INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

Effects of Fixed-Point DSP Processing on DMT Based xDSL Applications

by

Feng Deng, B. Eng.

**A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of**

Master of Electrical Engineering

**Ottawa-Carleton Institute for Electrical and Computer Engineering
Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada
K1S 5B6**

May 2001

© Copyright 2001, Feng Deng

**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-66878-9

Canada

The undersigned recommend to the Faculty of
Graduate Studies and Research acceptance of the thesis

**“Effects of Fixed-Point DSP Processing
on DMT Based xDSL Applications”**

submitted by

Feng Deng, B. Eng.

in partial fulfilment of the requirements
for the degree of Master of Engineering

Dr. M. S. El-Tanafry,
Thesis Co-Supervisor

Dr. R. A. Goubran,
Thesis Co-Supervisor

Chairman, J. W. Chinneck
Department of Systems
and Computer Engineering

Carleton University

May 2001

ABSTRACT

The growth of Internet traffic with data, audio, and video services has made the bandwidth for average computer users' traditional dial-up connections over the POTS the bottleneck. This prompted the development of various access technologies including FTTx, fixed wireless access, and broadband copper access techniques. The xDSL using the existing twisted copper telephone lines is one of the most promising methods to provide high speed access for homes and small businesses in the near future.

This thesis studies the implementation of the xDSL DMT modulated transceiver using a fixed-point DSP running fixed-point algorithms. The effects of the optimal scaling factor to be applied to the DMT coded QAM constellations and the resolution of the DAC become the focal points of this study.

A TI's high speed DSP board, TMS320C6201 EVM, is used to gather the experimental results. However, conclusions are applicable to other fixed-point DSPs.

ACKNOWLEDGEMENTS

Most important of all, I would like to thank my parents and my little sister for their many years of support and encouragement during my academic career. Mom, you can now stop worrying about my studies as your son has finished schooling.

I would like to thank my supervisors, Dr. M. S. El-Tanany and Dr. R. A. Goubran, for their guidance and support in developing this research project to its full potential. My thanks also goes to my research associates and friends, Rosa and Yongyi, for their input during the course of my thesis.

Matt and his father Myron, as well as Jeff and Christine, all deserve my deepest appreciation for your time and patients editing for my thesis.

The support staff at Carleton University deserves thanks for their assistance: Danny and the other lab technicians for their help in solving technical problems; the secretarial staff in the Department of Systems and Computer Engineering for all their help.

Finally, I would like to thank the Natural Sciences and Engineering Research Council of Canada, and Carleton University for their financial support.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	x
CHAPTER 1 Introduction	1
1-1 Problem Statement.....	1
1-2 Thesis Contribution	2
1-3 Thesis Organization	4
CHAPTER 2 Background	6
2-1 xDSL Technology Overview	7
2-2 Signal Impairment in the xDSL Environment.....	10
2-2.1 Crosstalk.....	12
2-2.2 Background Noise	14
2-2.3 Electronic Magnetic Capacity	15
2-3 Discrete Multitone.....	15
2-3.1 DMT Overview	15
2-3.2 QAM Multitone System	17
2-3.3 Signal Processing Steps.....	21
2-3.4 CAP and Comparison with DMT and QAM.....	26
2-3.4.1 CAP Overview	27
2-3.4.2 CAP vs. DMT	29

2-4	G.Lite.....	34
2-5	ANSI T1.413 Standard	35
CHAPTER 3 Review of Advances in DSPs		43
3-1	ASIC vs. DSP Implementation	43
3-2	Floating vs. Fixed Point Implementation	46
3-3	Fixed Point Representation	48
3-3.1	Fractional Q Formats	48
3-3.2	Q.15 format.....	49
3-3.3	Q.31 format.....	49
3-3.4	Q.F Conversion Example.....	50
3-4	C62x DSP Overview.....	51
CHAPTER 4 Experimental Model and Results		52
4-1	Experimental Model.....	52
4-1.1	Description of Simulation Model	54
4-2	Simulation Platform and Setup.....	54
4-2.1	Experiment Environment	54
4-2.2	Data Flow Diagram and Functional Description	56
4-3	Simulation Simplifications and Assumptions	64
4-3.1	Channel and Constellation Encoder	64
4-3.2	D/A and A/D Converters	65
4-3.3	Omissions	65
4-4	Experimental Results	66
4-4.1	Sources of Errors.....	66
4-4.2	Symbol Error Rate	70
4-4.3	Mean Square Errors	73
4-4.4	Envelope Distribution.....	83
4-4.5	Error Analysis	85
4-4.5.1	Magnitude	85
4-4.5.2	Angle.....	87
4-4.5.3	Frequency Domain Analysis of Errors	87

4-4.6	Signal to Noise Ratio	90
4-4.7	Effect of Number of Bits Used for D/A Conversion	93
4-4.8	DMT Transceiver with Different Sizes of FFT	96
4-4.8.1	Transmitted Signal Power.....	96
4-4.8.2	Signal to Noise Ratio	98
CHAPTER 5 Conclusions and Future Works		101
5-1	Conclusion.....	101
5-1.1	Symbol Error Rate	101
5-1.2	Error and Performance Analysis.....	102
5-1.3	Envelope Distribution.....	102
5-1.4	Other Results.....	103
5-2	Future Works.....	103
APPENDIX A Quantization Effects in the Computation of DFT		105
1-1	Quantization Errors in the Direct Computation of the DFT	105
1-2	Quantization Errors in FFT Algorithms	109
APPENDIX B Effect of DFT Size on Average Power of Signals		114
2-1	Average Power in Terms of FFT Size N.....	114
APPENDIX C Program Listing		116
3-1	Transceiver.....	116
3-2	IFFT Routine	122
3-3	FFT Routine	123
REFERENCES		129

LIST OF TABLES

Table 1.	Complexity comparison of CAP and DMT	28
Table 2.	Q.15 bit fields.....	49
Table 3.	Q.31 low memory location bit fields.....	49
Table 4.	Q.31 high memory location bit fields	50
Table 5.	QAM symbol error rate with different A1	71
Table 6.	MSE for different A1 and A2.....	75

LIST OF FIGURES

FIGURE 1.	F_A , W_{eff} and SNR_i 's physical interpretation	18
FIGURE 2.	A simplified DMT transceiver	20
FIGURE 3.	A specific ADSL transceiver.	25
FIGURE 4.	The CAP transmitter and receiver	28
FIGURE 5.	Constellation labels for $b=2$ and $b=4$	39
FIGURE 6.	Constellation labels for $b=3$	40
FIGURE 7.	Constellation labels for $b=5$	41
FIGURE 8.	Block diagram of the simulation model used on the 'C62x EVM DSP board.....	53
FIGURE 9.	TI Code Composer Studio user interface	55
FIGURE 10.	Data flow diagram.....	56
FIGURE 11.	An example of smearing of QAM constellation on fixed point DSPs	68
FIGURE 12.	Demodulated constellation with $A_1=512$, $A_2=1$	69
FIGURE 13.	Symbol error rate due to scale factor A_1	72
FIGURE 14.	Effect of A_1 and A_2 on the mean square error	76
FIGURE 15.	Performance with respect to A_1	77
FIGURE 16.	Performance at different values of A_2 , with $A_1=128$	79
FIGURE 17.	Applicable values of combinations of A_1 and A_2	82
FIGURE 18.	The histogram of magnitude of transmitted signal.....	84
FIGURE 19.	Normalized histogram of amplitude of the errors from their means	86
FIGURE 20.	Normalized histogram of distribution of the error angles	88
FIGURE 21.	Frequency domain analysis of the errors.....	89
FIGURE 22.	The effects of amplification factor A_1 on SNR.....	92
FIGURE 23.	The effect of quantization of A/D-D/A converter on linear scale	94
FIGURE 24.	The effect of quantization of A/D-D/A converter on logarithm scale	95
FIGURE 25.	Peak to average power ratio as a function of A_1	97
FIGURE 26.	Effect of size of DFT on SNR.....	100

LIST OF ABBREVIATIONS

A/D, ADC	Analog to Digital Convertor
ADSL	Asynchronous Digital Subscribers Line
AGC	Automatic Gain Controller
ANSI	American National Standards Institute
ASIC	Application Specific Integrated Circuit
AWGN	Additive white Gaussian Noise
BER	Bit Error Rate
CAP	Carrierless Amplitude Pulse Modulation
D/A, DAC	Digital to Analog convertor
DFD	Data Flow Diagram
DFT	Discrete Fourier Transform
DMT	Discrete Multitone
DSL	Digital Subscribers Line, also known as xDSL
DSP	Digital Signal Processing
EMC	Electronic Magnetic Capacity
ETSI	European Telecommunications Standards Institute
EVM	Evaluation Module
FEXT	Far-End Crosstalk
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
IBI	Interblock Interference
ICI	Intercarrier Interference
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISI	Intersymbol Interference
ITU-T	International Telecommunications Union — Telecommunication Standardization Sector
LAN	Local Area Network
MIPS	Million Instructions Per Second
modem	Modulator/Demodulator
MSE	Mean Square Error

NEXT	Near-End Crosstalk
PAR	Peak to Average Ratio
PC	Personal Computer
PSD	Power Spectrum Density
PSTN	Public Switched Telephone Network
POTS	Plain Old Telephone Service
QAM	Quadrature Amplitude Modulation
TEQ	Time Domain Equalizer
TI	Texas Instrument
transceiver	Transmitter/Receiver
VDSL	Very High-Speed Digital Subscriber Line
VLSI	Very Large-Scale Integration
VoIP	Voice over IP
VoD	Video on demand

CHAPTER 1 *Introduction*

1-1 Problem Statement

In the last decade, exponential growth of Internet usage and personal computers (PC) have prompted the expansion of the bandwidth in the backbone networks. However, the remote access techniques have not yet caught up with this high speed Internet traffic flow, and dial-up analog modems are still the only way to get on-line for the majority of Internet users. To change this situation, service providers have developed and rolled out a number of different access technologies in the last few years. These include fixed wireless access, cable modems, and ADSL access. In particular, ADSL and its successor technologies (xDSL) look very promising before the full migration towards fiber optical networks.

ADSL is not the only one of the technologies in xDSL family. Technical committees are already working on the standardization of other higher bandwidth techniques over the same twisted pairs, but for a shorter distance. To evaluate and achieve the time to market constraints for these new techniques, fast turnaround time is a must. Use of DSP based implementation is a solution for this

problem. It could also reduce the cost for customer side equipment and allow easy upgrades based on improved algorithms or an update to the new standards.

There are two flavours of DSPs available: fixed point and floating point. The cost for floating point DSPs are usually many times their fixed point version. Therefore, fixed point DSPs are chosen more often. However, for a DSL applications using DMT modulation, the FFT/IFFT routines require precise floating point calculations. When fixed point DSPs are used to perform these routines with fixed point representations of floating point values, errors are introduced into the system in the form of quantization errors, calculation overflows and underflows, and rounding errors.

This study attempts to characterize these errors in the absence of channel distortion, thus providing an understanding of the effects of these errors on the system level of a DMT based xDSL transceiver.

1-2 Thesis Contribution

A literature search was first conducted at the early stages of this thesis to prove that digital signal processors have already met or exceeded the real time requirement of basic ADSL transceivers. Upon the conclusion from the literature search, this thesis evaluates errors generated by fixed point DSP on DMT based DSL transceivers. A platform was set up using Texas Instruments' TMS320C62x, a high speed fixed point DSP processor with multiple processing units that has the capacity to perform the complex operations required by the DMT symbol modula-

tor and demodulators of an ADSL system. When real-time performance is not required, the platform can also be used to analyze other forms of DMT based DSL or OFDM systems.

During the course of this study, a significant amount of empirical data was collected using the DSP platform with the DMT based xDSL transceiver in focus. There are many factors that have shown impact on the result of the demodulated DMT signals. They include: the number of points in the QAM constellations, the automatic gain control at the transmitter side, the automatic gain control at the receiver side, and the number of bits used for the DAC and ADC. The thesis addresses these issues by changing these variables individually to show the effects of these factors on the demodulated signals.

After data collection, statistical analysis was performed to characterize the added noise introduced by these variables, which are more significant when a fixed point DSP and fixed point algorithm is used to perform the modulation and demodulation process. Signal to noise ratios were also computed for each of the variables, for the fixed point DSP implementation. A simple linear relationship between the SNR and automatic gain control was found on a semilog scale. This relationship can be used to determine the number of bits required for a DMT modulator/demodulator on fixed point devices.

Along with the literature search, the analysis of envelope of the DMT signal as affected by fixed point processing and clipping is also presented.

1-3 Thesis Organization

This thesis is organized in five chapters. The first chapter states the problems that arising from using DSP to implement the xDSL modems and the proposed topics of investigation. The engineering contribution from this thesis is also stated.

Chapter 2 gives background information for the various xDSL technologies and the noise sources that are associated with these broadband copper access techniques. Emphasis is given to a particular modulation method called discrete multitone (DMT) with a comparison to Quadrature Amplitude Modulation (QAM), and carrierless amplitude and pulse modulation (CAP), which are also two possible modulations for xDSL. A brief description of the current ADSL standard, ANSI T1.413, is also given in this chapter.

Chapter 3 reviews the advances in digital signal processors with a comparison of fixed and floating point DSPs. One of the formats of representing floating point values on fixed point DSPs (Q.F format) as well as TI TMS320C62x specific information are given at the end of Chapter 3.

At the beginning of Chapter 4, the simulation model used to collect the experimental data is given. A more detailed data processing block diagram, and the functional blocks' operations are provided after the simulation model. The simplification and assumptions used to develop the simulation is also listed in the following section. Finally, the experimental results and their analysis takes the latter half of the chapter.

Chapter 5 gives the conclusions gathered from the experimental results and a list of possible future work on related topics.

CHAPTER 2 *Background*

In the last decade, the explosive expansion of Internet based applications and digital services such as video on demand (VoD), video conferencing, on-line gaming, etc. have prompted an explosion of Internet traffic bandwidth. Vendors have responded to this change with the deployment of optical networks which can handling hundreds of Gigabits per second. In contrast, the evolution of remote access techniques lagged behind this growth, leaving many Internet users limited to the old dial-up modems, with a bandwidth only in the tens of kilo bits per second range.

However, this situation is to be changed in the very near future. Many access technologies will provide high speed connections covering the last mile to all the end users at homes or small businesses in North America and around the world. These technologies include optical fiber (fiber to the home or fiber to the curb), wireless (fixed wireless access or satellite connection), fast ethernet, and a number of copper access technologies. Due to the high installation cost of fiber [12], or the operating cost of wireless systems, two copper based access methods look very promising in the short run: the cable modem from the cable companies and xDSL from the telephone operators, both using existing wires that are reach-

ing to the homes. These two technologies are both available to end users and the red hot competition can be easily observed in the North American markets. Although the cable modems had a head start, the telephone companies' ADSL (G.Lite) subscribers are increasing at a very fast pace due to the shared media nature of cable network and increased concern of Internet security.

Although the deployment of DSL systems often requires the change of equipment on both sides of the line and the absence of loading coils which limit the bandwidth and performance of the DSL systems, when compared to the investment of time and money to set up a new access network, the various DSL solutions are very attractive.

This chapter is organized as follows: In Section 2-1, various DSL technologies are briefly introduced, followed by a discussion of the environment and noise sources for DSL systems. A detailed discussion of discrete multitone (DMT), which is the standard line code for ADSL, is given in Section 2-3. DMT is compared to the other line coding for ADSL, the carrierless amplitude and phase modulation (CAP). Section 2-4 deals with some practical issues in the ADSL environment; mainly full-rate ADSL versus G.Lite, in addition to electromagnetic radiation issues.

2-1 xDSL Technology Overview

Digital subscriber loop (DSL) technologies are broadband transmission systems utilizing the spectrum resources beyond the 4 kHz channel bandwidth, traditionally used for speech transmission, on the existing twisted-pair telephone wires. The current 4kHz signal bandwidth is limited by the central office (CO) ter-

minal equipment such as switch line cards. In addition, the use of loading coils (inductors) also significantly increases the cables' attenuation above 4 kHz. Although the improvement of signal processing techniques has pushed the analog modem faster and faster, they are still bounded by this 4 kHz bandwidth limit, thus approaching their physical capacity at 56 kbps under the Federal Communication Committee (FCC) maximum transmission power regulations. DSL systems can achieve much higher rate by removing this limiting factor and by the use of spectrum up to a few MHz.

Over the years, different transmission rates have been proposed and studied for the DSL technology. They are collectively called xDSL. These xDSL technologies include:

1. Basic rate DSL

Basic rate DSL is also known as ISDN DSL (IDSL). It supports a line rate of 160 kbps for up to 18 kft.

2. High-bit-rate DSL (HDSL)

HDSL has been in service for several years now. It provides T1 (1.544 Mbps) using two twisted pairs, each at 784 kbps, over the service range of 9-12 kft.

HDSL is a baseband technology requiring all of the bandwidth up to 500 kHz, including the 0 to 4 kHz needed for voice traffic. It therefore cannot support simultaneous telephone service and high-speed digital communications over the same telephone line.

3. Symmetrical DSL (SDSL)

SDSL is a single-pair application of the basic HDSL technology. Line rate could be lower or higher than the original 784 kbps that HDSL uses.

4. Rate Adaptive DSL (RADSL)

Different line conditions will result in different channel capacity for each line. Therefore, RADSL is proposed by supporters of CAP technology to find the maximum rate a loop will be able to support. Further discussion of RADSL will be given in Section 2-3.4.

5. Asymmetrical DSL (ADSL)

ADSL takes advantage of the asymmetric nature of video and Internet traffic. It assigns more bandwidth in the downstream than upstream direction. The line rate for the downstream is set to 6 to 8 Mbps over the CSA¹ range.

There are two different line coding technologies that are used for ADSL, namely DMT and CAP. Details and comparison of DMT and CAP will be discussed later in this chapter.

6. Very High Speed DSL (VDSL)

This is currently being studied by the ANSI standard group that standardized ADSL. It applies only in the drop portion for transmission rates greater than 10 Mbps. VDSL will support speed from T1 (1.544 Mbps) to OC1 (52 Mbps).

1. CSA, Carrier Serving Area, which in direct relation to the loop length, determines the maximum transmission rate for any digital subscriber loops. As the CSA increase, the transmission rate decreases. For ADSL, a CSA of 18000 kft or less is assumed.

2-2 Signal Impairment in the xDSL Environment

Copper access transmission systems face a variety of impairments that present barriers to their operation. For example, several published works, [2] and [7], have listed many of the xDSL impairments. Such impairments are

- intersymbol interference (ISI) and interblock interference (IBI), which result from the frequency dependent attenuation of the twisted pairs
- crosstalk noise coupled from adjacent loops within the same cable bundle. The two types of crosstalks are Near End Crosstalk (NEXT) and Far End Crosstalk (FEXT)
- impulsive noise caused by switching transients, lightning, and other electrical machinery
- echo noise resulting from the combination of imperfect hybrids and gauge changes on the line as well as signals reflected from bridged taps
- inductive noise at 60 Hz and its harmonics caused by nearby power lines
- single tone interference such as AM radio stations which broadcast from 525 kHz to 1.705 MHz in Canada. These frequencies coincide with xDSL spectra.

These impairments to the xDSL systems can be broadly classified as intrinsic or extrinsic to the twisted pair environment.

Thermal noise, echoes and reflections, attenuation, and crosstalks can all be classified as intrinsic noise. There are also other components that reside in the copper infrastructure that can impair the operation of DSL systems. These include surge protectors, radio frequency interference (RFI) filters, and, in some networks, bridged taps and loading coils. These intrinsic impairments depreciate DSL performance.

Examples of extrinsic impairments are impulsive noise originating from lightning strikes, electric fences, power lines, machinery, arc welders, switches, fluorescent lighting, and so on. These impulsive noises tend to have their spectrum concentrated at frequencies below 70 kHz. Therefore, they pose serious problems for baseband systems such as HDSL. However, since ADSL and VDSL have passband above 100 kHz, they are not limited by these impulsive noises. There is also radio interference from broadcasting and radio transmitters. However, like all practical digital communication systems, DSL systems use error correction codes and interleaving to mitigate the effect of impulsive noise.

The noise sources mentioned above can alternatively be classified as capacity or performance limiting. Capacity limiting noise is usually slowly changing, such as thermal noise and crosstalk. These noise levels are often predictable and relatively easy to take into account when the telephone companies create standards and/or deployment-planning rules.

Performance limiting noise such as impulses and RFI, is intermittent in nature. It is geographically variable and unpredictable, and therefore is usually accounted for in planning rules by using a safety margin.

Some of the most important noise sources for the DSL systems are detailed in the next few sections.

2-2.1 Crosstalk

Since the key limitation to DSL capacity is crosstalk, many researchers have invested considerable effort in understanding the crosstalk environment of the access network. Measurement results and empirical models are often used in the performance analysis of practical DSL systems. Such models also facilitate more accurate performance predictions. They are essential in the development of equipment deployment rules. Such models are often incorporated in various industry standards such as ANSI T1.413.

Crosstalk is by far the greatest contributor for capacity limiting noise. There are two types of crosstalk:

FEXT (Far End Crosstalk) is interference that appears on another pair at the opposite or far end of the cable relative to the source of the interference. Its level is attenuated at least as much as the signal itself if both have travelled the same distance.

FEXT is a well understood problem for telephony cables at low frequencies. However, it is not well characterized for higher frequencies in a real network.

NEXT (Near End Crosstalk) is interference that appears on another pair at the same end of the cable as the source of the interference. Its level is substantially independent of the length of the cable. The effect of NEXT is far greater than that of FEXT.

If NEXT is the only impairment, then theoretically, it can be completely eliminated if different frequency bands are used for the two directions of transmission. This is how ADSL and VDSL deal with NEXT by separating the upstream and downstream traffic in the frequency domain. Time compression multiplexing achieves the same result.

Another technique that would eliminate NEXT is crosstalk cancellation. This is used when the up-stream and down-stream signals overlap in the frequency domain.

In another new technique in NEXT suppression, a combination of linear pre-filtering and post-filtering will eliminate NEXT [5]. This technique, like NEXT cancellation, requires the knowledge of NEXT transfer functions between the pairs involved. For this reason, it is impractical because the identity of the disturbing transmitters is not known in most applications. An exception to this is in coordinated transmission, where both NEXT cancellation and post-filtering are used to eliminate FEXT between the constituent pairs of a two-pair HDSL are employed [5].

Because NEXT is not Gaussian, making the capacity of any NEXT impaired channel different than what it would be for a Gaussian channel, causing it more difficult to model. Many researchers proposed models to describe NEXT and the NEXT dominated channel as accurately as possible. One of the most accepted models is given in [8].

The NEXT transfer function $H_{NEXT}(f)$ is modelled using the equation

$$|H_{NEXT}(f)|^2 = K_{NEXT} f^{3/2} \quad (\text{EQ 1})$$

where K_{NEXT} for a 50-pair cable was approximately 10^{-13} .

2-2.2 Background Noise

Based on values of experimental data, it is shown that the background noise, like crosstalk, is not Gaussian either [6]. However, the Gaussian density is still used as a good engineering approximation to the background noise density. The residential loop background noise is not quite as damaging to transmission as Gaussian noise is. Therefore, the Gaussian approximation is likely to yield a performance bound that is more on the pessimistic side.

The data collected in [6] show that the levels of background noise measured at three New Jersey residences were reasonably low in the band of 8 kHz through 3 MHz. In the band that ADSL occupies, the background noise was generally below -140 dBm/Hz. Therefore, the background noise is usually modelled as additive white Gaussian noise (AWGN) with a power spectrum density (PSD) of -140 dBm/Hz. This approach is used in several references, as for example [2], [4], and [6]. The data in [6] also show that there were significant spikes of noise corresponding to local AM radio station at 570 kHz and above, proving that RFI is of great concern in DSL systems.

In addition to impulsive noise and crosstalk, induction noise is also relevant to DSL performance. One reason for this is that induction currents can be substantial enough to cause nonlinearities in system performance.

2-2.3 Electronic Magnetic Capacity

Electronic Magnetic Capacity (EMC) of the xDSL systems, if not carefully considered, could cause interference to the xDSL system as well as other communication systems. DSL lines have the capacity to act as antennas. This means that they can pick up radiated emissions which may become sources of interference to DSL systems; equally likely, there is the potential for DSL line signals to leak out of the cables and cause interference to other radio systems.

Radio Frequency Interference (RFI) pickup becomes an issue only when there are strong nearby transmitters such as AM radio stations.

2-3 Discrete Multitone

Using the same copper wire, different line codes used for transmission would result in different system performance in terms of maximum transmission rate, bit error rate, and service range. There are many possible line code schemes which can be applied to xDSL systems. They include QAM, DMT and CAP.

As mentioned earlier in this chapter, there are two competing line coding technologies for ADSL: DMT and CAP. DMT is the one selected by ANSI for ADSL standard in ANSI T1.413. It will be described in more detail in the following sections.

2-3.1 DMT Overview

Very efficient modulation and equalization techniques are necessary in ADSL transmission environments because of severe channel attenuation,

intersymbol interference and interblock interference, and a number of other line impairments, including crosstalk, AWGN, and impulse noise. As a result, the powerful DMT modulation technique is chosen by ANSI as the standard line code for ADSL. The ITU clearly endorsed the selection of DMT, paving the way for a single global standard. In this section, a brief overview of DMT is given.

DMT is closely related to orthogonal frequency division multiplexing (OFDM) or the Coded OFDM (C-OFDM), which has just been selected for Europe's Digital Audio Broadcast (DAB) to broadcast CD quality sound and multimedia over the airwaves for in-car and mobile applications.

The idea of DMT or OFDM is based on the fact that most of the realistic channels have non-flat frequency transfer characteristics. The theoretical limit for the channel capacity can be achieved with the water pouring solution [13]. This solution suggests an optimum power and bit allocation strategy, assuming the frequency response of the channel, and the additive noise density are known for all frequencies of interest. The effectiveness of the water pouring solution is highly dependent on the channel and noise tracking capability of the DSL transceiver.¹

Despite that DMT may sound complicated, based on the tremendous performance advantage to CAP and QAM, DMT was unanimously chosen as the standard ADSL line code by ANSI T1E1.4 members in March of 1993. Since then, the standard was updated in 1995 and 1998. The ANSI T1.413 standard for ADSL specifies 256 subcarriers, each with a 4 kHz bandwidth. They can be independently modulated using QAM. The QAM constellation size may vary from 4 points

1. The frequency response of the channel and the noise characteristics change with time at a speed that depends on the line and neighbor-line traffic conditions.

up to 2^{15} points. Generally, at low frequencies, where attenuation is low and the signal to noise ratio (SNR) is high, very dense constellations are used. Use of constellations greater than 10 bits/sec/Hz is typical. Under unfavorable line conditions, modulation can be relaxed to accommodate lower SNR, usually 4 bits/sec/Hz or less, to deliver the necessary noise immunity and keep the bit error rate (BER) at the required level. Furthermore, as the system measures line integrity, it can also avoid or compensate for cross-talk or single tone interference.

2-3.2 QAM Multitone System

The fundamental goal of all "multicarrier" modulation techniques is to partition a data transmission channel into a set of orthogonal, memoryless subchannels, each with its own carrier. It can be shown that DMT is a special case of optimal multicarrier modulation at the frequency domain, under certain sets of conditions and assumptions [3]. From a performance perspective, DMT is not optimal because of the redundant transmission power used by the DMT for cyclic prefix. However, in practice, DMT modulation is the preferred methodology because of its advantage in terms of computational complexity. A DMT system with N complex subchannels in the frequency domain requires a FFT size of $2N$, and the forced conjugate symmetry in the frequency domain will result in the desired real-valued time domain samples.

As described in the overview of DMT, different numbers of bits can be conveniently assigned to different subchannels. As result, more data can be transmitted over those subchannels with larger SNR's and less data can be transmitted over those subchannels with smaller SNR's. In this manner, the overall system performance can be optimized to achieve the maximum transmission rate at a

given BER. This bit assignment is done through a bank of Quadrature Amplitude Modulation (QAM) modulators.

FIGURE 1. F_A , W_{eff} and SNR_i 's physical interpretation

For the Multitone QAM system, each transmitted signal tone is given by

$$x_i(t) = \sum_{n=-\infty}^{\infty} a_n s(t - nT_i) \cos 2\pi f_i t + \sum_{n=-\infty}^{\infty} b_n s(t - nT_i) \sin 2\pi f_i t \quad (\text{EQ 2})$$

where $T_i = 1/W_i$, and f_i is the carrier frequency of the i th signal. The maximum bit rate the DMT system can handle is given by

$$R_{b_{\max}} \cong \int_{(f \in F_A)} \log_2 \{K_0 |H(f)|^2\} \quad (\text{EQ 3})$$

where $K_0 = \left(\frac{3}{2} \cdot \frac{P}{N_0 W_{\text{eff}}}\right) / (-\ln[Pr\{\varepsilon\}])$ and W_{eff} is the measure of the positive frequency range of F_A where the number of bits can be transmitted in that frequency sub-channel is greater or equal to 2.

For each of the subchannels, the number of bits that can be assigned is given by

$$b(i) = \log_2 \left(1 + \frac{SNR(i)}{\Gamma}\right) \quad (\text{EQ 4})$$

where $b(i)$ is the number of bits that can be supported in the i th subchannel, $SNR(i)$ is the signal-to-noise ratio of the i th subchannel, and Γ is the SNR gap, which is a function of the target bit error rate (BER), the chosen coding scheme, and the desired system performance margin.

Once the number of bits is assigned to each frequency bin, and the constellation determined, the bank of QAM modulators parses the bit stream and modulates them accordingly into frequency domain signals as shown in Figure 2.

FIGURE 2. A simplified DMT transceiver

The operation of the DMT modulator can now be summarized as follows:

1. **Tone encoding:**
 The n_i bits assigned to the tone i are mapped into a QAM complex constellation point.
 The size of the QAM constellation is 2^{n_i} .
 The constellation points to be transmitted over all tones are grouped into a complex vector.
 2. The N complex symbols generated by the tone encoder are appended with their Hermitian extension, thus yielding a $2N$ -point complex vector.
 3. The $2N$ -point vector is applied to a $2N$ -point IFFT block, which produces a $2N$ -point real vector.
-

4. A cyclic extension is added to the IFFT output. This extension is simply a repetition of the last N_p coefficients of the IFFT output.
5. The resulting $2N+N_p$ point vector is converted into a serial form, and then is applied to a D/A converter.

In [7], it has been reported that even for a channel with only two levels, when there is a deep null, large improvement is possible using multitone, while the improvement for the Gaussian channel is not as great. However, it is expected that multitone QAM shows greater improvement for channels with sharply decreasing amplitude characteristics, such as the twisted-pair channel. The authors of [7] also proved that multitone system substantially outperforms the linear equalization techniques for channels with deep nulls. This was the reason why DMT was selected for ADSL standards over CAP by ANSI and ITU.

2-3.3 Signal Processing Steps

The main functions of the receiver are described as follows:

FFT demodulator:

After proper analog filtering, the received signal is passed through an A/D converter which produces a sampled sequence. Assuming the receiver has the proper symbol timing, the sampled sequence is applied to a serial to parallel block which produces a $(2N+N_p)$ real vector for each DMT symbol. The cyclic prefix (N_p points) is then removed, and the remaining $2N$ points are applied to an FFT block which acts as a demodulator.

The FFT output is a $2N$ -point complex vector. The first N points is a scaled and rotated version of the transmitted N -QAM symbols. The scaling and rotation vary from one tone to the next, as dictated by the channel conditions.

Bit swapping:

DSL systems are faced with a quasi-stationary, slowly time varying transmission environment in ADSL. Although very slow in general, the frequency response of the channel does change over time. To maintain the maximum transmission rate at the required bit error rate, when channel conditions change, bit swapping is used [3].

The first step in this adaptive bit swap procedure is to keep track of the steady-state Mean Squared Error (MSE) of all subchannels in the receiver of a DMT-based ADSL system in the background during normal operation. When the MSE of one subchannel is over a certain predetermined threshold value, for example, more than 3 dB worse than another subchannel, a bit is swapped from the worse subchannel to the better subchannel, so that the overall BER will not be reduced.

Time-domain equalizer:

In a finite-length DMT system, the effective length of the channel impulse response, or the channel constraint length, will cause interblock interference (IBI), as the tail of the previous block multicarrier symbol will corrupt the beginning of the current block multicarrier symbol. As a result, the subchannels are not strictly independent of each other in the frequency domain. To solve this problem, prefix-

ing each block with the last few symbols of that block is used. However, the DMT performance becomes unacceptable if the cyclic prefix overhead can no longer eliminate the interblock interference [1].

A FIR time-domain equalizer (TEQ) is used to limit the effective memory of the channel. The purpose of TEQ is not to flatten the channel completely as in a traditional linear or decision feedback equalizer, but rather it is to reduce the length of cyclic prefix to a more manageable number, without sacrificing performance significantly.

Frequency-domain equalizer:

To compensate for phase distortion, a set of "frequency domain equalizers" (FEQ) that consists of N 1-tap complex Least-Mean-Square (LMS) adaptive filters is used following the FFT block before the received signal is decoded. These 1-tap equalizers offer the additional benefit of received signal scaling before decoding. From a practical standpoint, it is convenient to use these 1-tap equalizers to adjust the gain of each FFT output, so that a single decision element can be used for all subchannels regardless of the different subchannel attenuations and different constellation sizes.

A DMT-based ADSL system will use all the available subchannels on short loops because none of the subchannels are attenuated sufficiently to be silenced, unless strong AM radio interference or extremely severe near-end crosstalk is present. On such loops, it is desirable to use even more subchannels, or transmission bandwidth, than the 1.104 MHz transmission bandwidth specified by the current ANSI ADSL standard.

Channel decoding and de-interleaving:

One signal processing technique is using the error correction code and interleaving to mitigate the effect of impulsive noise. Actually, a Reed-Solomon forward error correction code is currently being standardized and implemented in DMT ADSL transceivers. Another signal processing technique that has been proposed is impulse noise cancellation [6].

FIGURE 3. A specific ADSL transceiver.

Although a simplified DMT based ADSL system is shown in Figure 2, a more complete ADSL transmitter and receiver block diagram can be implemented on Digital Signal Processors (DSPs) as detailed in the block diagram in Figure 3 taken from [9].

2-3.4 CAP and Comparison with DMT and QAM

Although the DMT modulation is chosen to be the standard for ADSL, some authors still favour the Carrierless Amplitude and Phase Modulation (CAP) [1]. They claim that the CAP products are tuned to support simultaneous POTS, ISDN, and high speed data, and DMT is tuned to support only high-speed data. Their point of view is seemingly supported by the fact that CAP is currently pre-dominates in the market place and is already in its third generation of chips.

But is it true that CAP is superior to DMT? Has ANSI and ITU made a mistake in choosing DMT for ADSL line coding? A more detailed comparison of DMT and QAM/CAP in terms of performance, equalization and implementation is shown in [8] backing up the DMT scheme.

QAM has emerged as a technique, extensively used in radio and voice band modems. The applicability of QAM to DSL applications might be somewhat surprising because QAM puts the transmission band at frequencies higher than baseband systems do, with consequent greater signal loss and lower crosstalk loss. However, if the effective carrier frequency is low enough, this is apparently a relatively small disadvantage. One reason for this is that most subscriber loops have severe amplitude and phase distortion in the baseband below 20 kHz. This is avoided by a passband system with its lower band-edge above 20 kHz. This makes QAM multitone systems more attractive candidates for DSL applications.

A DMT based system has been described in Section 2-3.2. In this section, a CAP system is briefly explained and then the two systems are compared in a number of different ways.

2-3.4.1 CAP Overview

CAP is also closely related to quadrature amplitude modulation. Mathematically, the two can be considered simple transforms of each other. QAM generates two signals with a sine/cosine mixer and combines them in the analog domain. In contrast, in CAP, orthogonal signal modulation is executed digitally, using two digital transversal bandpass filters with equal amplitude characteristics and a $\pi/2$ difference in phase response (the Hilbert Pair). These signals are combined, and fed to a digital to analog converter (DAC), then transmitted through the twisted pair channel.

The receiver compensates for the channel imperfections by using adaptive equalizers to invert the channel at the receiver end. From a theoretical standpoint, using a single carrier with a DFE at the receiver will lead to the same performance as using the multitone channel [1].

The CAP in reality works without using a carrier signal. Instead, two orthogonal signature waveforms are used to modulate the digital information. Using two signals instead of one doubles the system capacity, but it complicates the design of the receiver equalizer. Figure 4 shows the transmitter and the receivers for the CAP system.

FIGURE 4. The CAP transmitter and receiver

The receiver of the CAP system needs more processing power to invert the channel successfully. The complexities of CAP compared to DMT are shown in Table 1 taken from [1]. It can be clearly seen that the complexity of DMT is significantly less than half of that of CAP.

TABLE 1. Complexity comparison of CAP and DMT

Item	CAP	DMT
100Mbps Complexity: multipliers	26	12
100Mbps Complexity: adders	26	12
3Mbps Complexity: multipliers	7	3
3Mbps Complexity: adders	7	3

2-3.4.2 CAP vs. DMT

The choice of line-code (modulation) method is an engineering decision, and different cases imply different choices. There are applications where single-carrier technology is well suited, and should be considered. However, ADSL access to broadband services such as Internet access, local area network (LAN) access as well as video-conferencing, tele-learning and video-on-demand over the copper network is far better served by DMT. That was recognized by the consensus of experts around the world, and that is the reason the global standards bodies, including ANSI, ETSI and ITU have selected DMT [10].

Although CAP is currently the dominant¹ for ADSL in the market place, CAP modulation is not standardized. It is a single-source proprietary technology. No modem manufacturer using the CAP solution have yet demonstrated interoperability. CAP is a substantially incomplete solution, because it omits the error-correction operations and the framing as well as the management functions which are essential for any complete system. These must be performed by the external modem, therefore increasing its complexity and precluding interoperability.

ANSI spent a lot of efforts in evaluation of CAP vs. DMT for the standardization of ADSL. Various trial results showed that the existing field-proven DMT modems perform significantly better than even the theoretical CAP modem. While CAP modems needed different models to be optimized for either reach or rate, the DMT modems from different manufacturers are rate adapted better - achieving either 100% faster with at least 50% extra reach than one CAP model, or four times the maximum speed with 20% extra reach [14].

1. During the time of writing of this thesis work, more companies are implementing the ADSL transceivers using DMT, making DMT the dominant technology in the market.

In the following sections, various aspects of CAP and DMT are compared to show that the decision made by the international standard committees is correct for the ADSL applications.

Noise Immunity, RFI and Impulses

In principle, on any given channel, the two codes should achieve the same throughput. In practice, differences in the transmitter and receiver architecture as well as implementation limitations imposed by cost constraints affect real-world performance.

In telephone lines, high frequencies get significantly more attenuated than low frequencies and, with loops that have bridge taps in them, certain parts of the frequency band get notched or nulled. So transmitting signals through this channel is difficult in that it treats different frequencies very differently. Another complication is that the channel properties change slowly with time. A DMT transmitter can easily monitor the channel and then adapt its transmission to the characteristics of the telephone line, and continuously updates (bit-swapping) to maintain the optimum transmission. A CAP system cannot modify its transmitter and so it needs to try to undo all the attenuation and notches at the receiver - a tough challenge with channels with deep nulls. As a result, CAP is more sensitive to radio frequency interference (RFI) as well as impulsive noises. Therefore, CAP systems are often described in evaluations as not being as robust as DMT systems, and not as tolerant to bridge taps.

Power Consumption

A CAP device could be optimized for one very specific configuration, but if the versatility to cope with the huge variety of ADSL loops and rates is added, the signal processing load increases markedly. Some of this complexity (e.g. interleaving, or FEC) is not performed by the CAP chipset. However, it must be performed somewhere. The power consumption of this section will depend on implementation such as digital signal processing technology, but the two will be essentially the same.

Another concern for DMT and OFDM is their peak-to-average ratio (PAR). In general, CAP has a lower PAR compared to DMT. Although valid, this comparison ignores the fact that peak-to-average ratio is a ratio and that without knowing the average, no conclusion can be made about the result. A number of digital techniques now exist for gain scaling and peak mitigation that can significantly reduce this, which would reduce the peak power needs.

Power Spectral Density and Spectral Compatibility

The power spectral density (PSD) mask was defined by ANSI T1E1 and ITU for ADSL as a generic technology, and not of any one line code. Unfortunately, the current CAP document and existing implementations do not follow this mask, and breach the agreements [11].

For the upstream, rather than rolling off at 140 kHz, CAP continues up to 180 kHz. That additional 40kHz overlaps directly with other technologies, especially - and fatally - with standard DMT ADSL. As a result, CAP ADSL as currently

defined cannot be deployed in the same binder as ANSI standard ADSL. A similar overlap occurs at the upper end of the downstream band: instead of rolling off at 1.1 MHz this continues to 1.5 MHz - severely interfering with VDSL systems.

The crosstalk resulting from CAP systems renders it completely incompatible with a number of other communication systems, including, as mentioned, existing standard ADSL and VDSL.

Robustness and Error Correction

Virtually every communication system in the world today employs error correction coding, and it is almost inconceivable that any modern system would ignore the benefits from including them. Probably the only exception one could find is CAP chipsets. The existing chipset only supports error correction on the downstream[14].

When comparing data rates, the 6.1 Mbps defined in ANSI T1.413 is net effective payload - the line rate, with overhead, is higher (approximately 6.9 Mbps). In contrast, the 7 Mbps of CAP is a gross rate; the effective data rate will be very much less as all error-correction will need to be subtracted from this. However, since error-correction is not standardized, it is difficult to give a specific answer as to what this will be and it will vary by implementation.

Rate Adaptation

CAP supporters created the new acronym "RADSL" for Rate Adaptive ADSL, and somehow implying that DMT is not rate-adaptive. This strategy may

have achieved its early success in the market place; however, technically, the reverse is the case: DMT is inherently and straightforwardly rate adaptive, and it has always been so.

DMT achieves rate adaptation easily and flexibly. It uses hundreds of degrees of freedom (subchannels) to accomplish it, and delivers the maximum data rate for any given line. CAP can support rate adaptation by only varying the constellation and the bandwidth of a single carrier. This requires very careful analog design, and the rates have much worse gradularity. DMT steps smoothly in 32 kbps steps from 64 kbps to >8 Mbps actual payload, while CAP has coarse and erratic steps from 640 kbps to 7 Mbps. This coarse stepping matters more at lower rates as no service is available when the line condition can only support data rate of less than 640 kbps.

When DMT is not Suitable

Although DMT is substantially superior to CAP in the twisted pair copper environment, there are still times or situations that DMT is not suitable, such as when meeting strict timing and latency requirements is essential in some real-time applications. DMT has greater latency than single carrier technologies; this may actually breach some specifications for particular services such as ISDN, and it is why HDSL selected a single carrier technology.

CAP had a time advantage over DMT, as it did not need to work within the standards process. Now that the standards processes have delivered, and compliant DMT solutions are available, this is no longer relevant.

2-4 G.Lite

G.Lite is a variant of ADSL technology designed for easy, low-cost deployment in the home. With G.Lite technology, no special equipment or wiring is required. Using existing phone lines and ordinary phone jacks, G.Lite modems are as easy to use as traditional analog modems, but deliver transmission speeds over 27 times faster than today's top-of-the-line 56 kbps modems[15].

ADSL comes in two strengths: the industrial strength version, called full-rate ADSL or just ADSL, and a consumer version known as G.Lite that trades off speed for ease of installation[15]. With HDSL and full-rate ADSL, the telephone company must install a special kind of modem in the telephone central office and one in the subscriber's home or business. But for any new high-bandwidth residential service to succeed, it must be easily installable by both the service provider and the consumer. This has led to the development of a new form of DSL, the G.Lite. The key feature of G.Lite is that it does not require the use of a splitter at every home.

Instead of requiring a field technician to install a splitter at the home, G.Lite technology is embedded in the PC modem. These DSL modems are priced comparably with traditional analog modems, and work with existing phone jacks.

The advent of splitterless ADSL, or G.Lite, does not eliminate the need for full-rate ADSL. Rather, G.Lite only offers a logical stepping stone between the analog modems available today and the industrial strength solution of full-rate ADSL. Consumer demand for high-speed Internet access has motivated local exchange carriers to install central office equipment that can handle both full-rate ADSL and the new splitterless version, thus protecting customer and phone car-

rier investment, and reducing the threat of a fractured xDSL industry. In recent months, telephone companies around the world including Bell Atlantic, SBC Communications, USWest, BellSouth, British Columbia Telephone, Deutsche Telekom, and Telecom New Zealand have begun rolling out the G.Lite and full rate ADSL services to their customers [15].

This smooth transition will cause the accelerated deployment of ultrafast Internet access, unleashing a tidal wave of innovative new Web applications, incorporating CD-quality audio and DVD-quality video. The result will be unlimited possibilities for learning, communicating, working and conducting business on the Internet.

G.Lite is not without problems. Because it is not the full strength ADSL and it does not use a splitter, the voice band and ADSL signals are exposed to each other. The effect of the POTS signal can be avoided by using a highpass filter inside an ADSL transceiver. However, recent field trials have shown that without using a POTS splitter a percentage of customers may either experience very limited transmission throughput or audible ADSL interference.

2-5 ANSI T1.413 Standard

American National Standard Institute has setup a standard for ADSL in 1993. This was the ANSI T1.413 standard. Since then, the standard has been updated in 1995 and 1998.

Framing

Only the downstream signal framing (ATU-C transmitter) is shown in this study, for the reason given in Chapter 4.

Data symbols: Up to four downstream simplex data channels and up to three duplex data channels shall be synchronized to the 4 kHz ADSL DMT symbol rate, and multiplexed into two separate data buffers (fast and interleaved). A cyclic redundancy check (crc), scrambling, and forward error correction (FEC) coding shall be applied to the contents of each buffer separately, and the data from the interleaved buffer shall then be passed through an interleaving function. The two data streams shall then be tone ordered as defined in 6.5, and combined into a data symbol that is input to the constellation encoder. After constellation encoding, the data shall be modulated to produce an analog signal for transmission across the customer loop.

Scramblers

The binary data from the interleaved buffers are scrambled without reference to any framing or symbol synchronization. Descrambling in receivers can likewise be performed independent of symbol synchronization.

Forward error correction

R redundant check bytes shall be appended to K message bytes to form a Reed-Solomon code word of size $N=K+R$ bytes. The check bytes are computed from the message byte using the equation

$$C(D) = M(D)D^R \text{ modulo } G(D) \quad (\text{EQ 5})$$

where $M(D)$ is the message polynomial, and $C(D)$ is the check polynomial.

Interleaving

The Reed-Solomon codewords in the interleave buffer shall be convolutionally interleaved. The interleaving depth varies, but it shall always be a power of 2. Convolutional interleaving is defined by the rule:

Each of the N bytes $B_0, B_1, \dots, B_{N-1}$ in a Reed-Solomon codeword is delayed by an amount that varies linearly with the byte index. More precisely, byte B_i is delayed by $(D-1)*i$ bytes, where D is the interleave depth.

Tone ordering

A DMT time-domain signal has a high peak-to-average ratio (its amplitude distribution is almost Gaussian), and large values may be clipped by the digital-to-analog converter. The error signal caused by clipping can be considered as an

additive negative impulse for the time sample that was clipped. The clipping error power is almost equally distributed across all tones in the symbol in which clipping occurs. Clipping is therefore most likely to cause errors on those tones that, in anticipation of higher received SNRs, have been assigned the largest number of bits (and therefore have the densest constellations). These occasional errors can be reliably corrected by the FEC coding if the tones with the largest number of bits have been assigned to the interleave buffer.

The number of bits and the relative gains to be used for every tone are calculated in the ATR-R receiver, and sent back to the ATU-C according to a defined protocol (See 12.9.8 ANSI T1-413). The pairs of numbers are typically stored, in ascending order of frequency of tone number i , in a bit and gain table.

The "tone-ordered" encoding shall assign the first B_F bytes ($8B_F$ bits) from the symbol buffer (See 6.2) to the tones with the smallest number of bits assigned to them, and the remaining B_I bytes to the remaining tones.

The ordered bit table b'_i shall be based on the original bit table b_i as follows:

for $k = 0$ to 15

From the bit table, find the set of all i with the number of bits per tone

$b_i = k$

Assign b_i to the ordered bit allocation table in ascending order of i

A complementary de-ordering procedure should be performed in the ATU-R receiver. It is not necessary, however, to send the results of the ordering process to the receiver because the bit table was originally generated in the ATU-R,

and therefore that table has all the information necessary to perform the de-ordering.

Constellation encoder

For a given sub-channel, the encoder shall select an odd-integer point (X, Y) from the square-grid constellation based on the b bits.

Even values of b

For even values of b , the integer values X and Y of the constellation point (X, Y) shall be determined from the b bits as follows. X and Y are the odd integers with twos-complement binary representation respectively. Figure 5 shows constellations for $b=2$ and $b=4$.

FIGURE 5. Constellation labels for $b=2$ and $b=4$

The 4-bit constellation can be obtained from the 2-bit constellation by replacing each label n by the 2 x 2 block of labels:

$$\begin{array}{cc} 4n+1 & 4n+3 \\ 4n & 4n+2 \end{array} \quad (\text{EQ 6})$$

The same procedure can be used to construct the larger even-bit constellations recursively.

Odd values of b , $b=3$

$b=3$ is a special case and it is shown in Figure 6

FIGURE 6. Constellation labels for $b=3$

Odd values of b , $b>3$

If b is odd and greater than 3, the 2 MSBs of X and the 2 MSBs of Y are determined by the 5 MSBs of the b bits. Figure 7 shows the constellation for the case $b=5$.

FIGURE 7. Constellation labels for $b=5$

As with the even values of b , for larger odd-bit constellations can be constructed recursively by replacing each label n by the 2×2 block of labels of (EQ 6).

Gain scaling

A gain adjuster, g_i is used to effect a frequency-variable transmit power spectral density (PSD). It may have two factors:

- a gross gain adjustment of either 1.414 or 2.0 (i.e., 3 or 6 dB), which may be required for sub-carriers #51 and above
 - a fine gain adjustment with a range of approximately 0.8 to 1.2 (i.e., 0 to 1.5 dB), which may be used to equalize the expected error rates for all the sub-channels.
-

Each point, (X_i, Y_i) , or complex number, $Z_i = X_i + jY_i$, output from the encoder is multiplied by g_i :

$$Z'_i = g_i Z_i \quad (\text{EQ 7})$$

Modulation

Sub-carriers

The frequency spacing, Δf , between sub-carriers shall be 4.3125 kHz, with a tolerance of +50 ppm.

The channel analysis signal allows for a maximum of 255 carriers (at frequencies $n\Delta f$, $n=1$ to 255) to be used. Carrier #64 ($f=276$ kHz) shall be reserved for a pilot ($b_{64}=0$ and $g_{64}=1$).

Cyclic prefix

The last 32 samples of the output of the IDFT (x_k for $k=480$ to 511) shall be prepended to the block of 512 samples and read out to the digital-to-analog converter (DAC) in sequence. That is, the subscripts k , of the DAC samples in sequence are 480...511, 0...511.

CHAPTER 3 *Review of Advances in DSPs*

3-1 ASIC vs. DSP Implementation

Many companies which sell the electronic products do their designs and manufacturing using Application Specific Integrated Circuit, or ASIC.

ASIC has some noteworthy advantages over other ways of implementations:

- It is the fastest way to perform the operation which may be of major concern when digital hardware is needed to keep up with the A/D and D/A sampling rates.
- It gives the cheapest, lightest weight, smallest, lowest power way to do DSP for large volume runs. This is desired if building >1000-million units are planned.

However ASIC disadvantages may outweigh its advantages in some applications if other factors are in need of consideration.

- It can be difficult to design these devices. There are many design tools available to help in the development of these devices, but it is still a formidable and time-consuming task.
- Designs can be difficult to change.
- Most designs will use fixed point arithmetic, limiting their flexibility
- Can be very difficult to debug. Often special hardware must be designed into the ASIC just for testing.
- The learning process is time consuming and knowledge is difficult to master.

In these cases, programmable DSP implementation is a more desirable alternative which many of the difficulties related to ASIC design.

The general-purpose DSP was initially developed to serve as a platform to facilitate new designs and technology, allowing designers to quickly write and modify programs to implement algorithms. This approach is equivalent to building a special purpose computer for manipulating digital signals. The system has a central processing unit called a DSP chip, memory, input/output, and most of the other devices found in a simple general purpose computer.

Advantages of Programmable DSP

- MUCH more flexible than using dedicated hardware
 - Short development time. Often the same hardware can be used for a wide range of applications, all one has to do is write a new program.
-

- Easier to debug. Programs can be stopped and stepped to help in debugging. All modern DSP chips already have special hardware to make debugging easy.
- Typically the DSP program is developed on a standard PC or workstation and downloaded to the DSP chip's memory. The PC can also be used to control the chip and for debugging operations.
- Both floating point and fixed point DSP chips are available.

Disadvantages of Programmable DSP

Despite many of the advantageous qualities that are provided by the programmable DSPs, careful examination must be given to the limitations in implementation of certain applications as follows:

- Slower than ASIC chips. Some algorithms that can be implemented on ASIC can not be used on programmable hardware because there are not enough DSP clock cycles between samples.
- DSP chips can be programmed in a high level language like C, in the assembly language of the chip, or by using some custom software tools. To make the most efficient use of the chip's resources, it is often necessary to use assembly language.

Once the technology and algorithms had been thoroughly tested and verified, a custom ASIC chip would be fabricated to reduce the cost, size, and power consumption of the system. However, with the rapid increase in transistor density, it has become feasible to keep the functionality entirely in the programmable DSP, greatly improving time to market and allowing faster changes and upgrades. The

combination of advances in semiconductor technology and the need for rapid upgrades has transformed cellular phones and voice-band modems into programmable devices, allowing quick upgrades and backward compatibility with older units. Today, nearly all voice-band modems are implemented on a programmable platform. The advantages of using an ASIC for a voice-band modem have become insignificant. In the area of xDSL, the same trend is expected to occur: as DSP-based devices become even faster and more cost-effective, the advantages of DSP-based implementations begin to far outweigh the benefits of custom ASIC designs.

3-2 Floating vs. Fixed Point Implementation

Digital signal processors come in two flavours: floating-point and fixed-point DSPs. With the end uses in mind, there are many factors that designers use to come to a decision when choosing between fixed point versus floating point DSPs. Power consumption, performance, programmability, packaging and price are among the most important determining factors.

As the names imply, the fundamental differences between the two types of devices are their implementations of numeric format. The hardware of fixed-point DSPs uses integer arithmetic, whereas floating point DSPs have the added capability of real arithmetic. With the power of 32-bit-instruction-word width, floating-point devices' numeric format makes it ideal for more complex operation in high-performance applications. The power of 16-bit instruction word of fixed-point DSP dulls in comparison. What fixed-point DSP lacks in instruction power, it makes up

in speed. Its simple architectural difference and fewer speed paths gives it an edge in high-volume, lower-power applications.

Fixed-point DSPs have a smaller address reach to reduce the number of external pins and because of this are used primarily in embedded applications. Its smaller bus size allows for fitting into a small package and consumes relatively little power.

Floating point DSPs are optimized for system-level processing and are less sensitive to cost and require a large address reach. Thus floating-point DSPs have correspondingly large area of memory for greater data and program storage. But on the flip side, this adds up to greater power consumption.

Although the floating-point class DSPs are capable of efficiently performing floating point calculations, most of the real world applications use a fixed point DSP because it is only a fraction of the cost of that of the floating point DSP. Fixed point DSPs are usually capable of more computations compared to the floating point DSP with the same clock frequency.

The fixed-point digital signal processor (DSP) is frequently used instead of floating-point processors because the cost per million instructions per second (MIPS) is lower on the fixed-point version. Fixed-point representation of fractional numbers must be used to program these devices [28].

3-3 Fixed Point Representation

Although the functions required to perform the ADSL can be implemented in many different ways including ASIC design, they can be achieved more cost efficiently using digital signal processors (DSPs).

3-3.1 Fractional Q Formats

Because the fixed point DSPs can only perform integer calculations of a limited range, while most real world applications require non-integer number manipulation, therefore methods of converting from floating point numbers to fixed point numbers are needed. One of these methods is the widely adapted Q.F format for DSP programmers [29].

In a $Q_m.n$ format, there are m bits used to represent the two's complement integer portion of the number, and n bits used to represent the two's complement fractional portion. $m+n+1$ bits are needed to store a general $Q_m.n$ number. The extra bit is needed to store the sign of the number in the most-significant bit position. The representable integer range is specified by $(-2^m, 2^m)$ and the finest fractional resolution is 2^{-n} .

For example, the most commonly used format is Q.15. Q.15 means that a 16-bit word is used to express a signed number between +1 and -1. The most-significant binary digit is interpreted as the sign bit in any Q format number. Thus in Q.15 format, the decimal point is placed immediately to the right of the sign bit. The fractional portion to the right of the sign bit is stored in regular two's complement format.

3-3.2 Q.15 format

Q.15 format, or to be more exact Q0.15 format, places the sign bit at the left most binary digit, and the next 15 bits contain the two's complement fractional component. The approximate allowable ranges of numbers in Q.15 representation is (-1, 1) and the finest fractional resolution is $2^{-15} = 3.05 \times 10^{-5}$.

TABLE 2. Q.15 bit fields

Bit	15	14	13	12	11	10	9	...	0
Value	S	Q14	Q13	Q12	Q11	Q10	Q9	...	Q0

3-3.3 Q.31 format

Q.31 format spans two 16-bit memory words. The 16-bit word stored in the lower memory location contains the 16 least-significant bits, and the higher memory location contains the most significant 15 bits and the sign bit. The approximate allowable range of numbers in Q.31 representation is (-1, 1) and the finest fractional resolution is $2^{-31} = 4.66 \times 10^{-10}$.

TABLE 3. Q.31 low memory location bit fields

Bit	15	14	13	12	...	3	2	1	0
Value	Q15	Q14	Q13	Q12	...	Q3	Q2	Q1	Q0

TABLE 4. Q.31 high memory location bit fields

Bit	15	14	13	12	...	3	2	1	0
Value	S	Q30	Q29	Q28	...	Q19	Q18	Q17	Q16

Although Q.31 format is presented here in this thesis, no Q.31 is used for the experiments during this study. This is due to the fact that the IFFT and FFT routines were optimized for the 16-bit Q.15 represented values.

3-3.4 Q.F Conversion Example

The following conversion rule applies to converting a Q.F number into a fraction

$$N = D \times 2^{-F} \quad (\text{EQ 8})$$

where N is the number in fractional format between $[-1, 1 - 2^{-F}]$ and D is the Q.F number.

For example, if a 16-bit accumulator on a fixed point DSP contained the value 16384, then the corresponding fractional value would be

$$0.5 = 16384 \times 2^{-15} \quad (\text{EQ 9})$$

3-4 C62x DSP Overview

The C62x DSP considered is a 32-bit, fixed-point processor employing a very-long-instruction-word (VLIW) architecture. The processor has two multipliers and six arithmetic logic units (ALUs) operating in parallel. At every clock cycle the processor is able to execute eight instructions. The processing power, measured in MIPS, is therefore eight times the clock frequency of the device. Measured in multiply-and-accumulate operations per second (MACS), the processing power is two times the clock frequency.

In this section, C62x MIPS are used instead of MACS as a measure of complexity. There are two reasons for this. First, a multiplier on a DSP cannot generally be used at every instruction cycle. It is the six ALU units operating in parallel to the two multipliers on the C62x that frequently provide enough parallel processing power to keep both multipliers active at every cycle during multiply-intensive signal processing loops. Second, not all signal processing blocks for a DSL modem are multiply-intensive functions. For these processes, MACS is not a meaningful quantity.

As of the fourth quarter of 1999, existing C62x DSPs run at 250 MHz, providing a total processing power of up to 2000 MIPS. It should be pointed out here that since the number of MIPS required to perform a given task is processor-dependent, different processors may require a different number of MIPS to perform the same task. However, similar types of processors will usually require a similar number of MIPS for the same function.

CHAPTER 4 *Experimental Model and Results*

4-1 Experimental Model

The block diagram for the simulation model used for analysis is shown in Figure 8 with two different amplifiers at both the transmitter end and the receiver end. In this model, it would be useful to generate a rule of thumb to find the automatic gain controller (AGC) A1 and A2 that will produce the best results in the demodulated symbol, thus making a correct decision with or without the error correction coding.

FIGURE 8. Block diagram of the simulation model used on the 'C62x EVM DSP board

4-1.1 Description of Simulation Model

The effect of the amplifiers A1 and A2 is studied. A1 is the linear scaler to be performed on the DMT symbols before the IFFT modulator. A2 is a linear scaler to be used to scale the received signal. Because in this model, a perfect channel ($H_c=1$) is assumed and no quantization at the D/A and A/D conversions is used, the received signal is simply the transmitted signal, i.e. $d_r = d_x$. By independently studying the modulator and demodulator, the effects of quantization errors produced by the modulator/demodulator can be isolated.

The shaded area in Figure 8 denotes the functions performed by the TMS320C62x DSP processor, and blocks which are not shaded are performed in Matlab using data written through files. The values of d_r and d_x , X and Y are saved in files for further analysis using Matlab.

4-2 Simulation Platform and Setup

4-2.1 Experiment Environment

Although there are a number of choices in high speed digital signal processors, such as Texas Instruments' TMS320C62x and TMS320C64x, the platform being used for this thesis is the TI TMS320C6201 Evaluation Module (EVM) board running on a Pentium-based personal computer. The combination of the two provides easy generation of QAM symbols and analysis on the Pentium II computer with Matlab, while all the signal processing functions are actually performed on the TMS320C6201 DSP chip. The data to and from the DSP are piped through

files. The programs for all experiments are written in C programming language and compiled using the TI TMS320C6200 Code Composer Studio tool for optimized assembly code on the target platform. The Code Composer Studio is an integrated development environment for the C62x/67x class of DSP chips. It provides an editor, a compiler, an assembler, a debugger under one user-friendly interface for quick development and testing on the C62x DSPs. The Code Composer Studio also includes convenient file I/O and powerful graphing utilities, as well as tools for real-time performance analysis of the DSP programs. An example of the user interface with the partial driver program used for the simulation is shown in Figure 9.

FIGURE 9. TI Code Composer Studio user interface

FIGURE 10. Data flow diagram

4-2.2 Data Flow Diagram and Functional Description

Figure 10 describes all the stages that the digital data have passed through during the simulation. On the diagram, the words "PC Host" and "DSP" indicate either the operation is performed by the software packages running on the host

PC or on the TMS320C6201DSP processor respectively. The major functional blocks are described in detail.

Data Generator

This block generates random DMT symbols. A DMT symbol consists of M Complex integers, which in turn represent M Complex constellation points. Each constellation point is representative of a b -bit binary word where $b = \log_2 L$, and L is the number of points in the QAM constellation used.

The data generator operates as follows:

1. Generate $l \times (M - 1)$ random numbers uniformly distributed between -0.5 and +0.5, where l is the number of bits assigned to each subchannel.
 2. Pass these random numbers through a hard-limiter. The hard-limiter outputs 1's for positive inputs and 0's for negative inputs. The hard-limiter thus maps
-

the real random numbers of step 1 into a random binary sequence with equal probability of 0's and 1's.

3. The binary sequence of step 2 is now used to generate $M - 1$ Complex constellation points. Each constellation point is a complex integer determined by a l -bit binary word.

The mapping from binary word to QAM symbols is somewhat arbitrary. However, quite often gray coding rules are used in this step.

4. A DMT symbol is made up of M complex integers, the first of which is set to 0, while the remaining points are identical to the sequence generated in step 3.
5. The resulting DMT symbols are written to the file `dmt.txt` as real-imaginary integer pairs.

AGC A1

The Automatic Gain Controller A1 reads one DMT symbol at a time from the file `dmt.txt` on the host PC. It adjusts the amplitudes of M complex integer pairs by multiplying each complex pair by a scale factor $A1$, before they are stored in memory on the DSP. These scaled complex integers are passed to the Hermitian symmetry block for further processing. This factor $A1$ and the other factor AGC A2 are the two most important variables for this study.

Hermitian Symmetry

Hermitian symmetry is an important step in the modulation process. The Hermitian symmetry assures the output of the transmitter consists of real, time domain signals only. This block takes a DMT symbol consisting of M complex

QAM symbols as its input and then it outputs N complex integer pairs where $N = 2 \times M$. The operation of this Hermitian symmetry is straight forward:

1. Copy all M complex pairs of integers representing the scaled DMT symbol to the first M complex pairs of the output.
2. Copy the imaginary part of the first QAM symbol to the real part of $M+1$ th symbol for the output. Then set the imaginary portions of the first and $M+1$ th symbols to 0, making these two numbers real.

Because of the properties of the Data Generator, the first QAM symbol for the DMT symbol is always 0; this step can be simplified: just set the $M+1$ th symbol to 0.

3. Append the complex conjugates of the remaining $M-1$ symbols at the last half of the output. To be exact,

$$Re(S_{out}[M+h]) = Re(S_{in}[M-h]), \text{ and}$$

$$Im(S_{out}[M+h]) = -Im(S_{in}[M-h]) \text{ where } h \text{ is an index between } 1 \text{ and } M-1.$$

4. The output of Hermitian symmetrical complex numbers of size N is passed to the IFFT modulator.

IFFT/FFT Modulator and Demodulator

The Inverse Fast Fourier Transform (IFFT) along with the Fast Fourier Transform (FFT) are at the heart of the DMT based DSL transceiver. The IFFT modulation technique maps the maximum bit allocations in the frequency domain to time domain signals.

Many forms of FFT algorithms can be used to perform the Fourier Transforms. A collection of some of these algorithms is listed and implemented in C pro-

programming language in [20] for floating number computations including Radix-2 and Radix-4 algorithms. Texas Instruments has published a library of FFT routines based on these algorithms but implemented as fixed point calculations and optimized for their TMS320C62x fixed point DSP processors on their FTP site [24]. These FFT routines' inputs and outputs are 16-bit signed integers.

Although the Radix-4 algorithm reduces the computational complexity to almost half compared to Radix-2 algorithm, thus reducing the time required for each transform, it can be only applied with N is an integer value of power of 4 instead of 2. This requirement limits the number of frequency bins that the bandwidth can be divided into and the standard ADSL modulation does not meet this requirement. Therefore the Radix-2 routine was chosen for this study.

The operation of FFT demodulation routine is as follows:

1. Calculate the twiddling factor table according to the formulas:

$$Re(w_i) = \left\lfloor K \times \left(-\cos\left(\frac{2i\pi}{N}\right) \right) \right\rfloor \text{ and}$$

$$Im(w_i) = \left\lfloor K \times \left(-\sin\left(\frac{2i\pi}{N}\right) \right) \right\rfloor$$

where $0 \leq i < N$ and $K = 32767$ for Q.15 representation, and $\lfloor \cdot \rfloor$ denotes the flooring function which chooses the largest integer smaller or equal to the operand.

This step is performed only once and the resulting twiddling factors are stored for reuse for all later FFT and IFFT calculations.

2. A size N real input is transformed via Radix-2 FFT as per program list in Appendix 3.
3. The transformed data after step 2 is in bit-reversed order. Do bit reversal to reorder them in correct order.
4. The results of step 3 are then fed to the complex average block to reduce some large errors based on the Fourier Transform's property of a real series.

The IFFT modulation block uses the same FFT routine with the relationship of IDFT and DFT:

$$IDFT(X) = \frac{(DFT(X^*))^*}{N} \text{ where } X^* \text{ denotes the complex conjugates of } X. \text{ The}$$

size of IFFT N is usually large, e.g., for standard ADSL with 256 frequency bins, $N = 2^{11} = 512$. If the factor $1/N$ is applied at the end of DFT calculation, the intermediate values can easily overflow. Therefore, the IFFT routine uses a progressive division by 2 at every stage of the butterflies instead of a total division of N at the end of the IFFT routine. It is believed that by dividing only 2 at every butterfly, the risk of overflow can be reduced.

Realization of IFFT output

Ideally, the outputs of the IFFT modulator block should consist of real numbers only due to the Hermitian symmetry of the input to IFFT. However, on the DSP implementation used on the fixed-point algorithm, there are some small residues on the imaginary part of the inverse Fourier transformed DMT symbols. Although these residues may be important to recover the original symbols on the DSP implementation, they would take up extra bandwidth for transmission, thus they are undesired for any further processing. Therefore, a block for taking only the real parts of the resulting symbols from the IFFT block is needed as shown in Figure 10. This block just set the imaginary portion of IFFT outputs to zeros. In a real application implementation, this block can be done along with the next block by simply copying the real portions of the IFFT outputs to the transmission buffer, ready for D/A conversion.

Prefix Append/Remove

The N real symbols of each DMT symbols is then prefixed by the last 32 values, forming a string of signals with length of $N+32$, which is the frame for each DMT symbol to be transmitted. These 32 prefixes are used for the synchronization and clock recovery of the receiver circuits. The string of this transmitted DMT signal is recorded into another file `transdata.txt` for analysis of the transmitted signal envelopes.

As shown in Figure 10, after successfully recovering the transmitted symbols, a prefix removal block to strip the prefix is placed after the A/D converter for the receiver.

D/A and A/D Convertors

For this study, all data are in their digital format without ever being converted to analog format. The D/A and A/D converters are shown for the completeness of the data flow. The exception to this is when the quantization effects of the D/A converter were studied. Usually, the number of bits for D/A and A/D converters is limited compared to the precision of the DSP processor, which means that some of the least significant bits are lost when the digital signals are converted to analog form. This is exactly what the operation of the A/D block does when this quantization effect was investigated. The last few significant bits are zeroed out according to the number of bits available to the D/A and A/D converter.

On the receiver end, the A/D converter block is illustrated but not used because there is no actual analog signal used in this simulation. The transmitted signal is simply read back in digital form and multiplied by A^2 .

Complex Averaging

Because the input to the FFT routine is real, the output is ideally Hermitian symmetrical. However, with the non-reversible errors of the fixed-point algorithm in the FFT routine, the result from the FFT block are not exactly Hermitian symmetrical. This is the reason the Complex Averaging block is introduced to take the average value for each pair of the complex conjugates. It is an attempt to reduce the large deviation from the desired demodulated QAM symbol. Later studies showed very little improvement in error analysis with this complex averaging.

This block takes a sequence of complex signals of size N from the FFT block, and produces a size M sequence based on the algorithm described in the previous paragraph. The first symbol is set to 0 based on the modulation scheme (see Data Generator). Then the M QAM symbols are written to files `yreal.txt` and `yimag.txt`. These data preserve the relative errors of the demodulated QAM symbols and are used for further analysis with floating point calculations using software running on the host PC, such as Matlab. These same data are also passed to the hard decision block for a quick symbol error check.

Hard Decision

The results from the complex averaging are divided by the product of A_1 and A_2 on the TMS320C6201 DSP. A hard decision using rounding rules towards the nearest odd integers is made so a quick comparison of the demodulated symbols with the original ones can be done.

4-3 Simulation Simplifications and Assumptions

4-3.1 Channel and Constellation Encoder

A realistic physical channel would require each sub-channel to have its own bit assignment and constellation. This model would require additional blocks to perform these functionalities. To simplify the simulation at this early stage of development, a perfect, non-frequency selective channel is assumed, meaning that the loss for each of the sub-channels will be the same. To simplify the model even further, the channel has been idealized. There is no propagation loss or dis-

tortion, thus the signals being received on the twisted-pair are the same as the ones that are transmitted.

Based on the above assumptions, the constellations used for this simulation can be greatly simplified. Since the channel is chosen to be perfect, the number of bits to be assigned to each sub-channel is also assumed to be a constant. Because the 7- or 8-bit constellations are most common for ADSL transceivers, an 8-bit constellation is used in all experiments except where specified. Odd numbered bit assignments are not used during these experiments due to the additional complexity; therefore, all constellations used are square on the real-imaginary axes.

4-3.2 D/A and A/D Converters

All processing of the data is done with the TMS320C6201 DSP processor; therefore, it is all digital in nature. However, in the real applications employing the xDSL technologies, d_x and d_r must be converted to analog signals, thus requiring D/A and A/D converters. All D/A and A/D converters have manufacturers' specific precision, allowing only a fixed number of bits to be mapped or sampled. To simulate this conversion, a D/A precision block is used to limit the number of the most significant bits that are allowed to be received at the A/D block. Those unused bits are simply zeroed for the experiments.

4-3.3 Omissions

In this simulation model, the error correction coding and interleaver/deinterleaver are ignored. The omissions of these functional blocks are based on the

assumption that the channel was ideal and lossless. Therefore, there are no errors produced other than those introduced by the DSP itself.

4-4 Experimental Results

The experimental setup has been described in Section 4-1 on page 52 and 50. The algorithm used for the IDFT/DFT is the basic Radix-2 routine for fixed point computations. Although other forms of more sophisticated FFT routines [27], such as Radix-4 may be implemented on the TI TMS320C62xx DSP processors, Radix-2 was chosen for its simplicity and compatibility with the ADSL and VDSL standard.

4-4.1 Sources of Errors

The QAM constellation points are assigned at only the odd integer numbers on both the real and imaginary axis [ANSI T1.413], making the distance between any two constellation points greater or equal to two units. When using fixed-point calculations on fixed-point DSPs, these constellations must be scaled up to prevent truncation towards zero. This scaling also reduces the quantization errors which will result in smearing of the demodulated QAM symbols as shown in Figure 11 on page 68. Figure 12 illustrates a demodulated QAM constellation with a greater A1. It can be clearly seen that the constellation is much clearer with A1=512 compared to the case where A1=128.

This spreading of the symbol constellation is caused by the following factors:

1. The first factor is quantization of numerical values during IFFT/FFT modulator/demodulator. The solution to this problem is to scale the constellation by an as large as possible scaling factor;
-

2. The large scaling factor leads to the second error source — overflow. This effect counteracts the first one. Therefore, the scaling factor has to be chosen to be as large as possible, but with an upper limit.
 3. The quantization between the modulator/demodulator, i.e., the A/D converter, also causes the demodulated QAM symbols to spread. This acts as if adding a random noise to the transmitted signals. The clipping effect was investigated by some researchers [18], so it is not repeated here for this thesis.
 4. Forced zeroing of the imaginary parts of the results from the IFFT modulator introduces more error into the system. Theoretically, the imaginary parts of the IFFT block should all be zeros. However, the fixed point calculations divert the result from its theoretical values and produce non-zero imaginary values for Hermitian symmetric data.
 5. Throwing out half of the FFT results can have some implication on the errors. As for the same argument as in point 4, the results for a real-valued series should be Hermitian symmetric. However, the actual experiment suggests otherwise. This is due to the truncation errors occurring during the butterfly stages.
-

FIGURE 11. An example of smearing of QAM constellation on fixed point DSPs

FIGURE 12. Demodulated constellation with $A_1=512$, $A_2=1$

As shown in Figure 12, when the amplification factor A_1 is increased, the demodulated constellation becomes much clearer. Therefore, the AGC A_1 and A_2 have great effects on the DMT based xDSL signal demodulations when fixed-point DSP processors are used.

In the remainder of this chapter, each of these effects will be examined in more detail.

4-4.2 Symbol Error Rate

The QAM symbols used to make up the DMT symbols are formed by using a random number generator on the host PC. To illustrate the different constellations but keep calculations simple, only even numbers of bits are used to form the QAM symbols, i.e., $b=6, 8, 10$, e.t.c., resulting in the QAM constellations to be square for all the experiments.

It is well understood that the multiplication and division calculations with limited precision will produce unacceptable quantization errors when the inputs are too small, and overflow errors occur when inputs are too large. Fixed point DSP processors are not prone to these limitations. Therefore, a compromise must be made between quantization noise and overflow errors. Since all numbers must be represented by integers for the fixed point processors, the quantization effects and overflow become a greater issue. Therefore, proper scaling of the DMT symbols is required before the IDFT modulator and another scaling is required to be performed on the received signals before the DFT demodulation block. What

should the ranges for these scaling factors fall in? The following results identified the values that A1 and A2 should take.

During the simulations, it was noted that A1 is the most influential factor compared to A2; thus for the symbol error rate studies, only the value of A1 is changed while A2 is held constant at 1. A coarse step is also used during this phase of simulation to show the general effects of the scaling factor on the result of IFFT/FFT pairs on fixed point TI TMS320C62x DSPs. These values of the QAM symbol error rates are tabulated in Table 5 and plotted in Figure 13.

TABLE 5. QAM symbol error rate with different A1

Const\A1	16	32	64	128	256	400	512	800	1024	1600	2048
64 pt.	0.1649	0.0142	0.0041	0	0	0	0	0	0	0	0.0091
256 pt.	0.1623	0.0135	0.0041	0	0	0	0	0	0.0110	N/A ^a	0.7965
1024 pt.	0.1552	0.0139	0.0041	0	0	0	0.0116	N/A	0.7927	N/A	N/A

a. N/A, some of the error rates are not calculated due to either data unavailability or error rate so large to be meaningless.

As can be seen from Table 5 and Figure 13, the general trend of symbol error rate of the decoded QAM symbols is to first decrease towards zero while A1 is reduced; then after some critical values, the error rates shoot up. A very interesting point that shows up from this figure is that the lower boundaries for zero-symbol-error are almost invariant, while the upper boundaries of zero-symbol-error are greatly depend on the number of constellation points. This indicates that at lower boundary, the quantization errors are the dominating causes for symbols errors. A large enough factor of A1 must be chosen to reduce, if not completely eliminate the quantization-related QAM symbol errors, no matter what size of the constellation sizes are chosen for the sub-channels. It also proves the hypothesis

that the denser the constellation, the narrower the range of A_1 is for zero-error modulator/demodulator under perfect channel conditions.

FIGURE 13. Symbol error rate due to scale factor A_1

4-4.3 Mean Square Errors

For the symbol error analysis, of most interest is the mean square error (MSE) of X , the demodulated signal before decision, is due to the effect of scaling factors A_1 and A_2 . The two factors that can be applied to the transceiver both influence the mean square error after the FFT demodulator. In general, the greater A_1 and A_2 , the smaller the MSE would be after demodulation. Greater A_1 or A_2 can also cause overflow during the calculations which will cause symbol errors. Therefore, for the purpose of this thesis work, only the MSE for a bit error rate less than 10^{-5} is used at raw data transmission. A much better bit error rate can be achieved if various error correction codes are implemented as specified by the ANSI 1.413 standard.

Although the curves for the symbol error rate can be easily understood, as discussed in Section 4-2.2, the scalars that produce errors above the specified BER is not suitable in xDSL system designs. Therefore, the mean square error of the demodulated constellation points becomes a more important factor in analyzing the performance of the IDFT/DFT modulator/demodulator. During the experiments, the QAM symbols can be correctly decided while the demodulated constellations still clear enough for at least 200 DMT symbols. The correct demodulation of 200 DMT symbols for 256-point QAM constellation translates to a minimum raw bit error rate of 10^{-5} without any error correction coding. In fact, the bit error rate of 10^{-6} or better was achieved during the experiments.

The MSE is defined in (EQ 10) below:

$$MSE = \frac{\sum_m \sum_i |s_i^m - s_{ref}^m|^2}{\sum_m |d_{ref}^m|^2} \quad (EQ 10)$$

where s_i^m is the m^{th} symbol at i^{th} location which is to be decoded to the point at the reference point s_{ref}^m . The value of d_{ref}^m is the minimum distance between any two reference constellation points. When the minimum distance is normalized to 2 as per standard, (EQ 10) can be reduced to

$$MSE = \frac{\sum_m \sum_i |s_i^m - s_{ref}^m|^2}{4m} \quad (EQ 11)$$

where m is the number of QAM points used for calculation.

In Figure 14 (a), the MSE at different values of A1 while holding A2=1 is plotted for the 256-point constellation, i.e. b=8. The values of A1 ranges from 128 to 800. Values of A1 greater than 800 are not plotted because symbol errors started occurring for A1 greater than 800, and the MSE takes no meaningful significance. If the performance is defined as 1/MSE, then it can be seen that the performance improves as A1 increases in Figure 15 on page 77.

TABLE 6. MSE for different A1 and A2

	A2												
	1	2	3	4	5	6	7	8	9	10	11	12	13
A1													
128	1.3741 ^a	0.9808	0.9138	0.8950	0.8851	0.8812	0.8791	0.8776	0.8774	0.8765	0.8766	0.8773	0.8769
256	0.3758	0.2677	0.2493	0.2430	0.2405	0.2393	N/A ^b	N/A	N/A	N/A	N/A	N/A	N/A
350	0.1936	0.1394	0.1302	0.1268	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
500	0.0952	0.0687	0.0642	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
600	0.0666	0.0484	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
700	0.0498	0.0365	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
800	0.0382	0.0280	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

a. Values of MSE are to be multiplied by 10^{-2} .

b. N/A means that symbol errors which occur prevent correctly deciding the demodulated QAM symbols.

(a)

(b)

FIGURE 14. Effect of A1 and A2 on the mean square error

FIGURE 15. Performance with respect to A1

It can be seen that the performance improves at an exponential rate until it hits the critical point where overflow has caused errors in decoding the symbols. This suggests that in fixed point implementation of xDSL using DMT modulation, the scaling factor A_1 should be chosen as large as possible without introducing overflow errors.

The scaling factor A_2 also plays an important role in improving performance and reducing MSE on the decoded symbols. In general, the curve shows that the greater the A_2 is, the less the MSE for the decoded QAM symbols becomes. This can be clearly seen in Figure 14 (b), where MSE vs. A_2 is plotted while holding A_1 constant at 128 for the same 256 point constellation. Thus, a larger A_2 is also desirable. However, from Figure 16, the performance at different values of A_2 , it can be seen that the performance improvement flattens at about $A_2=5$, making larger A_2 unnecessary. This suggests that the incremental improvement diminishes as the value of A_2 becomes greater. Similar to A_1 , A_2 is also bounded. In Figure 14 (b), no values are plotted for $A_1 > 13$. The reason is that the performance deteriorates for A_2 greater than 13, when symbol errors were introduced by overflow during the demodulation process.

FIGURE 16. Performance at different values of A2, with A1=128

Summary

From the experimental results shown earlier, it was noticed that only a number of combinations of A1 and A2 can produce symbol-error-free modulation/demodulation within the specified raw bit-error-rate (BER). The product of maximum value of A2 at any specific point of A1 is approximately constant. In the above case of an 8-bit constellation, the constant is approximately 1700. There are times that A2 must be greater than 1 in order to correctly decode the transmitted DMT signals when the value of A1 is relative small.

This region of valid combinations of A1 and A2 is shown in the shaded area in Figure 17 for the 8-bit constellation. A note should be taken that a lower bound of A2=1 is drawn on the graph. This is because although some fractional values of A2 may produce error-free demodulation for large values of A1. By reducing the transmitted/received signals, more quantization noises are added, resulting in greater MSE. Therefore, no values of A2 less than 1 should be selected. Also note that the boundaries at A1=80 and A1=700 are only approximate values where underflow or overflow already happens in the IFFT modulator block.

To generalize, the maximum value of A2 at any given A1 can be calculated easily once the constant is found through a few experiments.

$$A2_{max} = \frac{C}{A1} \quad (EQ\ 12)$$

where C is a constant based on a specified IDFT/DFT algorithm and the precision of the fixed point implementation.

In the case where $A1$ and $A2$ are integers on fixed-point DSPs,

$$A2_{max} = \left\lfloor \frac{C}{A1} \right\rfloor \quad (\text{EQ 13})$$

where $\lfloor \cdot \rfloor$ denotes the flooring function which chooses the largest integer smaller or equal to the operand.

In Figure 17, besides the blue curve where the upper boundary of $A2$ can be chosen if decimal numbers can be selected, the cyan line represents where only fixed-point calculation is allowed as defined by (EQ 13).

FIGURE 17. Applicable values of combinations of A1 and A2

4-4.4 Envelope Distribution

The distribution of the transmitted signals that are subject to the physical channel distortion is very important. This distribution can be used to help the DMT receiver equalizer to compensate for the channel variation. As noted in several papers [18][25][26], the modulated signals have a Gaussian distribution with centre at zero. The Gaussian distribution of the signals makes it possible to clip the high amplitude signals which have very low probability of occurring. [18] states that by limiting the maximum amplitude of the transmitted signal, the peak power can be reduced with little effects on the bit error rate. However, this must be accompanied by error correction coding. Since there is no error correction incorporated into the simulation model, no conclusion can be reached on this issue for the fixed-point DSP implementation. However, the effects of a limited number of bits used by the A/D and D/A converters will be studied later in Section 4-4.7 on page 93.

For the fixed point implementation used for the experiments, the histogram of the transmitted signal without modification is plotted in Figure 18. The horizontal axis is the magnitude of the signal to be transmitted, and the vertical axis is the relative frequency of occurrences at that magnitude. It is very close to Gaussian distribution with a peak at zero. The peak at zero is caused by the fact that the fixed point algorithm uses the method of truncating instead of rounding during divisions. The truncation causes the signal amplitudes to be reduced toward zero from both positive and negative side, marking a rise of probability of transmitting this signal with magnitude of zero. Although not necessary, this spike at zero may be smoothed by modifying the rounding rules for the digital processing, such as rounding up when the least significant bit is 1. However, these changes in the rounding rules requires more processing power and may not be desirable.

FIGURE 18. The histogram of magnitude of transmitted signal

4-4.5 Error Analysis

Error analysis reveals some insight into the errors produced by the fixed-point DSP and its implementations. In the following sections, unless specified, the errors refer to the distances of the demodulated QAM symbols from their complex means. This deviation from the mean values is a very important factor in soft decision during the demodulation and decoding stages. From the analysis of the properties of the errors, some error-reducing DSP algorithm may be developed. This is subject to further studies. For this purpose, the statistics of the errors are collected and histograms are plotted to aid the presentation of the analysis.

4-4.5.1 Magnitude

One of the most important aspects of the QAM symbol errors is their magnitudes and distributions. Because the absolute magnitudes are directly affected by the amplification factors A_1 and A_2 , which were discussed in previous sections, it is not repeated in this section. However, the distributions of the magnitudes of the errors also present great significance. It may be used for error mitigation purposes if it is at all possible. As can be seen from Figure 19, the histogram of the magnitudes for the symbol errors has a shape of Rayleigh distributions. This indicates that the errors introduced by DSP are randomly distributed but with a concentration a short distance from the centre of the demodulated QAM symbols. This result suggests the quantization errors generated by DSP are Gaussian.

FIGURE 19. Normalized histogram of amplitude of the errors from their means

4-4.5.2 Angle

Another aspect of the errors is the distribution of the angles from their means. In Figure 20, a normalized histogram is plotted for the distribution of error angles on a polar plane for a better illustration. From the data gathered through the experiments, it is noticed that the angles are not uniformly distributed as in the case for the additive white Gaussian noise. If the peaks and valleys are smoothed, the angle distribution is closer to elliptical than circular.

Combining the results from both the amplitude and angle distributions of the errors, it can be said that although the errors are centred around their means, they are skewed and rotated to a certain angle. Although the explanation for this phenomenon requires more investigation, the results are useful for a soft decision rule that might be applied.

4-4.5.3 Frequency Domain Analysis of Errors

The above results for both magnitudes and angles of the errors can be summarized by frequency domain analysis. In Figure 21 on page 89, it is more obvious that the errors from the fixed point DSP processing are very white noise-like across all frequency bands. However, the phase, which is related to the angle of the errors, is not linear. This fact further proves that the noise produced by the fixed point DSP is not white.

FIGURE 20. Normalized histogram of distribution of the error angles

FIGURE 21. Frequency domain analysis of the errors

4-4.6 Signal to Noise Ratio

The magnitudes of the errors lead to the next topic to be discussed, the signal to noise ratio (SNR). The noise power is calculated by squaring the magnitude of the errors. The signal power refers to the average power of the constellation of the QAM symbols to be used. Assuming all points on the QAM symbol constellation have equal probability of occurrence, the average power is the sum of the magnitudes squared of all QAM symbols.

$$P_{avg} = \frac{\sum_{i=1}^M P_i}{M} = \frac{\sum_{i=1}^M |A_{s,i}|^2}{M} \quad (\text{EQ 14})$$

where M is the number of points on the QAM constellation and P_i is the power of the i th symbol, $A_{s,i}$ is the amplitude of the i th QAM symbol from the origin. And the average power of the noise computed based on K QAM symbols is

$$N_{avg} = \frac{\sum_{i=1}^K |A_{e,i}|^2}{K} \quad (\text{EQ 15})$$

The amplitude of error $A_{e,i}$ is the magnitude of i th error as defined in Section 4-4.5 on page 85. The SNR is the ratio of the two quantities

$$SNR = 10 \times \log_{10} \frac{P_{avg}}{N_{avg}} \quad (\text{EQ 16})$$

expressed in dB.

The effects of amplifiers A1 and A2 have on the signal to noise ratio is similar to that of mean square error (MSE) described in Section 4-4.3, because of the direct relationship between MSE and the average power of the errors. This becomes evident in Figure 22 (a), where the curve resembles the plot of performance shown at the beginning of this chapter. Nevertheless, the SNR plot clearly demonstrates that the ratio is better than 50 dB for most of the operating region of amplifier A1 for the 8-bit constellation under ideal conditions stated in Section 4-3, "Simulation Simplifications and Assumptions," on page 64 for the experiments under study. Since the ADSL transceivers require only approximately 20 to 30 dB to compensate for the channel distortion, the fixed-point DSP implementation clearly meets the link margin budgets of the operational requirements.

Figure 22 (b) is the same SNR vs. A1 curve. This time, the amplification factor A1 is plotted on a logarithmic scale. Interestingly, the curve becomes a straight line. This provides an method to accurately estimate for the gain controller to adjust the AGC A1 to achieve the level of SNR required. The maximum improvement of A2 on the SNR curves is approximately 1 dB better than the plotted lines against A1.

(a)

(b)

FIGURE 22. The effects of amplification factor A1 on SNR
(a) A1 on linear scale, (b) A1 on log scale

4-4.7 Effect of Number of Bits Used for D/A Conversion

Ideally, the exact values that are calculated from the DSP processor should be transmitted on to the twisted copper wire. To do so, the digital-to-analog and analog-to-digital converters must have the same level of precision as the DSP processor to handle the greatest value that can be generated by the fixed-point DSP processor. This requires 16-bit D/A and a 16-bit A/D converters for the experimental model that was set up. However, in practice, designers tend to use the least number of bits possible with enough SNR margin, or to choose commercially available off the shelf digital/analog converters. This practice can reduce the total cost of the system. However, it does not come for free. By reducing the precision of the D/A and A/D converters, there is an added quantization noise in the system. In this case, system designers must provide enough signal to noise ratio for the link budget before the total system becomes inoperative. For this reason, it is also important to investigate the effects on the demodulated QAM symbols with a number of different bits being used by the D/A converters.

When there are more bits from the DSP processor compared to the D/A converter, usually the last few significant bits that are simply dropped. To simulate this effect on the TMS320C6201 DSP chipset, the D/A precession block is inserted before the modulated DMT symbols are passed on to the channel. This block shifts the value of the transmitted signal to the right, dropping the last few bits in each value of the DMT symbols according to the number of bit for the D/A converter. On the receiver end, the A/D converter moves the bit back to the left for the same number of shifts. By doing this, the magnitude of the transmitted and received signals will not be much altered from the signals with no shifts. For the perfect channel assumed in these experiments, the shifts done in the converters are equivalent to zeroing the least significant bits.

FIGURE 23. The effect of quantization of A/D-D/A converter on linear scale

FIGURE 24. The effect of quantization of A/D-D/A converter on logarithm scale

The effect of quantization of the transmitted signal at the D/A converter is also plotted. With every bit loss, the SNR reduces between 3 to 4 dB except in the

case for 7 bits plus the sign bit. In this case, the symbols start to break up and more error occurs. This graph also indicates that the selected FFT/IFFT routines can produce enough SNR for the ADSL applications.

4-4.8 DMT Transceiver with Different Sizes of FFT

ANSI T1.413 specifies the number of frequency bins to be $M=256$ for DMT based ADSL; however, with the emerging technologies like VDSL to be standardized and implemented, it is worth investigating the fixed point transceiver with different numbers of frequency partitions, thus different sizes of DFT.

4-4.8.1 Transmitted Signal Power

A well known problem with multitone transmission systems is the high Peak to Average Ratio (PAR) for the transmitted symbols. The PAR is defined as

$$PAR_i = \frac{P_{i,max}}{P_{i,mean}} \quad (EQ 17)$$

where PAR_i is the peak to average power ratio for the i th DMT symbol, $P_{i,max}$ is the maximum instantaneous power to be transmitted during the i th DMT symbol, and the average power for the i th symbol $P_{i,mean}$ is defined as the algorithm average for all data that are transmitted, including the prefixes.

FIGURE 25. Peak to average power ratio as a function of A1

Figure 25 has plotted the effects of amplification factor on the peak to average power ratio. The ratio at each point of A_1 are taken to be the mean of the peak to averages of 200 DMT symbols. The plot includes not only the region where the values of A_1 are valid, it also shows some points where the DMT symbol errors occurred due to overflowing during the transmitting/receiving stages. Even with the increase of the ratio in the region of overflow, the peak to average ratio remained fairly constant with very little variation.

On the other hand, the power of transmitted signals is directly affected by the amplification scaling factor A_1 . Since the power of the transmitted signals is only dependent on A_1 and the size of the IFFT modulation block, A_1 was fixed during each of the comparisons. It was found that when the size of the IDFT doubles, the average of the modulated signal to be transmitted is halved. This is shown in the experimental data collected, and a mathematical proof is given in Appendix B.

The same argument can be applied to the peak power of each of the DMT symbols as well. It was demonstrated that the peak power is also inverse-proportional to the size of FFT. Therefore, the peak to average power ratio is independent of the IFFT size N .

4-4.8.2 Signal to Noise Ratio

The size of FFT and IFFT is directly related to the number of stages in the butterflies required for the computations. With each additional level of the butterflies, there is more quantization noise added; therefore, the SNR is reduced with

the increasing size of FFT and IFFT. In Figure 26, this is clearly demonstrated by the experimental data. Theoretically, the signal to quantization noise is inversely proportional to the size of FFT; thus with every doubling of the size, there is a 3 dB decrease as suggested in [19]. The expression is also given in (EQ 18) for reference here,

$$\begin{aligned} \frac{\sigma_X^2}{\sigma_q^2} &= \frac{1}{2N} \cdot 2^{2b} \\ &= 2^{2b-v-1} \end{aligned} \quad (\text{EQ 18})$$

Therefore, it should be suspected that the SNR should be decreased by 6 dB for the butterflies added at both IFFT and FFT modules. However, the experimental data suggest otherwise: for every increase in the size by a factor of 2, the SNR only drops by approximately little more than 3 dB for the combined effects as shown in Figure 26 on page 100. It was suggested that the 6 dB drop may be seen if random noise is added on the channel before the receiver. However, the reason behind this result requires more investigation in depth. It is subjected to further studies.

Figure 26 also brings up another point with the increase in sizes of the IFFT and FFT. As the size of IFFT/FFT N increases, the suitable region of A1 becomes narrower on the upper limit. This can be explained by the overflow of errors occurring during processing on the transmitter side.

FIGURE 26. Effect of size of DFT on SNR

CHAPTER 5 *Conclusions and Future Works*

5-1 Conclusion

5-1.1 Symbol Error Rate

The QAM symbol error rate is directly affected by the automatic gain controllers. During the course of this study, two scaling factors A1 and A2 are used to simulate these AGCs. Despite the fact that the scaling factor A2, on the receiver side, does contribute to the reduction of overflow and quantization problem for a fixed point DSP implementation of xDSL demodulator, the scaling factor at the transmitter side, which is A1, is still the most dominant factor in the reduction of symbol error rate.

If the scaling factors are too small, the constellation is too close together and the factor of $1/N$ reduces the significant numbers; if the scaling factors are too large, then the DSP could overflow, resulting in symbol errors.

5-1.2 Error and Performance Analysis

Even when the QAM symbol error rate or the bit error rate meets the requirement of xDSL, the quantization errors during the IFFT/FFT still propagate to the resulting demodulated symbols, making each demodulated constellation point smear.

During this study, it was found that the increase of A1 exponentially improves the performance of the transceiver, while the increase of A2 has a diminishing effect on the performance improvement. There are break points for both A1 and A2 when the demodulated QAM symbols are in error.

The analysis of the errors later shows that they are almost like white noise that is added to the system. The non-white characteristic comes from the fact that the distribution of the angles of the errors is skewed.

The signal to noise ratio study proves that the fixed point DSP implementation satisfies the SNR requirement of the DMT based DSL applications. A linear relation between SNR and scaling factors is also found on the logarithm scale.

5-1.3 Envelope Distribution

In theory, the envelope of the transmitted signals should have a Gaussian distribution centred at zero. However, the experimental data showed that although most data agree with this distribution, there is a concentration of zero, which is a direct result of the way fractional numbers are truncated up to their precision.

5-1.4 Other Results

The effect of the digital to analog converter on the SNR for DMT based xDSL transceiver corresponds with the general knowledge that for each bit lost, there is a 3 dB degradation in the SNR ratio.

The size of FFT/IFFT also has effects on the signal envelope and the SNR. Every time the size of FFT/IFFT is doubled, the average power of the transmitted signal after modulation is halved, and the signal to noise ratio drops by approximately 3 dB. The size change, however, does not affect the peak to average power ratio.

5-2 Future Works

The work done for this thesis has laid the foundation for some interesting topics in the works that could follow. They can be in the following areas:

1. Effect of different IFFT/FFT routines that can be implemented on the TMS320C62x DSPs. Only one 16-bit Radix-2 IFFT/FFT routine was used during this study. Other implementations of IFFT/FFT with different precisions can be used to compare the scaling factor ranges and SNR ratios.
 2. Mitigation of the fixed point quantization errors. The quantization errors from the fixed point implementation are non-revertible. However, with error statistics of the error distribution, an equalization algorithm that reduces these errors may be found.
 3. Real-time transceiver. It is proven that DSP implementation of DMT based xDSL application is possible. A real-time ADSL downstream transceiver can be
-

designed for demonstration. However, high speed A/D and D/A converters must be used instead of the ones on the EVM board.

APPENDIX A Quantization Effects in the Computation of DFT

The DFT plays an important role in many digital signal processing applications, including FIR filtering the computation of the correlation between signals, and spectral analysis. For this reason it is important for us to know the effect of quantization errors in its computation. In particular, the effect of round-off errors due to the multiplications performed in the DFT with fixed-point arithmetic is considered in [19].

The model to be adopted for characterizing round-off errors in multiplication is the additive white noise model. Although the statistical analysis is performed for rounding, the analysis can be easily modified to apply to truncation in two's-complement arithmetic. Of particular interest is the analysis of round-off errors in the computation of the DFT via the FFT algorithm. However, the direct computation of the DFT is first established by determining the round-off errors in.

1-1 Quantization Errors in the Direct Computation of the DFT

Given a finite-duration sequence $\{x(n)\}$, $0 \leq n \leq N-1$, the DFT of $\{x(n)\}$ is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad k = 0, 1, \dots, N-1 \quad (\text{EQ 19})$$

where $W_N = e^{-j2\pi/N}$. Assume in general, $\{x(n)\}$ is a complex-valued sequence, and real and imaginary components of $\{x(n)\}$ and $\{W_N^{kn}\}$ are represented by b bits. Consequently, the computation of the product $x(n)W_N^{kn}$ requires four real multiplications. Each real multiplication is rounded from $2b$ bits to b bits, and hence there are four quantization errors for each complex-valued multiplication.

In the direct computation of the DFT, there are N complex-valued multiplications for each point in the DFT. Therefore, the total number of real multiplications in the computation of a single point in the DFT is $4N$. Therefore, there are $4N$ quantization errors.

To evaluate the variance of the quantization errors in a fixed-point computation of the DFT, the following assumptions about the statistical properties of the quantization errors are made.

1. The quantization errors due to rounding are uniformly distributed random variables in the range $(-\Delta/2, \Delta/2)$ where $\Delta=2^{-b}$.
2. The $4N$ quantization errors are mutually uncorrelated.
3. The $4N$ quantization errors are uncorrelated with the sequence $\{x(n)\}$.

Each of the quantization errors has a variance

$$\sigma_e^2 = \frac{\Delta^2}{12} = \frac{2^{-2b}}{12} \quad (\text{EQ 20})$$

the variance of the quantization errors from the $4N$ multiplications is given by

$$\begin{aligned} \sigma_q^2 &= 4N\sigma_e^2 \\ &= \frac{N}{3} \cdot 2^{-2b} \end{aligned} \quad (\text{EQ 21})$$

Therefore, the variance of the quantization error is proportional to the size of DFT. Note that when N is a power of 2 (i.e., $N=2^v$), the variance can be expressed as

$$\sigma_q^2 = \frac{2^{-2(b-v/2)}}{3} \quad (\text{EQ 22})$$

This expression implies that for every fourfold increase in the size N of the DFT, an additional bit is required in computational precision of offset the additional quantization errors.

To prevent overflow, the input sequence to the DFT requires scaling. For this reason, an upper bound on $|X(k)|$ must be less than

$$|X(k)| \leq \sum_{n=0}^{N-1} |x(n)| \quad (\text{EQ 23})$$

If the dynamic range in addition is between -1 and 1, then the condition $|X(k)| < 1$ requires that

$$\sum_{n=0}^{N-1} |x(n)| < 1 \quad (\text{EQ 24})$$

If $|x(n)|$ is initially scaled such that $|x(n)| < 1$ for all stages, then all points in the sequence must be divided by N to ensure that (EQ 24) is satisfied. This condition of (EQ 24) is extremely severe. For example, if the signal sequence $\{x(n)\}$ is white and, after scaling, each value $|x(n)|$ of the sequence is uniformly distributed in the range $(-1/N, 1/N)$, then the variance of the signal sequence is

$$\sigma_x^2 = \frac{(2/N)^2}{12} = \frac{1}{3N^2} \quad (\text{EQ 25})$$

and the variance of the output DFT coefficients $|X(k)|$ is

$$\sigma_X^2 = N\sigma_x^2 = \frac{1}{3N} \quad (\text{EQ 26})$$

Thus the signal-to-noise power ratio is

$$\frac{\sigma_X^2}{\sigma_q^2} = \frac{2^{2b}}{N^2} \quad (\text{EQ 27})$$

From (EQ 27), scaling reduces the SNR by N and the combination of scaling and quantization errors results in a severe penalty on the signal-to-noise ratio in the DFT.

Instead of scaling the input sequence $\{x(n)\}$, if only $|x(n)| < 1$ is required, then a sufficiently large dynamic range for addition such that $|X(k)| < N$ must pro-

vided. In this case, the variance of the sequence $\{|x(n)|\}$ is $\sigma_x^2 = \frac{1}{3}$, and hence the variance of $|X(k)|$ is

$$\sigma_X^2 = N\sigma_x^2 = \frac{N}{3} \quad (\text{EQ 28})$$

Consequently, the SNR is

$$\frac{\sigma_X^2}{\sigma_x^2} = 2^{2b} \quad (\text{EQ 29})$$

1-2 Quantization Errors in FFT Algorithms

As shown in many textbooks, FFT algorithms require significantly fewer multiplications than the direct computation of the DFT. In view of this, one might conclude that the computation of the DFT via an FFT algorithm will result in smaller quantization errors. Unfortunately, this is not the case, as will be demonstrated by the following:

The use of fixed-point arithmetic in the computation of a radix-2 FFT algorithm is under consideration. To be specific, the radix-2, decimation-in-time algorithm for the case $N=8$ is selected. The results on quantization errors were obtained for this radix-2 FFT algorithm are typical of the results obtained with other radix-2 and higher radix algorithms.

As in the case of direct computation, each butterfly computation involves one complex-valued multiplication or four real multiplications. The fact that some butterflies contain a trivial multiplication by ± 1 is ignored. For the butterflies that

affect the computation of any one value of the DFT, in general, there are $N/2$ in the first stage of the FFT, $N/4$ in the second stage, $N/8$ in the third state, and so on, until the last stage, where there is only one. Consequently, the number of butterflies per output point is

$$\begin{aligned}
 (2^{v-1} + 2^{v-2} + \dots + 2 + 1) &= 2^{v-1} \left[1 + \left(\frac{1}{2}\right) + \dots + \left(\frac{1}{2}\right)^{v-1} \right] \\
 &= 2^v \left[1 - \left(\frac{1}{2}\right)^v \right] \\
 &= N - 1
 \end{aligned}
 \tag{EQ 30}$$

The quantization errors introduced in each butterfly propagate to the output. Note that the quantization errors introduced in the first stage propagate through $(v-1)$ stages, those introduced in the second stage propagate through $(v-2)$ stages, and so on. As these quantization errors propagate through a number of subsequent stages, they are phase shifted by the phase factors W_N^{kn} . These phase rotations do not change the statistical properties of the quantization errors and, in particular, the variance of each quantization error remains invariant.

Under the assumption that the quantization errors in each butterfly are uncorrelated with the errors in other butterflies, then there are $4(N-1)$ errors that affect the output of each point of the FFT. Consequently, the variance of the total quantization error at the output is

$$\sigma_q^2 = 4(N-1) \frac{\Delta^2}{12} \approx \frac{N\Delta^2}{3}
 \tag{EQ 31}$$

where $\Delta=2^{-b}$. Hence

$$\sigma_q^2 = \frac{N}{3} \cdot 2^{-2b} \quad (\text{EQ 32})$$

This is exactly the same result that is obtained for the direct computation of the DFT.

The result in (EQ 32) is surprising. In fact, the FFT algorithm does not reduce the number of multiplications required to compute a single point of the DFT. It does, however, exploit the periodicities in W_N^{kn} and thus reduces the number of multiplications in the computation of the entire block of N points in the DFT.

As in the case of the direct computation of the DFT, scaling of the input sequence must be used to prevent overflow. If $|x(n)| < N, 0 \leq n \leq N-1$, then $|X(k)| < 1$ for $0 \leq k \leq N-1$. Thus overflow is avoided. With no reduction of the number of multiplications for each FFT point, the same SNR is obtained as the direct computation.

Since the FFT algorithm consists of a sequence of stages, where each stage contains butterflies that involve pairs of points, it is possible to devise a different scaling strategy that is not as severe as dividing each input point by N . This alternative scaling strategy is motivated by the observation that the intermediate values $|X_n(k)|$ in the $n = 1, 2, \dots, v$ stages of the FFT algorithm satisfy the conditions

$$\begin{aligned} \max[|X_{n+1}(k)|, |X_{n+1}(l)|] &\geq \max[|X_n(k)|, |X_n(l)|] \\ \max[|X_{n+1}(k)|, |X_{n+1}(l)|] &\leq 2\max[|X_n(k)|, |X_n(l)|] \end{aligned} \quad (\text{EQ 33})$$

The total scaling of $1/N$ can be distributed into each of the stages of the FFT algorithm. In particular, if $|x(n)| < \frac{1}{2}$, then the output of each subsequent stage in the FFT algorithm is scaled by $\frac{1}{2}$, so that after v stages an overall scale factor of $\left(\frac{1}{2}\right)^v = 1/N$ can be achieved. Thus overflow in the computation of the DFT is avoided.

This scaling procedure does not affect the signal level at the output of the FFT algorithm, but it significantly reduces the variance of the quantization errors at the output. Specifically, each factor of $\frac{1}{2}$ reduces the variance of a quantization error term by a factor of $\frac{1}{4}$. Thus the $4(N/2)$ quantization errors introduced in the first stage are reduced in variance by $(\frac{1}{4})^{v-1}$, the $4(N/4)$ quantization errors introduced in the second stage are reduced in variance by $(\frac{1}{4})^{v-2}$, and so on. Consequently, the total variance of the quantization errors at the output of the FFT algorithm is

$$\begin{aligned}
 \sigma_q^2 &= \frac{\Delta^2}{12} \left\{ 4\left(\frac{N}{2}\right)\left(\frac{1}{4}\right)^{v-1} + 4\left(\frac{N}{4}\right)\left(\frac{1}{4}\right)^{v-2} + 4\left(\frac{N}{8}\right)\left(\frac{1}{4}\right)^{v-3} + \dots + 4 \right\} \quad (\text{EQ 34}) \\
 &= \frac{\Delta^2}{3} \left\{ \left(\frac{1}{2}\right)^{v-1} + \left(\frac{1}{2}\right)^{v-2} + \left(\frac{1}{2}\right)^{v-3} + \dots + \frac{1}{2} + 1 \right\} \\
 &= \frac{2\Delta^2}{3} \left[1 - \left(\frac{1}{2}\right)^v \right] \approx \frac{3}{2} \cdot 2^{-b}
 \end{aligned}$$

where the factor $(\frac{1}{2})^{-v}$ is negligible.

Observe that (EQ 34) is no longer proportional to N . On the other hand, the signal has the variance $\sigma_X^2 = 1/(3N)$, as given in section 6.4.8 of [19]. Hence the SNR is

$$\begin{aligned} \frac{\sigma_X^2}{\sigma_q^2} &= \frac{1}{2N} \cdot 2^{2b} \\ &= 2^{2b-v-1} \end{aligned} \quad (\text{EQ 35})$$

By distributing the scaling of $1/N$ uniformly throughout the FFT algorithm, an SNR that is inversely proportional to N instead of N^2 is achieved.

APPENDIX B *Effect of DFT Size on Average Power of Signals*

2-1 Average Power in Terms of FFT Size N

If the transmitted signals in $X(k)$ are the IFFT of $x(n)$ and they can be expressed by

$$X(k) = \frac{1}{N} \sum_n x(n) e^{j\frac{2\pi}{N}nk} \quad (\text{EQ 36})$$

and

$$x(n) = \sum_k X(k) e^{-j\frac{2\pi}{N}nk}, \quad (\text{EQ 37})$$

$$\begin{aligned} |x(n)|^2 &= \sum_k \sum_l X(l) X^*(k) e^{-j\frac{2\pi}{N}nl} e^{j\frac{2\pi}{N}nk} \\ &= \sum_k \sum_l X(l) X^*(k) e^{j\frac{2\pi}{N}n(k-l)} \end{aligned} \quad (\text{EQ 38})$$

Therefore,

$$\begin{aligned}
 \sum_n |x(n)|^2 &= \sum_k \sum_l \sum_n X(l) X^*(k) e^{j \frac{2\pi}{N} n(k-l)} \\
 &= \sum_k \sum_l X(l) X^*(k) \sum_{n=0}^{N-1} e^{j \frac{2\pi}{N} n(k-l)}
 \end{aligned}
 \tag{EQ 39}$$

$$\text{because } \sum_{n=0}^{N-1} e^{j \frac{2\pi}{N} n(k-l)} = \begin{cases} N & k-l = 0 \\ 0 & k-l \neq 0 \end{cases},$$

$$\begin{aligned}
 \sum_n |x(n)|^2 &= \sum_l N X(l) X^*(l) \\
 &= N \sum_l |X(l)|^2
 \end{aligned}
 \tag{EQ 40}$$

therefore

$$\sum_l |X(l)|^2 = \frac{1}{N} \sum_n |x(n)|^2
 \tag{EQ 41}$$

and

$$\frac{\sum_l |X(l)|^2}{N} = \frac{1}{N} \left(\frac{\sum_n |x(n)|^2}{N} \right)
 \tag{EQ 42}$$

which means that the average power is inversely proportional to the size of FFT.

APPENDIX C *Program Listing*

This section presents the program listings of the most important blocks used during the course of this thesis. Many of the parameters that to be studied are either explicitly defined at the top of the programs or they can be inputted during run time.

3-1 Transceiver

```
#define Q          0xFFFFA
#define NN          256 /*512*/ /* number of point of the FFT */
#define HalfN      NN/2    /* number of frequency bins */
#define K          32767    /* for Q.15 conversion */
#define PREFIX0    /*32*/   /* length of prefix used */

#define AMPLIFY    1
#define TX         1

/* Run-time Support (RTS) Library Include Files
*/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

/* Math Function Include file */
#define _TI_ENHANCED_MATH_H 1

#include <math.h>
#include <mathf.h>
#include "complex.h"
```

```

/* EVM Driver Include Files */
#include <common.h>
#include <board.h>

#define PRN_DEBUG1

/* function declearation */
void modulate();
void demodulate();

#pragma DATA_ALIGN(x,512); /* align x on word boundry */
#pragma DATA_ALIGN(w,512); /* align w on word boundry */
COMPLEX x[NN], y[NN];
COMPLEX w[NN];

short index[32];      /* the index used for bit reversal */
short max;            /* the max value after ifft */
int i, j, h;          /* index for looping */
int dummy;            /* dummy variable to probing */
int A1, A2;           /* A1 is the amplifier at the modulate stage
                      A2 is the amplifier at demoulate stage */

/* variables used to write to file */
#if TX
    int txBuffer[PREFIX+NN]; /* the tx buffer */
#endif
int tempXreal[HalfN], tempXimag[HalfN], tempYreal[HalfN], tempY-
imag[HalfN];

void main()
{
    FILE *fidr;

    charinFilename[25];
    doubledelta;
    int aLow, aUp, step;

    /* Initialize tables for FFT digit reversal function */
    digitrev_index(index, NN, 2);

```

```

/* creating the twddling factor table */
delta = 2*PI/NN;
for (i=0; i<NN; i++)
{
    w[i].real=(short)round(K*(-cos(i*delta)));
    w[i].imag=(short)round(K*(-sin(i*delta)));
}

evm_init();

/* Enter the file name for input file.
The file should be formatted short integers written in Real and Imag-
inary pairs
on each line.
*/

/* Show program Loaded */
led_blink(2,200);
#ifdef PRN_DEBUG
    printf("Program Started \n");
#endif

#ifdef AMPLIFY
    printf("Please enter the First Amplifier: ");
    scanf("%d", &A1);

    printf("Please enter lower bound of scaling factor: ");
    scanf("%d", &aLow);
    printf("now upper bound: ");
    scanf("%d", &aUp);
    printf("Enter the step size: ");
    scanf("%d", &step);
#endif

for (A2=aLow; A2<=aUp; A2+=step)
{
    fidr = fopen("dmt_sym.txt","r");
    if(fidr == NULL)
    {
        printf("Cannot open input file. Exiting...\n");
        exit(1);
    }

    #ifdef PRN_DEBUG
    printf("Files are opened, now processing ... \n");
    #endif

```

```

    evm_led_enable(1);

    while(1)
    {
        /* This inner loop reads the data from the file as input for the
        IFFT modulators */
        for (j=0;j<(HalfN);j++)
        {
            if (feof(fidr))
            {
                /* zero padding should there are not enough data availi-
able */
                x[j].real=0;
                x[j].imag=0;
                continue;
            }

            /* read data into the array x in real and imaginary pairs */
            fscanf(fidr, "%hd %hd\n", &(x[j].real), &(x[j].imag));
            #if AMPLIFY
                x[j].real = x[j].real*A1;
                x[j].imag = x[j].imag*A1;
            #endif
        }

        modulate();

        dummy = 0;

        demodulate();

        dummy = 0;

        if (feof(fidr))
        { /* if end of file is reached, then quit reading and exit the
loop */
            break;
        }
    }

    fclose(fidr);
}

evm_led_disable(1);

#if PRN_DEBUG
printf("Finished writing to file\n");

```

```

    #endif
}

void modulate()
{
    /* make Hermitian of the block so IFFT will produce only real number
    */

    /* splitting the DC componet */
    x[HalfN].real=0;
    x[HalfN].imag=0;

    /* copy the complex conjugates of the first half to the last half in
    reverse order */
    for (h=1; h<HalfN; h++)
    {
        x[HalfN+h].real = x[HalfN-h].real;
        x[HalfN+h].imag = -x[HalfN-h].imag;
    }

    /* N point complex IFFT */
    r2_ifft((short *)x, NN, (short *)w);
    bitrev(x, index, NN);

    /* zeroing all the imaginary points, no need if file I/O is used
    instead of real transmission, the max for the sequence is
    also found for nomaliation*/
    max=0;
    for (h=0; h<NN; h++)
    {
        x[h].imag=0;
        if (abs(x[h].real) > max)
            max=abs(x[h].real);
    }

    /* adding prefix and copying only the real part of the signal
    to the buffer for transmission */
    #if TX
    for (h=0; h<PREFIX; h++)
        txBuffer[h] = x[480+h].real;

    for (h=0; h<NN; h++)
        txBuffer[PREFIX+h] = x[h].real;
    #endif
}

void demodulate()
{
    int A;          /* A is the product of A1 and A2 */

```

```

A = A1*A2;
/*remove PREFIX */
for (h=0; h<NN; h++)
{
    x[h].real=(txBuffer[PREFIX+h]&Q) *A2;
    x[h].imag=0;
}

/* doing fft to convert the data back */
r2_fft((short *)x, NN, (short *)w);
bitrev(x, index, NN);

/* forcing the DC to be zero because we know it's 0 */
x[0].real=0;
x[0].imag=0;
y[0].real=0;
y[0].imag=0;

/* Use the hermitian symtry to reduce some errors */
for (h=1; h<HalfN; h++)
{
    x[HalfN-h].real=(x[HalfN-h].real+x[HalfN+h].real)/2;
    x[HalfN-h].imag=(x[HalfN-h].imag-x[HalfN+h].imag)/2;

    /* Do the decisions here, rounding to closest odd integer */
    if (x[HalfN-h].real > 0)
    {
        y[HalfN-h].real = x[HalfN-h].real/A;
        if (!(y[HalfN-h].real%2))/* if even */
        { y[HalfN-h].real++;}
    }
    else
    {
        y[HalfN-h].real = x[HalfN-h].real/A;
        if (!(y[HalfN-h].real%2))/* if even */
        { y[HalfN-h].real--;}
    }

    if (x[HalfN-h].imag > 0)
    {
        y[HalfN-h].imag = x[HalfN-h].imag/A;
        if (!(y[HalfN-h].imag%2))/* if even */
        { y[HalfN-h].imag++;}
    }
    else
    {
        y[HalfN-h].imag = x[HalfN-h].imag/A;
        if (!(y[HalfN-h].imag%2))/* if even */

```

```

        { y[HalfN-h].imag--;}
    }
}

/* prepare to write output to file */
for (h=0; h<HalfN; h++)
{
    tempXreal[h]=x[h].real;
    tempXimag[h]=x[h].imag;
    tempYreal[h]=y[h].real;
    tempYimag[h]=y[h].imag;
}
}

```

3-2 IFFT Routine

```

/* File r2_ifft.c
   Created by Feng Deng on Feb. 29, 2000 to calculate the inverse FFT of
   a sequence of complex numbers using radix 2. This routine was modi-
   fied
   from the FFT routine given in radix2.asm supplied from TI's website.

   Prameters:
   xy      the input and output of the complex numbers. See complex.h for
           defination. In place computation is used.
   n       the size of xy and the IFFT.
   w       twiddle factors used for the IFFT routine. Same as the ones
   for
           FFT.
*/

void r2_ifft(short xy[],short n,short w[])
{
    short n1,n2,ie,ia,i,j,k,l;
    short m, nn;
    short xt,yt,c,s;
    short loop1, loop2, loop3;

    nn=n<<1;                      /* double of n */

    /* complex conj of the input sequence */
    for (k=0; k<n; k++)
    {
        xy[2*k+1]=-xy[2*k+1];
    }

    n2 = n;

```

```

ie = 1;
for (k=n; k > 1; k = (k >> 1) ) {
    n1 = n2;
    n2 = n2>>1;
    ia = 0;
    for (j=0; j < n2; j++) {
        c = w[2*ia];
        s = w[2*ia+1];
        ia = ia + ie;

        for (i=j; i < n; i += n1) {
            l = i + n2;
            xt = xy[2*l] - xy[2*i];
            xy[2*i] = xy[2*i] + xy[2*l];
            yt = xy[2*l+1] - xy[2*i+1];
            xy[2*i+1] = xy[2*i+1] + xy[2*l+1];
            xy[2*l] = (c*xt + s*yt)>>15;
            xy[2*l+1] = (c*yt - s*xt)>>15;
        }
    }
    ie = ie<<1;

    /*half the result at each iteration */
    for (m=0; m<nn; m++) {
        xy[m] = xy[m] >>1;
    }

}

for (k=0; k<n; k++)
{
    xy[2*k+1] = -xy[2*k+1];
}

}

```

3-3 FFT Routine

```

*****
*****/
*
*   TEXAS INSTRUMENTS INC.
*
*   COMPLEX FFT (Radix 2)
*
*   Revision Data: 04/14/97
*

```

```

*  USAGE This routine is C callable and can be called as
*
*  void r2_fft(short xy[],short n,short w[])
*
*  xy[] --- input and output sequences (dim-n) (input/output)
*  n    --- FFT size(input)
*  w[]  --- FFT coefficients (dim-n/2)(input)
*
*  If the routine is not to be used as a C callable function,
*  then all instructions relating to stack should be removed.
*  Refer to comments of individual instructions. You will also
*  need to initialize values for all the values passed as these
*  are assumed to be in registers as defined by the calling
*  convention of the compiler, (refer to the C compiler reference
*  guide.)
*
*  DESCRIPTION
*  This routine is used to compute FFT of a complex sequece
*  of size n, a power of 2, with "decimation-in-frequency
*  decomposition" method, ie, the output is in bit-reversed
*  order. Each complex value is with interleaved 16-bit real
*  and imaginary parts.
*
*  TECHNIQUES
*  1. Loading input xy as well as coefficient w in word.
*  2. Both loops j and i shown in the C code are placed in the
*     INNERLOOP of the assembly code.
*
*  ASSUMPTIONS
*  8 <= n <= 32768
*  Both input xy and coefficient w should be aligned on word
*  boundary.
*  w coef stored ordered is  $k*(-\cos[0*\delta])$ ,  $k*(-\sin[0*\delta])$ ,
*   $k*(-\cos[1*\delta])$ , ... where  $\delta = 2*PI/N$ ,  $k = 32767$ 
*
*  MEMORY NOTE
*  Align xy and w on different word boundaries to minimize
*  memory bank hits.
*
*  CYCLES
*   $\text{Log}(\text{base } 2)n * (4*n/2+7) + 9 + n/4$  (memory bank hits)
*  (= 4225 for n = 256)
*
*****

.global _r2_fft
.text

```

_r2_fft:

*** BEGIN Benchmark Timing ***

B_START:

```

    LMBD.L21,B4,B8; outer loop count calculation
| | SUB.L1XB15,4,A9; copy stack pointer
| | STW.D2B3, *B15--[2]; push return address on stack
| | MV .S2B4,B5 ; XY index setup

    SHL.S2B4,2,B3; calculating reset offset
| | MV .L2XA4,B9; setup ptr for X[i] & Y[i]
| | STW.D1A10, *A9--[2]; push A10 on stack
| | STW.D2B11, *B15--[2]; push B11 on stack

    SHR.S2B4,1,B4; used for inner loop count
| | SUB.D2B3,4,B3; calculating reset offset
| | MV .L1XB4,A7; XY index setup
| | MV .L2XA6,B6; permanent ptr for W[0]
| | STW.D1A11, *A9--[3]; push A11 on stack

    ADDAH.D1A4,A7,A3; setup ptr for X[i+N2] & Y[i+N2]
| | MVK.S131,A1 ; outer loop count calculation
| | STW.D2A13, *B15--; push A13 on stack

    MV .S2B9,B8 ; setup store ptr for X[i] & Y[i]
| | MV .L2B9,B11; permanent ptr for X[0] & Y[0]
| | SUB.L1XA1,B8,A1; outer loop counter
| | STW.D1A12, *A9--[2]; push A12 on stack
| | STW.D2B12, *B15--[2]; push B12 on stack
| | MVK.S11,A2 ; IE = 1

    LDW.D2*B9++[B5],B7; X[i] & Y[i]
| | LDW.D1*A6++[A2],A5; CS = W[IA] & W[IA+1]

    SUB.L2XA2,1,B2; decrement twiddle factor counter
| | LDW.D1*A3++[A7],A13; X[i+N2] & Y[i+N2]
| | MV .L1XB3,A8; move reset offset
| | SUB.S2B4,3,B4; inner loop count

    [!B2] SUBB9,B3,B9; reset load X[i] & Y[i] ptrs
| | STW.D2B10, *B15--[2]; push B10 on stack
| | STW.D1A14, *A9--; push A14 on stack
| | SHL.S1A7,1,A0; calculating reset offset

    [!B2] SUBA3,A8,A3; reset load X[i+N2] & Y[i+N2] ptrs
| | STW.D1A15, *A9; push A15 on stack

```

OUTLOOP:

```

    LDW.D2*B9++[B5],B7;* X[i] & Y[i]
| | [!B2] MV.L2XA2,B2;* reset twiddle factor counter
| | [!B2] LDW.D1*A6++[A2],A5;* CS = W[IA] & W[IA+1]
| | MPY.M2B2,1,B0; move to next iteration

    [B2] SUBB2,1,B2;* decrement twiddle factor counter
| | LDW.D1*A3++[A7],A13;* X[i+N2] & Y[i+N2]

    SUB2.S1XA13,B7,A4; XYT = X[i]-X[i+N2] & Y[i]-Y[i+N2]
| | MV    A5,A10; move to other file
| | ADD2.S2XB7,A13,B7; X[i]+X[i+N2] & Y[i]+Y[i+N2]
| | [!B2] SUBB9,B3,B9;* reset load X[i] & Y[i] ptrs

    MPYHL.M1A4,A5,A14; (C)*(YT)
| | STW.D2B7,*B8++[B5]; XY[i] = X[i]+X[i+N2] & Y[i]+Y[i+N2]
| | [!B2] SUBA3,A8,A3;* reset load X[i+N2] & Y[i+N2] ptrs

    MPYLH.M1A4,A5,A15; (S)*XT
| | LDW.D2*B9++[B5],B7;** X[i] & Y[i]
| | [!B2] MV.L2XA2,B2;** reset twiddle factor counter
| | [!B2] LDW.D1*A6++[A2],A5;** CS = W[IA] & W[IA+1]
| | SUB.L1XB8,A0,A9; set store X[i+N2] ptr
| | MPY.M2B2,1,B0;* move to next iteration
| | MV .S2B4,B1 ; set inner loop counter

    MPY.M1A4,A10,A13; (C)*XT
| | [!B0] SUBB8,B3,B8; reset store X[i] & Y[i] ptrs
| | [B2] SUBB2,1,B2;** decrement twiddle factor counter
| | LDW.D1*A3++[A7],A13;** X[i+N2] & Y[i+N2]
| | [B1]B.S1INNERLOOP; branch to the inner loop

    MPYH.M1A4,A10,A14; (S)*(YT)
| | SUB2.S1XA13,B7,A4;* XYT = X[i]-X[i+N2] & Y[i]-Y[i+N2]
| | MV .L1A5,A10;* move to other file
| | ADD2.S2XB7,A13,B7;* X[i]+X[i+N2] & Y[i]+Y[i+N2]
| | [!B2] SUB.D2B9,B3,B9;** reset load X[i] & Y[i] ptrs

```

INNERLOOP:

```

    SUB.L1A14,A15,A15; Y[i+N2] = C*YT - S*XT
| | MPYHL.M1A4,A5,A14;* (C)*(YT)
| | STW.D2B7,*B8++[B5];* XY[i] = X[i]+X[i+N2] & Y[i]+Y[i+N2]
| | [!B2] SUB.D1A3,A8,A3;** reset load X[i+N2] & Y[i+N2] ptrs
| | MV .S1A9,A12; set store X[i+N2] ptr
| | [!B2] MPY.M2XA2,1,B2;*** reset twiddle factor counter

    SHR.S1A15,15,A11; scale Y[i+N2]
| | ADD.L1A14,A13,A13; X[i+N2] = C*XT + S*YT
| | MPYLH.M1A4,A5,A15;* (S)*XT

```

```

| LDW.D2*B9++[B5],B7;*** X[i] & Y[i]
| [!B2] LDW.D1*A6++[A2],A5;*** CS = W[IA] & W[IA+1]
| [B1]SUB.L2B1,1,B1; decrement inner loop counter
| MPY.M2B2,1,B0;** move to next iteration
| ADD.S2XA12,2,B12; set store Y[i+N2] ptr

[B1]B.S1INNERLOOP; for (i=j; i < N; i += N1)
| SHR.S2XA13,15,B10; scale X[i+N2]
| SUB.L1XB8,A0,A9;* set store X[i+N2] ptr
| MPY.M1A4,A10,A13;* (C)*XT
| [!B0] SUB.L2B8,B3,B8;* reset store X[i] & Y[i] ptrs
| [B2]SUB.D2B2,1,B2;*** decrement twiddle factor counter
| LDW.D1*A3++[A7],A13;*** X[i+N2] & Y[i+N2]

| STH.D1B10,*A12; X[i+N2] = C*YT + S*XT
| STH.D2A11,*B12; Y[i+N2] = C*YT - S*XT
| MPYH.M1A4,A10,A14;* (S)*(YT)
| SUB2.S1XA13,B7,A4;*** XYT = X[i]-X[i+N2] & Y[i]-Y[i+N2]
| ADD.L10,A5,A10;** move to other file
| ADD2.S2XB7,A13,B7;*** X[i]+X[i+N2] & Y[i]+Y[i+N2]
| [!B2] SUB.L2B9,B3,B9;*** reset load X[i] & Y[i] ptrs
| ; INNERLOOP ENDS HERE

| SUB.D1A14,A15,A15;* Y[i+N2] = C*YT - S*XT
| MV .L1A9,A12;* set store X[i+N2] ptr
| MPYHL.M1A4,A5,A14;*** (C)*(YT)
| STW.D2B7,*B8++[B5];** XY[i] = X[i]+X[i+N2] & Y[i]+Y[i+N2]
| SHL.S1A2,1,A2; IE << 1
| MV .L2B11,B9; set up load ptr to X[0] & Y[0]

| ADD.D1A14,A13,A13;* X[i+N2] = C*XT + S*YT
| MPYHL.M1A4,A5,A15;*** (S)*XT
| SUB.L1XB8,A0,A9;** set store X[i+N2] ptr
| SHR.S2XA15,15,B10;* scale Y[i+N2]
| MV .L2B11,B8; set up store ptr to X[0] & Y[0]
| [B1]SUB.D2B1,1,B1;** dec i lp cntr
| SUB.S1A1,1,A1; decrement outer loop counter

| SHR.S1A13,15,A11;* scale X[i+N2]
| ADD.L2XA12,2,B12;* set store Y[i+N2] ptr
| MPY.M1A4,A10,A13;*** (C)*XT
| [A1] B.S2OUTLOOP; branch to next stage
| [!A1]LDW.D2*B15[9],B3; pop B3 off stack
| ADD.L1XA7,B9,A3; set up load ptr to X[N2] & Y[N2]
| MV .D1A7,A0 ; calculating reset offset

| STH.D1A11,*A12; X[i+N2] = C*YT + S*XT
| STH.D2B10,*B12; Y[i+N2] = C*YT - S*XT
| MPYH.M1A4,A10,A10;* (S)*(YT)

```

```

| | SHR.S2B5,1,B5; load index >> 1
| | MV .L1XB6,A6; reset W ptr
| | MV .S1A9,A12;* set store X[i+N2] ptr
| | MV .L2XA2,B2; reset twiddle factor counter

| | LDW.D2*B9++[B5],B7; X[i] & Y[i]
| | LDW.D1*A6++[A2],A5; CS = W[IA] & W[IA+1]
| | SHR.S1A7,1,A7; load index >> 1
| | [!A1]ADD.L1XB15,4,A4; copy stack pointer

| | ADD.L1A10,A13,A13;* X[i+N2] = C*XT + S*YT
| | SUB.S1A14,A15,A15;* Y[i+N2] = C*YT - S*XT
| | [A1]LDW.D1*A3++[A7],A13; X[i+N2] & Y[i+N2]
| | ADD.L2XA12,2,B12;* set store Y[i+N2] ptr

| | SHR.S1A13,15,A11;* scale X[i+N2]
| | SHR.S2XA15,15,B10;* scale Y[i+N2]
| | [B2]SUB.L2B2,1,B2; decrement twiddle factor counter
| | [!A1]LDW.D2*B15++[2], A15; pop A15 off stack

| | STH.D1A11,*A12; X[i+N2] = C*YT + S*XT
| | STH.D2B10,*B12; Y[i+N2] = C*YT - S*XT
| | [!B2] SUB.L2B9,B3,B9; reset load X[i] & Y[i] ptrs
| | [!B2] SUB.S1A3,A8,A3; reset load X[i+N2] & Y[i+N2] ptrs
| | ; OUTLOOP ENDS HERE

B_END:
*** END Benchmark Timing ***

| | LDW.D2*B15++[2], B10; pop B10 off stack
| | LDW.D1*A4++[2], A14; pop A14 off stack

| | LDW.D2*B15++[3], B12; pop B12 off stack
| | LDW.D1*A4++[2], A12; pop A12 off stack

| | LDW.D1*A4++, A13; pop A13 off stack

| | B .S2B3
| | LDW.D2*B15++[2], B11; pop B11 off stack
| | LDW.D1*A4++[2], A11; pop A11 off stack

| | LDW.D1*A4, A10; pop A10 off stack

NOP 4

```

REFERENCES

- [1] A. Shalash, K. K. Parhi, Comparison of discrete multitone and carrierless AM/PM techniques for line equalization, *Circuits and Systems, 1996. ISCAS '96., Connecting the World., 1996 IEEE International Symposium on*, vol. 2, 1996, pp. 560-563.
- [2] J.W. Cook, R.H. Kirkby, M.G. Booth, K.T. Foster, D. Clarke, G. Young, The noise and crosstalk environment for ADSL and VDSL systems, *IEEE Comm. Magazine*, May, 1999, pp. 73-78.
- [3] P.S. Chow, J.M. Cioffi, J.A. Bingham, DMT-based ADSL: concept, architecture, and performance, *High speed Access Technology and Services, Including Video-on-Demand (Digest No. 1994/192), IEE Colloquium on*, 1994, pp. 371-376.
- [4] M. Barton, L. Chang, T.R. Hsing, Performance study of high-speed asymmetric digital subscriber lines technology, *IEEE Trans. on Comm.*, vol. 44, no. 2, Feb. 1996, pp. 156-157.

- [5] J. W. Lechleider, High bit rate digital subscriber lines: a review of HDSL progress", *IEEE JSAC*, vol. 9, no. 6, August 1991, pp. 769-784.
 - [6] C.F. Valenti, K. Kerpez, Analysis of wideband noise measurements and implications for signal processing in ADSL systems, *Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications.'* *IEEE International Conference on*, 1994, pp. 826 -832.
 - [7] J.S. Chow, J.C. Tu, J.M. Cioffi, A discrete multitone transceiver system for HDSL applications, *IEEE JSAC*, vol. 9, no. 6, Aug. 1991, pp. 895-908.
 - [8] D.W. Lin, C.T. Chen, and T.R. Hsing, Video on phone lines: technology and applications", *Proc. of IEEE*, vol. 83, no. 2, Feb. 1995, pp. 175-192.
 - [9] D.C. Jones, Signal processing design for an ADSL high speed equalizer, *Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications', IEEE International Conference on*, 1994, pp. 283-287.
 - [10] W.Y. Chen, The development and standardization of asymmetrical digital subscriber line", *IEEE Communications Magazine*, May 1999, pp. 68-92.
 - [11] M. Barton, M.L. Honig, Optimization of discrete multitone to maintain spectrum compatibility with other transmission systems on twisted copper pairs, *IEEE JSAC*, vol. 13 9, Dec. 1995, pp. 1558-1563.
 - [12] M. Humphrey, J. Freeman, How xDSL supports broadband services to the home", *IEEE Network*, Jan/Feb 1997, pp. 14-23.
 - [13] I. Kalet, The multitone channel, *IEEE Trans. on Communications*, vol. 37, no. 2, Feb. 1989, pp. 119-124.
-

- [14] Aware, Discrete multitone (DMT) vs. carrierless amplitude/phase (CAP) line codes, *Whitepapers*, <http://www.aware.com/technology/whitepapers/dmt.html>
- [15] Aware, G.Lite - making the Internet fast enough for consumers, *Whitepapers*, http://www.aware.com/technology/whitepapers/g_lite.pdf
- [16] L. de Clercq, M. Peeters, S. Schelstraete, T. Pollet, Mitigation of radio interference in xDSL transmission, *IEEE Communications Magazine*, March 2000, pp. 168-173.
- [17] Brian R. Wiese, Jacky S. Chow, Programmable implementations of xDSL transceiver systems, *IEEE Communications Magazine*, May 2000, pp. 114-119.
- [18] D.J.G. Mestdagh, P. Spruyt, B. Biran, Analysis of clipping effect in DMT-based ADSL systems, *IEEE International Conference on Communications*, 1994, vol 1, pp. 293-300.
- [19] J. Proakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd ed. Upper Saddle River, N.J., Prentice Hall. 1996.
- [20] W. Vetterling, S. Teukolsk, W. Press, B. Flannery, *Numerical Recipes Example Book*, Cambridge University Press, Cambridge, 1988.
- [21] R. Strum, D. Kirk, *First Principles of Discrete Systems and Digital Signal Processing*, Addison-Wesley Publishing, 1988.
- [22] P. Vetter, D. Goderis, L. Verpooten, A. Granger, Systems aspects of APON/VDSL deployment, *IEEE Communications Magazine*, May 2000, pp. 66-72.
- [23] P. Odling, B. Mayr, S. Palm, The technical impact of the unbundling process and regulatory action, *IEEE Communications Magazine*, May 2000, pp. 74-80.
-

- [24] Texas Instruments Incorporated, <ftp://ftp.ti.com/>.
- [25] R. Gross, D. Veeneman, SNR and spectral properties for a clipped DMT ADSL signal", *IEEE International Conference on Communications*, 1994, vol. 2, pp. 843-847.
- [26] D. Mestdagh, P. Spruyt, A method to reduce the probability of clipping in DMT-based transceivers, *IEEE Transactions on Communications*, Oct. 1996, pp. 1234-1238.
- [27] He, Shoushen, Torkelson, Mats, Designing pipeline FFT processor for OFDM (de)modulation, *International Symposium on Signals, Systems, and Electronics*, 1998, pp. 257-262.
- [28] T. Yensen, Structures and Interfaces for Voice Over IP, *Master Thesis*, Dept. of Systems and Computer Engineering, Carleton University, 1998.
- [29] Texas Instruments, TMS320C54x DSP Library programmer's Reference, Literature Number: SPRU518, Texas Instruments, April, 2001.
-