Modeling and Simulation of an Asynchronous Digital Subscriber Line Transceiver Data Transmission Subsystem

Elmustafa Erwa

ABSTRACT

Recently, there has been an increase in demand for digital services provided over the public telephone line network. Asymmetric digital subscriber line (ADSL) transmit high bit rate data in the forward direction to the subscriber, and lower bit rate data in the reverse direction to the central office, both on a single copper telephone loop. I implemented a Synchronous Dataflow (SDF) model for an ADSL transceiver's data transmission subsystem in LabVIEW. My implementation enables designers to simulate and optimize different ADSL transceiver designs. The overall implementation is compliant with the European Telecommunications Standards Institute's ADSL specification.

1. INTRODUCTION

With the emergence of the Internet as the cornerstone of communications in this age, the demand for high speed Internet access has only been increasing. Asymmetric digital subscriber lines (ADSL) is one of the technologies that provide high-speed Internet access in residences and offices [1]. It facilitates the use of normal telephone services, Integrated Services Digital Network (ISDN), and high-speed data transmission simultaneously. Hence, bandwidth-demanding technologies, such as video-conferencing and video-on-demand, are enabled over ordinary telephone lines.

ADSL standards use discrete multi-tone (DMT) modulation [2]. DMT divides the effectively bandlimited communication channel into a larger number of orthogonal narrowband subchannels. This allows for maximizing the transmitted bit rate and adapting to changing line conditions.

Designing ADSL systems is inherently complex. However, advances in the digital signal processor (DSP) technology allowed programmable DSP based solutions to replace application-specific integrated circuit based implementations. DSP based

solutions allow rapid upgrades to the system after it is shipped. This necessitates a high-level abstraction of the system that can be reused and shared between product releases.

The Synchronous dataflow (SDF) model of computation [3] is well suited for data-driven, communication, and signal processing applications such as ADSL. Efficient DSP code can be generated from statically scheduled SDF programs [4].

My project contributes a high-level SDF abstraction and corresponding implementation of an ADSL transceiver data transmission subsystem in an electronic design automation tool. Furthermore, my implementation is compliant with the European Telecommunications Standards Institute's ADSL specification [2].

2. BACKGROUND

2.1. ADSL Impairments

The performance of ADSL systems degrades due to severe channel attenuation, intersymbol interference (ISI) and interblock interference (IBI), and other line impairments including crosstalk and additive noise. The effective length of the channel impulse response causes IBI as the tail of the previous ADSL block symbol corrupts the beginning of the current block symbol, hence the name IBI [5].

Crosstalk is caused by electromagnetic coupling due to other signals that are traveling in adjacent or nearby wires. Crosstalk interference increases with frequency. My literature review [6] described these impairments in more detail.

2.2. Discrete Multi-tone Modulation

DMT and the transceiver functions were also explained in detail in my literature survey [6]. DMT partitions a data transmission channel with ISI into a set of orthogonal,

memoryless subchannels, each with its own "carrier" [5]. Data is transmitted through each subchannel independently of other subchannels. Previous research has shown that such a system is capable of transmitting at the highest data rate when allocating more bits and powers to subchannels with higher signal-to-noise ratio (SNR) [7, 8]. DMT modulation is one of many multicarrier techniques including Vector Coding, Structured Channel Signaling, and Discrete Wavelet Multi-tone Modulation.

In ADSL, the number of bits that can be carried by each subchannel ranges from 0 to 15. The number of subcarriers used in the system is 256. The assignment of the number of bits for each subchannel, in the form of a *bit allocation table*, is exchanged between the transmitter and the receiver during *initialization* and *retraining*.

Figure 1 illustrates a DMT based transceiver along with the channel. In a DMT based transceiver, a bit stream is encoded/decoded into a set of Quadrature Amplitude Modulation (QAM) symbols, called *sub-symbols*. Each sub-symbol is a complex number that carries the number of bits determined by the bit allocation table in one subcarrier.

DMT modulation and demodulation are implemented using the Inverse Discrete Fourier Transform (IDFT) and Discrete Fourier Transform (DFT), respectively. To implement an $N/2$ subchannel DMT system, an N size IDFT/DFT is required. The size is doubled by mirroring the data to impose conjugate symmetry in the frequency domain, which results in real-valued signal in the time domain after applying the IDFT. The IDFT and DFT are implemented very efficiently using the well known Inverse Fast Fourier Transform (IFFT) and Fast Fourier Transform (FFT) algorithms.

The other functionalities of the ADSL transceiver also include serial-to-parallel/parallel-to-serial conversion, cyclic extension, digital-to-analog/analog-to-digital conversion, time-domain equalization (TEQ), and frequency-domain equalization (FEQ).

A *T*-tap pre-filter TEQ, is implemented if the channel duration is too long compared to the length of the cyclic extension [9]. TEQ suppresses the radio frequency interference from amateur radio bands, while FEQ compensates for magnitude and phase distortion caused by the channel and the TEQ.

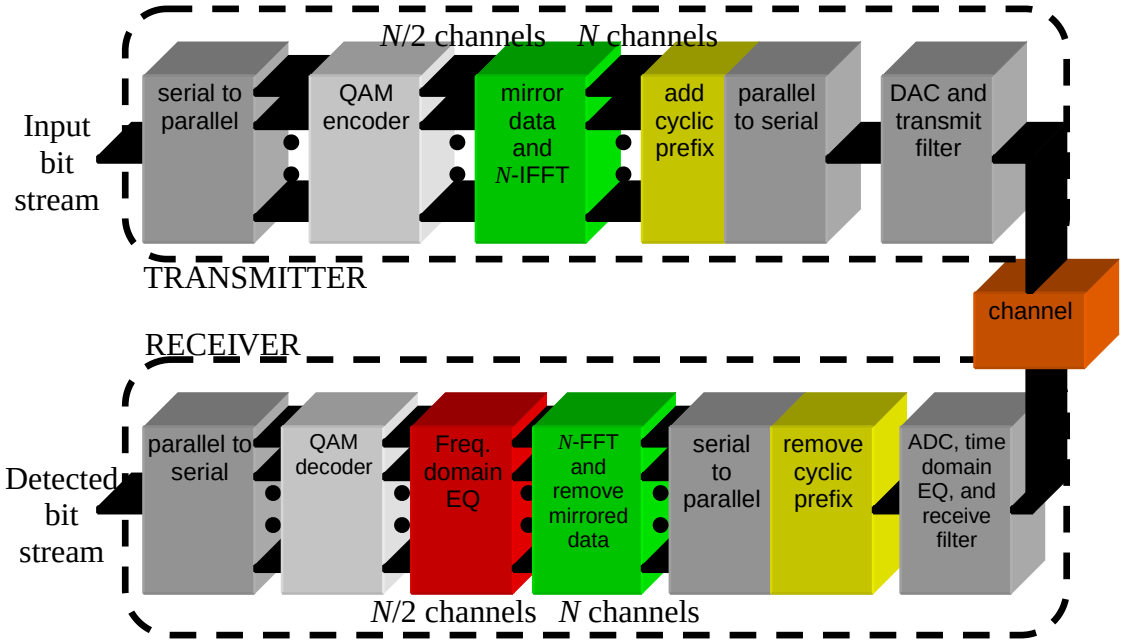


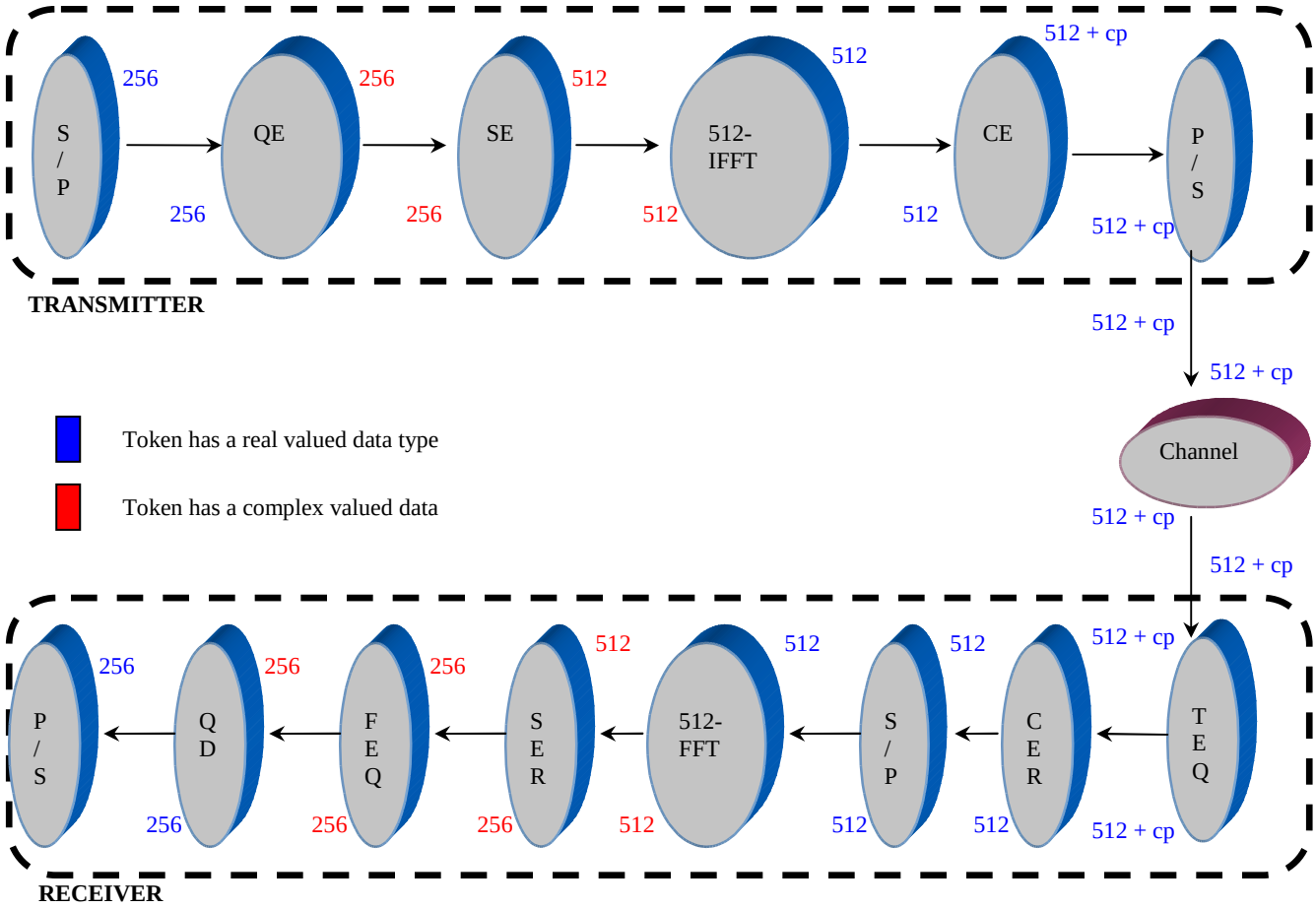
Figure 2. A block diagram for data transmission on an ADSL transceiver [10].

3. MODELING

I use SDF [3] to model the ADSL transceiver blocks as well as the channel, as illustrated in figure 2. SDF is well suited to model signal processing and communication systems in which a stream of data is processed by signal processing functional blocks. Inputs and outputs of the SDF blocks subsystems are the sub-symbols and the data consumption and production in SDF is pre-determined.

The total number of bits encoded/decoded by the QAM encoder/decoder bank is fixed since my implementation does not retrain and dynamically update the bit allocation

table. Each QAM encoder/decoder encodes/decodes a number of bits determined by the bit allocation table. The number of QAM encoders/decoders in the QAM bank is equal to the number of subcarriers used by the system. Figure 3 shows the SDF blocks for an individual QAM encoder/decoder.



S/P: Parallel-to-Serial	CE: Cyclic Extension	SER: SE Remover
QE: QAM Encoder	P/S: Parallel-to-Serial	FEQ: Frequency Domain Equalizer
SE: Conjugate Semmetric Extension	TEQ: Time-domain Equalizer	QD: QAM Decoder
FFT: Fast Fourier Transform	CER: CE Remover	

Figure 2. SDF model for the discretzed ADSL transceiver that uses 256 subcarriers. 'cp' is the size of the cyclic prefix.

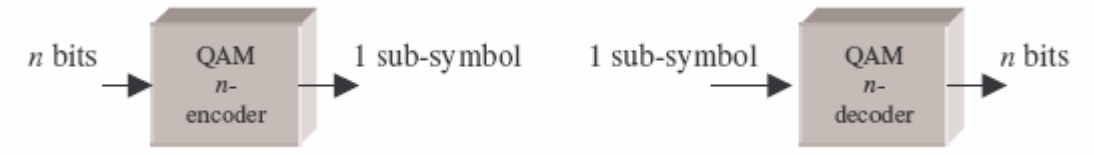


Figure 3. SDF blocks for a QAM encoder and a QAM decoder.

For this project, I chose not to implement the digital-to-analog and analog-to-digital conversion blocks. Instead, I implement a discretized ADSL data transmission subsystem. I made this decision to simplify the modeling and simulation by removing any notion of time. Otherwise, I would have used timed synchronous dataflow to model the digital-to-analog and analog-to-digital conversion blocks along with the channel block.

The ADSL channel is modeled as a Finite impulse Response (FIR) filter. The tokens in the input and the output arcs of the channel's SDF block are real valued as shown in Figure 2. This is because the conjugate symmetric extension produces tokens that are complex valued, which causes the IFFT block to produce tokens, which are real valued.

4. IMPLEMENTATION

I implemented the ADSL data transmission subsystem using LabVIEW. I assumed that the transmitter and receiver were perfectly synchronized. I also assumed that the bit allocation table did not change after initialization.

The design of the system was a top-down design. I implemented a top level *virtual instrument* (VI) that 1) takes in all the user input parameters, such as the channel type and the equalization method; 2) calls the transmitter, channel, and receiver VI's, respectively; and 3) calculates, plots, and displays the bit error rate (BER). Both the transmitter and receiver VI's call the sub VI's that implement each of the blocks shown in figure 2. The QAM encoders/decoders bank was implemented using National Instrument's modulation library components.

The ADSL simulator enables the user to choose one of eight different channel models. Each one of those models one of the eight different carrier service area (CSA) loops that are defined in the ADSL standard. CSA loops have different configurations of bridge taps, gauges, and lengths. These configurations yield different impulse responses once the system has been linearized [11]. Each of these channel models is implemented as a 512-tap FIR filter. I obtained the coefficients of the FIR filters from the MATLAB *Discrete Multitone Time-domain Equalizer* (DMTTEQ) Toolbox that was implemented by the Embedded Signal Processing Lab at the University of Texas [12]. The *linemod* [13] program from Stanford University was used to generate the coefficients. The simulator obtains the coefficients for the desired channel model, from a text file, based on the user's choice.

Furthermore, the simulator allows the user to choose one of five TEQ methods:

1. Minimum mean squared error -- unit energy constraint (MMSE-1) [14];
2. Minimum mean squared error -- unit tap constraint (MMSE-2) [14];
3. Maximum shortening signal to noise ratio (MSSNR) [15];
4. Maximum bit rate (MBR) [16]; and
5. Minimum intersymbol interference (MISI) [16].

To implement the TEQ FIR filter, I added a new function, *teqcoeff*, to the DMTTEQ toolbox. This function returns the values of the coefficients that were produced by the DMTTEQ toolbox. For every TEQ method, I obtained the coefficients that correspond to each of the eight channel models and stored the results in a text file. Each TEQ is implemented as a 16-tap FIR filter. The TEQ VI takes the channel model and TEQ method as inputs then performs a table lookup to obtain the corresponding coefficients.

5. RESULTS AND CONCLUSION

I simulated the data transmission of 10 million bits, to verify the functionality of the transceiver components. I set the signal-to-noise ratio (SNR) to 25 dB. Table 2 shows the results of my simulations for selected channel models and TEQ methods. The lowest bit error rate that I was able to achieve was approximately 10^{-5}. This is two orders of magnitude higher than the ADSL specification requirement of a bit error rate of at most 10^{-7}. The results of my simulation could be greatly improved to meet the specification requirement by implementing error correction codes and dynamic allocation of bits by retraining the transmitter. These tasks could be implemented in the future to further enhance the simulator.

Table 1. Bit error rates for selected channel models and TEQ methods when transmitting 10 million bits. The size of the cyclic prefix is 32 sub-symbols. SNR is 25 dB

Channel	TEQ method	Number of errors	Bit error rate
CSA-1	MBR	136	1.360×10^{-5}
CSA-2	MMSE-2	167	1.670×10^{-5}
CSA-3	MISI	2452	2.452×10^{-4}
CSA-4	MSSNR	2924	2.924×10^{-4}
CSA-5	MBR	164	1.640×10^{-5}
CSA-6	MSSNR	854	8.540×10^{-5}
CSA-7	MMSE-1	1062	1.062×10^{-4}
CSA-8	MSSNR	659	6.59×10^{-5}

The LabVIEW implementation of the discretized data transmission subsystem is a framework for research in ADSL systems. The framework provides a user-friendly simulation and test environments for evaluating ADSL transceivers design.

5. REFERENCES

[1] Thomas Starr, John M. Cioffi, and Peter J. Silverman, *Understanding Digital Subscriber Line Technology* (CD-ROM included), Prentice Hall PTR, 1999, ISBN 0-13-780545-4.

[2] "Transmission and Multiplexing (TM); Asymmetric Digital Subscriber Line (ADSL); Requirements and performance," *ETSI ETR 328 ed.1 (1996-11)*, Technical Specificattion.

[3] E. A. Lee and D. Messerschmitt, "Synchronous data flow", *Proceedings of the IEEE*, vol. 75, no. 9, pp. 1235-1245, Sep. 1987.

[4] E. A. Lee and D. Messerschmitt, "Static scheduling of synchronous data flow programs for digital signal processing", *IEEE transactions on Computers*, vol. C-36, no. 1, pp. 24-35, Jan. 1987.

[5] P. S. Chow, J. M. Cioffi, and J. A. C. Bingham, "DMT-based ADSL: concept, architecture, and performance," *IEE Colloquium on High speed Access Technology and Services*, Oct. 19, 1994.

[6] Elmustafa Erwa "Modeling of an ADSL Transceiver Data Transmission Subsystem", Literature Survey for EE 382C-9 Embedded Software Systems, The University of Texas at Austin, Austin, TX, March 2002.

[7] P. S. Chow, J. M. Cioffi, and J. C. Tu, "A discrete multitone transceiver system for HDSL applications," *IEEE Journal on Selected Areas in Communications*, Volume 9, Issue 6, pp. 895-908, Aug. 1991.

[8] P. S. Chow, J. M. Cioffi, and J. A. C. Bingham, "A practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels," *IEEE Transactions on Communications*, Vol. 43, Issue 2, Part 3, pp. 773-775, Feb.-March-April 1995.

[9] N. Al-Dhahir, and J. M. Cioffi, "Optimum finite-length equalization for multicarrier transceivers," *IEEE Transactions on Communications*, Vol. 44, no. 1, pp. 56-64, Jan. 1996.

[10] G. Arslan, "ADSL Transceivers," Presentation for EE 379K-17 Real-Time Digital Signal Processing Laboratory, The University of Texas at Austin, Austin, TX, November 15, 1999.

[11] M. Milosevic, "Maximizing Data Rate of Discrete Multitone Systems using Time Domain Equalization Design," Ph. D. E. E. dissertation, University of Texas at Austin, Austin, TX, May 2003.

[12] MATLAB *Discrete Multitone Time-domain Equalizer* (DMTTEQ) Toolbox, University of Texas at Austin, **http://www.ece.utexas.edu/~bevans/projects/adsl/dmtteq/dmtteq.html**.

[13] *Linemod*, Stanford Unversity, **http://www.stanford.edu/class/ee379a**.

[14] N. Al-Dhahir and J. M. Cioffi, "Efficiently computed reduced-parameter input-aided MMSE equalizers for ML detection: A unified approach," *IEEE Trans. on Info. Theory*, vol. 42, pp. 903-915, May 1996.

[15] P. J. W. Melsa, R. C. Younce, and C. E. Rhors, "Impulse response shortening for discrete multitone transceivers," *IEEE Trans. on Communications*, vol. 44, pp. 1662-1672, Dec. 1996.

[16] G. Arslan , B. L. Evans, and S. Kiaei, "Equalization for Discrete Multitone Receivers To Maximize Bit Rate," *IEEE Trans. on Signal Processing*, vol. 49, no. 12, pp. 3123-3135, Dec. 2001.