocorrelation function of $z_i[m]$ and the result is

$$\phi_{z_q z_q}[m;m-k] = \phi_{z_q z_q}[k]= N_0\cdot\frac{\sin(\pi WkT_c)}{\pi kT_c} \qquad (3.6.2.11)$$

Equations 3.6.2.10 and 3.6.2.11 show that $z_i[m]$ and $z_q[m]$ have the same autocorrelation function and their autocorrelation depend only on the difference of the two time indices. This suggests that both $z_i[m]$ and $z_q[m]$ are wide-sense stationary random processes. In addition, Equations 3.2.6 and 3.6.2.10 indicate that $\phi_{z_i z_i}[k]$ is a sampled version of $\phi_{xx}(\tau)$ with a sampling rate of $1/T_c$. This implies that the noise processes $z_i[m]$ and $z_q[m]$ can be white or non-white processes depending on the bandwidth of the shaping pulse. With a 100% excess bandwidth duobinary pulse, the total bandpass bandwidth, W, is

$$W=\frac{2}{T_c} \qquad (3.6.2.12)$$

Therefore the autocorrelation functions, $\phi_{cc}(\tau)$ and $\phi_{vv}(\tau)$, become

$$\phi_{cc}(\tau)=\phi_{vv}(\tau)=N_0\cdot\frac{\sin\left(\pi\cdot\frac{2}{T_c}\cdot\tau\right)}{\pi\tau} \tag{3.6.2.13}$$

With a sampling rate of $1/T_c$, the autocorrelation functions, $\phi_{z_i z_i}[k]$ and $\phi_{z_q z_q}[k]$, are

$$\begin{aligned}\phi_{z_i z_i}[k]=\phi_{z_q z_q}[k]&=N_0\cdot\frac{\sin\left(\pi\cdot\frac{2}{T_c}\cdot kT_c\right)}{\pi kT_c}\\&=N_0\cdot\frac{\sin(2\pi k)}{\pi kT_c}=\frac{2N_0}{T_c}\cdot\delta[k]\end{aligned} \tag{3.6.2.14}$$

The relationship between Equation 3.6.2.13 and 3.6.2.14 is shown in Figure 3.6.2.1.

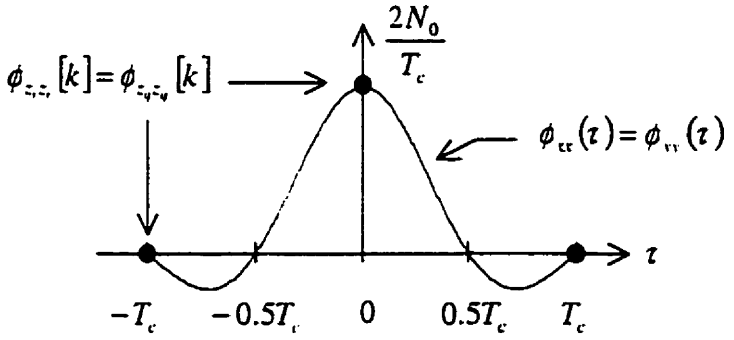


Figure 3.6.2.1 Relationship between $\phi_{cc}(\tau)$, $\phi_{vv}(\tau)$, $\phi_{z_i z_i}[k]$, and $\phi_{z_q z_q}[k]$ for 100% excess bandwidth duobinary pulse.

Since $\phi_{z_i z_i}[k]$ and $\phi_{z_q z_q}[k]$ are equal to a scaled discrete-time delta function, both $z_i[m]$ and $z_q[m]$ are white random processes.

In the frequency domain, the power density spectra of $x(t)$ and $y(t)$ are given as

$$\Phi_{xx}(f)=\Phi_{yy}(f)=\begin{cases}N_0, & |f|<\dfrac{1}{T_c}\\ 0, & \textit{otherwise}\end{cases} \qquad (3.6.2.15)$$

By periodically replicating $\Phi_{xx}(f)$ with a period of $1/T_c$ and multiplying the spectrum by $W = 2/T_c$, the power density spectra of $z_i[m]$ and $z_q[m]$ are obtained as

$$\Phi_{z_i z_i}\left(e^{j2\pi fT_c}\right)=\Phi_{z_q z_q}\left(e^{j2\pi fT_c}\right)=\frac{2N_0}{T_c} \qquad (3.6.2.16)$$

All of these four power density spectra are shown in Figure 3.6.2.2.

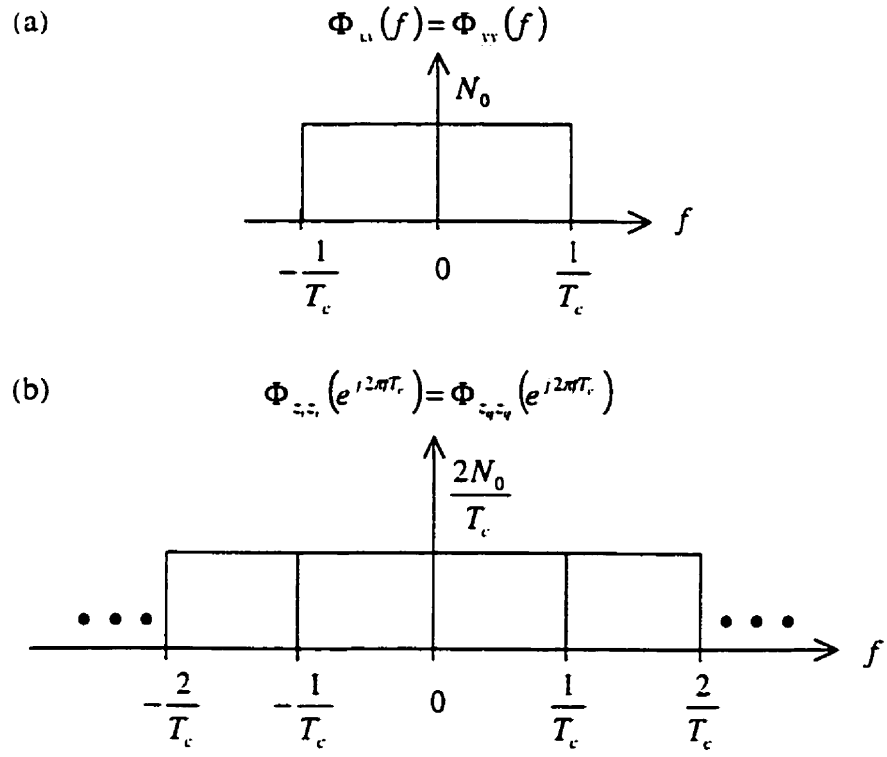


Figure 3.6.2.2 (a) Power density spectra of $x(t)$ and $y(t)$.
(b) Power density spectra of $z_i[m]$ and $z_q[m]$.
(100% excess bandwidth duobinary pulse)

The flat frequency spectrum for $\Phi_{z_i z_i}\left(e^{j2\pi fT_s}\right)$ and $\Phi_{z_q z_q}\left(e^{j2\pi fT_s}\right)$ further proves that $z_i[m]$ and $z_q[m]$ are white processes.

The total average power of a continuous-time random process is equal to its autocorrelation function evaluated at zero time shift. Therefore,

$$\text{Total Average Power of } x(t) = \phi_{xx}(\tau = 0) = N_0 W \qquad (3.6.2.17)$$

and

$$\text{Total Average Power of } y(t) = \phi_{yy}(\tau = 0) = N_0 W \qquad (3.6.2.18)$$

For a discrete-time random process, the total average power is also equal to its autocorrelation function evaluated at zero time shift. Therefore,

$$\text{Total Average Power of } z_i[m] = \phi_{z_i z_i}[k = 0] = N_0 W \qquad (3.6.2.19)$$

and

$$\text{Total Average Power of } z_q(t) = \phi_{z_q z_q}[k = 0] = N_0 W \qquad (3.6.2.20)$$

Notice that the two discrete-time noise sequences, $z_i[m]$ and $z_q[m]$, have the same average power as the two continuous-time lowpass noise processes, $x(t)$ and $y(t)$. As a result, as the bandpass noise propagates through the IF-sampling receiver, the noise is neither reduced nor enhanced. In other word, the noise power propagates through the IF-sampling receiver without modification.

The cross-correlation function between $z_i[m]$ and $z_q[m]$ is defined as

$$\phi_{z_i z_q}[m; m-k] = E\{z_i[m] \cdot z_q[m-k]\} \qquad (3.6.2.21)$$

By substituting Equations 3.6.2.1 and 3.6.2.2 into Equation 3.6.2.21, the cross-correlation between $z_i[m]$ and $z_q[m]$ is given as

$$\phi_{z_i z_q}[m;m-k] = E\left\{x(mT_c - t_d)\cdot x(mT_c - kT_c + 0.5T_c - t_d)\cdot\frac{1}{2}\sin\left(-\frac{2\pi t_d}{T_c}+2\theta\right)\right\}$$
$$+ E\left\{x(mT_c - t_d)\cdot y(mT_c - kT_c + 0.5T_c - t_d)\cdot\cos^2\left(-\frac{\pi t_d}{T_c}+\theta\right)\right\}$$
$$- E\left\{y(mT_c - t_d)\cdot x(mT_c - kT_c + 0.5T_c - t_d)\cdot\sin^2\left(-\frac{\pi t_d}{T_c}+\theta\right)\right\}$$
$$- E\left\{y(mT_c - t_d)\cdot y(mT_c - kT_c + 0.5T_c - t_d)\cdot\frac{1}{2}\sin\left(-\frac{2\pi t_d}{T_c}+2\theta\right)\right\}$$

(3.6.2.22)

Since $\phi_{xy}(\tau)=0$ (Equation 3.2.9), the middle two terms are equal to zero. Also since $\phi_{xx}(\tau)=\phi_{yy}(\tau)$, the first term cancels out the last term. As a result, the cross-correlation between $z_i[m]$ and $z_q[m]$ is always zero for all time shifts, that is,

$$\phi_{z_i z_q}[m;m-k]=\phi_{z_i z_q}[k]=0 \qquad \text{for all } k \qquad (3.6.2.23)$$

The last statistical quantity being studied is the probability density functions of $z_i[m]$ and $z_q[m]$. Due to the fact that both $z_i[m]$ and $z_q[m]$ are linear combinations of two uncorrelated Gaussian processes, it follows that $z_i[m]$ and $z_q[m]$ are also Gaussian processes [17, pp.41].

3.6.3 Simulation Setup and Results

In general, the reliability of a digital communication system is measured by its probability of bit error. To show the performance of the DS/SS IF-sampling system, with

the presence of sampling error and bandpass white noise, a bit-error-rate (BER) graph is generated from the simulation in [30]. During the simulation, a random number generator is used to generate the information bits and the information bits are obtained by the following rule: a bit is 0 if the uniform random deviate is from 0 (exclusive) to 0.5 (inclusive); while it is 1 if the uniform random deviate is from 0.5 (exclusive) to 1.0 (exclusive).

The shaping pulse being simulated is the 100% excess bandwidth duobinary pulse with duration of 98 chips, that is,

$$g(t)=\begin{cases}\dfrac{0.25T_c^2\cdot\sin(2\pi/T_c)}{\pi(0.5T_c-t)}\cdot\dfrac{1}{\sqrt{T_c}}, & -49T_c\le t\le 49T_c\\ 0, & otherwise\end{cases} \qquad (3.6.3.1)$$

With this long duration, nearly all ISI introduced by the tails of the duobinary pulse are taken into account. Notice that an extra scaling factor is added in the definition of $g(t)$. As will be shown later, this scaling factor is used to make the SNR independent of the chip duration T_c. Since $g(t)$ is a truncated duobinary pulse with a long duration, the Fourier transform of $g(t)$ can be approximated as

$$G(f)\approx\begin{cases}T_c\cdot\cos\dfrac{\pi fT_c}{2}\cdot e^{-j\frac{\pi fT_c}{2}}\cdot\dfrac{1}{\sqrt{T_c}}, & |f|\le\dfrac{1}{T_c}\\ 0, & otherwise\end{cases} \qquad (3.6.3.2)$$

By applying the Parseval's Theorem into the definition of energy of $g(t)$,

$$E_g=\int_{-\infty}^{\infty}g^2(t)dt=\int_{-\infty}^{\infty}|G(f)|^2df \qquad (3.6.3.3)$$

the energy of $g(t)$ in found to be,

$$E_g\approx\int_{-\frac{1}{T_c}}^{\frac{1}{T_c}}T_c\cdot\cos^2\left(\frac{\pi fT_c}{2}\right)df=1 \qquad (3.6.3.4)$$

Referring to Section 3.1, the transmitted signal $s(t)$ is give as

$$s(t)=\mathrm{Re}\left\{v(t)\cdot e^{j2\pi f_c t}\right\} \tag{3.6.3.5}$$

where

$$v(t)=\sum_{n=-\infty}^{\infty}(X_n+jY_n)\cdot g(t-nT_c)$$

It can be shown that the lowpass equivalent signal $v(t)$ is a cyclostationary process with an average power density spectrum given by

$$\Phi_{vv}(f)=\frac{1}{T_c}\cdot|G(f)|^2\cdot\frac{1}{2}E\left\{|X_n+jY_n|^2\right\} \tag{3.6.3.6}$$

Since the total average power of $s(t)$ is the total area under $\Phi_{ss}(f)$ and

$$\Phi_{ss}(f)=\frac{1}{2}\Phi_{vv}(f-f_c)+\frac{1}{2}\Phi_{vv}(-f-f_c) \tag{3.6.3.7}$$

then

$$\begin{aligned}\text{Average power of } s(t) &= \int_{-\infty}^{\infty}\Phi_{ss}(f)df\\ &=\frac{1}{2}\int_{-\infty}^{\infty}\Phi_{vv}(f-f_c)df+\frac{1}{2}\int_{-\infty}^{\infty}\Phi_{vv}(-f-f_c)df\\ &=\frac{1}{2}\int_{-\infty}^{\infty}\Phi_{vv}(f)df+\frac{1}{2}\int_{-\infty}^{\infty}\Phi_{vv}(f)df\\ &=\int_{-\infty}^{\infty}\Phi_{vv}(f)df\\ &=\frac{1}{2T_c}\cdot E\left\{|X_n+jY_n|^2\right\}\cdot\int_{-\infty}^{\infty}|G(f)|^2df\end{aligned} \tag{3.6.3.8}$$

Due to the fact that the complex chips X_n+jY_n are obtained from $\pi/4$-shifted-DQPSK symbols, the expectation term in Equation 3.6.3.9 becomes

$$E\left\{|X_n+jY_n|^2\right\}=E\{1\}=1 \tag{3.6.3.9}$$

By substituting Equation 3.6.3.4 and 3.6.3.9 into Equation 3.6.3.8, the average power for $s(t)$ becomes

$$\text{Average power of } s(t) = \frac{1}{2T_c} \cdot (1) \cdot (1) = \frac{1}{2T_c} \tag{3.6.3.10}$$

The SNR for a communication system is defined as

$$SNR = \frac{Average\ power\ of\ transmitted\ bandpass\ signal}{Average\ power\ of\ in-band\ bandpass\ noise} \tag{3.6.3.11}$$

As mentioned in Section 3.6.2, average power for the in-band bandpass noise sequences, $z_i[m]$ and $z_q[m]$, are

$$\phi_{z_i z_i}[0] = \phi_{z_q z_q}[0] = \frac{2N_0}{T_c} \tag{3.6.3.12}$$

By substituting Equations 3.6.3.10 and 3.6.3.12 into Equation 3.6.3.11, the SNR for this system is found to be

$$SNR = \frac{1/(2T_c)}{2N_0/T_c} = \frac{1}{4N_0} \tag{3.6.3.13}$$

Equivalently the SNR expressed in decibel (dB) is

$$SNR = 10 \cdot \log_{10}\left(\frac{1}{4N_0}\right) \tag{3.6.3.14}$$

Notice that the SNR is independent of the chip rate with the extra scaling term in the shaping pulse (Equation 3.6.3.1) and therefore the chip period is arbitrarily chosen to be 1 in the simulation.

In addition, the phase error, $-\frac{\pi t_d}{T_c}+\theta$, in the system is ignored throughout the simulation. This is because the phase error does not affect the system performance of the DQPSK detector as demonstrated in Section 3.3.

The simulation is performed for timing error $t_d = 0, 0.1T_c, 0.15T_c, 0.2T_c, 0.25T_c$ and the results are shown in Figure 3.6.3.1. For t_d equal to 0, the bandpass white noise is the only distortion degrading the system performance. When t_d is not zero, the system suffers from both timing error and noise. Therefore the bit error curve for $t_d = 0$ is used as a reference in our investigation. At a bit error rate of 10^{-3}, the DS/SS system suffers about 1 dB loss for $t_d = 0.1T_c$. As t_d increases to its worst value of $0.25T_c$, the degradation is 7 dB. The performance degradation mainly comes from three sources: (1) the residual ISI, (2) the discard of signal power in the delayed path and (3) gain imbalance between the two noise sequences. Since the autocorrelation of the Barker sequence is not exactly zero for non-zero time shift (Table 2.2.3.1), the interfering paths are suppressed but not completely removed. Therefore a residual amount of ISI still exists at the output of the despreader which degrades the performance. Another factor is the discard of signal power in the delayed path. Instead of using the signal power from the delayed path, the despreader simply suppresses or discards it to alleviate ISI. As a result, the power carried by the delayed path signal is not recovered. In addition, the bandpass noise sequences, $z_i[m]$ and $z_q[m]$, are also multiplied by G_i and G_q respectively. Unless the two gains are exactly the same, gain imbalance is introduced on the two noise sequences that also degrades the system performance.

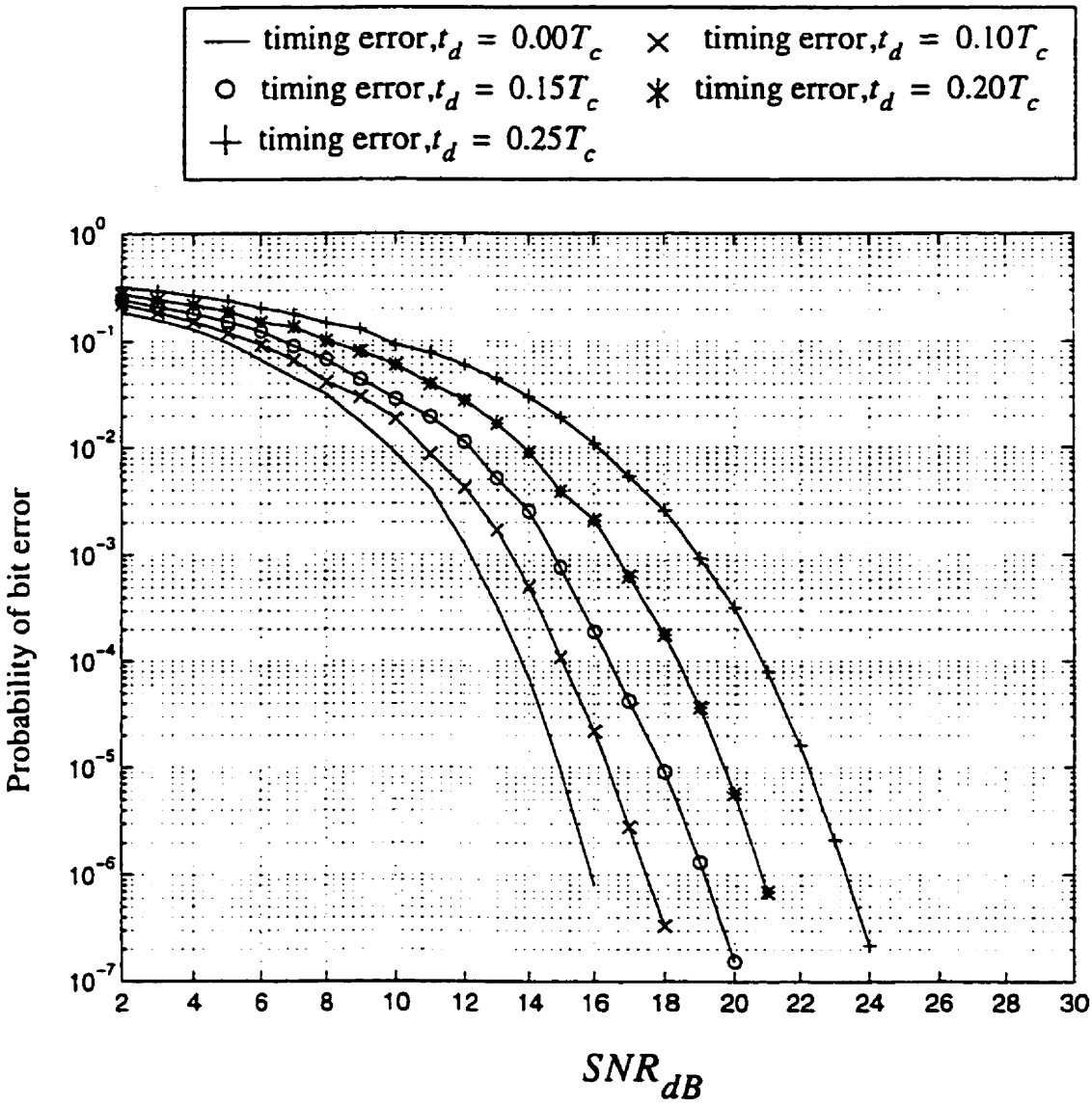


Figure 3.6.3.1 Simulated performance of the DS/SS IF-sampling system (from [30]).

Although the DS/SS IF-sampling system still suffers degradation from timing error, the system performance does not display an error floor. Even with the worst case timing error $t_d = 0.25T_c$, the bit error curve still has a waterfall shape which means that

the system recovers almost all the transmitted information bits perfectly when the noise is negligible.

3.7 Consideration for System Implementation

In [30], the transmitter is a standard DS/SS digital transmitter which can be used in any DS/SS system. Therefore cost and complexity is not an issue. The receiver, however, is unique to IF-sampling system. Basically, its IF-stage can be implemented with an ADC, a despreader and two variable gains as shown in Figure 3.7.1.

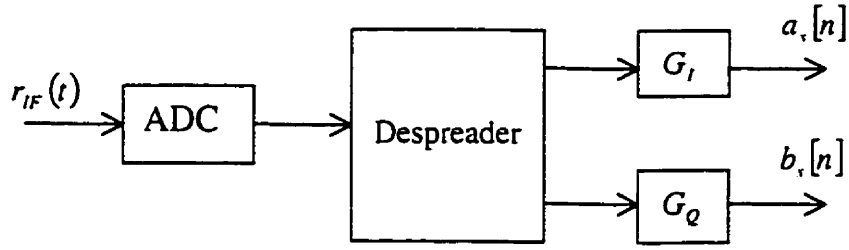


Figure 3.7.1 IF-stage of the DS/SS IF-sampling receiver.

The cost and complexity of the ADC depend on its precision and sampling rate. This is common to all digital receivers. The despreader performs despreading, sign conversion and decimation. All these operation can be carried out by digital components such as adders, subtracters and logic gates. Therefore the cost and complexity of the despreader is limited. However, the variable gains, G_I and G_Q, are costly to implement. These gains require multiplication, lookup table and training. Multipliers are the most

expensive digital component. They are complex and must be fast. The lookup table requires memory and the training takes time. As a result, the receiver cost and complexity increases dramatically with the presence of the two variable gains, G_I and G_Q. Consequently, G_I and G_Q are not acceptable for a low cost implementation of the IF-sampling system.

3.8 Summary

This chapter contains a detailed analysis of the DS/SS IF-sampling system in [30] with a sampling rate of 2 samples/chip. Throughout the analysis, we showed that the presence of a sampling timing error causes asymmetric ISI and amplitude imbalance problem between the I and Q channels. We also showed that the DS/SS IF-sampling receiver is capable of handling these problems. The BER graph (Figure 3.6.3.1), generated by the simulation, showed that the difference between the worst case sampling timing error ($t_d = 0.25T_c$) and the best case sampling timing error ($t_d = 0$) was about 7 dB.

To implement the system, all components must be common and easy to build. To save cost the two variable gains, G_I and G_Q must be eliminated.

Chapter 4 The Proposed DS/SS IF-sampling System

In the previous chapter, we showed the architecture and the performance analysis of DS/SS IF-sampling system according to [30]. Although such a system is relatively simpler than conventional IF-sampling systems, its complexity is still high due to two variable gains, G_I and G_Q, at the receiver back end (Figure 3.3.1). The goal of the purposed DS/SS IF-sampling system is to find an alternative way to achieve a acceptable performance without the two variable gains, G_I and G_Q. This chapter starts with a discussion regarding removing G_I and G_Q. Then it presents a performance analysis for the proposed DS/SS IF-sampling system with a bandpass white Gaussian channel. Finally, it ends with a brief summary.

4.1 Removing G_I and G_Q

In Section 3.6.1, we showed that G_I and G_Q are used to remove the amplitude imbalance between the received I and Q symbols in the presence of sampling timing error. We also showed that the values of G_I and G_Q depend on the sampling timing error t_d and the shaping pulse $g(t)$, that is,

$$G_I = \frac{1}{g(-t_d)} \tag{4.1.1}$$

$$G_Q = \frac{1}{g(0.5T_c - t_d)} \tag{4.1.2.}$$

where

$$-0.25T_c \le t_d \le 0.25T_c .$$

If the difference between $g(-t_d)$ and $g(0.5T_c - t_d)$ is small, then G_I and G_Q can be ignored. Otherwise, G_I and G_Q must be used in order to maintain the system performance. In [30], the shaping pulse $g(t)$ (Equation 3.5.16) is a duobinary pulse with 100% excess bandwidth. Using this pulse, the variations between $g(-t_d)$ and $g(0.5T_c - t_d)$ (Table 3.6.1.1 and 3.6.1.2) are too large especially when $|t_d| \approx 0.25T_c$. Therefore, the system in [30] will experience a performance degradation if G_I and G_Q are not used. In other words, the shaping pulse must be reselected in order to remove G_I and G_Q.

To select the shaping pulse, $g(t)$, for the proposed DS/SS IF-sampling system, the main goal is to reduce the variations between $g(-t_d)$ and $g(0.5T_c - t_d)$ when a limited bandwidth is used. This can be achieved only if the new pulse has a wider pulse width. However, a wider pulse width normally introduces more ISI, which means that the receiver has to spend more effort to fight such an ISI. As a result, the pulse width should be kept to a minimum but wide enough to keep the variations between $g(-t_d)$ and $g(0.5T_c - t_d)$ reasonably small. To satisfy all the requirements for the shaping pulse, a time-shifted version of the duobinary pulse is used in the proposed DS/SS IF-sampling system and the equation of the pulse is

$$g(t)=\frac{T_c^2\cdot\sin[\pi(t+0.25T_c)/T_c]}{\pi(t+0.25T_c)\cdot[T_c-(t+0.25T_c)]} \tag{4.1.3}$$

The frequency spectrum of $g(t)$ is

$$G(f)=\begin{cases}2T_c\cdot\cos(\pi fT_c)\cdot e^{j1.5\pi fT_c}, & |f|\le\dfrac{1}{2T_c}\\ 0, & |f|>\dfrac{1}{2T_c}\end{cases} \tag{4.1.4}$$

The impulse and frequency response of $g(t)$ are plotted in Figure 4.1.1 and 4.1.2 respectively.

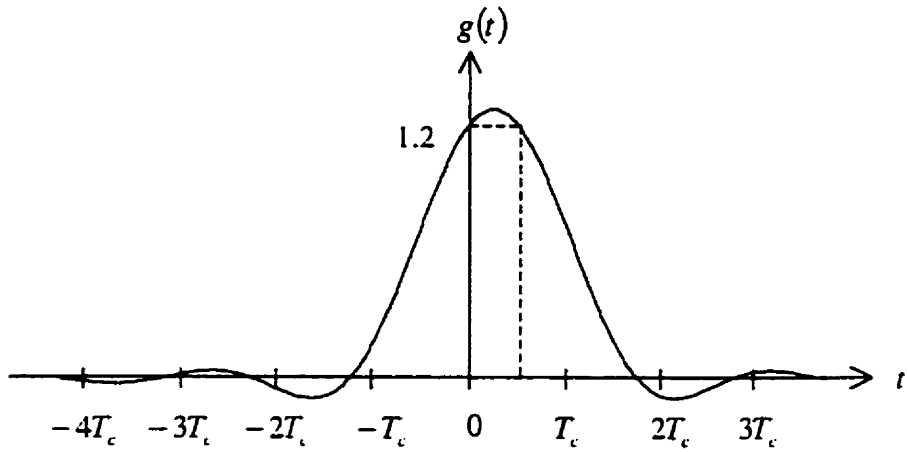


Figure 4.1.1 Impulse response of the time-shifted duobinary pulse $g(t)$.

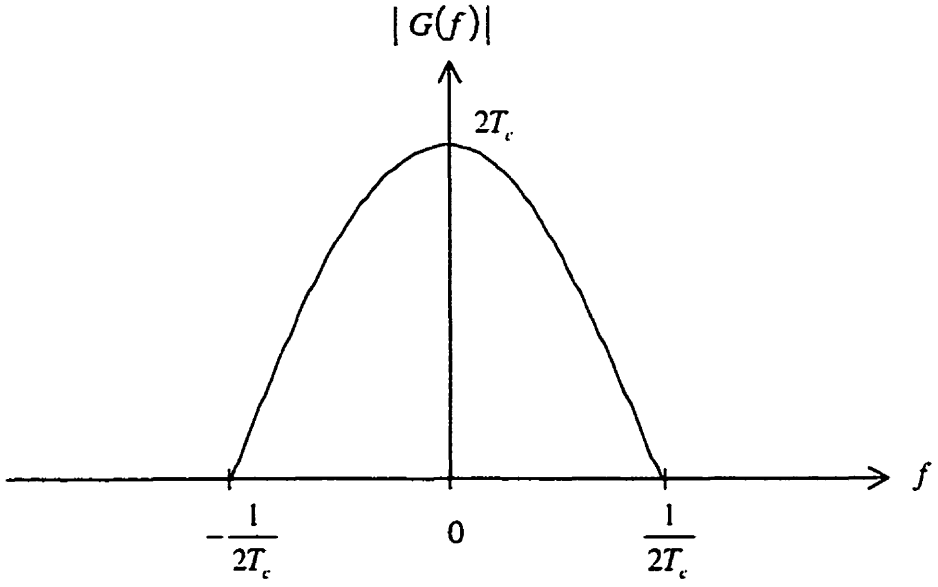


Figure 4.1.2 Frequency response of the time-shifted duobinary pulse $g(t)$.

t_d	$g(-t_d)$	$g(0.5T_c - t_d)$
$-0.25T_c$	1.2732	1.0
$-0.20T_c$	1.2703	1.0483
$-0.15T_c$	1.2614	1.0929
$-0.10T_c$	1.2467	1.1334
0	1.2004	1.2004
$0.10T_c$	1.1334	1.2467
$0.15T_c$	1.0929	1.2614
$0.20T_c$	1.0483	1.2703
$0.25T_c$	1.0	1.2732

Table 4.1.1 Amplitude variations of the time-shifted duobinary pulse $g(t)$.

Comparing the impulse responses shown in Figure 3.5.4 and Figure 4.1.1, the result shows that the amplitude variation between $t = -0.25T_c$ and $t = 0.25T_c$ is less significant for the time-shifted duobinary pulse. This result is shown more clearly in Table 4.1.1.

4.2 Performance Analysis for the Bandpass White Gaussian Channel

The architecture of the proposed DS/SS IF-sampling system is identical to the system in [30] except for the receiver. In the proposed receiver, the variable gains, G_I and G_Q, are eliminated as shown in Figure 4.2.1. In this section, the performance analysis is carried out the same way as in Section 3.6. Firstly, the section begins with the analysis of ISI suppression during the despreading process. Then it discusses the effect of the bandpass white noise in the proposed IF-sampling receiver. Finally, this section shows the simulation setup and results for the proposed IF-sampling system.

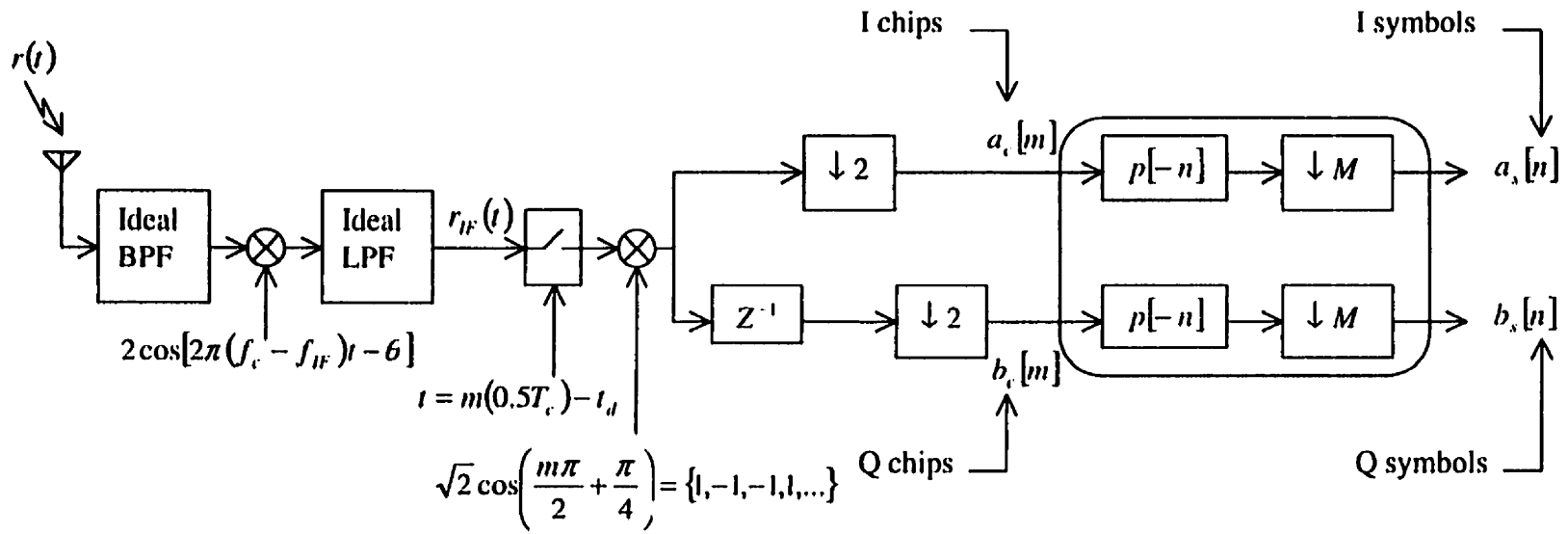


Figure 4.2.1 Architecture of the proposed DS/SS IF-sampling receiver.

4.2.1 Despreading

Since the proposed receiver is identical to the receiver in [30] up to the despreader, most of the equations in Section 3.6.1 can be reused. Therefore the received I and Q chip sequences are

$$a_c[m] = \sum_{n=-\infty}^{\infty} X_n \cdot g(mT_c - t_d - nT_c) \tag{4.2.1.1}$$

$$b_c[m] = \sum_{n=-\infty}^{\infty} Y_n \cdot g(mT_c + 0.5T_c - t_d - nT_c) \tag{4.2.1.2}$$

where

$$X_n = \sum_{k=-\infty}^{\infty} I_k \cdot p[n - kM] \tag{4.2.1.3}$$

$$Y_n = \sum_{k=-\infty}^{\infty} Q_k \cdot p[n - kM] \tag{4.2.1.4}$$

The shaping pulse $g(t)$ is a time-shifted version of the duobinary pulse with truncated tails and the impulse response of $g(t)$ is

$$g(t) = \begin{cases} \dfrac{T_c^2 \cdot \sin[\pi(t + 0.25T_c)/T_c]}{\pi(t + 0.25T_c) \cdot [T_c - (t + 0.25T_c)]}, & -1.25T_c \le t \le 1.75T_c \\ 0, & \textit{otherwise} \end{cases} \tag{4.2.1.5}$$

After despreading, the recovered I and Q symbol sequences are obtained as

$$a_s[n] = \sum_{m=-\infty}^{\infty} a_c[m] \cdot p[m - nM] \tag{4.2.1.6}$$

$$b_s[n] = \sum_{m=-\infty}^{\infty} b_c[m] \cdot p[m - nM] \tag{4.2.1.7}$$

Recall that the system in [30] does not suffer from ISI when $t_d = 0$. However ISI always exists in the proposed system independent of the value of t_d as shown in Figure 4.2.2. This is because the shaping pulse in the proposed system has a much wider pulse width.

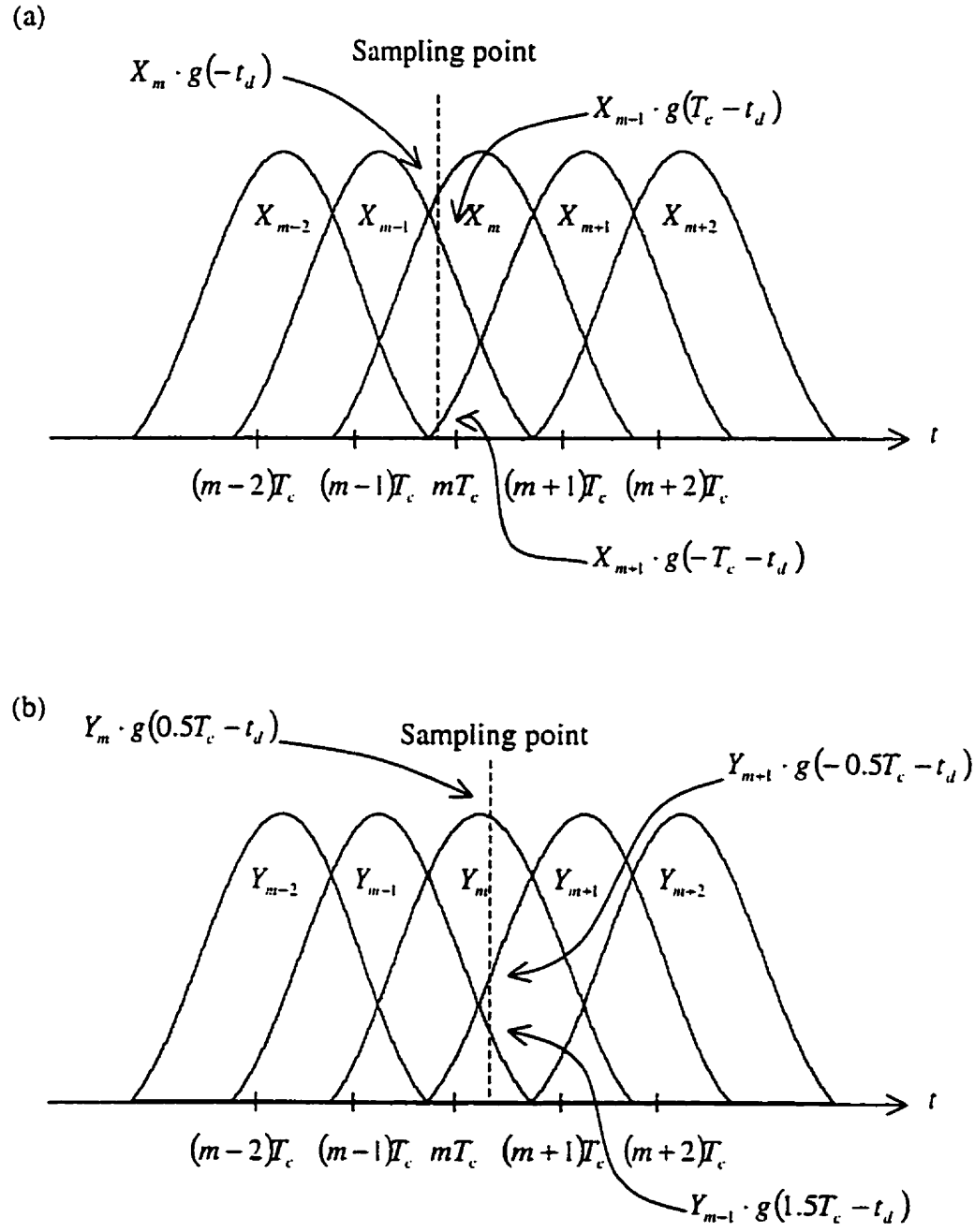


Figure 4.2.2 (a) ISI in the received inphase chip sequence.

(b) ISI in the received quadrature chip sequence.

As shown in Figure 4.2.2, the I and Q chip sequences (Equation 4.2.1.1 and 4.2.1.2) are non-zero only when $n = m$, $n = m-1$, and $n = m+1$. Therefore the I and Q chip sequences are reduced to

$$\begin{aligned} a_c[m] &= X_m \cdot g(-t_d) \\ &\quad + X_{m-1} \cdot g(T_c - t_d) \\ &\quad + X_{m+1} \cdot g(-T_c - t_d) \end{aligned} \tag{4.2.1.8}$$

$$\begin{aligned} b_c[m] &= Y_m \cdot g(0.5T_c - t_d) \\ &\quad + Y_{m-1} \cdot g(1.5T_c - t_d) \\ &\quad + Y_{m+1} \cdot g(-0.5T_c - t_d) \end{aligned} \tag{4.2.1.9}$$

Substituting Equation 4.2.1.3 and 4.2.1.8 into Equation 4.2.1.6, the recovered I symbol is given as

$$\begin{aligned} a_s[n] &= g(-t_d) \cdot \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} I_k \cdot p[m-kM] \cdot p[m-nM] \\ &\quad + g(T_c - t_d) \cdot \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} I_k \cdot p[m-1-kM] \cdot p[m-nM] \\ &\quad + g(-T_c - t_d) \cdot \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} I_k \cdot p[m+1-kM] \cdot p[m-nM] \\ &= g(-t_d) \cdot M \cdot I_n + g(T_c - t_d) \cdot (\pm 1) \cdot I_n + g(-T_c - t_d) \cdot (\pm 1) \cdot I_n \\ &\approx g(-t_d) \cdot M \cdot I_n \end{aligned} \tag{4.2.1.10}$$

Similarly, the recovered Q symbol is obtained by substituting Equation 4.2.1.4 and 4.2.1.9 into Equation 4.2.1.7 and the result is

$$
\begin{aligned}
b_s[n] &= g(0.5T_c - t_d)\cdot \sum_{m=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} Q_k \cdot p[m-kM]\cdot p[m-nM] \\
&+ g(1.5T_c - t_d)\cdot \sum_{m=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} Q_k \cdot p[m-1-kM]\cdot p[m-nM] \\
&+ g(-0.5T_c - t_d)\cdot \sum_{m=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} Q_k \cdot p[m+1-kM]\cdot p[m-nM] \\
&= g(0.5T_c - t_d)\cdot M\cdot Q_n + g(1.5T_c - t_d)\cdot(\pm 1)\cdot Q_n + g(-0.5T_c - t_d)\cdot(\pm 1)\cdot Q_n \\
&= g(0.5T_c - t_d)\cdot M\cdot Q_n \qquad (4.2.1.11)
\end{aligned}
$$

As a result, the effect of ISI is successfully removed from both I and Q symbols after despreading. Note that the I and Q symbols still experience different gains in the receiver since $g(-t_d) \neq g(0.5T_c - t_d)$. However the difference between $g(-t_d)$ and $g(0.5T_c - t_d)$ (Table 4.1.1) is small enough to be ignored. Hence the performance of the proposed system will not be affected without the gain correction factors, G_I and G_Q.

4.2.2 The Effect of Bandpass White Noise in the Proposed Receiver

Similar to Section 3.6.2, the received I and Q bandpass white noise is

$$
\begin{aligned}
z_s[m] &= x(mT_c - t_d)\cdot \cos\left(-\frac{\pi t_d}{T_c} + \theta\right) \\
&- y(mT_c - t_d)\cdot \sin\left(-\frac{\pi t_d}{T_c} + \theta\right) \qquad (4.2.2.1)
\end{aligned}
$$

and

$$z_q[m] = x(mT_c + 0.5T_c - t_d)\cdot\sin\left(-\frac{\pi t_d}{T_c}+\theta\right) + y(mT_c + 0.5T_c - t_d)\cdot\cos\left(-\frac{\pi t_d}{T_c}+\theta\right) \quad (4.2.2.2)$$

respectively. According to Section 3.2, the mean of $x(t)$ and $y(t)$ are

$$E\{x(t)\}= E\{y(t)\}=0 \qquad \text{for all } t \quad (4.2.2.3)$$

and the autocorrelation of $x(t)$ and $y(t)$ are

$$\phi_{xx}(\tau)=\phi_{yy}(\tau)= N_0\cdot\frac{\sin(\pi W\tau)}{\pi\tau} \quad (4.2.2.4)$$

According to Section 3.6.2, the mean of $z_i[m]$ and $z_q[m]$ are

$$E\{z_i[m]\}= E\{z_q[m]\}=0 \qquad \text{for all } m \quad (4.2.2.5)$$

and the autocorrelation of $z_i[m]$ and $z_q[m]$ are

$$\phi_{z_iz_i}[k]=\phi_{z_qz_q}[k]= N_0\cdot\frac{\sin(\pi WkT_c)}{\pi kT_c} \quad (4.2.2.6)$$

Note that Equations 4.2.2.4 and 4.2.2.6 imply that $\phi_{z_iz_i}[k]$ is a sampled version of $\phi_{xx}(\tau)$ with a sampling rate of $1/T_c$ and the same relationship applies to $\phi_{z_qz_q}[k]$ and $\phi_{yy}(\tau)$. In addition, Section 3.6.2 also shows that the cross-correlation between $z_i[m]$ and $z_q[m]$ is zero and the total average power of $z_i[m]$ and $z_q[m]$ are

$$\text{Total Average Power of } z_i[m]=\phi_{z_iz_i}[k=0]= N_0W \quad (4.2.2.7)$$

$$\text{Total Average Power of } z_q(t)=\phi_{z_qz_q}[k=0]= N_0W \quad (4.2.2.8)$$

Since the proposed system uses the time-shifted duobinary pulse (Equation 4.1.4), the bandwidth of the system is $W = 1/T_c$. Therefore Equation 4.2.2.4 becomes

$$\phi_{cc}(\tau) = \phi_{vv}(\tau) = N_0 \cdot \frac{\sin\left(\pi \cdot \frac{1}{T_c} \cdot \tau\right)}{\pi\tau} \tag{4.2.2.9}$$

and Equation 4.2.2.6 becomes

$$\phi_{z_i z_i}[k] = \phi_{z_{i-1} z_{i-1}}[k] = N_0 \cdot \frac{\sin\left(\pi \cdot \frac{1}{T_c} \cdot kT_c\right)}{\pi k T_c} = N_0 \cdot \frac{\sin(\pi k)}{\pi k T_c} = \frac{N_0}{T_c} \cdot \delta[k] \tag{4.2.2.10}$$

The relationship between Equation 4.2.2.9 and 4.2.2.10 is shown in Figure 4.2.2.1.

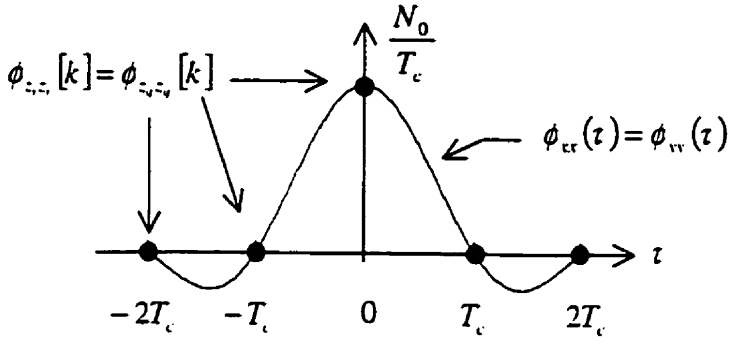


Figure 4.2.2.1 Relationship between $\phi_{cc}(\tau)$, $\phi_{vv}(\tau)$, $\phi_{z_i z_i}[k]$, and $\phi_{z_{i-1} z_{i-1}}[k]$ for the time-shifted duobinary pulse.

In the frequency domain, the power density spectra of $x(t)$ and $y(t)$ change to

$$\Phi_{xx}(f)=\Phi_{yy}(f)=\begin{cases} N_0, & |f|<\dfrac{1}{2T_c} \\ 0, & otherwise \end{cases} \qquad (4.2.2.11)$$

with the new shaping pulse. By periodically replicating $\Phi_{xx}(f)$ with a period of $1/T_c$ and multiplying the spectrum by $W = 1/T_c$, the power density spectra of $z_i[m]$ and $z_q[m]$ are obtained as

$$\Phi_{z_i z_i}\left(e^{j2\pi fT_c}\right)=\Phi_{z_q z_q}\left(e^{j2\pi fT_c}\right)=\frac{N_0}{T_c} \qquad (4.2.2.12)$$

All of these four power density spectra are shown in Figure 4.2.2.2.

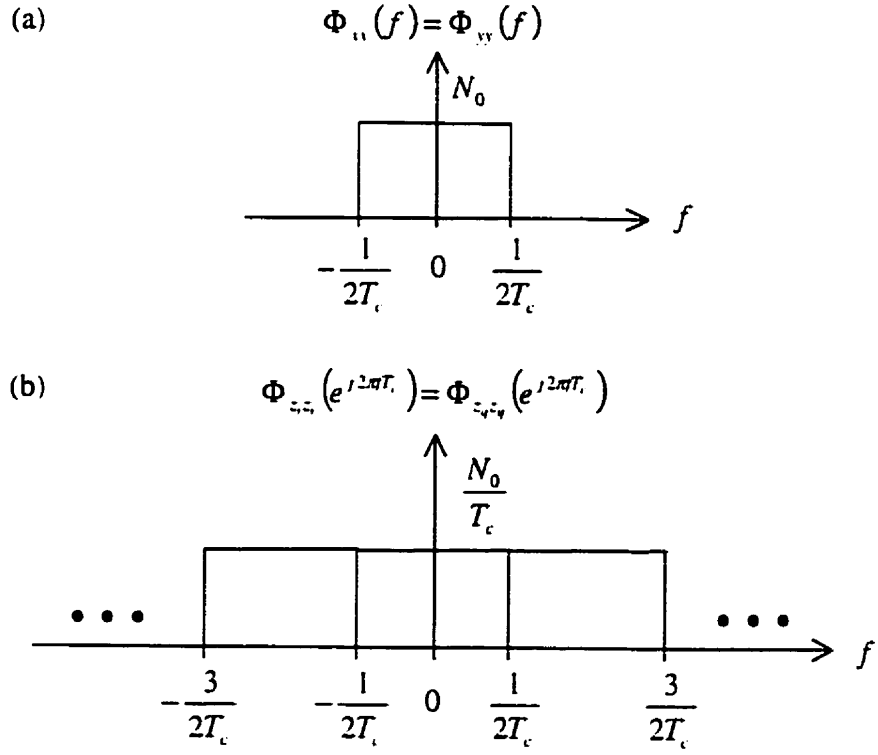


Figure 4.2.2.2 (a) Power density spectra of $x(t)$ and $y(t)$.
(b) Power density spectra of $z_i[m]$ and $z_q[m]$.
(Time-shifted duobinary pulse)

Since $\phi_{z_i z_i}[k]$ is a scaled discrete-time delta function and $\Phi_{z_i z_i}(e^{j2\pi fT_s})$ has a flat frequency spectrum, $z_i[m]$ is a white Gaussian random process. Since $\phi_{z_q z_q}[k] = \phi_{z_i z_i}[k]$ and $\Phi_{z_q z_q}(e^{j2\pi fT_s}) = \Phi_{z_i z_i}(e^{j2\pi fT_s})$, $z_q[m]$ is also a white Gaussian random process. As a result, the effect of the bandpass noise is the same in both the proposed and the receiver in [30].

4.2.3 Simulation Setup and Results

To verify the performance of the proposed DS/SS IF-sampling system, in the presence of sampling error and bandpass white noise, a Matlab simulation program is used to generate bit-error-rate (BER) curves. In the simulation program, the Matlab function 'randn' is used to generate the information bits and the information bits are obtained the same way as in the simulation in [30] (Section 3.6.3).

The shaping pulse used in the simulation is the time-shifted duobinary pulse. As shown in Section 4.2.1, the ISI, caused by the main lobe of the shaping pulse, is insignificant. Therefore the ISI caused by the tails of the shaping pulse is negligible. As a result, the simulation program uses the time-shifted duobinary pulse with truncated tails, that is,

$$g(t)=\begin{cases}\dfrac{T_c^2\cdot\sin[\pi(t+0.25T_c)/T_c]}{\pi(t+0.25T_c)\cdot[T_c-(t+0.25T_c)]}\cdot\dfrac{1}{2\sqrt{T_c}}, & -1.25T_c\le t\le 1.75T_c\\ 0, & \textit{otherwise}\end{cases}$$

(4.2.3.1)

Similar to Section 3.6.3, an extra scaling factor is added to the definition of $g(t)$. In this case, the scaling factor serves two purposes. The first purpose is to make the SNR independent of the chip duration T_c. The second purpose is to make the SNR the same as the system in [30]. In this way, the simulation results in [30] can be used as a reference.

To perform the SNR calculation, the Fourier transform of $g(t)$ is approximated as

$$G(f)\approx\begin{cases}2T_c\cdot\cos\pi fT_c\cdot e^{-j1.5\pi fT_c}\cdot\dfrac{1}{2\sqrt{T_c}}, & |f|\le\dfrac{1}{2T_c}\\ 0, & \textit{otherwise}\end{cases}\qquad(4.2.3.2)$$

By applying Parseval's Theorem into the definition of energy of $g(t)$,

$$E_g=\int_{-\infty}^{\infty}g^2(t)dt=\int_{-\infty}^{\infty}|G(f)|^2df\qquad(4.2.3.3)$$

the energy of $g(t)$ is found to be,

$$E_g=\int_{-\frac{1}{2T_c}}^{\frac{1}{2T_c}}T_c\cdot\cos^2(\pi fT_c)df=\frac{1}{2}\qquad(4.2.3.4)$$

In Section 3.6.3, the average power of the transmitted signal, $s(t)$, is found to be

$$\text{Average power of } s(t)=\frac{1}{2T_c}\cdot E\left\{|X_n+jY_n|^2\right\}\cdot\int_{-\infty}^{\infty}|G(f)|^2df\qquad(4.2.3.5)$$

where $E\left\{|X_n+jY_n|^2\right\}=1$. By substituting Equation 4.2.3.4 into Equation 4.2.3.5, the average power of the transmitted signal becomes

$$\text{Average power of } s(t) = \frac{1}{2T_c} \cdot (1) \cdot \left(\frac{1}{2}\right) = \frac{1}{4T_c} \tag{4.2.3.6}$$

Using Equations 3.6.3.11, 4.2.3.6 and 4.2.2.12, the SNR of the system is obtained as

$$SNR = \frac{1/(4T_c)}{N_0/T_c} = \frac{1}{4N_0} \tag{4.2.3.7}$$

Equivalently the SNR expressed in decibel (dB) is

$$SNR = 10 \cdot \log_{10}\left(\frac{1}{4N_0}\right) \tag{3.6.3.14}$$

The simulation results are shown in Figure 4.2.3.1. Comparing the ideal performance of the proposed DS/SS IF-sampling system (square pulse) and the DS/SS IF-sampling system in [30] ($t_d = 0.00T_c$), it seems that the proposed DS/SS IF-sampling system outperforms the system in [30] by approximately 1.5 dB. This means that one of us has made a mistake in our simulations. However the fact is that we are using a different random number generator in our simulations. Thus our results could vary slightly depending on the different performances of the random number generators that we used in our simulations. In this case, the performance of our systems can be compared by the *SNR* difference between the best case curve and the worst case curve at a low bit error rate ($\approx 10^{-6}$). This way, the system with a smaller *SNR* difference, which means that it is less sensitive to the timing error, will be a better system.

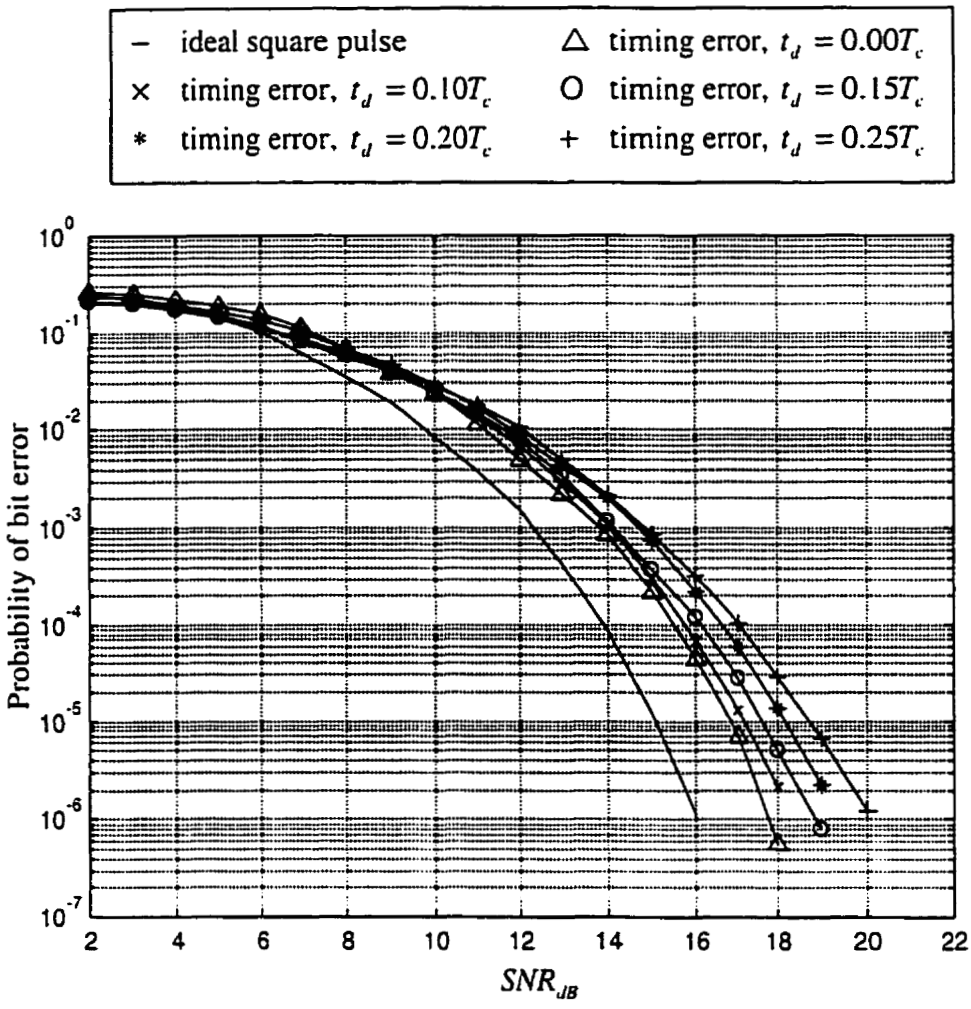


Figure 4.2.3.1 Simulated performance of the proposed DS/SS IF-sampling system.

Unlike the system in [30], the proposed system suffers from residual ISI even when $t_d = 0.00T_c$. However the performance degradation from $t_d = 0.00T_c$ to $t_d = 0.25T_c$ is reduced to approximately 2 dB and the overall performance degradation from the ideal curve to $t_d = 0.25T_c$ is reduced to approximately 4 dB. As a result, the proposed system outperforms the system in [30] by approximately 3 dB. This further

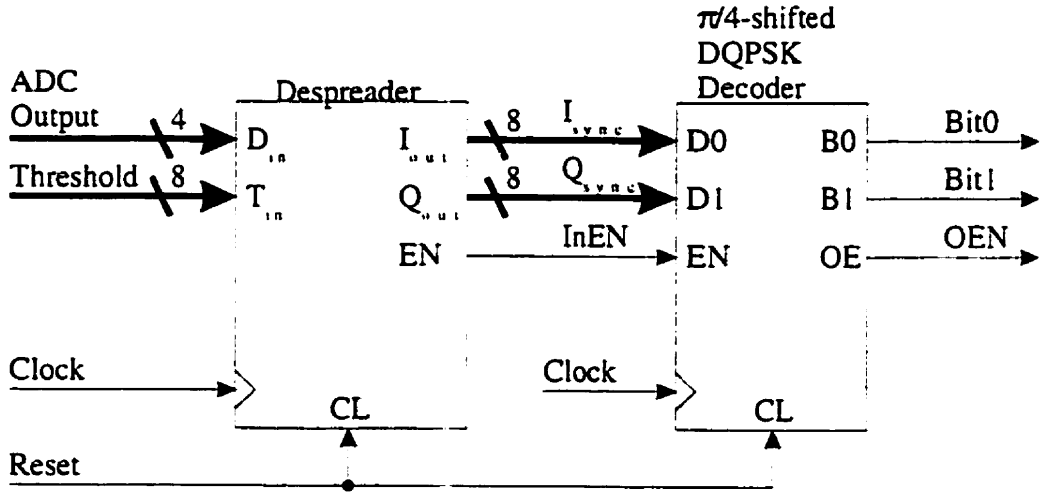
proves that the performance of the 2 samples/symbol DS/SS IF-sampling system is more sensitive to the distortion caused by the I/Q amplitude imbalance than by the residual ISI.

4.3 Summary

To reduce the implementation complexity and improve the performance of the 2 samples/symbol DS/SS IF-sampling system in [30], the proposed system uses the time-shifted duobinary pulse as its shaping pulse. Using this new shaping pulse, the I/Q amplitude imbalance is greatly reduced but the residual ISI is increased. Despite the increase of the residual ISI, the simulation results show that the proposed system has approximately 3 dB performance improvement over the system in [30]. This further verifies that the I/Q amplitude imbalance is the major source of the performance degradation of the 2 samples/symbol DS/SS IF-sampling system.

Chapter 5 Implementation of the Proposed DS/SS IF-sampling Receiver

To implement the proposed receiver (not including the RF front end), dedicated hardware is necessary to perform the task of despreading, data synchronization and decoding. For many similar applications, a digital signal processing (DSP) microprocessor based system is used for implementation. With all the effort spent in Chapter 4 to simplify the receiver structure, it makes sense to implement the proposed receiver with discrete logic components which is a cheaper solution.



ADC Output - Output of the Analog-to-Digital converter
Threshold - I/Q magnitude threshold
Clock - Master clock signal
Reset - Master reset signal

I_{sync} - Synchronized inphase symbol
Q_{sync} - Synchronized quadrature symbol
InEN - Symbol ready signal
Bit0 - Inphase bit
Bit1 - Quadrature bit
OEN - Bit ready signal

Figure 5.1 The block diagram of the proposed receiver.

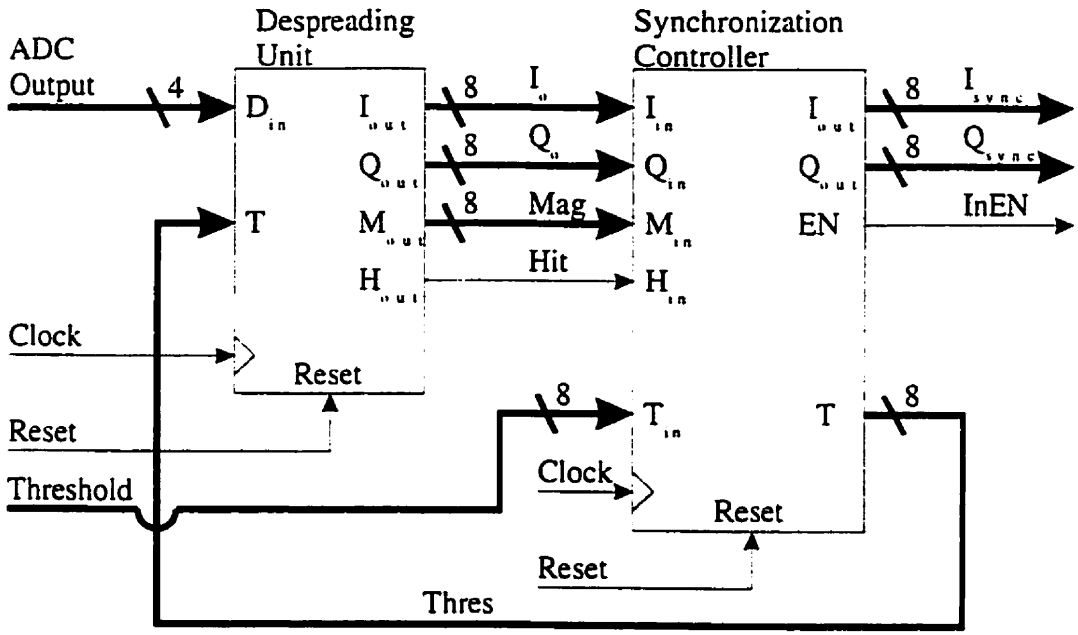
The proposed receiver consists of two functional blocks: a despreader and a $\pi/4$-shifted DQPSK decoder. The block diagram of the proposed receiver is shown in Figure 5.1. In the following sections, the details of each functional block are given. To test the receiver functionality, a software tool called Logsim is used to simulate the receiver. The description of this tool can be found in Appendix A.

5.1 The Proposed Receiver

The function of the proposed receiver is to extract the information bits from the ADC output samples (4-bit) given an input I/Q magnitude threshold (8-bit). The proposed receiver is divided into two parts: the despreader and the $\pi/4$-shifted DQPSK decoder. The details about the despreader and the $\pi/4$-shifted DQPSK decoder are given in Sections 5.2 and 5.3.

5.2 The Despreader

The despreader, as shown in Figure 5.2.1, consists of two major functional blocks: a despreading unit and a synchronization controller. It will take 4-bit samples from the ADC and an 8-bit I/Q magnitude threshold to reconstruct synchronized I/Q symbols (I_{sync} and Q_{sync}) along with a symbol ready signal ($InEN$). The details about the despreading unit and the synchronization controller are given in Section 5.2.1 and 5.2.2.



I_o - Output inphase symbol
Q_o - Output quadrature symbol
Mag - Estimated I/Q magnitude
Hit - Magnitude hit signal (Magnitude excess threshold)

Figure 5.2.1 The block diagram of the despreader.

5.2.1 The Despreading Unit

The despreading unit was originally designed by Titus Mathews from the University of Calgary. It is designed for a 2 samples/symbol DS/SS baseband sampling system with a tested data speed as high as 27MHz [32]. It is modified to fit the 2 samples/symbol DS/SS IF-sampling system; however, its basic concepts remain the same. The primary function of the despreading unit is to reconstruct I/Q symbols (I_n and

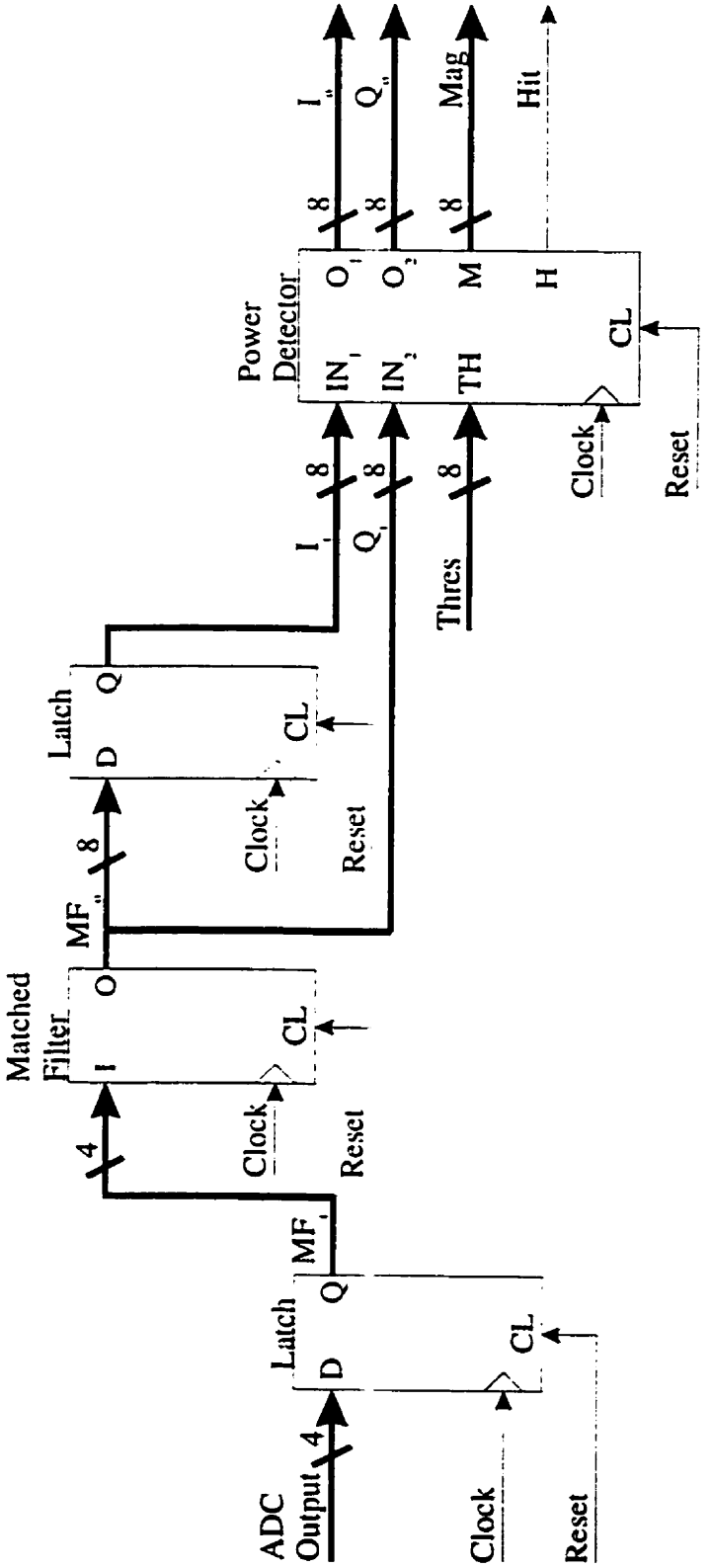


Figure 5.2.1.1 The block diagram of the despreading unit.

Q_n) from the ADC output samples (PN despreading process). The secondary function of the despreading unit is to estimate the magnitude of a pair of I/Q symbols (*Mag*). In addition, it is also required to compare the output magnitude to a magnitude threshold (*Thres*) in order to assert the hit signal (*Hit*) when the output magnitude is greater than the magnitude threshold. Therefore the hit signal simply indicates that the signal strength is strong enough at that time instance. Consequently, each hit signal represents a valid system synchronization point. The block diagram of the despreading unit is shown in Figure 5.2.1.1.

As shown in Figure 5.2.1.1, the despreading unit consists of a matched filter, a power detector and two registers (Latches). The purpose of a register is (1) to temporarily store the output of the previous stage, and (2) to synchronize the output of the previous stage to the master system clock signal. The matched filter is used to reconstruct the I/Q symbols from the ADC output samples. The power detector calculates the I/Q symbol magnitude and generates the hit signal. The details about the matched filter and the power detector are given in Section 5.2.1.1 and 5.2.1.2.

The timing diagram of the despreading unit is shown in Figure 5.2.1.2 and the operation step of the despreading unit are listed as follows:

1. ADC output samples (*ADC Output*) are latched by a 4-bit latch.
2. The matched filter takes the output of the 4-bit latch and forms an output symbol (8-bit) in 4 clock cycles.

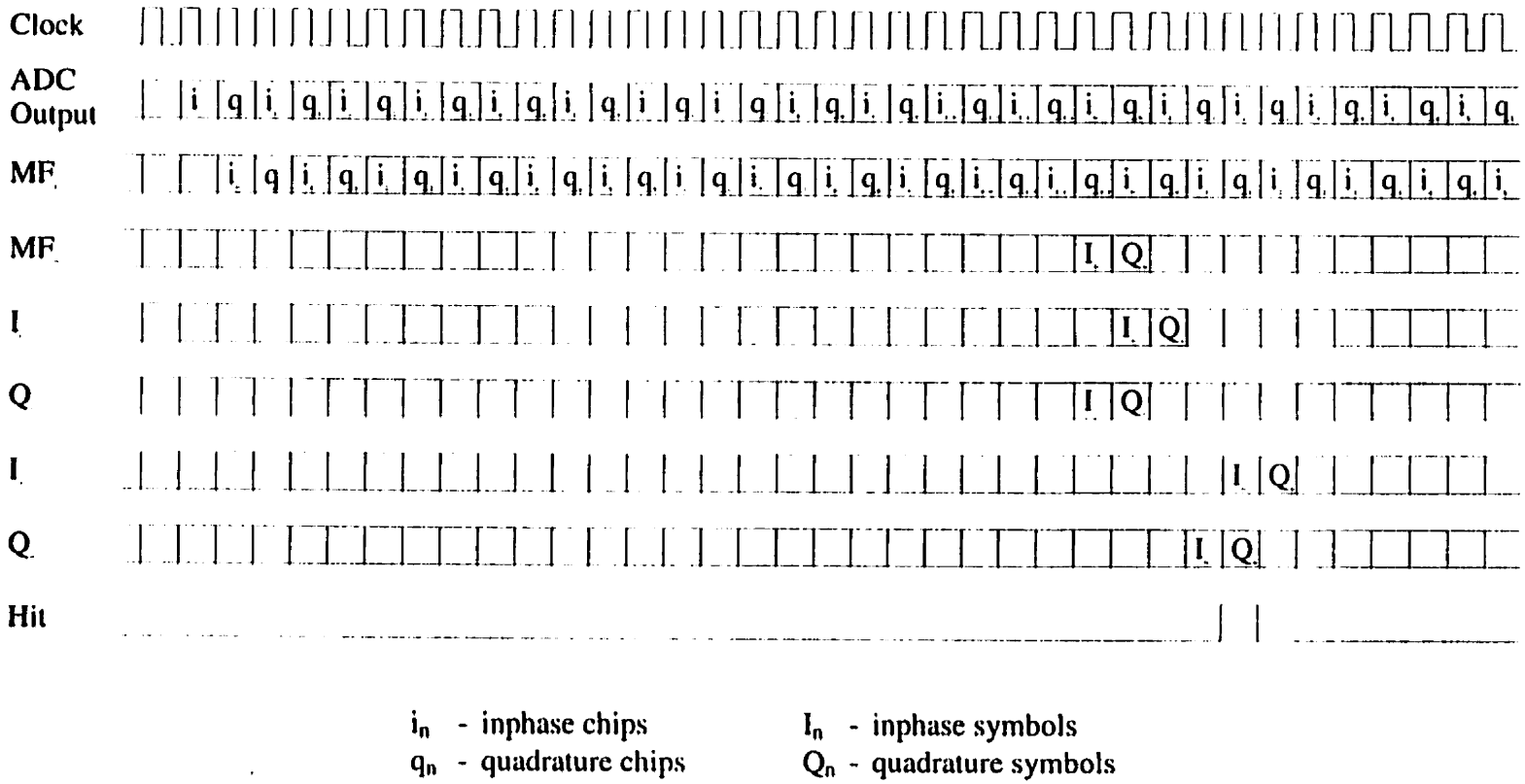


Figure 5.2.1.2 Timing diagram of the despreading unit.

3. The matched filter output goes into IN_2 of the power detector and a delayed matched filter output goes into IN_1 of the power detector. This way the I and Q symbols always line up at the input of the power detector at the same instance in time.
4. The power detector takes the input symbols (from IN_1 and IN_2) and an input threshold (*Thres*) to generate the output I/Q symbols (I_n and Q_n), the I/Q symbol magnitude (*Mag*), and the hit signal (*Hit*). Note that the hit signal is asserted only when the I/Q symbol magnitude is greater than the input threshold.

5.2.1.1 The Matched Filter

The detailed structure of the matched filter is shown in Figure 5.2.1.1.1. In this design, the matched filter consists of a delay (or register) chain (Row 1), a ladder structure of adder/subtracter modules (Row 2, 4, 6, 8), and a pipeline stage register after each adder/subtracter module (Row 3, 5, 7, 9). The main function of the matched filter is to calculate the sum of product between the input samples ($d0, d1, ..., d10$) and their coefficients ($a0, a1, ..., a10$). Mathematically, this sum of products can be expressed as:

$$[d0, d1, ..., d10] \cdot [a0, a1, ...a10] = a0 \cdot d0 + a1 \cdot d1 + ... + a10 \cdot d10$$
$$= Output\ Symbol \qquad (5.2.1.1.1)$$

where

$d0, d1, ..., d10$ are 4-bit two's complement words,

$d0$ is the latest arrived sample. and

$a0, a1, ..., a10$ is the filter coefficients.

The operation described by Equation 5.2.1.1.1 performs the despreading process described by Equation 4.2.1.6 and 4.2.1.7. Therefore the filter coefficients should be set as the reversed Barker sequence stated in Equation 2.2.3.2. However the filter input data stream, MF_i, is (Equation 3.3.13):

$$\begin{aligned} MF_i = \{&X_0, -Y_0, -X_1, Y_1, ... X_{10}, -Y_{10}, && \text{(Symbol 0} \rightarrow I_0 \text{ and } Q_0\text{)} \\ &-X_{11}, Y_{11}, X_{12}, -Y_{12}, ..., -X_{21}, Y_{21}, && \text{(Symbol 1} \rightarrow I_1 \text{ and } Q_1\text{)} \\ &X_{22}, -Y_{22}, -X_{23}, Y_{23}, ..., X_{32}, -Y_{32}, ...\} && (5.2.1.1.2) \end{aligned}$$

where

X_m is the I samples (chip) at time index m

Y_m is the Q samples (chip) at time index m

Thus the sign correction must be applied to MF_i before the despreading process. . The sign correction is performed by setting the filter coefficients as follow:

$$\begin{array}{cccccccccccc} -1 & -1 & -1 & +1 & +1 & +1 & -1 & +1 & +1 & -1 & +1 & \rightarrow \text{ Reversed Barker sequence} \\ \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 & +1 & \rightarrow \text{ Sign correction sequence} \\ \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \\ -1 & +1 & -1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & \rightarrow \{a0, a1, a2, a3, a4 ..... a10\} \end{array}$$

(5.2.1.1.3)

Using the filter coefficients in Equation 5.2.1.1.3, the I symbols at the data synchronization points are obtained as follow:

$$d0 = (-1)^n \cdot X_{n+10}, d1 = (-1)^n \cdot X_{n+9}, \ldots d10 = (-1)^n \cdot X_n$$

$$\begin{aligned} I_n &= [d0, d1, \ldots, d10] \cdot [a0, a1, \ldots, a10] \\ &= (-1)^n \cdot [X_{n+10}, X_{n+9}, \ldots, X_n] \cdot \underbrace{[-1,-1,-1,1,1,1,-1,1,1,-1,1]}_{\text{Reversed Barker sequence}} \\ &= \{I_0, -I_1, I_2, -I_3, \ldots\ldots\} \end{aligned} \qquad (5.2.1.1.4)$$

Similarly, the Q symbols are obtained as follow:

$$d0 = (-1)^{n+1} \cdot Y_{n+10}, d1 = (-1)^{n+1} \cdot Y_{n+9}, \ldots d10 = (-1)^{n+1} \cdot Y_n$$

$$\begin{aligned} Q_n &= [d0, d1, \ldots, d10] \cdot [a0, a1, \ldots, a10] \\ &= (-1)^{n+1} \cdot [Y_{n+10}, Y_{n+9}, \ldots, Y_n] \cdot \underbrace{[-1,-1,-1,1,1,1,-1,1,1,-1,1]}_{\text{Reversed Barker sequence}} \\ &= \{-Q_0, Q_1, -Q_2, Q_3, \ldots\ldots\} \end{aligned} \qquad (5.2.1.1.5)$$

Since MF_i consists of two separate interlaced data streams – one corresponding to the I data stream and the other corresponding to the Q data stream to calculate the output symbol, the matched filter cannot mix up these two data streams. This separation of the two streams is achieved using the double delays between each tap in the delay chain of the filter. The pipeline stage registers are used to reduce the critical path throughout the filter, thereby permitting a narrower clock pulse width. As a result, the matched filter can operate at a higher speed. Note that the matched filter needs four clock cycles to produce an output symbol since there are four pipeline stages in it.

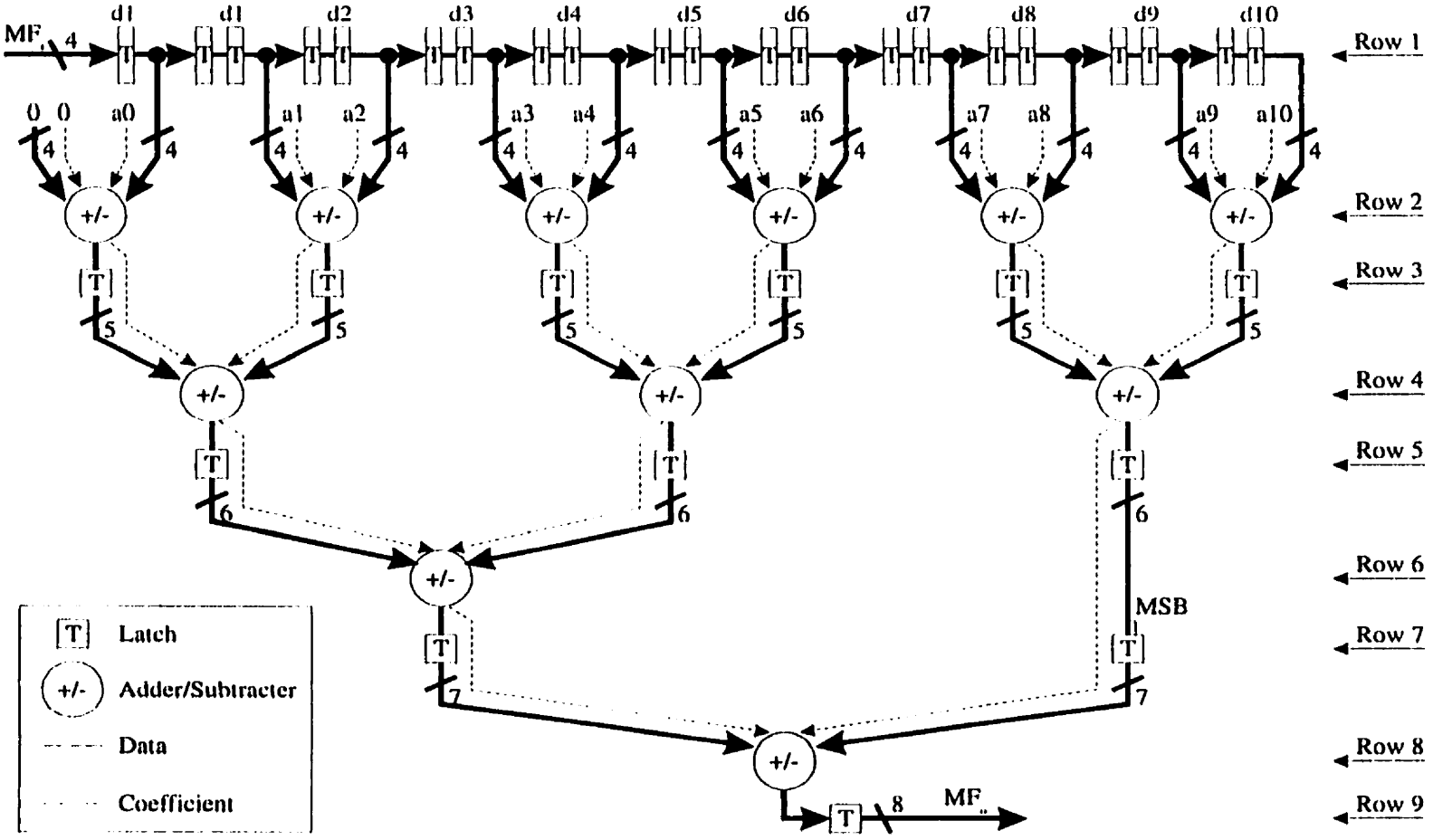


Figure 5.2.1.1.1 The block diagram of the matched filter.

Since the filter coefficients are binary, the multiplication and addition in Equation 5.2.1.1.1 are both performed in the adder/subtracter module. Figure 5.2.1.1.2 shows both the detailed structural diagram of the adder/subtracter module and the register used in the matched filter.

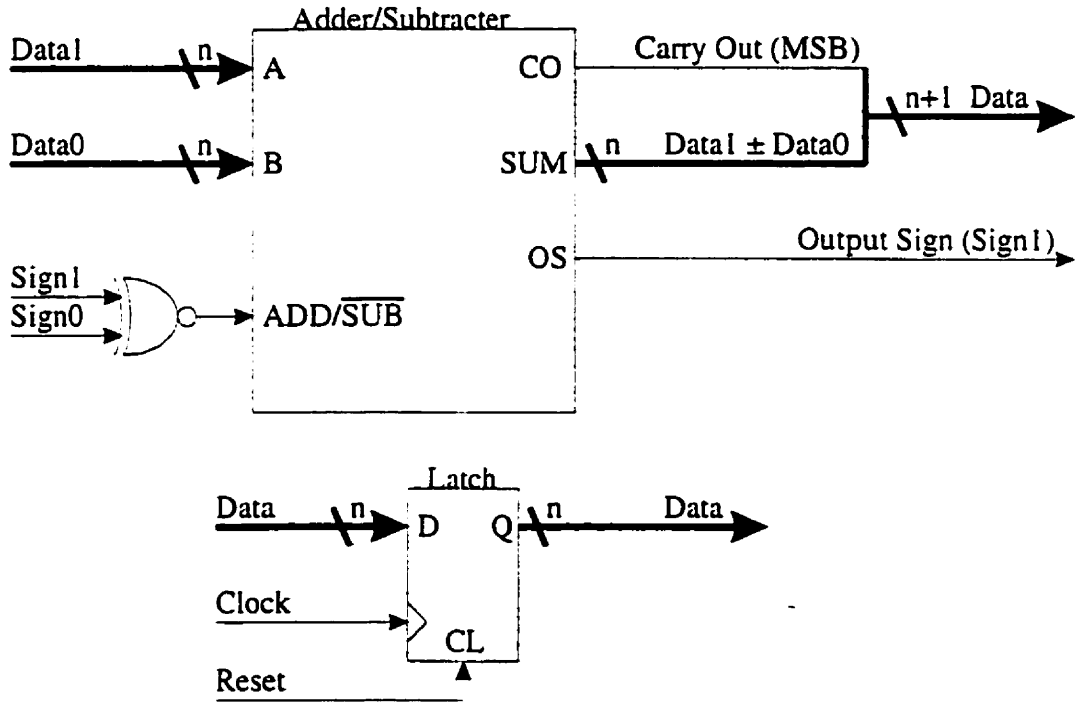


Figure 5.2.1.1.2 The structure of the adder/subtracter module and the register.

The input of the adder/subtracter module consists of two data samples (*Data*1 and *Data*0 in two's complement format) and their signs (*Sign*1 and *Sign*0 in binary format, 0 is + and 1 is -). Rather than applying two's complement conversion to both input data according to their sign, the adder/subtracter module determines the output sign (*Output Sign*) and the type of operation (addition/subtraction) from the input signs using an *XNOR* gate. To avoid overflow in the output number (*Data*), the wordlength increases by one bit from input to output. As a result, the wordlength grows from 4-bit in the uppermost level (Row 1) of the ladder structure to 8-bit in the final level (Row 9) in order to preserve accuracy. The coefficients of the input data is applied onto the input data simply by making these coefficients the input signs of each adder/subtracter module at the highest level of the ladder structure. When the operation of the adder/subtracter module finishes, both the output data (*Data*) and the output sign (*Output Sign*) propagates down to the next stage of the ladder structure. The operation of the adder/subtracter module is demonstrated more clearly in the following example:

Addition: $-2-(-5)=3$

$Data1 = -5$, $Data0 = 2$, $Sign1 = 1$, *and* $Sign0 = 1$

Output Sign = $Sign1 = 1$ = Negative

The output of *XNOR* = 1 *XNOR* 1 = 1 = Addition

$Data = (-5)+2 = -3$

The resulting number = $(Output\ Sign)(Data) = -(-3) = 3$

Subtraction: $-4+3=-1$

$Data1=3,\ Data0=4,\ Sign1=0,\ and\ Sign0=1$

$Output\ Sign = Sign1 = 0 = \text{Positive}$

The output of $XNOR = 0\ XNOR\ 1 = 0 = \text{Subtraction}$

$Data = 3-4=-1$

The resulting number $= (Output\ Sign)(Data) = +(-1) = -1$

5.2.1.2 The Power Detector

The detailed structural diagram of the power detector is shown in Figure 5.2.1.2.1. The architecture of the power detector includes 11 pipeline registers, an adder, and a comparator. The primary function of the power detector is to estimate the I/Q symbol magnitude (*Mag*) and to generate the hit signal (Hit). The secondary function of the power detector is to synchronize the input I/Q symbols (I_i and Q_i) to their estimated magnitude at its output. For simplicity, the magnitude of I/Q symbols is calculated as the sum of the absolute value of the I and Q symbols, that is:

$$Mag = |I_i| + |Q_i| \qquad (5.2.1.2.1)$$

The hit signal is asserted if the I/Q symbol magnitude is greater than the input threshold (*Thres*). The synchronization of the I/Q symbols and their estimated magnitude is achieved by the six delay registers on the I_i to I_{i6} and Q_i to Q_{i6} paths. Note that there are totally three pipeline stages inside the power detector; therefore, it takes three clock cycles for the power detector to generate its outputs.

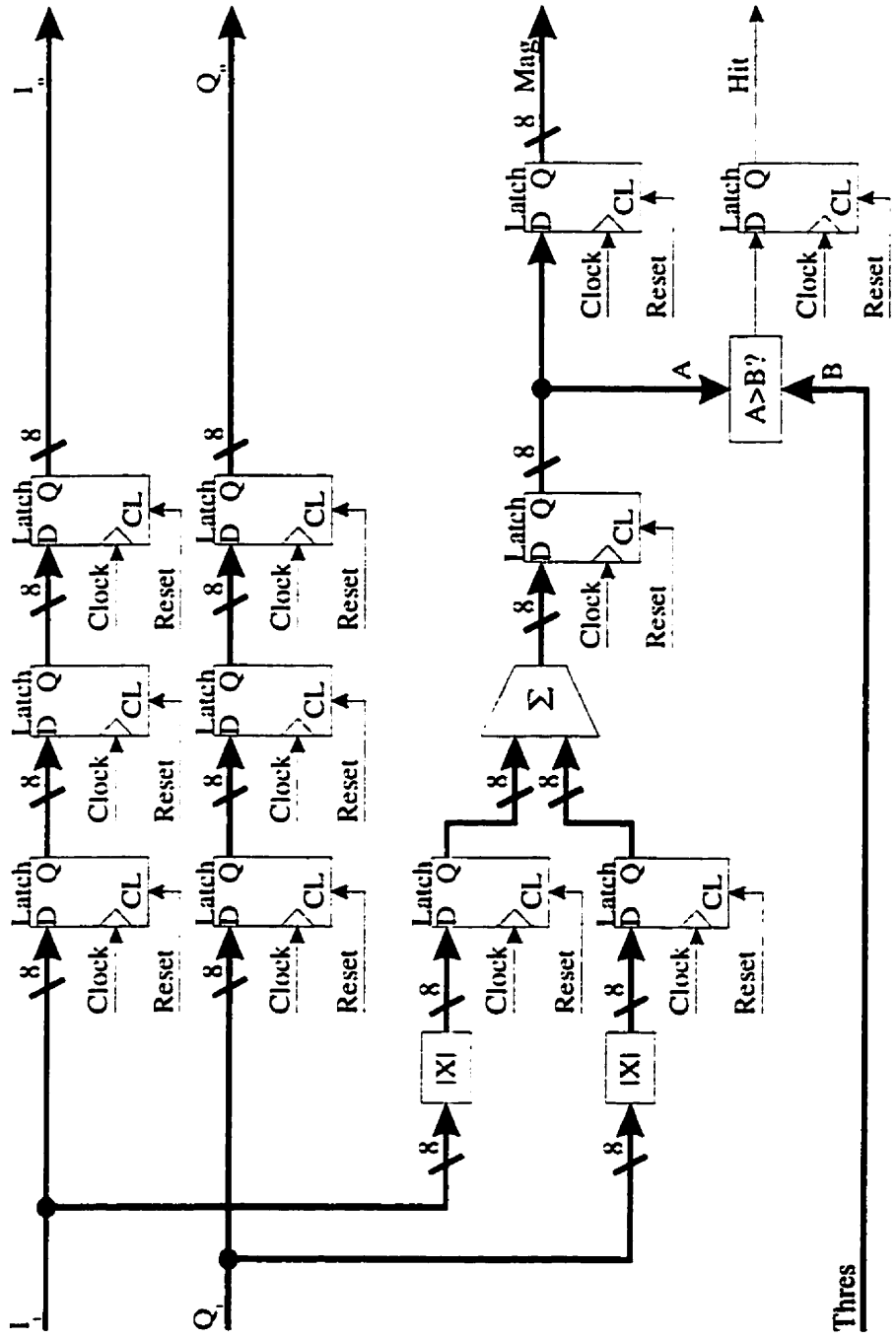


Figure 5.2.1.2.1 The block of the power detector.

magnitude (*Mag*) and to generate the hit signal (Hit). The secondary function of the power detector is to synchronize the input I/Q symbols (I_i and Q_i) to their estimated magnitude at its output. For simplicity, the magnitude of I/Q symbols is calculated as the sum of the absolute value of the I and Q symbols, that is:

$$Mag = |I_i| + |Q_i| \qquad (5.2.1.2.1)$$

The hit signal is asserted if the I/Q symbol magnitude is greater than the input threshold (*Thres*). The synchronization of the I/Q symbols and their estimated magnitude is achieved by the six delay registers on the I_i to I_n and Q_i to Q_n paths. Note that there are totally three pipeline stages inside the power detector; therefore, it takes three clock cycles for the power detector to generate its outputs.

5.2.2 The Synchronization Controller

The detailed architecture of the synchronization controller is shown in Figure 5.2.2.1. The synchronization controller circuit consists of a counter controller, a timer controller, 2 registers ($R1$ and $R2$), 3 multiplexers ($M1$ and $M2$), and a 3-input *AND* gate. The function of the synchronization controller is (1) to determine the presence of information signal at the output of the ADC, (2) to find the synchronization point of the received data, and (3) to handle data termination. To do that, the synchronization controller takes the outputs of the despreading unit (I_n, Q_n, *Mag* and *Hit*) and the input threshold (*Threshold*) and generates synchronized I/Q symbols (I_{sync} and Q_{sync}) and an output enable signal (*InEN*).

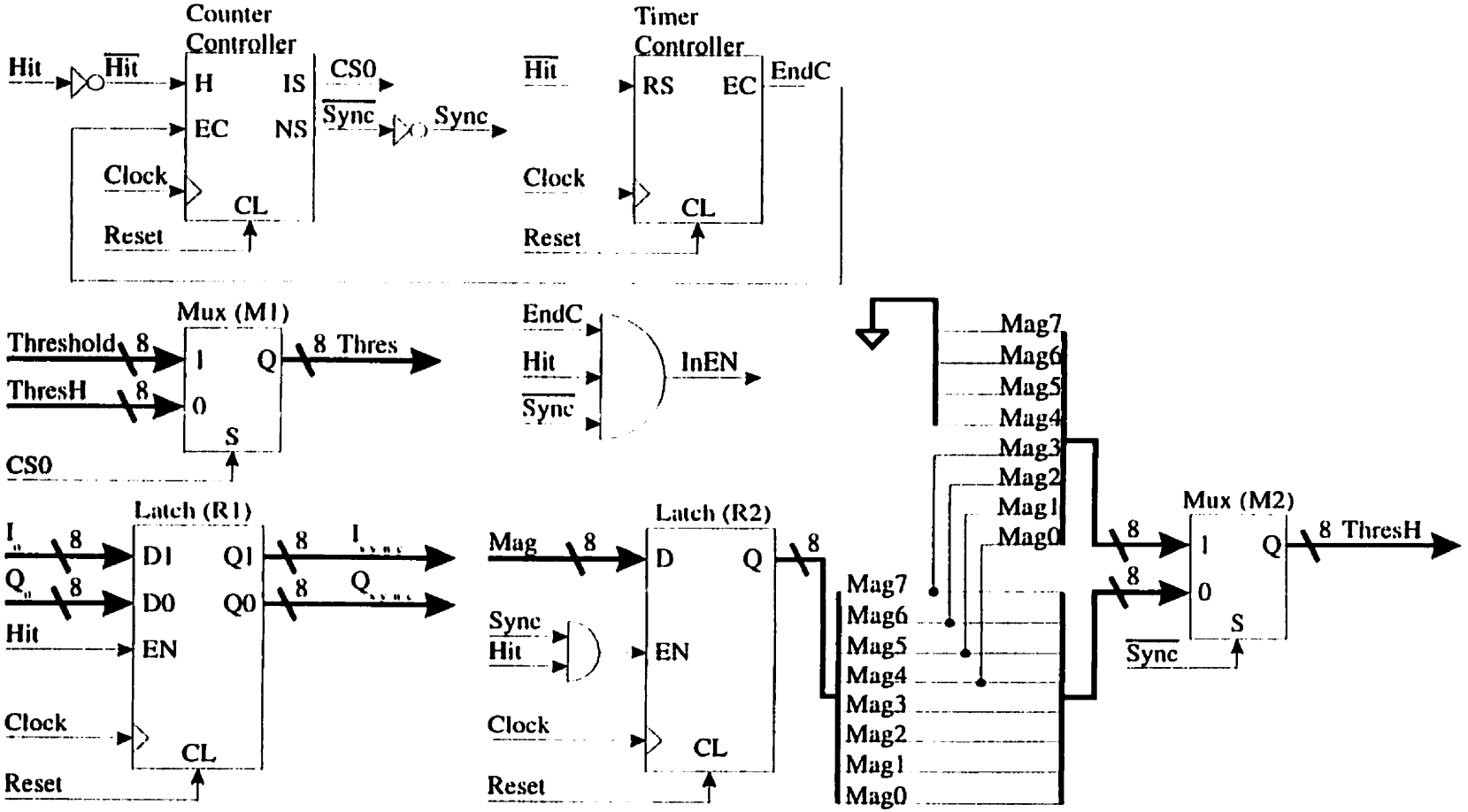


Figure 5.2.2.1 The block diagram of the synchronization controller.

To understand how the synchronization controller works, let us look at the following diagram:

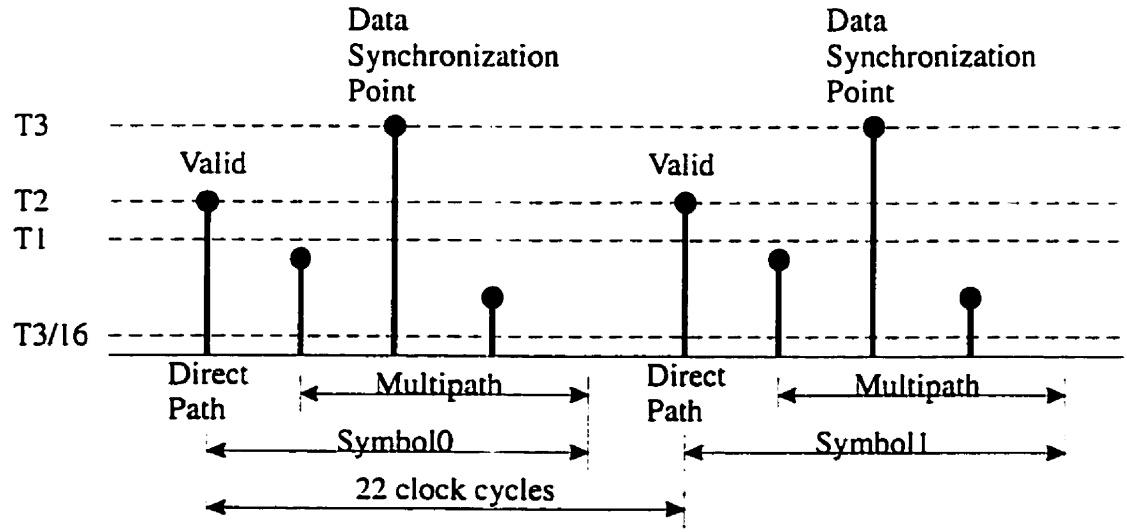


Figure 5.2.2.2 A sample output magnitude of the despreading unit.

Assuming that Figure 5.2.2.2 represents the output magnitude of the despreading unit, then the synchronization controller needs to determine the presence of the information signal from its first peak (*Direct Path* of *Symbol*0). Next, the synchronization controller has to decide that the highest peak (3rd peak from left) during the first symbol is the data synchronization point. Finally, the synchronization controller also needs to determine the termination of the data stream. To achieve these tasks, the synchronization controller is divided into two states: data acquisition (*Sync* asserted) and decoding ($\overline{Sync}$ asserted). The first two tasks are performed in the data acquisition state and the third task is performed in the decoding state. The sequence of events during the two states are listed as follows:

Data acquisition state:

1. *Sync* asserts and stays high to indicate that the synchronization controller is in data acquisition state. *CS0* asserts and stays high so that the input threshold (*Threshold*) is used as the magnitude threshold (*Thres* = *T1*) for the despreading unit.
2. *Hit* asserts for one clock cycle which indicates the presence of the information signal. The output I/Q symbols (I_n and Q_n) are loaded into *R1* in the next clock cycle. Since *Hit* asserts while *Sync* is high, the output magnitude of the despreading unit (*Mag*) is loaded into *R2* in the next clock cycle. *CS0* drops and stays low so that *ThresH* becomes the new magnitude threshold (*Thres* = *ThresH* = *Mag* = *T2*) in the next clock cycle. $\overline{Hit}$ drops for one clock cycle to reset the timer controller in the next clock cycle so that this hit position is marked as the data synchronization point.
3. *Hit* asserts again for one clock cycle and the output I/Q symbols (I_n and Q_n) are loaded into *R1* in the next clock cycle. Since *Hit* asserts while *Sync* is high, the output magnitude of the despreading unit (*Mag*) is loaded into *R2* in the next clock cycle. *CS0* drops and stays low so that *ThresH* becomes the new magnitude threshold (*Thres* = *ThresH* = *Mag* = *T3*) in the next clock cycle. $\overline{Hit}$ drops for one clock cycle to reset the timer controller in the next clock cycle so that this hit position is marked as the data synchronization point.
4. If there are no more hits within the next 22 clock cycles from the first hit, then the last hit position becomes the final data synchronization point. $\overline{Sync}$ asserts and stays high to indicate that the synchronization controller enters the decoding state. Since *CS0* stays low and $\overline{Sync}$ stays high, the new magnitude threshold (*Thres*) is set to *Mag* / 16 = *T3* / 16.

Decoding state:

1. Both *Hit* and *EndC* assert for one cycle which indicates that the next pair of I/Q symbols are ready at the input of $R1$. Since *Hit* asserts for one cycle, I_n and Q_n are loaded into $R1$ in the next cycle. $\overline{Hit}$ drops for one clock cycle to reset the timer controller in the next cycle. *InEN* asserts for one cycle indicates that the I/Q symbols in $R1$ are synchronized and ready for the decoder.
2. If *EndC* goes high and *Hit* stays low, which indicates that the output I/Q symbols at the data synchronization point have a magnitude lower then the threshold ($T3/16$), it is the end of the data stream. The counter controller resets itself in the next cycle which puts the synchronization controller back to the data acquisition state.

A detailed timing diagram for the example shown in Figure 5.2.2.2 is shown in Figure 5.2.2.3.

The heart of the synchronization controller is the counter and timer controllers. Both the counter and timer controllers are state machines and their state diagrams are included in Appendix B (Figure B.1 and B.2). The counter controller is used to counter the number of cycles passed since the first *Hit* assertion (synchronization period = 22 clock cycles) during the data acquisition state. It will turn the synchronization controller into the decoding state when the synchronization period ends by setting *Sync* low. In the decoding state, the counter controller determines the termination of the data stream by monitoring both *EndC* and *Hit*. It will turn the synchronization controller back to the data acquisition state by resetting itself which in turn sets *Sync* high. The timer

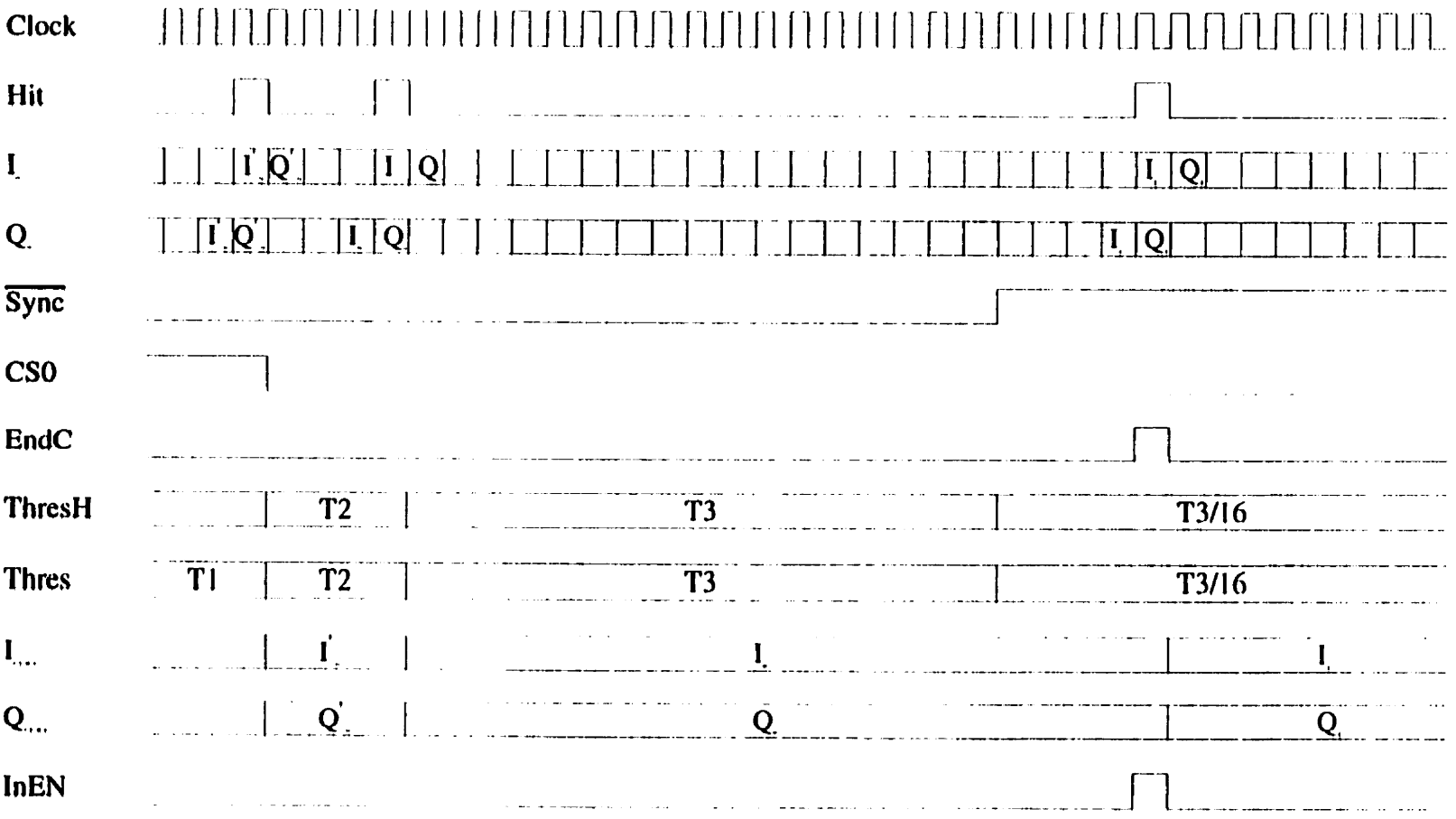


Figure 5.2.2.3 The timing diagram of the synchronization controller.

controller remembers the data synchronization points. It sets *EndC* high for one cycle at each data synchronization point. Note that the timer controller is reset whenever *Hit* asserts. The architectures of the counter and timer controllers (Figure B.3, B.4, and B.5) are also attached in Appendix B.

5.3 The π/4-shifted-DQPSK Decoder

The basic function of the decoder is to extract the information bits encoded inside the phase difference between two consecutive output complex symbols from the despreader. Mathematically, the operation of the decoder is given as:

$$(I_n + jQ_n)\cdot(I_{n-1} + jQ_{n-1})^* = (I_n + jQ_n)\cdot(I_{n-1} - jQ_{n-1})$$
$$= (I_n I_{n-1} + Q_n Q_{n-1}) + j(I_{n-1}Q_n - I_n Q_{n-1}) \quad (5.3.1)$$

where

I_n = current I symbol.

Q_n = current Q symbol.

I_{n-1} = last I symbol.

Q_{n-1} = last Q symbol.

Therefore the most significant bit is equal to the sign bit of $(I_n I_{n-1} + Q_n Q_{n-1})$ and the least significant bit is equal to the sign bit of $(I_{n-1}Q_n - I_n Q_{n-1})$ as shown in Equation 2.1.2.4 and 2.1.2.5. However the despreader output complex symbols, obtained from Equation 5.2.1.1.4 and 5.2.1.1.5, are given as:

$$\{(I_0 - jQ_0),(-I_1 + jQ_1),(I_2 - jQ_2),.....\} \tag{5.3.2}$$

Therefore the sign of the I and Q symbols are different as compared to Equation 5.3.1. At symbol index n, a pair of consecutive complex symbols are either $[(I_n - jQ_n),(-I_{n-1} + jQ_{n-1})]$ or $[(-I_n + jQ_n),(I_{n-1} - jQ_{n-1})]$. If the pair of complex symbols is $[(I_n - jQ_n),(-I_{n-1} + jQ_{n-1})]$, then the decoder gives:

$$\begin{aligned}(I_n - jQ_n)\cdot(-I_{n-1} + jQ_{n-1})^* &= (I_n - jQ_n)\cdot(-I_{n-1} - jQ_{n-1}) \\ &= -(I_nI_{n-1} + Q_nQ_{n-1}) + j(I_{n-1}Q_n - I_nQ_{n-1})\end{aligned} \tag{5.3.3}$$

If the pair of complex symbols is $[(-I_n + jQ_n),(I_{n-1} - jQ_{n-1})]$, then the decoder gives:

$$\begin{aligned}(-I_n + jQ_n)\cdot(I_{n-1} - jQ_{n-1})^* &= (-I_n + jQ_n)\cdot(I_{n-1} + jQ_{n-1}) \\ &= -(I_nI_{n-1} + Q_nQ_{n-1}) + j(I_{n-1}Q_n - I_nQ_{n-1})\end{aligned} \tag{5.3.4}$$

Comparing Equation 5.3.3 and 5.3.4, both equations give the same result which means that the decoding logic for the two possible input complex symbols is the same. However Equation 5.3.3 and 5.3.4 have opposite real part sign compared to Equation 5.3.1. Therefore the decoder must perform a negation on the output most significant bit after the $\pi/4$-shifted DQPSK decoding.

The $\pi/4$-shifted DQPSK decoder can be implemented in two ways: (1) using multiplier and adder/subtracter modules, or (2) using lookup table (ROM). To achieve a good accuracy with a lookup table, the size of the table will be large. Thus, the first solution is preferable. However, a multiplier module is large in size and takes too much space on a single chip. So it is necessary to minimize the size of the multiplier module.

The size of the multiplier module is usually determined by the number of bits of the multiplicand and multiplier as shown in the following example:

Example: $C = A \times B$ where

$$\left.\begin{aligned} A &= x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \ldots + x_0 \quad (n \text{ bits multiplicand}) \\ B &= y_{n-1}2^{n-1} + y_{n-2}2^{n-2} + \ldots + y_0 \quad (n \text{ bits multiplier}) \\ C &= z_{2n-1}2^{2n-1} + z_{2n-2}2^{2n-2} + \ldots + z_0 \quad (2n \text{ bits product}) \end{aligned}\right\} \{x, y, z \in 1,0\}$$

$$\therefore C = A \times B = \underbrace{Ay_{n-1}2^{n-1} + Ay_{n-2}2^{n-2} + \ldots + Ay_0}$$

n-term requires $n-1$ adder/subtracter modules

or (expanded form)

$$\begin{array}{cccccccccc}
 & 2^{2n-1} & 2^{2n-2} & \cdots & 2^{n-1} & 2^{n-2} & \cdots & 2^0 & & \\
 & & & & x_{n-1} & x_{n-2} & \cdots & x_0 & \rightarrow & A \\
 & & & \times & y_{n-1} & y_{n-2} & \cdots & y_0 & \rightarrow & B \\
\hline
 & & & & x_{n-1}y_0 & x_{n-2}y_0 & \cdots & x_0y_0 & \rightarrow & Ay_0 \\
 & & + & x_{n-1}y_1 & x_{n-2}y_1 & \cdots & x_0y_1 & & \rightarrow & Ay_1 2^1 \\
 & \iddots & \iddots & & \vdots & \iddots & & & \vdots & \vdots \\
+ & x_{n-1}y_{n-1} & x_{n-2}y_{n-1} & \cdots & x_0y_{n-1} & & & & \rightarrow & Ay_{n-1}2^{n-1} \\
\hline
 & z_{2n-1} & z_{2n-2} & \cdots & \cdots & \cdots & \cdots & z_0 & \rightarrow & C
\end{array}$$

In general, multiplying two n-bit numbers requires $n-1$ shift-and-add operations because there are $n-1$ terms in the product's equation. Each shift-and-add operation requires an adder/subtracter module. Consequently, the number of adder/subtracter module required for an $n \times n$ multiplier module is $n-1$ which determines the size of the multiplier module. Therefore the size of the multiplier module is reduced if the number of adder/subtracter modules in the multiplier module is reduced. To reduce the number of adder/subtracter modules in a multiplier module, the Modified Booth's algorithm [33][34] is used. To demonstrate the Modified Booth's algorithm, the previous example is used and the product is given as:

$$\begin{aligned} C = A \times B &= A\left(y_{n-1}2^{n-1} \cdot 2\right) + A\left(-y_{n-1}2^{n-1} + y_{n-2}2^{n-2} + y_{n-3}2^{n-3} \cdot 2\right) + \ldots \\ &\quad + A\left(-y_3 2^3 + y_2 2^2 + y_1 2^1 \cdot 2\right) + A\left(-y_1 2^1 + y_0\right) \\ &= A y_{n-1}2^n + A\left(-2y_{n-1} + y_{n-2} + y_{n-3}\right)2^{n-2} + \ldots \\ &\quad + A\left(-2y_3 + y_2 + y_1\right)2^2 + A\left(-2y_1 + y_0\right) \end{aligned} \quad \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \begin{array}{l} n/2+1\text{-term requires} \\ n/2 \text{ adder/subtracter} \\ \text{modules} \end{array}$$

(5.3.5)

By applying the Modified Booth's algorithm, the number of terms in the product's equation is reduced by almost half. As a result, the number of adder/subtracter modules required for the multiplier module is greatly reduced. Note that the multiplier is used to generate the Booth's coefficient and the Booth's coefficients in Equation 5.3.5 are obtained as:

Booth's Coefficients	*a*	*b*	*c*
$(-2y_1 + y_0)$	y_1	y_0	0
$(-2y_3 + y_2 + y_1)$	y_3	y_2	y_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(-2y_{n-1} + y_{n-2} + y_{n-3})$	y_{n-1}	y_{n-2}	y_{n-3}
(y_{n-1})	0	0	y_{n-1}

and all possible Booth's coefficients are listed in Table 5.3.1.

a	*b*	*c*	Booth's Coefficients	2× (shift left)	*Neg* (negative)	*Z* (zero)
0	0	0	0	0	0	1
0	0	1	1	0	0	0
0	1	0	1	0	0	0
0	1	1	2	1	0	0
1	0	0	-2	1	1	0
1	0	1	-1	0	1	0
1	1	0	-1	0	1	0
1	1	1	0	0	0	1

Table 5.3.1 The Booth's coefficients.

Note that $2\times$, *Neg* and Z, which are generated by the Booth's coefficients, are the actual control signals used in a Modified Booth's multiplier module.

In addition, it is always possible to further reduce the number of adder/subtracter module in a multiplier module if the multiplier module can have more clock cycles to complete the operation. In Section 5.2.2, it shows that the number of clock cycle between two consecutive data synchronization points are 22. Therefore the decoder has 22 clock cycles to complete the calculation described in Equation 5.3.1 (4 multiplications and 2 additions/subtractions). Since all I/Q symbols are 8-bit numbers, one product term in Equation 5.3.1 requires four additions/subtractions using the Modified Booth's algorithm. Therefore the total number of addition/subtraction required to complete the calculation described in Equation 5.3.1 is 18. If a single adder/subtracter module is used in the decoder, it takes 18 clock cycles to complete all operations in Equation 5.3.1 which is affordable in this case. Consequently, the decoder is implemented as a single cycling adder/subtracter module which can perform addition, subtraction and multiplication.

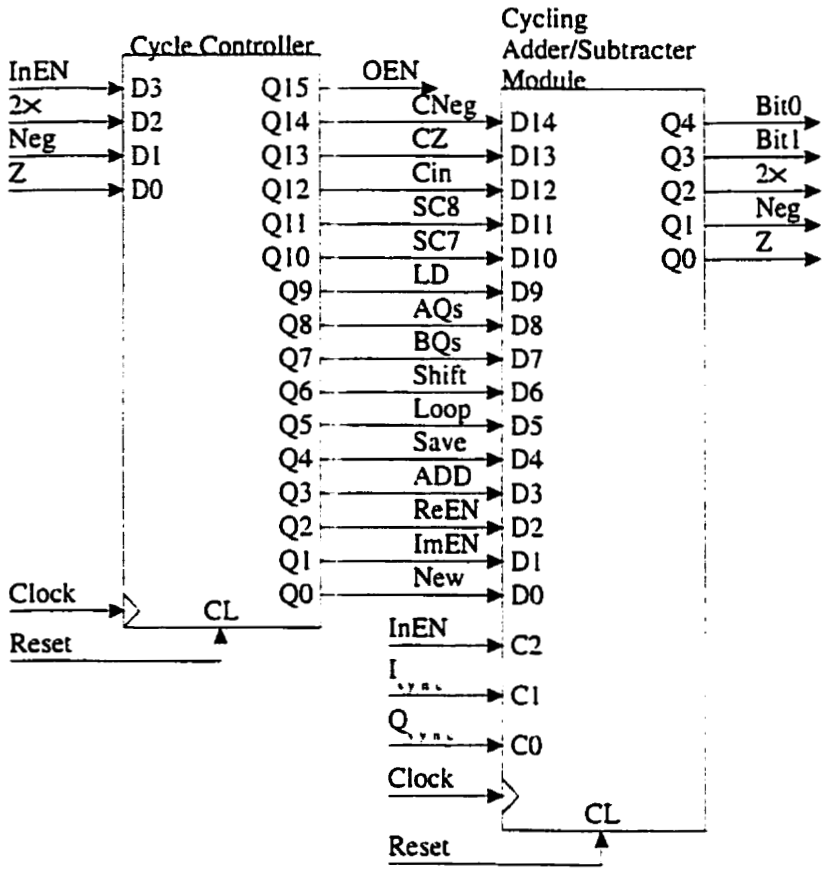


Figure 5.3.1 The structure of the $\pi/4$-shifted DQPSK decoder.

The structure of the decoder, as shown in Figure 5.3.1, contains two function blocks: a cycle controller and a cycling adder/subtracter module.

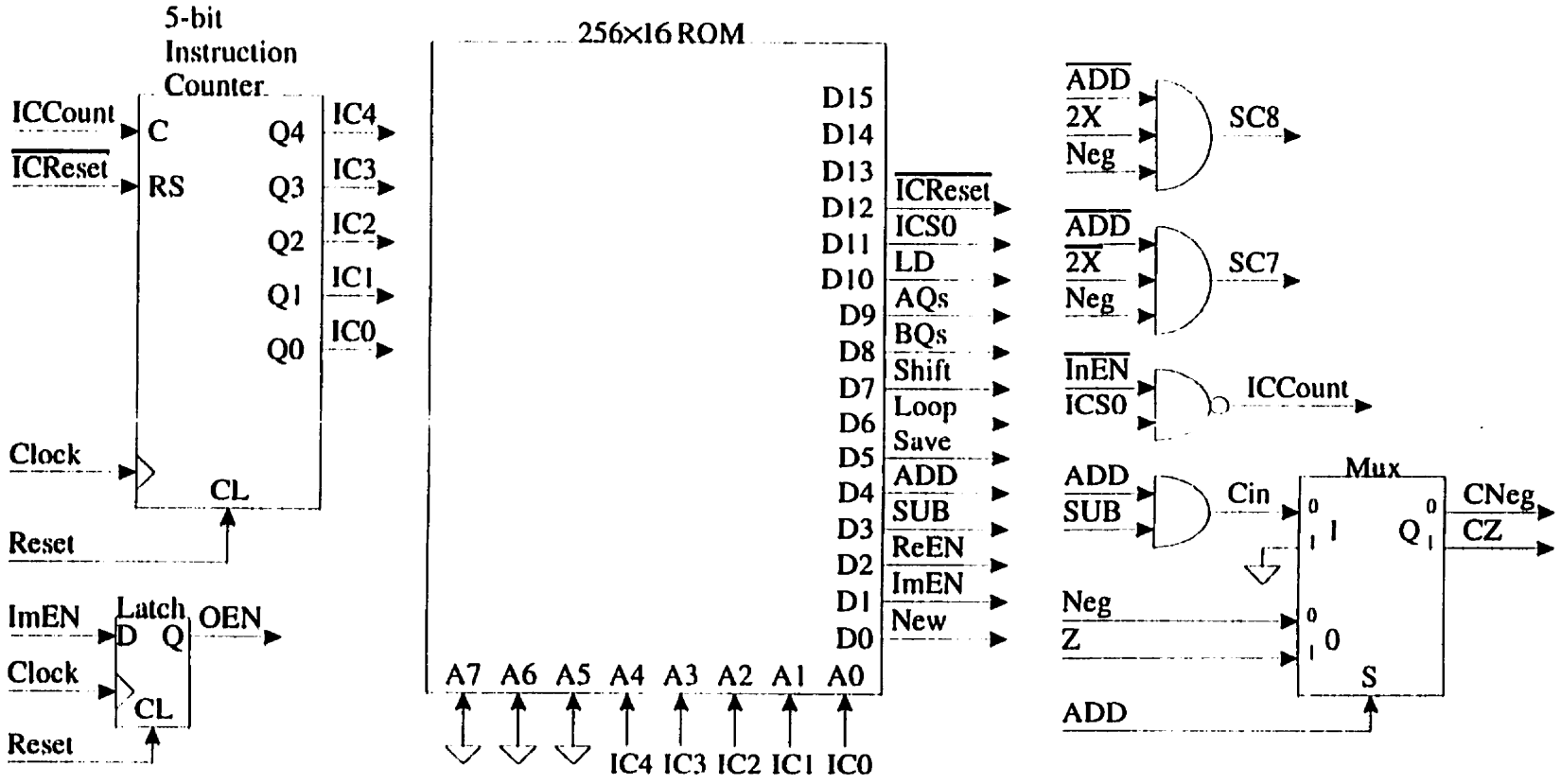


Figure 5.3.2 The block diagram of the cycle controller.

The cycle controller, as shown in Figure 5.3.2, takes the data ready signal (*InEN*) from the despreader and the Booth's control signals (2×, *Neg* and *Z*) from the cycling adder/subtracter module to generate the remaining control signals (*CNeg* , *CZ* , *Cin* , *SC8* , *SC7* , *LD* , *AQs* , *BQs* , *Shift* , *Loop* , *Save* , *ADD* , Re *EN* , Im *EN* , and *New*) for the cycling adder/subtracter module as well as the output bit ready signal, *OEN* . Basically, it is a state machine and is formed by a 256×16 ROM and a 5-bit instruction counter. The ROM is used to store all the necessary control signal patterns. The instruction counter is used to provide the memory address at each clock cycle in order to select the corresponding control signal pattern for both the cycling adder/subtracter module and itself. The state diagram of the cycle controller is available in Appendix B (Figure B.6).

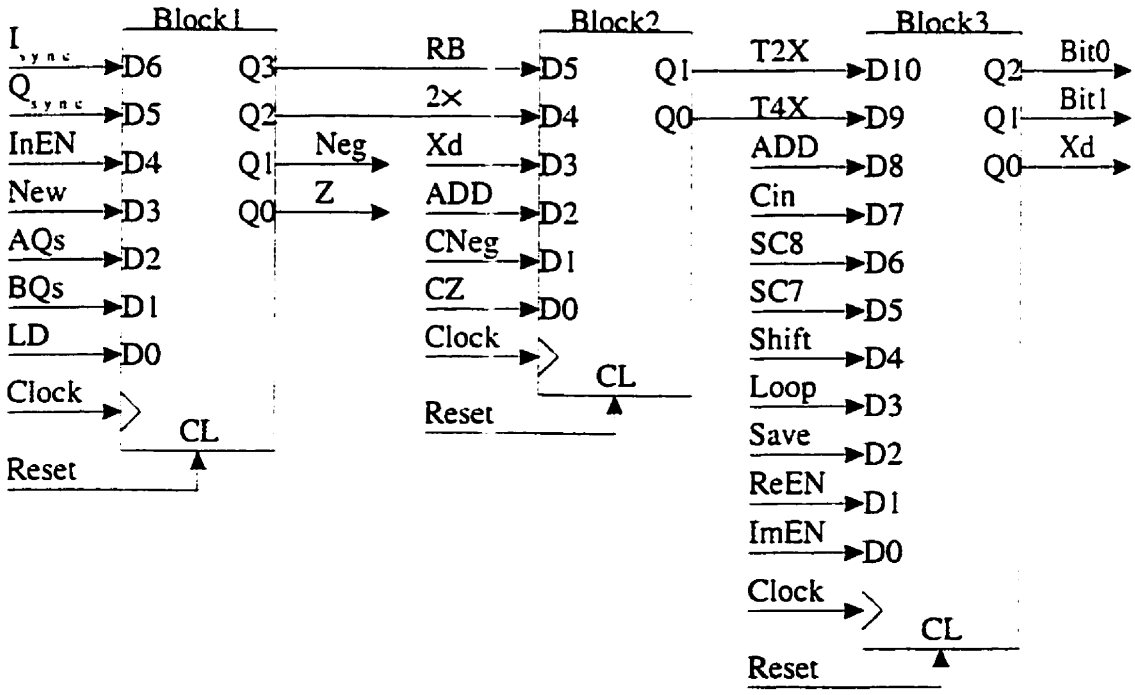


Figure 5.3.3 Block diagram of the cycling adder/subtracter module.

To simplify the discussion of the cycling adder/subtracter module, the structure of the cycling adder/subtracter module is subdivided into three blocks as shown in Figure 5.3.3. The first block, *Block*1, as shown in Figure 5.3.4, takes the despreader outputs (I_{sync}, Q_{sync} and *InEN*) and four control signals (*New*, *AQs*, *BQs* and *LD*) to generate the multiplicand (*RB*) and the Booth's control signals (2×, *Neg* and *Z*).

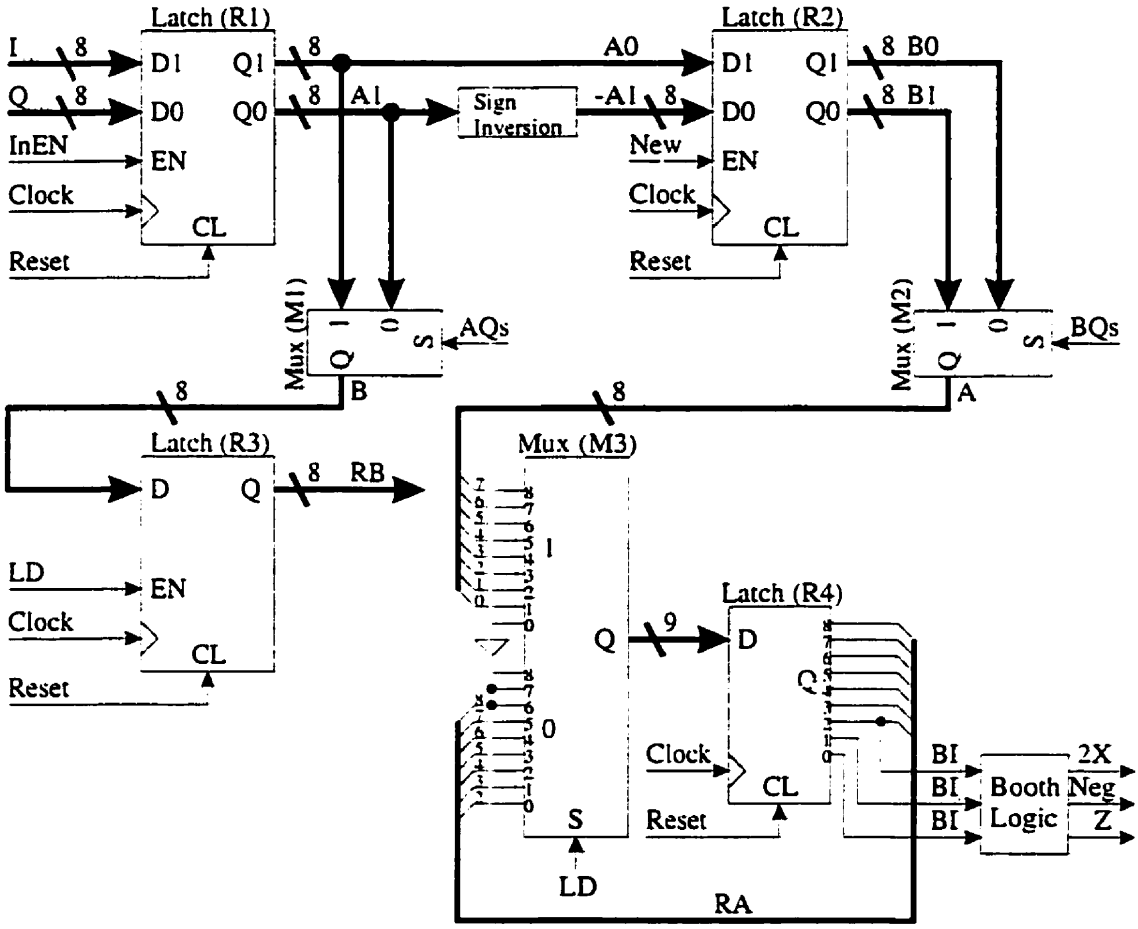


Figure 5.3.4 Detail structure of *Block*1.

The functions of *Block*1 are (1) to store the current complex symbol ($R1$), (2) to store the complex conjugate of the previous symbol (*Sign Inversion* and $R2$), (3) to select a multiplicand ($M1$ and $R3$) and a multiplier ($M2$, $M3$ and $R4$), and (4) to generate the Booth's control signals ($M3$, $R4$ and *Booth Logic*). The *Booth Logic*, according to Table 5.3.1, can be easily implemented using few NAND gates as shown in Figure 5.3.5.

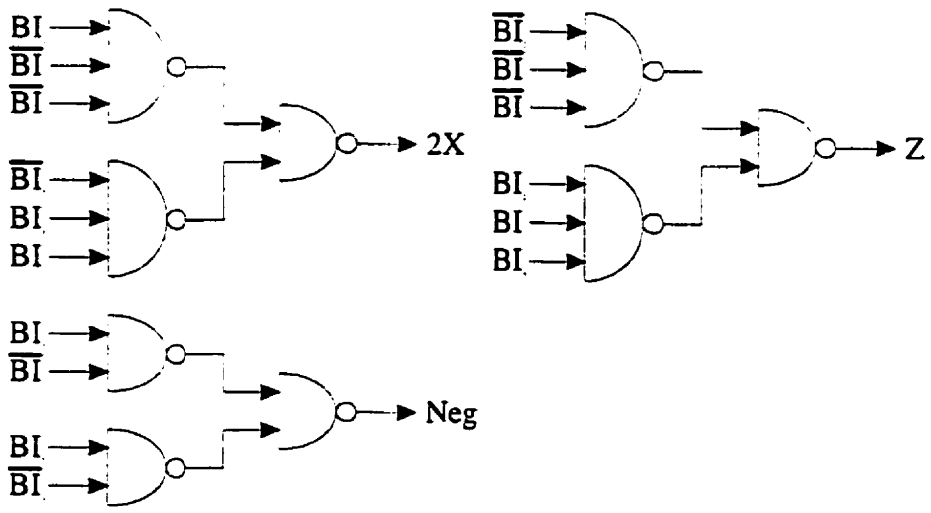


Figure 5.3.5 The structure of *Booth Logic*.

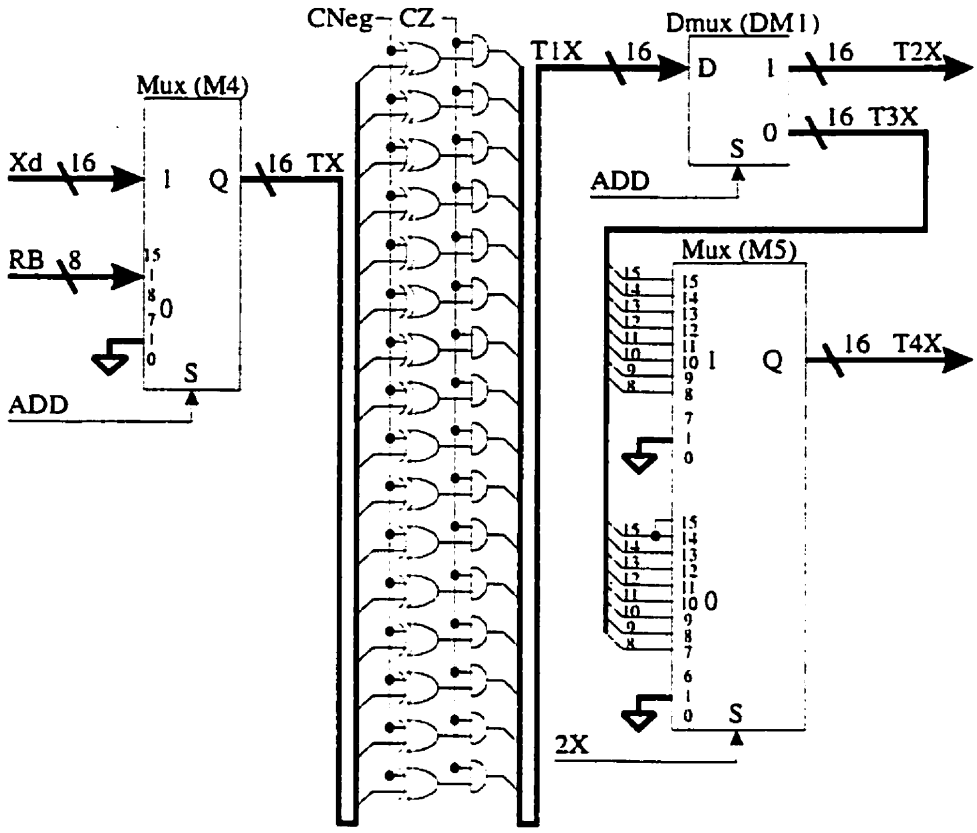


Figure 5.3.6 Detailed structure of *Block2*.

The second block, *Block2*, as shown in Figure 5.3.6, takes the multiplicand (*RB* from *Block1*), a product number (*Xd* from *Block3*) and four control signals (2×, *ADD*, *CNeg* and *CZ*) to generate two output numbers (*T2X* and *T4X*). The functions of *Block2* are (1) to perform proper 2's complement sign inversion on *Xd* to produce *T2X* during the addition/subtraction cycle (*ADD* low), and (2) to apply Booth's operation on *RB* to produce *T4X* during the multiplication cycle (*ADD* high).

The last block, *Block*3, as shown in Figure 5.3.7, takes *Block*2 outputs (*T*2*X* and *T*4*X*) and nine control signals (*ADD*, *Cin*, *SC*8, *SC*7, *Shift*, *Loop*, *Save*, Re*EN* and Im*EN*) to generate the output dibit (*Bit*0 and *Bit*1) and a product number (*Xd*). The functions of *Block*3 are listed as follows:

Multiplication Cycle: (*ADD* low)

1. Performs cycling addition/subtraction on *T*4*X* and *LS*.
2. Completes the 2's complement inversion operation when *SC*8 or *SC*7 asserts.
3. Stores the intermediate result of the multiplication (*R*5).
4. Stores the product (*Yd*) in *R*6 or outputs the product (*Xd*) to *Block*2 when the multiplication completes.

Addition/Subtraction Cycle: (*ADD* high)

1. Performs addition/subtraction on *T*2*X* and *Yd*.
2. Completes the 2's complement inversion operation when *Cin* asserts.
3. Converts the sign of the sum to the output bit and stores the output bit in either *R*7 or *R*8 depending on Re*EN* and Im*EN* signals.

To demonstrate the operation of the decoder, an example is given in terms of the timing diagram assuming that the current and previous symbols are $(I_n - jQ_n)$ and $(-I_{n-1} + jQ_{n-1})$ respectively as shown in Figure 5.3.8.

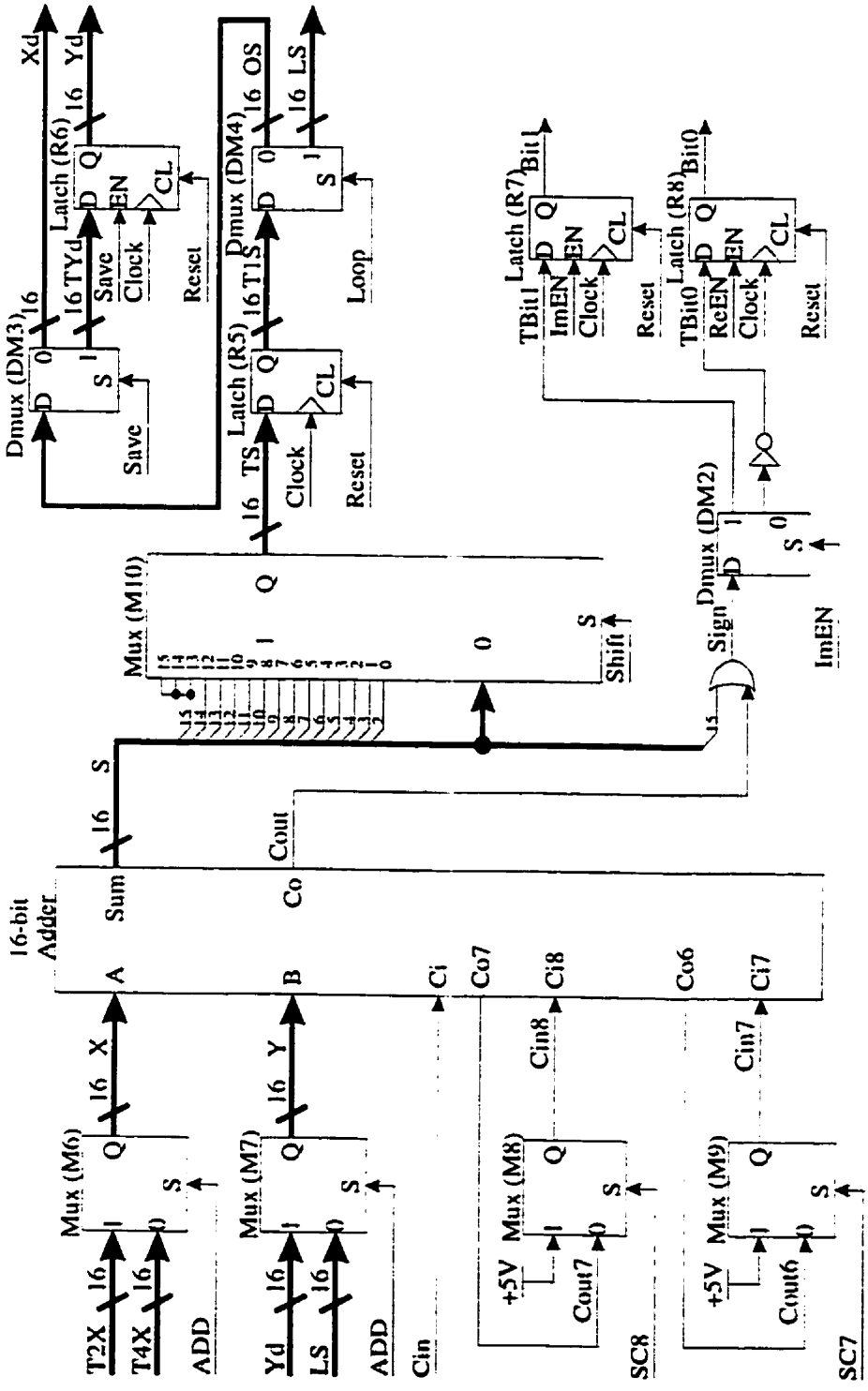
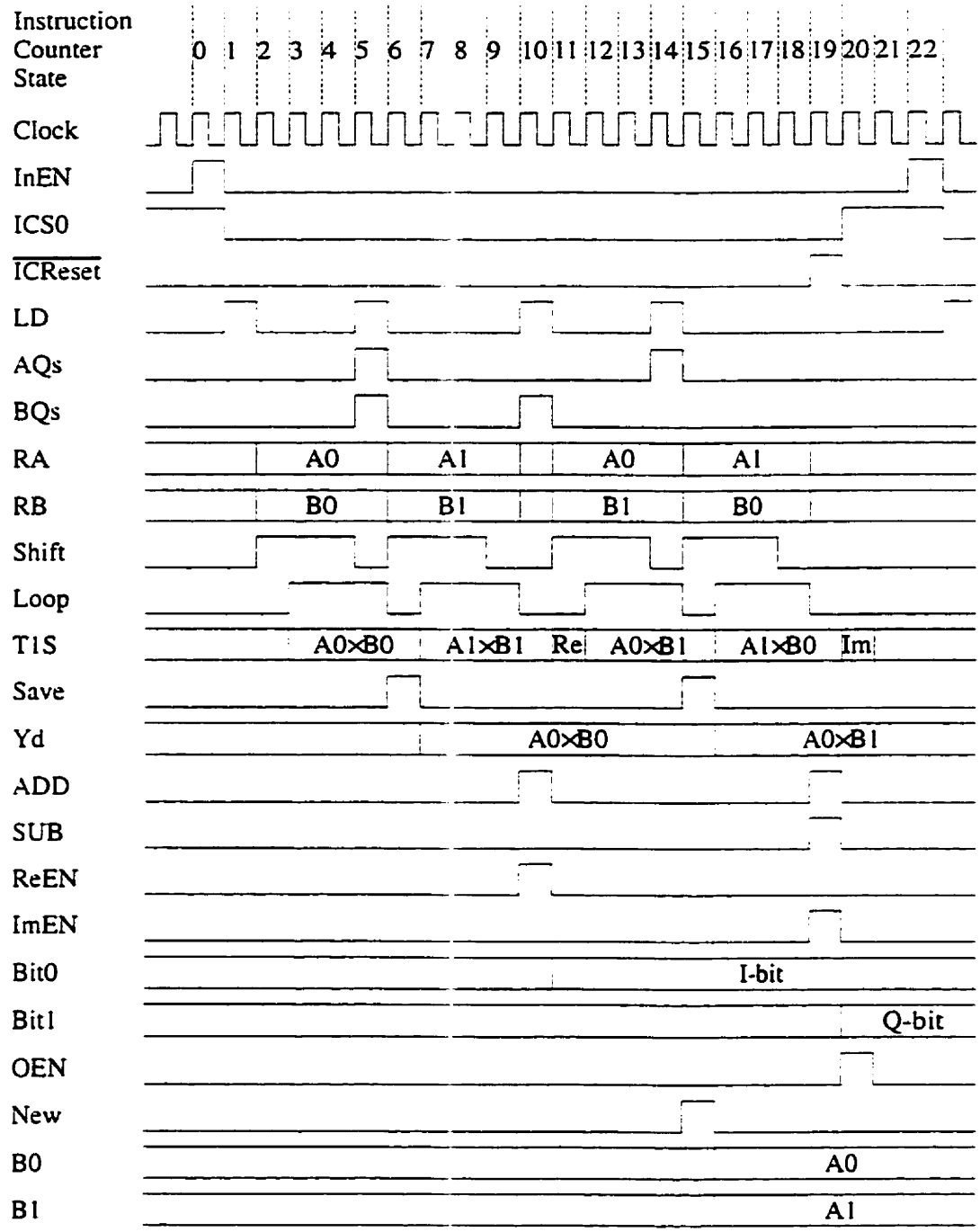


Figure 5.3.7 Detail structure of *Block3*.



$$A0 = I_n \qquad A1 = -Q_n \qquad \mathrm{Re} = (I_n I_{n-1} + Q_n Q_{n-1})$$
$$B0 = -I_{n-1} \qquad B1 = Q_{n-1} \qquad \mathrm{Im} = (I_{n-1} Q_n - I_n Q_{n-1})$$

Figure 5.3.8 The timing diagram of the $\pi/4$-shifted DQPSK decoder.

5.4 Summary

The proposed receiver is implemented with simple discrete components. The structure of the receiver contains two major functional blocks: the despreader and the $\pi/4$-shifted DQPSK decoder. The depreader performs the signal detection and data synchronization. The decoder extracts the information bits from the $\pi/4$-shifted DQPSK symbols. The entire design and testing of the receiver is achieved using Logsim and the results from the simulation verify the functionality of the receiver.

Chapter 6 Conclusion and Future Work

The goal of Chapter 2 is to review the five digital communication concepts, which serve as the foundations in the design of the 2 samples/symbol DS/SS IF-sampling system. The five concepts are $\pi/4$-shifted DQPSK, DS/SS, $F_s/4$-downconversion, Nyquist sampling theorem and the Nyquist pulse for minimum ISI. $\pi/4$-shifted DQPSK is used in the IF-sampling system as the modulation scheme. It protects the system from channel distortions such as amplitude attenuation and constant phase error. The only drawback of using $\pi/4$-shifted DQPSK is that the system will suffer an additional 2.1 dB loss in SNR compared to normal QPSK modulation scheme. DS/SS is used to protect the system from ISI. Generally, it uses much more bandwidth than the data rate for the communication system to reduce the effect of ISI. To use DS/SS, an additional process is added to both the transmitter and receiver. The process added to the transmitter is called the spreading process and the process added to the receiver is called the despreading process. The despreading process is the opposite process of the spreading process. The combination of these two processes can greatly reduce the effect of ISI.

$F_s/4$-downconversion is the common technique used in an IF-sampling system to obtain the I/Q samples directly from a bandpass IF signal. The rule for $F_s/4$-downconversion is that the IF center frequency must be an odd integer multiple of a quarter of the sampling rate. Using $F_s/4$-downconversion, the IF stage of the receiver can be moved to the digital domain and hence, the overall complexity of the receiver is reduced. The drawback of using $F_s/4$-downconversion is that it introduces a timing

misalignment between the I and Q samples which can result in an amplitude imbalance between the I and the Q symbols if the shaping pulse is not carefully chosen. Nyquist sampling theorem and the Nyquist pulse for minimum ISI explain the effect of the signal spectrum after the sampling process. They help choose the most desired sampling rate for the IF-sampling system.

After going through the major concepts of the IF-sampling system in Chapter 2, Chapter 3 begins the discussion on the 2 samples/symbol DS/SS IF-sampling system in [30]. It provides a complete mathematical model of the transmitter, channel and receiver in order to show the signal flow throughout the system. By doing that, it also shows how the theories in Chapter 2 apply to the system in [30]. In addition, it provides a simulation setup and results, and shows that the sampling timing error is the major source of error that can cause severe performance degradation to the system. The results also show that the performance difference between the best case scenario and the worst case scenario is about 7 dB at low BER. Finally, Chapter 3 points out that [30] uses two variable gains in the I and in the Q signal paths to compensate the amplitude imbalance between the I and Q symbols. These two variable gains create implementation difficulty and increase the overall receiver complexity.

Chapter 4 starts the discussion of the proposed 2 samples/symbol DS/SS IF-sampling system. This system is designed based on the system in [30] but uses a new shaping pulse to eliminate the need for the variable gains in the I and in the Q signal paths. The same system analysis in Chapter 3 is performed for the proposed system and the simulation results show that the performance difference between the best case

scenario and the worst case scenario is about 4 dB at low BER. Therefore, the proposed system has not only a lower system complexity but also a better performance compared to the system in [30].

Chapter 5 talks about the actual implementation of the proposed IF receiver. It shows that the whole receiver can be implemented simply using discrete logic components. The receiver contains only two functional blocks: the despreader and the decoder. The function of the despreader is to perform signal detection and data synchronization. It is formed by two state machines, a matched filter and a power detector. The function of the decoder is to extract the information bits from the $\pi/4$-shifted DQPSK symbols. It is formed by a state machine and a cycling adder/subtracter module. To reduce the size of the decoder, the Modified Booth algorithm and cycling addition concept are used. The receiver is designed and tested successfully using a software tool called Logsim which provides a fast and easy way to analyze a digital logic system.

From the research point of view, the analysis for the proposed system is done based on a Gaussian channel. It is definitely worth some future effort to carry out a similar analysis for this system with a multipath channel. In addition, further effort is also needed to complete the implementation and obtain the maximum throughput of the receiver. This involves translating the Logsim netlist to an existing hardware format, programming that particular hardware and testing the receiver in real life.

APPENDIX A:

Implementation Tool

The implementation and simulation of the proposed receiver is done by Logsim [35], which is a text-based digital logic netlist description and event-driven simulation package. Logsim is used because it is very easy to learn and gives enough flexibility to enable the design and testing of complex circuits. Logsim also provides a text-based interactive user interface during the simulation stage which further reduces the debugging effort for a complex system. Logsim uses three files with different file extensions to store the description of a digital system. The file extensions are .mcr, .sim, and .tem. The .mcr file stores the description of all hardware modules. The .sim file specifies the signal flow throughout the system and the mapping between input/output signals and their display symbols. The .tem file describes the location of input/output signals on the simulator screen in terms of their display symbols. A design of a simple 4-bit ripple-carry adder module, shown in Figure 5.4.1, using Logsim is given in Figure 5.4.2, 5.4.3 and 5.4.4 as an example.

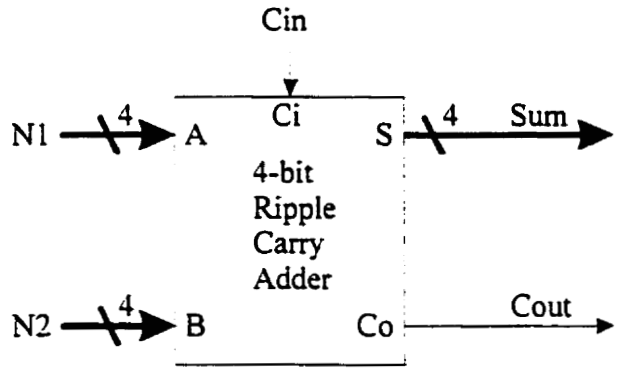


Figure A.1 The block diagram of the 4-bit ripple-carry adder module.

```
template file RC4.tem

define 1BitRCAdder A,B,Cin,S,Cout
        xor inputs $A,$B,$Cin output $S
        nand inputs $A,$B output $nab
        nand inputs $A,$Cin output $nac
        nand inputs $B,$Cin output $nbc
        nand inputs $nab,$nac,$nbc output $Cout
end

define 4BitRCAdder A3,A2,A1,A0,B3,B2,B1,B0,Cin,S3,S2,S1,S0,Cout
        1BitRCAdder $A0,$B0,$Cin,$S0,$t0
        1BitRCAdder $A1,$B1,$t0,$S1,$t1
        1BitRCAdder $A2,$B2,$t1,$S2,$t2
        1BitRCAdder $A3,$B3,$t2,$S3,$Cout
end
```

Figure A.2 The .mcr file for the 4-bit ripple-carry adder module.

```
include RC4.mcr

4BitRCAdder BitA3,BitA2,BitA1,BitA0,BitB3,BitB2,BitB1,BitB0,\
        CarryIn,Sum3,Sum2,Sum1,Sum0,CarryOut

switch output BitA3 screen a3 key Q
switch output BitA2 screen a2 key W
switch output BitA1 screen a1 key E
switch output BitA0 screen a0 key R
switch output BitB3 screen b3 key A
switch output BitB2 screen b2 key S
switch output BitB1 screen b1 key D
switch output BitB0 screen b0 key F
switch output CarryIn screen cin key Z

led input Sum0 screen s0
led input Sum1 screen s1
led input Sum2 screen s2
led input Sum3 screen s3
led input CarryOut screen cout
```

Figure A.3 The .sim file for the 4-bit ripple-carry adder module.

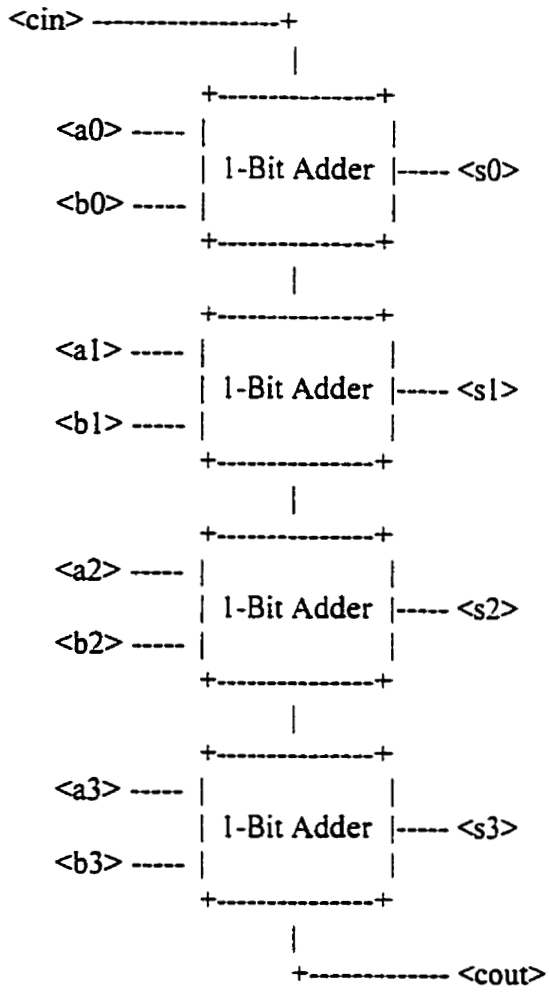


Figure A.4 The .tem file for the 4-bit ripple-carry adder module.

APPENDIX B:

Extra Diagrams

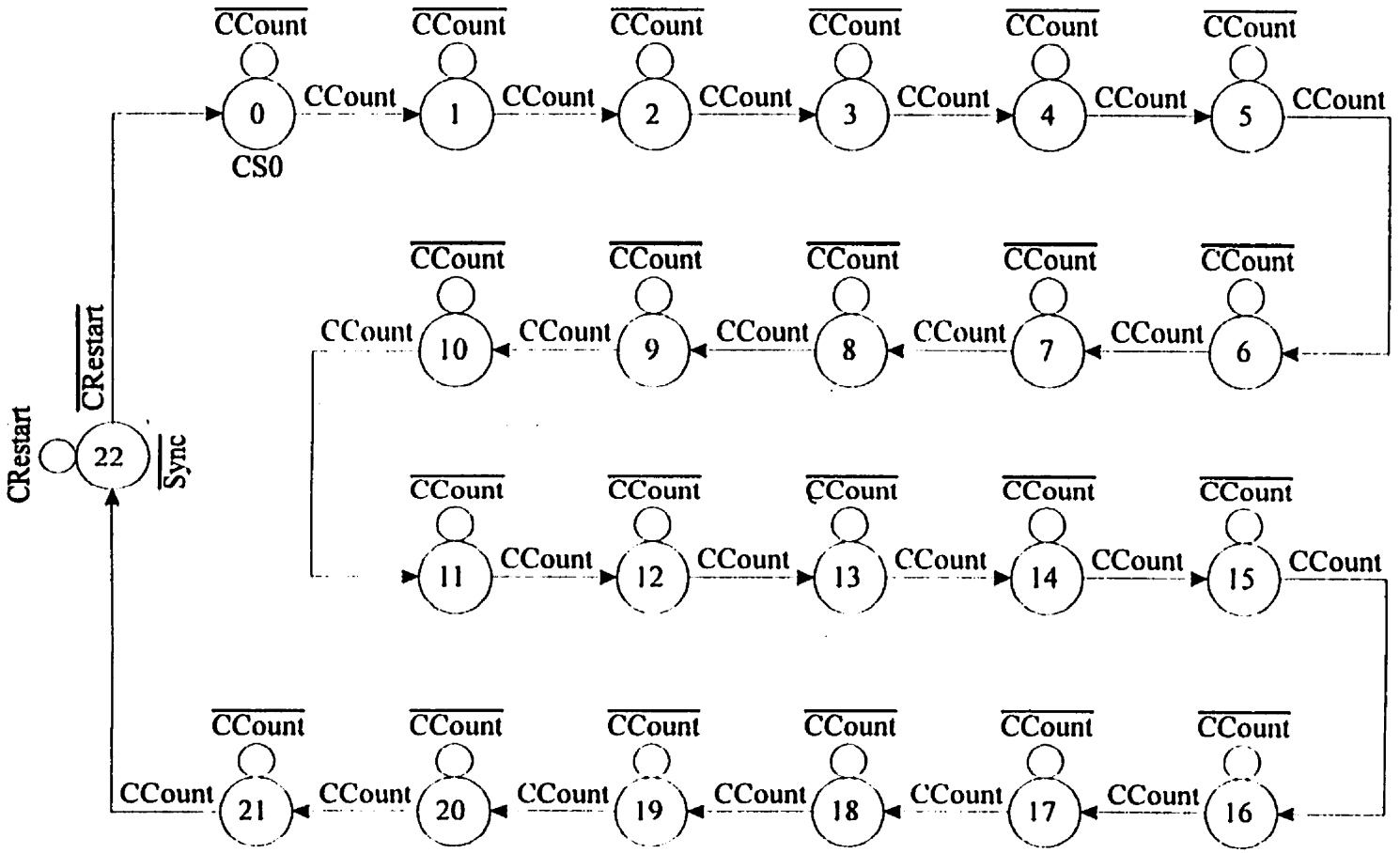


Figure B.1 The state diagram of the counter controller.

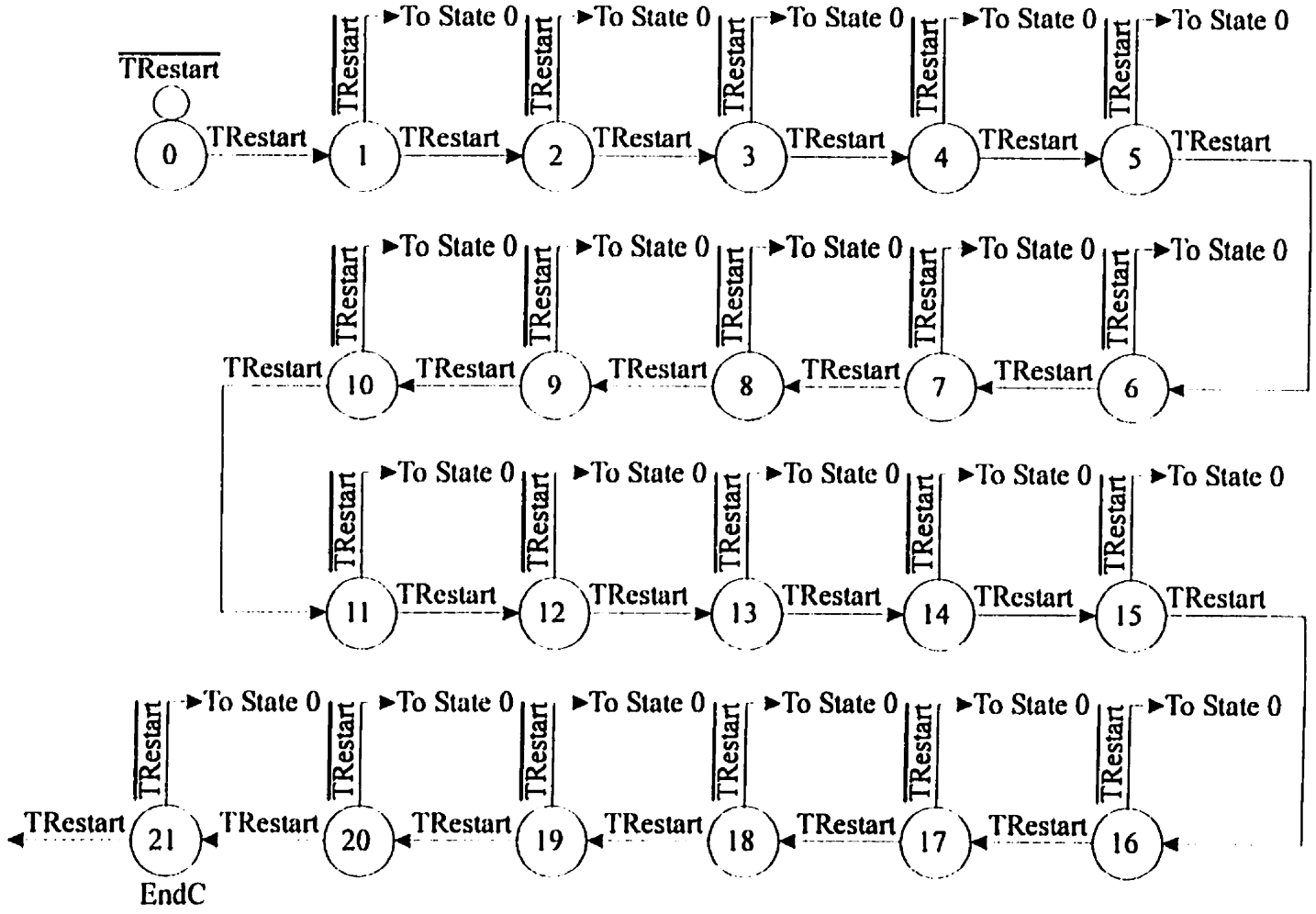


Figure B.2 The state diagram of the timer controller.

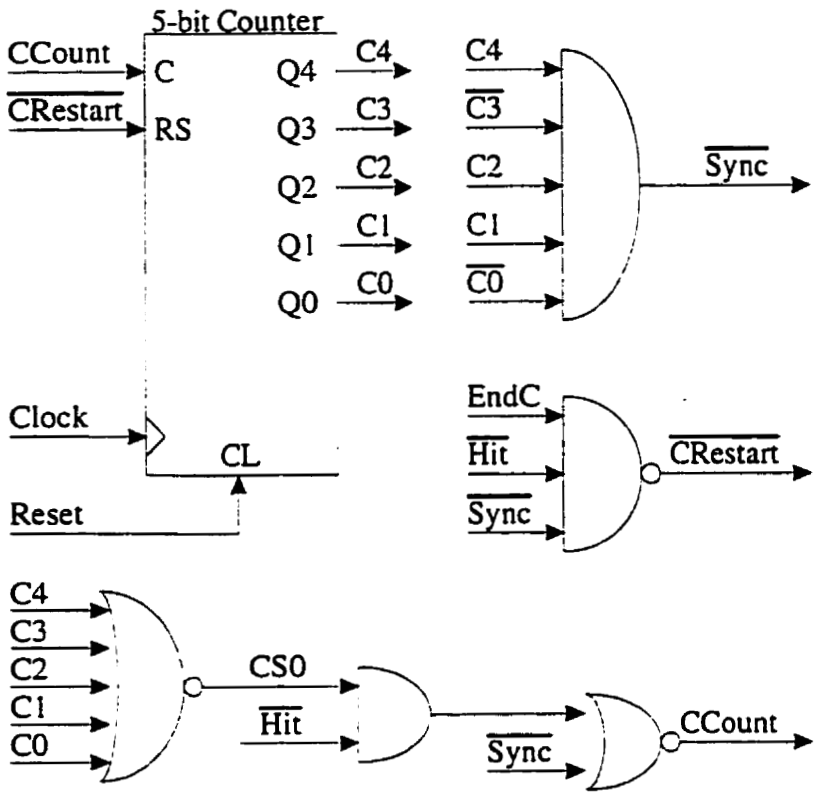


Figure B.3 The block diagram of the counter controller.

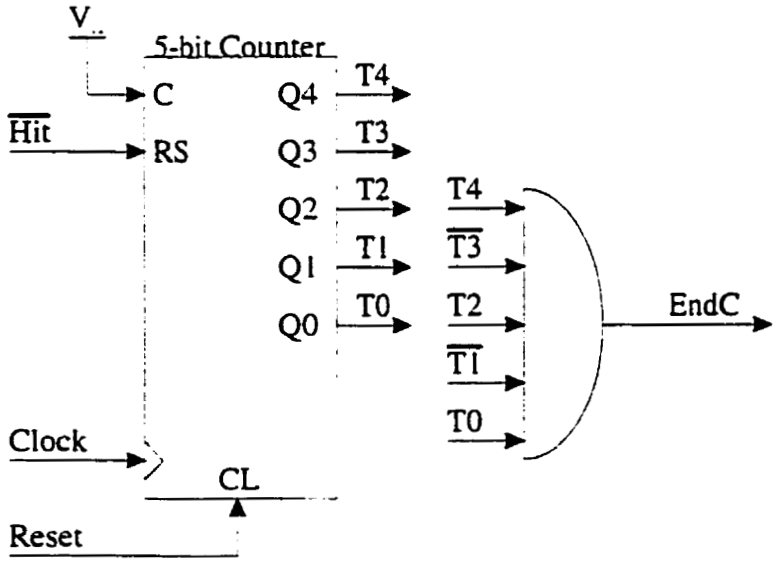


Figure B.4 The block diagram of the timer controller.

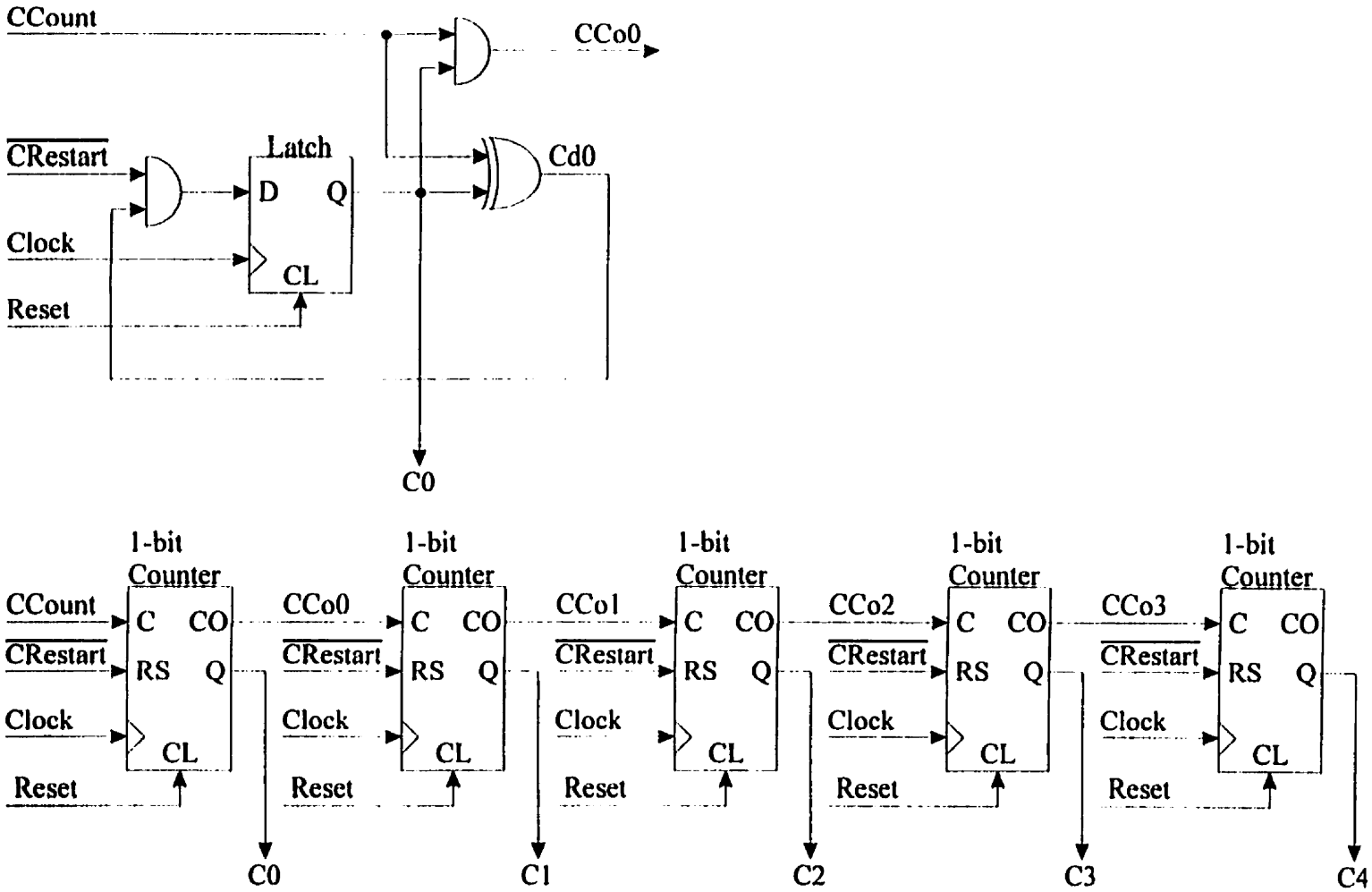


Figure B.5 Detail block diagram of a 5-bit counter unit.

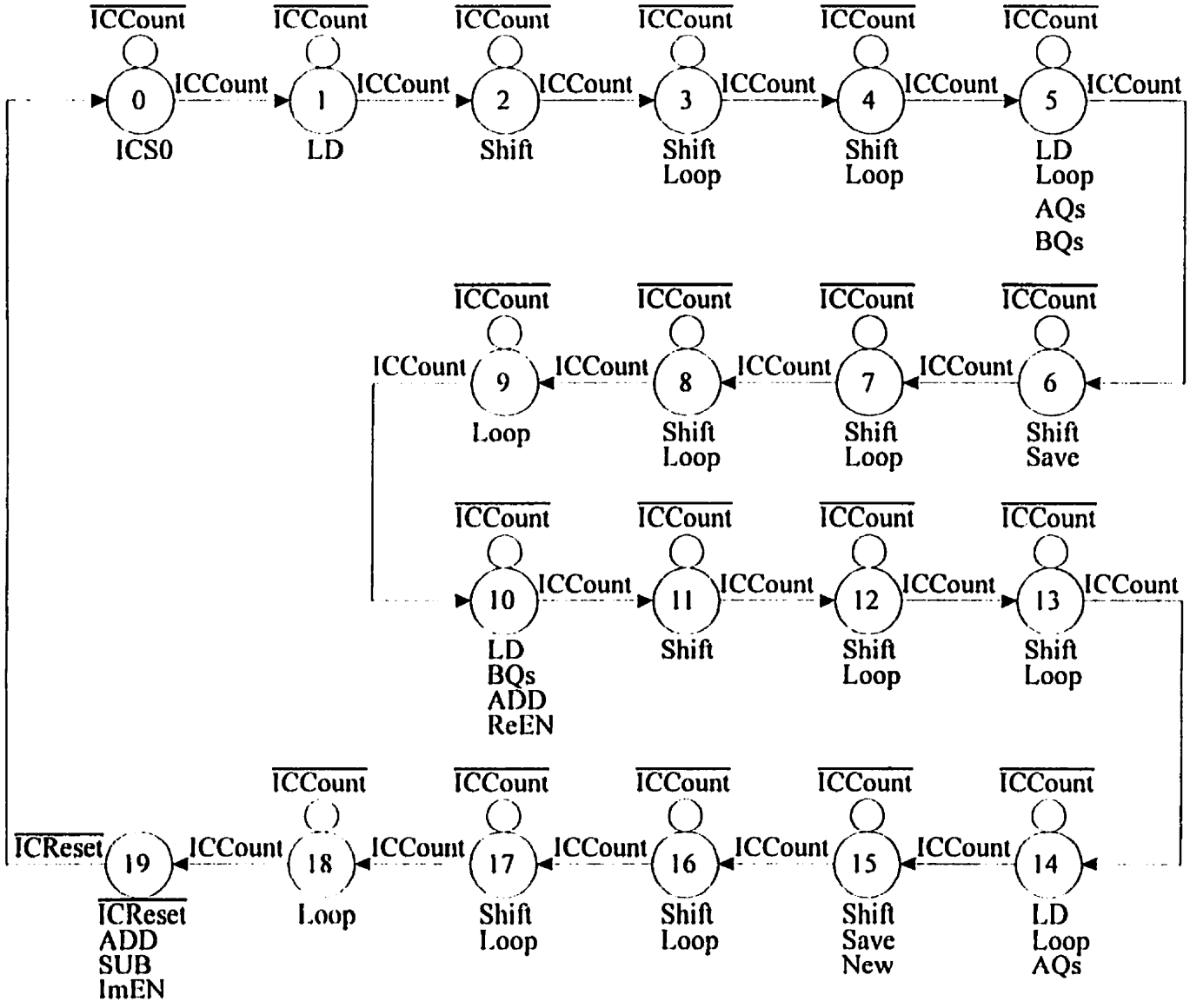


Figure B.6 The state diagram of the cycle controller.

References

[1] J.K. Cavers and M.W. Liao, "Adaptive Compensation for Imbalance and Offset Losses in Direct Conversion Transceivers", IEEE Transactions on Vehicular Technology, vol.42, pp. 581-588, no. 4, November 1993.

[2] J.K. Cavers and S.P. Stapleton, "A DSP-Based Alternative to Direct Conversion Receivers for Digital Mobile Communications", in Proc. IEEE Global Telecommun. Conf., San Diego, CA, Dec 1990.

[3] G.J. Saulnier, C.M. Puckette, IV, R.C. Gaus, JR., R.J. Dunki-Jacobs, and T.E. Thiel, " A VLSI Demodulator for Digital RF Network Applications: Theory and Results", IEEE Journal on Selected Areas in Communications, vol. 8, pp. 1500-1511, no. 8, October 1990.

[4] J.H. Lodge, "Digital Implementation of Narrowband Radio", Report from Communications Research Center, 1989.

[5] E. Patton, "Hardware Implementation of the Digital Rake Transceiver", Master Thesis, pp. 39-58, University of Calgary, 1993.

[6] "HSP43216 Manual of Harris Semiconductor", Jan 1994, pp. 3-43 to 3-59.

[7] "Z2000 Spread-Spectrum Transceiver Manual of Zilog", pp. A-1 to A-11.

[8] H. Meyr and R. Subramanian, "Advanced Digital Receiver Principles and Technologies for PCS", IEEE Communications Magazine, pp. 68-78, January 1995.

[9] Floyd M. Gardner, "A BPSK/QPSK Timing-Error Detector for Sampled Receivers", IEEE Transactions on Communications, vol. COM-34, pp. 423-429, no. 5, May 1986.

[10] Brad Brannon, "IF Sampling Chip Sets Shift Receivers To The Digital Realm", Wireless System Design, vol. 2, num. 10, pp. 34-47, September 1997.

[11] S. Chennakeshu, and G.J. Saulnier, "Differential Detection of $\pi/4$-shifted-DQPSK for Digital Cellular Radio", IEEE Transactions on Vehicular Technology, vol. 42, pp. 46-57, no. 1, February 1993

[12] Y. Akaiwa, and Y. Nagata, "Highly Efficient Digital Mobile Communications with a Linear Modulation Method", IEEE Journal on Selected Areas in Communications, vol. SAC-5, pp. 890-895, June 1987.

[13] P.C. Wong and P.T. Mathiopoulos, "Nonredundant Error Correction Analysis and Evaluation of Differentially Detected pi/4-shifted DQPSK Systems in a Combined CCI and AWGN Environment", IEEE Transactions on Vehicular Technology, vol. 41, pp. 35-48, no. 1, February 1992.

[14] C.S. Ng, T.T. Tjhung, F. Adachi and K.M. Lye, "On the Error Rates of Differentially Detected Narrowband pi/4-DQPSK in Rayleigh Fading and Gaussian Noise", IEEE Transactions on Vehicular Technology, vol. 42, pp. 259-265, no. 3, August 1993.

[15] K.A. Hamied and G.L. Stuber, "An Adaptive Truncated MLSE Receiver for Japanese Personal Digital Cellular", IEEE Transactions on Vehicular Technology, vol. 45, pp. 41-50, no. 1, February 1996.

[16] S. Benedetto, E. Biglieri, and V. Castellani, "Digital Transmission Theory", Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1987.

[17] J.G. Proakis, "Digital Communications", 3rd Edition, McGraw Hill Inc., 1995.

[18] C.L. Liu, and K. Feher, "Noncoherent Detection of $\pi/4$-QPSK Systems in a CCI-AWGN Combined Interference Environment", Proc. IEEE Vehicular Technology Conference, San Franciso, CA, May 1989.

[19] R.L. Pickholtz, D.L. Schilling, and L.B. Milstein, "Theory of Spread-Spectrum Communications --- A Tutorial", IEEE Transactions on Communications, vol. com-30, pp. 855-884, no. 5, May 1982.

[20] Jerry D. Gibson, "Principles of Digital and Analog Communications", 2nd Edition, Macmillan Publishing Company Inc., 1993.

[21] G.W. Wornell, "Emerging Applications of Multirate Signal Processing and Wavelets in Digital Communications:, Proceedings of the IEEE, vol. 84, no. 4, April 1996.

[22] G.R. Cooper, C.D. McGillem, "Modern Communications and Spread Spectrum", 1986 McGraw-Hill, pp. 290.

[23] J.K. Holmes, "Coherent Spread Spectrum Systems", 1990 Robert E. Krieger Publishing Company, Inc., pp. 564, 618.

[24] H. Samueli and B.C. Wong, "A VLSI Architecture for a High-Speed All-Digital Quadrature Modulator and Demodulator for Digital Radio Applications", IEEE Journal on Selected Areas in Communications, vol. 8, pp. 1512-1519, no. 8, October 1990.

[25] Andreas Antoniou, "Digital Filters", 2nd Edition, McGraw-Hill Inc., 1996.

[26] A.V. Oppenheim and R.W. Schafer, "Discrete-Time Signal Processing", Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1989.

[27] Dennis M. Akos and James B.Y. Tsui, "Design and Implementation of a Direct Digitization GPS Receiver Front End", IEEE Transactions on Microwave Theory and Techniques, vol. 44, no. 12, December 1996.

[28] E.O. Brigham, "The Fast Fourier Transform and its Applications", Englewood Cliffs, N.J., Prentice-Hall, 1988.

[29] Simon Haykin, "An Introduction to Analog and Digital Communications", John Wiley & Sons Inc., 1989.

[30] C.Y. Hung, "A Novel Low-Complexity IF-Sampling Receiver", M. Sc thesis, University of Calgary, Department of Electrical and Computer Engineering, December 1996.

[31] D.A. Shnidman, "A Generalized Nyquist Criterion and an Optimum Linear Receiver for a Pulse Modulation System", The Bell System Technical Journal, vol. 46, pp. 2163-2177, no. 9, November 1967.

[32] T. Mathews, L.E. Turner, P.J.W. Graumann and M. Fattouche, "An FPGA Implementation of a Matched Filter Detector for Spread Spectrum Communications Systems", University of Calgary, Department of Electrical and Computer Engineering, October 1996.

[33] A D. Booth, "A Signed Binary Multiplication Technique", Quart. J. Mech. Appl. Math., Vol 4, Part 2, 1951, pp. 236- 240.

[34] S. Kawahito, M. Kameyama, T. Higuchi, H. Yamada, "A High-Speed Compact Multiplier Based on Multiple-Valued Bi-directional Current-Mode Circuits", Proc. 17th ISMVL, May 1987, pp. 172-174.

[35] B.Kish, J.M. Bauer L.E. Turner, and R. Wheatley. Logsim user's guide. Technical report, University of Calgary, Department of Electrical and Computer Engineering, 1989.