

**High Speed Digital Signal Compensation
on Printed Circuit Boards**

**A Thesis Submitted to the College of
Graduate Studies and Research
in Partial Fulfilment of the Requirements
for the Degree of Master of Science
in the Department of Electrical Engineering
University of Saskatchewan
Saskatoon, Saskatchewan, Canada**

by

Bernard Eugene (Bernie) Boos

PERMISSION TO USE

In presenting this thesis, in partial fulfillment of the requirements for a degree of Master of Science from the University of Saskatchewan, the researcher agrees that the Libraries of this University may make it freely available for inspection. The researcher further agrees that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor(s) who supervised this thesis work. In their absence, this permission can also be given by the Head of the Department, or the Dean of the College of Graduated Studies and Research at the University of Saskatchewan. Any copying, publication, or use of this thesis or parts thereof for financial gain without the researcher's written permission is strictly prohibited by law. Due recognition shall be given to the researcher and to the University of Saskatchewan in any scholarly use which may be made of any material in this thesis.

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

Head of the Department of Electrical Engineering

57 Campus Drive

University of Saskatchewan

Saskatoon, Saskatchewan, Canada

S7N 5A9

University of Saskatchewan

High Speed Digital Signal Compensation on Printed Circuit Boards

Candidate: Bernie Boos

Supervisor: D.E. Dodds

Supervisor: R.J. Bolton

M.Sc. Thesis Submitted to the
College of Graduate Studies and Research

ABSTRACT

In 1965 Gordon Moore postulated that the transistor density of a computer chip would double every year or so, which has held reasonably true up to this point in time. With the increase in transistor density, the data throughput of intra- and inter-chip communication has also increased dramatically, but has currently reached a plateau. The ever-increasing data transfer rates are quickly approaching speeds that challenge the material limits of today's technology. Several problems need to be addressed to significantly increase data transfer rates in chip-to-chip digital communication on printed circuit boards (PCB).

Currently data rates have reached 3.125 Gb/s on standard circuit board interconnect. Until optical circuit board interconnect becomes a reality, current interconnect technology requires improvements to exceed the current throughput limits. These improvements could employ digital or analog designs to compensate for distortions and interferences whose characteristics are well known. The design process is challenging because of the extremely high data rates present in high-speed systems. Both digital and analog devices are believed to have fast enough characteristics to correct for certain signal distortions.

This thesis addresses problems associated with dc wander and crosstalk by employing digital components to provide compensation in the circuit. Two circuits have been designed with simple architectures which are tested with simulations to verify their operation. This thesis gives a detailed account of the design of the two circuits and reports the results of the simulations. The designs demonstrate improved performance over the standard system.

ACKNOWLEDGEMENTS

The researcher wishes to acknowledge and thank his two co-supervisors Dr. Ron Bolton and Prof. David Dodds, and his extremely knowledgeable colleague Eric Pélet. Without their expertise, guidance and advice, the goals would have taken much longer to reach.

The researcher would like to extend his appreciation to the management and staff of TR Labs and in particular the office in Saskatoon, Saskatchewan, where much support and technical help was provided by Garth Wells, Dan Aspel, Jack Hanson, and Trevor Hamm. The researcher's thanks go out to the professors from the Electrical Engineering Department at the University of Saskatchewan that provided needed guidance in finishing this project, namely Dr. Eric Salt and Dr. Brian Daku.

Finally, the researcher would like to thank his friends and family for the support, encouragement, patience, and witty nerd jokes they provided during the high stress times as an engineering undergraduate and during graduate school. Special thanks go out to the author's parents, Peter and Helena Boos, girlfriend Pamela Hildebrandt and friends Adriana, Scott, Celina and Carter Brown. Without your presence and love this would have been a much more difficult and stressful task. Cheers.

TABLE OF CONTENTS

PERMISSION TO USE	i
ABSTRACT	ii
ACKNOWLEDGEMENTS.....	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES.....	viii
LIST OF TABLES.....	xii
LIST OF TABLES.....	xii
LIST OF ABBREVIATIONS	xiii
1 Introduction	1
1.1 Progression from Analog to Digital Communication.....	1
1.2 Baseband Digital Signal Transmission.....	2
1.3 Bandpass Digital Signal Transmission.....	2
1.4 Signal Transmission on Circuit Boards.....	3
1.4.1 Ac Coupling and Dc Wander	4
1.4.2 Crosstalk	6
1.5 Thesis Main Objectives	7
1.6 Thesis Organization.....	9
2 High Speed Digital Communications.....	10
2.1 Transmission Lines.....	10
2.1.1 Microstrip	11
2.1.2 Stripline	13
2.1.3 Transmission Line Modeling.....	14
2.2 High Speed Transmission on PCB Traces.....	15
2.2.1 Reflections and Transmission Line Termination	16
2.3 Thresholds and Noise Margins	18
2.4 Distortion.....	20
2.5 Filters.....	22
2.5.1 Digital IIR Filter	24
2.5.2 Digital FIR Filter	26
2.6 Issues with Digital Circuits	28

2.6.1	Digital Multipliers	28
2.6.2	Digital Adders	30
3	Dc Wander (Intersymbol Interference).....	31
3.1	Possible Solutions to Dc Wander	32
3.1.1	Dc Balanced Signals.....	32
3.1.2	Dc Balancing Bits.....	32
3.1.3	Dc Balanced Codes.....	33
3.1.4	Scrambling Codes.....	34
3.1.5	Baseline Restoration and Decision Feedback.....	36
3.2	Comparison of Dc Wander Solutions.....	37
3.3	Signal Flow in a Decision Feedback System	38
4	Decision Feedback for Dc Wander Compensation.....	40
4.1	Digital Filter Structures	40
4.2	Filter Development.....	40
4.2.1	Infinite Impulse Response (IIR) Filter Design	42
4.2.2	Finite Impulse Response (FIR) Filter Design.....	43
4.3	Simulation Strategy	44
4.4	Simulation Results.....	48
5	Signal Interference (Crosstalk)	60
5.1	Capacitive Coupling	60
5.2	Inductive Coupling	62
5.3	Solutions for Signal Interference (Crosstalk) Reduction.....	65
5.4	Compensating Near End Crosstalk using a Digital Equalizer.....	68
5.5	Compensating Near End Crosstalk using 2-Bit Selection Circuit.....	70
5.6	Compensation Circuit Design Considerations.....	71
6	Digital Circuits for Crosstalk Compensation.....	73
6.1	System Parameters.....	73
6.2	Edge Detection	74
6.3	Digital Equalizer Design	74
6.4	2-Bit Selection (NEXT Compensation) Design	78
6.5	Simulation Strategy	82

6.6	Simulation Results.....	84
7	Conclusions and Future Work	91
7.1	Summary.....	91
7.2	Results	91
7.3	Future Work.....	93
	References	94
A.	Calculations.....	96
A.1	Digital IIR Filter	96
A.2	Digital FIR Filter	97
A.3	Energy per Bit to Noise Spectral Density Ratio (E_b/N_0)	98
B.	Simulink® Simulation Blocks and Parameters	99
B.1	Bipolar Signal Transmitter	99
B.2	Ac Coupling.....	99
B.3	Analog to Digital Converter	100
B.4	Transmission Line Simulation.....	100
B.5	Additive White Gaussian Noise (AWGN) Channel.....	101
B.6	Receiver.....	101
B.7	Digital Filter for Decision Feedback.....	102
B.8	Discrete-Time Eye Diagram Scope.....	102
B.9	Error Rate Calculation and Display.....	103
B.10	Edge Detector Block.....	103
B.11	Crosstalk Equalizer.....	104
B.12	2-bit Compensation Circuit	105
B.13	Generation of Crosstalk Interference.....	105
C.	Results Tables	107
C.1	Decision Feedback Results Tables	107
C.2	Crosstalk Compensation Results Tables	110
D.	Program Code and Block Diagrams	112
D.1	Matlab® Code	112
D.1.1	Transmission Line Modeling (tlinemodel.m).....	113
D.1.2	Calculating Filter Tap Multipliers (tapmultipliers.m).....	116

D.1.3	Calculating Crosstalk Parameters (crosstalkparams.m)	119
D.2	Simulink® Block Diagrams of Complete Systems	120
E.	Circuit Diagrams from Quartus II®	123

LIST OF FIGURES

Figure 1.1	Eliminating dc offset using ac coupling.	4
Figure 1.2	Transmitted signal (top) and ac coupled signal (bottom).	5
Figure 2.1	Equivalent circuit of an incremental length of a 2 conductor transmission line.	10
Figure 2.2	Microstrip transmission line. (a) Geometry. (b) Electric and magnetic field lines [7].	12
Figure 2.3	Stripline transmission line. (a) Geometry. (b) Electric and magnetic field lines [7].	13
Figure 2.4	Slow rise time (top) and fast rise time (bottom) on unterminated transmission line [5].	17
Figure 2.5	Incident signal being reflected by a mismatched load [6].	18
Figure 2.6	Example of noise margins [5].	19
Figure 2.7	Example of an eye diagram.	20
Figure 2.8	(a) Original spectrum of a square pulse and resulting ladder (rectangular eye) diagram and (b) spectrum and eye diagram after loss of high frequency component [10].	21
Figure 2.9	Spectrum and eye diagram from a (a) loss of high frequency component due to transmission line effects and (b) loss of low frequency due to ac coupling [10].	22
Figure 2.10	Basic digital filter building blocks.	22
Figure 2.11	Block diagram of a digital filter with recursive and non-recursive parts.	23
Figure 2.12	Typical form of IIR filter.	24
Figure 2.13	(a) Discrete impulse, (b) highpass impulse response for IIR filter.	25
Figure 2.14	First order IIR (recursive) digital filter.	26
Figure 2.15	Simple FIR (non-recursive) digital filter.	27
Figure 2.16	(a) Discrete impulse, (b) highpass impulse response for an FIR filter.	27
Figure 2.17	Increasing accuracy of BFP multiplier using the concept of <i>bit shifting</i>	29
Figure 2.18	A 16-tap equalizer to illustrate the number of adders needed.	30
Figure 3.1	Example of a step input response on an ac coupled transmission line.	31
Figure 3.2	ISDN local networking frame format [4].	33

Figure 3.3	Binary signaling formats.	34
Figure 3.4	SONET scrambler/descrambler.	35
Figure 3.5	Baseline restoration for unipolar codes [2].	36
Figure 3.6	Structure of decision feedback equalizer.	36
Figure 3.7	Decision feedback equalization [2].	39
Figure 4.1	Unit step input responses (a) highpass filter, (b) lowpass filter, (c) sum of step input responses of (a) and (b).	41
Figure 4.2	First order lowpass IIR digital filter.	42
Figure 4.3	FIR digital filter with $M = 6$	43
Figure 4.4	Theoretic BER for polar (NRZ), unipolar (NRZ), and bipolar line codes [2].	46
Figure 4.5	Transmission line magnitude and phase response (1 meter).	48
Figure 4.6	(a) Spectrum of 320 ns pulse, (b) magnitude response of 1 meter of stripline transmission line in FR4 circuit board, and (c) spectrum of 320 ns pulse after traveling through 1 meter transmission line.	49
Figure 4.7	Time domain representation of pulse with 90 ns rise time at transmitter and showing attenuation after traveling through a transmission line.	50
Figure 4.8	Spectrum of a pseudo-random polar NRZ signal (a) with 90 ps rise/fall times and (b) after ac coupling and propagating through 20 cm of (stripline) transmission line.	51
Figure 4.9	Eye diagram from simulation affected by dc wander and transmission line effects.	52
Figure 4.10	Eye diagrams of non ac-coupled signal attenuated by transmission line effects and added AWGN of (a) $E_b/N_0 = 14$ dB, (b) $E_b/N_0 = 12$ dB, (c) $E_b/N_0 = 10$ dB and (d) $E_b/N_0 = 8$ dB.	53
Figure 4.11	Eye diagrams of ac coupled, unbalanced ($P_{zero} = 0.3$) signal with added noise ($E_b/N_0 = 14$ dB). (a) Time constant ($\tau = 100T$) uncompensated and (b) signal compensated with digital IIR filter. (c) Time constant ($\tau = 200T$) uncompensated and (d) signal compensated with digital IIR filter.	54
Figure 4.12	Eye diagrams of unbalanced signal ($P_{zero} = 0.3$). (a) Uncompensated ($\tau =$ $100T$) and (b) uncompensated ($\tau = 200T$). (c) Signal compensated with 10-	

	tap FIR filter ($\tau = 100T$) and (d) compensated with 10-tap FIR filter ($\tau = 200T$). (e) Signal compensated with 50-tap FIR filter ($\tau = 100T$) and (f) compensated with 50-tap FIR filter ($\tau = 200T$).	55
Figure 4.13	Bit error rate (BER) for dc wander compensation with balanced signal ($P_{zero} = 0.5$).	57
Figure 4.14	Bit error rate graph for dc wander compensation system with unbalanced signal ($P_{zero} = 0.4$).	58
Figure 4.15	Bit error rate graph for dc wander compensation with unbalanced signal ($P_{zero} = 0.3$).	59
Figure 5.1	Behaviour of NEXT and FEXT generated on a transmission line from capacitive coupling [5].	61
Figure 5.2	Behaviour of NEXT and FEXT generated on a transmission line from inductive coupling [5].	63
Figure 5.3	(a) Transmitted signal generates (b) NEXT on neighbouring line. (c) Receive signal is (d) affected by crosstalk.	64
Figure 5.4	Odd and even mode electric and magnetic field patterns for a simple two-conductor system [6].	68
Figure 5.5	Crosstalk compensation using digital feed-forward circuit.	69
Figure 5.6	Block diagram for proposed NEXT compensation circuit.	71
Figure 6.1	Edge detection circuit for proposed compensation schemes.	74
Figure 6.2	Geometry and topology of dual stripline configuration.	76
Figure 6.3	Impedance values versus trace separation for odd (Z_{oo}), even (Z_{oe}), and single line characteristic impedance (Z_o) of stripline interconnect [18].	76
Figure 6.4	Reverse voltage crosstalk coefficient α_R as a function of stripline separation [18].	77
Figure 6.5	(a) Near end crosstalk (NEXT) tri-state circuit and (b) state machine.	79
Figure 6.6	NEXT compensation circuit (I).	80
Figure 6.7	NEXT compensation circuit (II).	81
Figure 6.8	Timing diagram for crosstalk compensation circuit in Figure 6.6.	86
Figure 6.9	Timing diagram for crosstalk compensation circuit in Figure 6.7.	86

Figure 6.10	Eye diagrams without AWGN showing (a) signal affected by crosstalk, (b) unsuccessful compensation of crosstalk, (c) partial compensation, and (d) full compensation.	87
Figure 6.11	Bit Error Rate (BER) graph showing incoming ac coupled signal with crosstalk, with and without compensation.	89
Figure 6.12	Bit Error Rate (BER) graph for non ac coupled signal showing crosstalk with and without compensation.....	90
Figure B.1	Transmitter block and contents in simulations.....	99
Figure B.2	Ac coupling block for simulations.	99
Figure B.3	The A-D conversion block (left) contains a sampled quantizer encode block (right).....	100
Figure B.4	Additive white gaussian noise block for simulations.....	101
Figure B.5	Simulink [®] blocks (a) downsample, (b) integer delay, (c) signum, and (d) complex to real conversion.....	102
Figure B.6	Eye diagram scope block used in simulations.....	103
Figure B.7	Error rate calculation and display blocks.	103
Figure B.8	Edge detection block.	103
Figure B.9	Crosstalk equalizer block diagram for simulation in Simulink [®]	104
Figure B.10	Edge detection and delay simulation circuit for 2-bit adder.....	104
Figure B.11	2-bit crosstalk "adder" simulation circuit.....	105
Figure B.12	Crosstalk generator for Simulink [®] simulations.....	106
Figure D.1	Waveform generator, with ac coupling block and A-D conversion.....	120
Figure D.2	Complete circuit for dc wander compensation with decision feedback.....	121
Figure D.3	Crosstalk compensation circuit with equalizer technique.	121
Figure D.4	Crosstalk correction using 2-bit selection compensation technique.....	122
Figure E.1	Crosstalk compensation simulation for functionality and timing (I).....	124
Figure E.2	Crosstalk compensation simulation for functionality and timing (II).	125

LIST OF TABLES

Table 4.1	Simulation results demonstrating increases in data depending on E_b/N_0 level.	47
Table 5.1	Digital 2-bit values representing levels of NEXT.	70
Table 6.1	State table for 2-bit selection circuit.	79
Table 6.2	Crosstalk simulation results demonstrating increases in data used depending on E_b/N_0 level.	83
Table 6.3	Propagation delays through 2-bit selection circuit.	85
Table C.1	Dc wander compensation with balanced signal ($P_z = 0.5$).	107
Table C.2	Dc wander compensation with unbalanced signal ($P_z = 0.4$).	108
Table C.3	Dc wander compensation with unbalanced signal ($P_z = 0.3$).	109
Table C.4	Crosstalk compensation circuit with ac coupled data.	110
Table C.5	Crosstalk compensation circuit with non ac coupled data.	111

LIST OF ABBREVIATIONS

ac	Alternating Current
A-D	Analog to Digital
AMI	Alternate Mark Inversion
AWGN	Additive White Gaussian Noise
BBG	Bernoulli Binary Generator
BER	Bit Error Rate
BFP	Block Floating Point
BLT	Bilinear Transform
BPRZ	Bipolar Return-to-Zero
CMOS	Complimentary Metal-Oxide Semiconductor
dB	Decibel
dc	Direct Current
DSL	Digital Subscriber Line
E_b/N_0	Energy per bit (joules) to noise spectral density (watts per hertz)
EM	Electromagnetic
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FEXT	Far End Crosstalk
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FR4	Common type of printed circuit board material
Gb/s	Giga-bits per second (Giga = 10^9)
HPF	Highpass Filter
IC	Integrated Circuit
IIR	Infinite Impulse Response
ISDN	Integrated Services Digital Network
ITU-T	International Telecommunications Union
in	Inch
kb/s	Kilo-bits per second (kilo= 10^3)
LAN	Local Area Network
LPF	Lowpass Filter

LSB	Least Significant Bit
LUT	Look-up Table
LVDS	Low Voltage Differential Signaling
mil	10^{-3} inches
MSB	Most Significant Bit
NEXT	Near End Crosstalk
NMOS	N-type Metal-Oxide Semiconductor
NRZ	Non Return-to-Zero
ns	Nanosecond (Nano = 10^{-9})
PCB	Printed Circuit Board
PMOS	P-type Metal-Oxide Semiconductor
PN	Pseudo-noise
ps	Picosecond (Pico = 10^{-12})
PSK	Phase Shift Keying
p.u.l.	Per Unit Length
P_{zero}	Probability of Zero
QAM	Quadrature Amplitude Modulation
RZ	Return-to-Zero
SERDES	Serializer / Deserializer
SNR	Signal-to-Noise Ratio
SONET	Synchronous Optical Network
TEM	Transverse Electromagnetic
μm	Micrometers (micro = 10^{-6})

1 Introduction

1.1 Progression from Analog to Digital Communication

The field of communications has undergone significant changes in the last century. From wired/wireless telegraph to cellular telephones and the Internet, all are examples of communication systems that have added to the addiction the modern world has for the exchange of, and access to, information.

Taking discrete, quantized samples of the original waveform digitizes an analog signal. This discrete data can then be used to reconstruct the original signal with minimal loss in content. Traditionally, wired and wireless communications began by modulating analog information (like a voice on a telephone) onto a carrier waveform and transmitting the composite signal to a receiver. There, the composite signal is demodulated and the original signal recovered. In the analog domain, frequency domain multiplexing (FDM) is a method of transmitting information from multiple users on the same channel by allocating a frequency band to each user. This multiplexing method is very effective, however FDM requires a coherent demodulator for each frequency being used, which is expensive. Digital communications uses time domain multiplexing (TDM) which is a much simpler method for integrating multiple users, and TDM equipment costs are quite small by comparison. TDM assigns time slots to each user on the same frequency carrier, which has a wider bandwidth than FDM.

Without the benefit of coding to reduce redundancy, digital communication in the form of linear pulse code modulation uses more bandwidth than analog communication. Although digital communication uses more bandwidth, its benefits far outweigh this cost. Digital communications has gained support because of its simplicity, low power consumption, lower cost components, and operability at low signal-to-noise (SNR) ratio levels. Digital communication can be further broken up into two subsections, *Bandpass* and *Baseband*.

1.2 Baseband Digital Signal Transmission

Baseband digital transmission is non-modulated binary data which is conceptually very simple. Digital data is made up of groups of binary values which consists of two distinct values, logic 1 and logic 0. In its simplest form, one voltage level represents a logic 1 and another represents logic 0. For example a unipolar non-return-to-zero (NRZ) code could have 5 volts representing a logic 1 and 0 volts representing a logic 0, while polar NRZ could have logic 1 and 0 values represented by +5 and -5 volts, respectively. Other data codes, such as alternate mark inversion (AMI) and Manchester use combinations of discrete voltage levels to represent the two logic levels.

Ethernet local area networks (LAN) use baseband signals to transmit data along twisted pair wires across relatively shorter distances, for example, within buildings or businesses. It uses Manchester encoding, also known as biphase level encoding, to make use of its clocking information. Data rates are very high and special data coding schemes are used to increase throughput.

On a smaller scale, binary data transmitted within an integrated circuit (IC), across a printed circuit board (PCB) trace, or on a backplane inside a computer are also examples of baseband signals. Logic 1's and logic 0's travel on the tracks between two communicating IC's at data rates that have been steadily increasing over the years. Currently on PCB's, transfer rates of 3.125 gigabits per second (Gb/s) are common [1]. With these high transfer rates, problems that could previously be ignored with lower data rate systems must now be addressed.

1.3 Bandpass Digital Signal Transmission

Bandpass digital communication means that the information bearing signal is constrained to a relatively range of frequencies. In this communication method, digital information is used to modulate a sine wave before it is transmitted. Modulation allows information to be transmitted through a medium that does not carry low frequencies. An example of such a medium is a radio channel that uses small antennae. Many systems

use bandpass communications such as computer modems, digital subscriber lines (DSL), television cables, and cellular telephones.

Modems modulate and demodulate signals for sending and receiving data on twisted pair telephone lines, television cables, and radio channels. Some examples are

- A "voiceband" modem is designed to connect a computer to the Internet over a standard telephone line using the voice frequency range of 300 Hz to 3400 Hz.
- A cable modem sends and receives Internet data through a 6 MHz frequency band cable connection, which could simultaneously transmit cable television.
- Cellular telephones use digitized voice signals to modulate a carrier before sending the information through the air to a base station. They also demodulate incoming wireless signals from a cellular base station and convert it back into a voice signal.

Further benefits of digital communications come from multilevel signaling of data codes and modulation methods. Multilevel signaling uses several voltage levels to represent groups of bits. Modulation methods such as phase shift keying (PSK) use the different levels of the multilevel signal to change the phase of the carrier. The changes in phase represent the information bits. Quadrature amplitude modulation (QAM) also uses multilevel binary signaling and two quadrature carriers to transmit multiple bits per symbol [2].

1.4 Signal Transmission on Circuit Boards

Digital signals are exchanged on a printed circuit board (PCB) using conductive paths called *interconnects*, or *traces*. In order to ensure proper transmission of signals on the interconnect, signal transitions (edge rates) should be fast. As the bit rate on an interconnect line increases, so must the transition times. The faster the edge rate, the higher the frequencies contained in the signal and at high enough frequencies, the capacitance, inductance, and frequency dependent resistance of the interconnect become important. With slow edge rates, an interconnect line can be modeled as a lumped element, but fast edge rates require treating the line as a distributed parasitic element exactly like a transmission line. This depends on the spatial length of logic transition

and the physical length of the interconnect (trace). When logic transitions become spatially short with respect to the trace length or the trace length gets long with respect to the spatial length of the transition, they reach a point where the PCB trace can no longer be modeled as one simple system. When modeling a trace as a transmission line attention must be paid to impedance matching to alleviate reflections, echoes, or ringing, which distort the information-carrying signal.

Currently extensive research in optical interconnect technology is hoping to overcome the limitations of metal interconnects at the board and backplane levels. Optical data transmission has already been proven to transmit at data rates of 10 Gb/s which is three times higher than current metal interconnect [2]. Significant inroads have been made for optical interconnect technology on chip-to-chip and backplane domains. With advancements in the packaging of optical receivers with optical channels, higher bandwidth densities and efficiencies are possible compared to metal interconnect [3].

1.4.1 Ac Coupling and Dc Wander

When components on a PCB communicate with each other, all the information is in the form of binary data (logic 1's and logic 0's) which are represented by voltage levels. The region where a digital receiver distinguishes between a high and a low voltage is known as the *threshold region*. The component receiving the data uses the threshold to decide whether the incoming voltage represents logic 1 or logic 0; these voltage levels are not always common between manufacturers. One significant difference between the voltage levels of competing IC companies is the presence (or absence) of a dc offset. So how can two dissimilar IC's communicate?

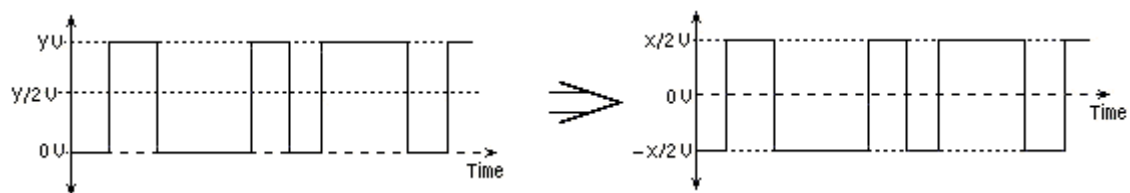


Figure 1.1 Eliminating dc offset using ac coupling.

A reasonable solution to this problem is to remove the dc offset on the trace between the IC's, giving the signal a zero volt threshold. The desired result is shown in Figure 1.1. The plot on the left shows a digital baseband signal with a $+y/2$ volts dc offset of (average voltage of 0V and yV). The plot on the right is the same signal with the dc offset removed. Dc offset removal is also known as ac coupling.

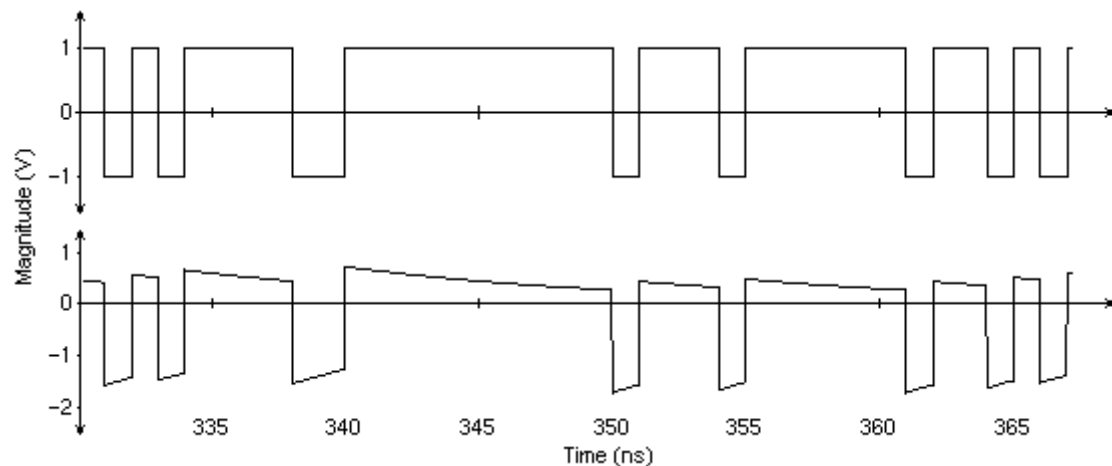


Figure 1.2 Transmitted signal (top) and ac coupled signal (bottom).

Ac coupling is a simple concept that has been adopted as an industry standard for solving the dc offset problem, and it works very well. The configuration consists of a capacitor in series with the line, and a shunt termination resistance. A transmission line also requires the termination resistance to eliminate reflections, echoes and ringing. The combination of capacitor and resistor forms a simple first order highpass filter. As seen in Figure 1.2, the voltage on the receiver side of the capacitor decays with respect to the input signal. This happens when the charge on the capacitor drains through the termination resistor to ground, causing the signal voltage levels to approach the threshold level. In the presence of noise, the signal voltage can cross the threshold level, causing receiver bit errors. The adverse effects due to ac coupling on a transmission line are noticeable if large numbers of consecutive identical values are transmitted. Over time, the dc level of the received signal constantly drifts up (positive dc offset due to an overabundance of logic low values) and down (negative dc offset due to an overabundance of logic high values). This phenomenon is called “dc wander”.

Some proposed solutions to alleviate dc wander are the use of dc balanced codes such as Manchester, Bipolar alternative mark inversion (AMI), or Bipolar return-to-zero (BPRZ) [2]. Integrated Services Digital Network (ISDN) uses dc balancing bits to improve signal balance [4]. This method uses several bits within each byte of data to keep the signal balanced. Line codes (4B5B or 8B10B) also use redundant bits to improve signal balance. Decision feedback circuits correct dc wander by using the history of the received binary sequence to determine the amount of feedback needed. Currently, there are existing dc restoration circuits in commercial IC's, but the designs are protected as proprietary information so no published work is available.

1.4.2 Crosstalk

Crosstalk is a problem because the added noise (in the form of voltage and current injected by changing electromagnetic fields of neighbouring traces) can drive the signal across the decision threshold and cause bit errors. When communications occur between chips on a PCB, the signals propagate on metal trace lines. As a consequence, electric and magnetic fields are present around the lines. As technology progresses, logic switching times have become increasingly fast. As the rate of change of voltage on a line increases, the rate of change of electric flux also increases. Additionally, the increase in the rate of change of driving current in a circuit corresponds to an increase in the rate of change of magnetic flux around the trace. The increase in the rates of change of electric and magnetic flux induces currents and voltages, respectively, on traces in close proximity of the driving circuits. This causes additive errors to any existing signal on the victim line. At the end of a trace, an IC receives the distorted signal which, depending on the severity of the distortion, could result in a bit error. This type of interference is called "crosstalk". Although the electric and magnetic coupling can seem quite small, they become increasingly significant as switching times decrease, PCB geometries shrink, and interconnect lines get closer together.

There are two types of EM interference. The first is due to mutual *capacitance* between the neighbouring traces. This capacitance injects a current from the driving trace (sometimes called the "aggressor") to the adjacent receiving trace (called the

“victim”). The magnitude of the injected current is highly dependent on the separation of the two lines (having an inverse relationship to the separation distance), the rise time of the digital signal, and the resistance of the victim trace.

The second type of EM interference is due to the mutual *inductance* between adjacent circuit loops, which consist of the interconnect line and the return ground plane. The two circuit loops can be modeled as the primary and secondary windings in a transformer. The inductance value has the same dependence on the line separation as with mutual capacitance. A voltage is induced on the victim circuit proportional to the rate of change of current on the aggressor circuit, which is directly related to the rise time of the digital signal and the resistance of the driving circuit.

Interference, due to mutual capacitance and inductance, combine to create “crosstalk” on neighbouring traces. There are two types of crosstalk referred to as *far end crosstalk* (FEXT) and *near end crosstalk* (NEXT). FEXT is also called ‘forward-traveling’ crosstalk because it travels in the same direction as the pulse that induced it. FEXT has contributions from both mutual capacitance and inductance which are induced constantly as the rising/falling edge propagates on the aggressor line. This allows the crosstalk to build linearly on the victim trace for the entire length of the (closely spaced) lines. The sum of this crosstalk appears as a small pulse at the receiver where it can cause the receiver to misread the data. NEXT is also known as ‘reverse-traveling’ crosstalk as it propagates in the opposite direction of the pulse that induced it. NEXT also includes contributions from both mutual capacitance and inductance between the closely spaced traces. This crosstalk also occurs at all points between the interconnect traces and the result is a higher (or lower) than normal voltage at the near end (where the aggressor pulse originated) of the victim line. Both FEXT and NEXT will be investigated in detail in Chapter 5.

1.5 Thesis Main Objectives

The ultimate outcome of this project is to help increase the transmission bit rate on a PCB interconnect line. The requirements of the proposed compensation schemes are that they must operate at very high speeds and provide compensation in real time. A

“compensation signal” is generated and used to correct for a given distortion. Complex filter and equalizer designs require too much computation time and, therefore, can not be used here.

While information is transmitted between communicating devices, interference and noise sources corrupt the signal. These external influences are additive and can make it difficult for the receiver to distinguish between the different logic levels. Two of the interference sources are dc wander and crosstalk. Noise comes in the form of thermal noise, power supply noise, and external electromagnetic (EM) radiation.

The dc wander problem, a form of inter-symbol interference, is caused by the ac coupling on a transmission line. The researcher proposes to alleviate the effects of dc wander by using digital filters to add a complementary waveform (having an opposite effect than the ac coupling) to the original signal. This filter is designed with prior knowledge of the termination resistance and the ac coupling capacitor. With this information, a signal can be created that cancels the signal decay (due to ac coupling) experienced on the interconnect. Both finite impulse response (FIR), and infinite impulse response (IIR) filters have been considered.

The crosstalk investigated in this thesis is caused by electromagnetic radiation between PCB traces due to sub-nanosecond digital transition times. The researcher proposes a novel CMOS (complementary metal-oxide semiconductor) logic circuit to mimic the NEXT crosstalk effect and subtract this mimic signal from the incoming signal in real time.

The above problems have been identified as limiting factors before significant increases in bit rates can be made [5]. Along with timing, silicon space is also an important issue. This project will also minimize the space taken on the IC to implement the proposed designs.

Because interconnect problems involve very high data rates, it is difficult to model the system by creating a physical PCB with high speed IC's exchanging information. It is also impractical to fabricate small digital filter designs on silicon for testing purposes, as fabrication is very expensive. Therefore, Matlab[®], Simulink[®], and

Altera[®] Quartus II[®] are used to create models that represent actual communication systems to simulate the effects of the proposed compensation schemes.

1.6 Thesis Organization

The body of this thesis contains an introductory chapter which outlines the concepts of digital communication. This is followed by a discussion of transmission lines, signal thresholds, distortion, and digital circuits in Chapter 2.

Chapter 3 analyzes dc wander in more depth and discusses several possible solutions to reduce its effects. A filter is designed to address dc wander in Chapter 4 and the Matlab[®] and Simulink[®] software packages are introduced. A simulation system is developed and the proposed filters are incorporated to run within the simulation system and correct for the dc wander. Chapter 4 also presents the results of the dc wander compensation which are then analyzed and discussed.

Chapter 5 analyzes coupling between adjacent interconnect lines and how crosstalk is generated. Possible solutions to crosstalk are presented and the proposed solutions are presented. In Chapter 6 the proposed circuits are designed and tested within a simulation system created in Simulink[®] and Matlab[®]. The circuits are also created using Altera[®] software and the simulations show the functionality of the designs. The results of the crosstalk compensation are presented and analyzed which are compared to ideal performance. Finally, conclusions and suggestions for further investigation are given in Chapter 7.

2 High Speed Digital Communications

Making the transition to higher-speed digital circuit design requires knowledge that digital design engineers have previously not needed. This knowledge required for high speed digital communications has been available for decades in other electrical engineering disciplines, such as microwave design [6]. High speed systems operate at frequencies such that conductors no longer behave as simple wires. Instead, they behave as transmission lines and are therefore affected by high-frequency anomalies such as “skin effect”. System performance can be seriously overestimated if the transmission lines are not modeled properly [5],[6].

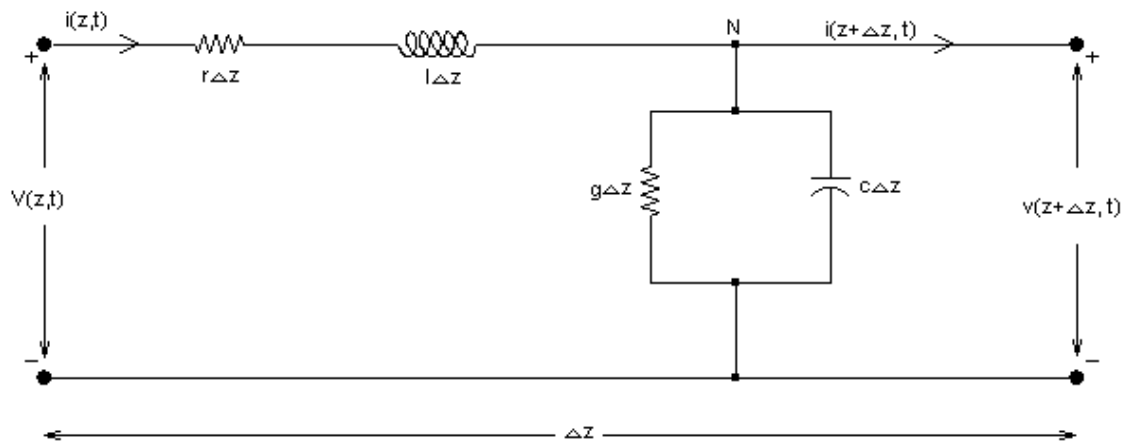


Figure 2.1 Equivalent circuit of an incremental length of a 2 conductor transmission line.

2.1 Transmission Lines

As with any communication system, performance depends on the transmission channel. On a printed circuit board, an interconnect line must be treated as a transmission line if the digital rise/fall times become very fast (e.g. $T_r < 1$ ns). This makes analysis of the communication system more complicated. Figure 2.1 shows a

model of an incremental length of transmission line. The series resistor, $r\Delta z$, represents the losses due to the finite conductivity of the conductor. The shunt conductance, $g\Delta z$, represents the losses due to the finite resistance of the dielectric material separating the conductor and the ground plane. The series inductor, $l\Delta z$, represents the losses due to self-inductance of the line and a shunt capacitance, $c\Delta z$, represents the self-capacitance of the line.

These parameters are important for accurately modeling of the line and for calculating the losses and interactions between closely spaced high speed data lines. The *per unit length* (p.u.l.) parameters (R, G, L, and C) can be measured and used to determine the characteristic impedance (Z_0) of a PCB trace using the following formula:

$$Z_0 = \frac{V}{I} = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \quad (2.1)$$

Characteristic impedances typically used on printed circuit boards are 50Ω single ended and 100Ω differential [5]. The propagation delay of the system (as dictated by electromagnetic wave theory) is given by

$$Delay(ps / p.u.l.) = 10^{12} \sqrt{LC} \quad (2.2)$$

It is also important to treat interconnect traces as transmission lines in high speed circuits to properly model crosstalk. With a lumped model, a neighbouring trace is affected by crosstalk across the entire line simultaneously. With a transmission line (distributed) model, the inductive and capacitive crosstalk occurs sequentially along the line. There are two configurations of interconnect line that are commonly used for high speed transmission of data, microstrip and stripline.

2.1.1 Microstrip

Microstrip is a versatile, easily fabricated configuration for an interconnect line. A microstrip is a thin conducting metal line with the top exposed to air and the bottom separated from a solid ground plane by a dielectric substance (Figure 2.2). Since there are two different dielectrics surrounding the microstrip, the environment is considered non-homogeneous which cannot support a true transverse electromagnetic (TEM) wave

[7]. This means the phase velocity of the radiated TEM fields in the two dielectrics travel at different speeds. The consequence of this is signal dispersion which affects the eye opening of the eye diagram. The amount of dispersion is proportional to the length of line, so shorter lengths are desired.

Another disadvantage to microstrip is the amount of EM fields radiated to the surrounding environment. Since the top of the interconnect is exposed to air, the EM fields radiate outwards and can be picked up by other electronic components in the system, causing noise. In particular, the neighbouring lines can pick up this EM radiation as crosstalk.

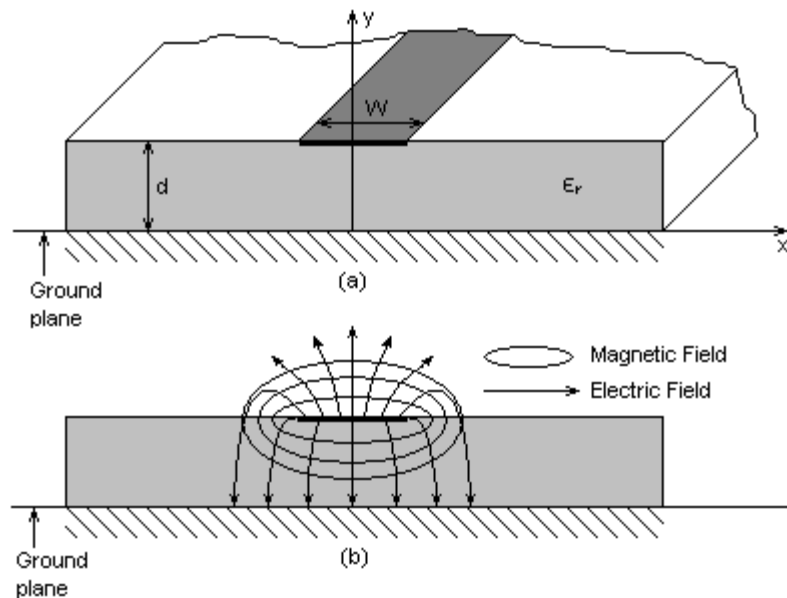


Figure 2.2 Microstrip transmission line. (a) Geometry. (b) Electric and magnetic field lines [7].

Microstrip also has several advantages. Since the top is exposed to air it is easy to test the voltages on the line. This also makes it much easier to manufacture and to make changes to a system after it has been manufactured. Another advantage involves the use of *vias*. A via is a drilled hole in a circuit board used to connect two components (e.g. two interconnect lines) from one circuit level to another on multi-level printed circuits boards. When connecting two integrated circuits on a PCB using microstrip,

vias are not needed to route the line between them. At high speeds, vias add signal distortion which can affect signal integrity [8], [9].

2.1.2 Stripline

Stripline is another configuration of interconnect line that addresses some of the shortcomings of microstrip. However it also has shortcomings of its own. The stripline configuration consists of a metal line that is ‘sandwiched’ between two identical dielectric substances (making it a homogeneous medium) with solid ground planes on top and bottom. Ground planes are sometimes used for dc power supply, however still function as an ac ground plane. This configuration is shown in Figure 2.3. The homogeneous dielectric medium means pure TEM waves are allowed to propagate on the stripline [7], which avoids the dispersion experienced by microstrip lines.

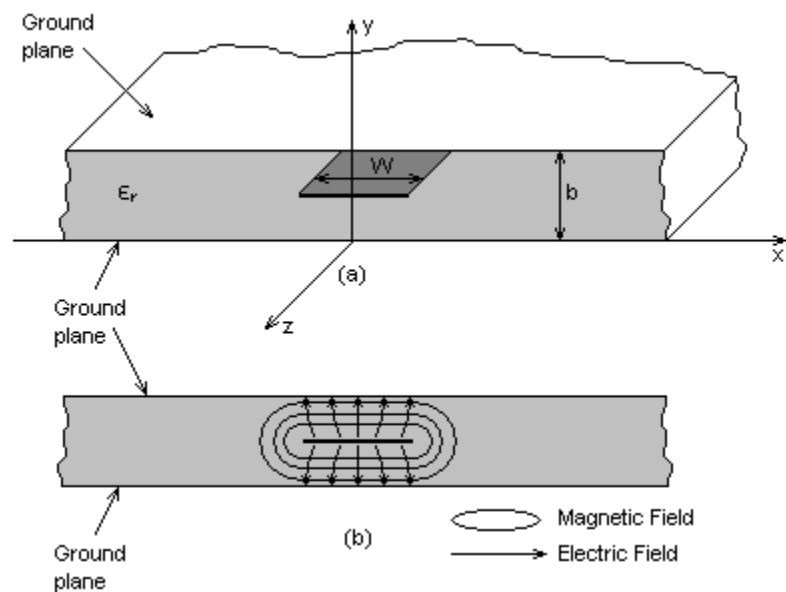


Figure 2.3 Stripline transmission line. (a) Geometry. (b) Electric and magnetic field lines [7].

Also, because of placement of the stripline solid ground planes, there are no EM fields radiated into the surrounding environment. Although this eliminates the EMI to other components on the PCB, it does not eliminate the electromagnetic interference (EMI) between striplines within the same dielectric medium.

The existence of ground planes is not always an advantage since this topology makes the line inaccessible for testing. Vias are needed to bring the signal up to the top layer to connect with ICs. Even with the several shortcomings of stripline, it is the one chosen for this thesis project because of its signal transmission properties.

2.1.3 Transmission Line Modeling

Modeling a transmission line in the simulations is very important since the signal is distorted as much by the transmission line effects as it is by dc wander or crosstalk. These transmission line effects are needed to present an accurate model and accurate results. From transmission line theory, the general expression describing the distortion on a transmission line is called the propagation constant, and is given by [7]

$$\gamma = \alpha + j\beta = \sqrt{(R + j\omega L)(G + j\omega C)}. \quad (2.3)$$

A good transmission line has low-loss and approximations can be applied under these conditions. A reasonable approximation is obtained by assuming it operates in a low-loss frequency range ($\omega L \gg R$). This results in equations which are known as the high-frequency, low-loss approximations for transmission lines and are given by [7]

$$\gamma \approx j\omega\sqrt{LC} \left[1 - \frac{j}{2} \left(\frac{R}{\omega L} + \frac{G}{\omega C} \right) \right] \quad (2.4)$$

and

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \approx \sqrt{\frac{L}{C}}. \quad (2.5)$$

This means that the attenuation and phase parts of the propagation constant are now given by

$$\alpha \approx \frac{1}{2} \left(R\sqrt{\frac{C}{L}} + G\sqrt{\frac{L}{C}} \right) = \left(\frac{R}{2Z_0} + \frac{GZ_0}{2} \right) \quad (2.6)$$

and

$$\beta \approx \omega\sqrt{LC}. \quad (2.7)$$

These are important expressions and are used in the simulations as the per unit length transfer function

$$H_{p.u.}(j\omega) = e^{-\left(\frac{R}{2Z_0} + \frac{GZ_0}{2}\right)} e^{-j\omega\sqrt{LC}}. \quad (2.8)$$

As will be shown in the following chapters, transmission line attenuation is a serious issue when trying to increase the throughput of a communication system. A transmitted signal can be seriously degraded at the receiver depending on the length of transmission line. This is particularly important when trying to correct for near end crosstalk (NEXT), as the amplitude of the crosstalk voltage (induced by the adjacent line) is similar to that of the received signal.

2.2 High Speed Transmission on PCB Traces

Historically, with long signal transition times, PCB traces were simply viewed as an electrical short. However, as transition times decrease, important parameters to consider are *rise time* and *propagation delay*. For a given rise time and depending on the length of line, track geometry, and PCB board material, the transmission line can either be analyzed as *lumped* (one value for each of resistance, capacitance, and inductance) or *distributed* (as in a transmission line).

- Transition time is defined as the amount of time it takes for the logic signal to change state (measured from 10% to 90% of full transition). Transition times have been getting increasingly shorter as technology progresses and depend on the fabrication process used in the IC manufacturing. Propagation delay is defined as the time it takes for a signal to propagate one unit of distance. The delay increases in proportion to the square root of the dielectric constant of the surrounding medium [5].
- A common printed circuit board material is FR4, which is a type of fiberglass [2]. It has a dielectric constant range of 2.8 - 4.5 and causes a signal delay of $D = 55 - 70$ picoseconds/cm (ps/cm) [5]. The low cost FR4 material is a very popular choice with PCB manufacturers and, as a consequence, specifying a different board material to decrease the dielectric constant is not desirable.

- PCB trace length is the physical length of the interconnect on which the signals propagate. As the level of integration on each chip increases, PCB's get smaller, the components get much closer together, and trace lines get shorter. Shorter trace lengths are desirable, and this can partially compensate for faster transition times.

The response of any system of conductors to an incoming signal depends greatly on whether the physical system (PCB interconnect) is smaller or larger than the effective spatial length of the fastest electrical feature in the signal. The effective length of such an electrical feature, like a rising edge, depends on the time duration of the feature and its per unit propagation delay. According to [5] the length of a rising edge is given by:

$$l = \frac{T_r}{D} \quad (2.9)$$

where D is delay per unit length (p.u.l.). As a rule of thumb, if a PCB trace length is shorter than $l/6$, the system can be viewed as a lumped system; otherwise it should be viewed as a distributed system and transmission line theory applies. Some researchers use $l/\sqrt{2\pi}$; others use $l/4$ or even as small as $l/10$. The idea is that “electrically” short structures are lumped circuits, while big ones are distributed [5].

By analyzing the interconnect length, rise time, and propagation delay, high speed communication effects such as reflections, echoes, and ringing can be analyzed and compensated. These effects are covered in the next section. Another consequence of faster rise times is the increased electromagnetic (EM) radiation emanating from PCB interconnect which can have adverse effects on neighbouring traces. Displacement currents are induced on adjacent interconnect lines by sudden changes of voltage. Additionally, fast changes in current cause fast changes in the magnetic flux that couple closely spaced interconnect, which in turn induces voltages on adjacent lines. These are two different components of crosstalk which will be covered in more detail in Chapter 5.

2.2.1 Reflections and Transmission Line Termination

To understand the purpose of transmission line termination, the behavior of an unterminated line is examined first. An unterminated transmission line is similar to an

echo chamber such as a church organ pipe. If a voltage step is transmitted down an unterminated transmission line, it will bounce back and forth between the ends of the line until losses in the line eventually cause the echo to die out. The resulting reflections appear as undesirable ringing on the step signal. Ringing includes both overshoot (going past the steady state value) and undershoot (falling back below the steady state value before stabilizing). Although ringing can happen in lumped systems, it *usually* happens in distributed systems unless properly terminated Figure 2.4.

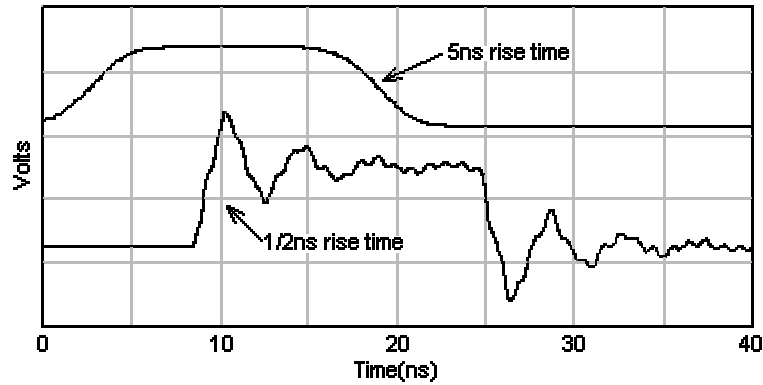


Figure 2.4 Slow rise time (top) and fast rise time (bottom) on unterminated transmission line [5].

Properly chosen termination resistors placed at either the source side or the load side of a transmission line can eliminate this ringing. The purpose of termination resistors is to dissipate the energy in the transmitted signal so that reflections do not occur. Figure 2.5 illustrates how a signal is reflected on a transmission line from an impedance mismatch at the load. The magnitude of the incident wave (V_i) is determined by the voltage divider between the source impedance and line impedance:

$$V_i = V_s \frac{Z_0}{Z_s + Z_0} \quad (2.10)$$

The amount of signal reflected depends on the termination resistance, and is represented by the voltage reflection coefficient, ρ , defined as:

$$\rho = \frac{V_{\text{reflected}}}{V_{\text{incident}}} = \frac{Z_t - Z_0}{Z_t + Z_0} \quad (2.11)$$

where Z_0 is the impedance of the line and Z_t is the impedance of the termination resistance. Note that if $Z_t = Z_0$ (a matched termination), the reflection coefficient is zero (i.e. there is no reflection).

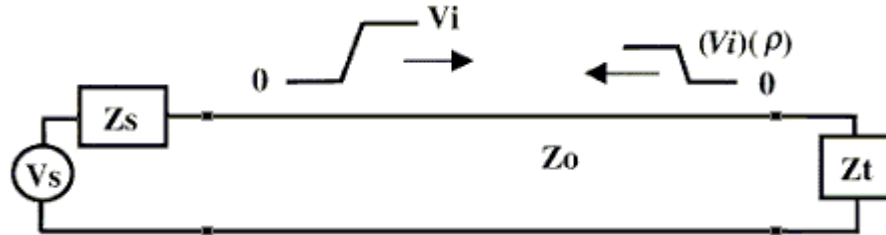


Figure 2.5 Incident signal being reflected by a mismatched load [6].

When the forward traveling and reflected signals interact, they add and subtract according to the principle of superposition. A detailed description and analysis of electromagnetic and transmission line theory can be found in [11].

2.3 Thresholds and Noise Margins

As mentioned in a previous section, a *threshold* is a voltage where a digital receiver distinguishes between high and low logic levels. For example, a NRZ binary signal with no dc offset would have a decision threshold near 0 Volts. Figure 2.6 shows the threshold voltage is uncertain. The precise value can be anywhere within a certain range. Since there is always noise present in electronic circuits, buffer zones must be created around the threshold region to protect against bit errors. Most noise can be approximated and modeled by well-known statistical distributions. If the particular noise can be identified and modeled, it can be incorporated into a design to create buffer zones, called *noise margins*. The term “noise margin” refers to the difference between $V_{OH\ min}$ and $V_{IH\ max}$, or the difference between $V_{OL\ max}$ and $V_{IL\ min}$, whichever is least (Figure 2.6). Noise margins employ the variance of the noise distribution to move the safe input level and guaranteed output level away from the decision threshold. This is an effective method for protecting a system from bit errors provided the signal’s dc level remains constant. Since the actual decision threshold tends not to be at exactly zero, the noise margin is already reduced due to the threshold variation.

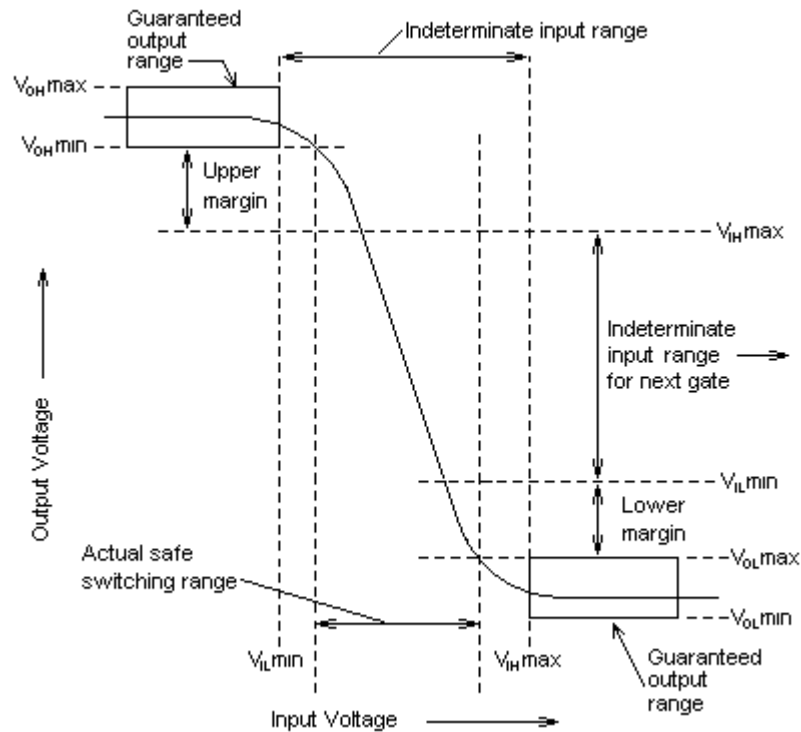


Figure 2.6 Example of noise margins [5].

As seen in Figure 2.6, there is no overlap between the guaranteed output range and the indeterminate input range. The absence of overlap means that static transmitted values, whether 0 or 1, will always be properly received.

The effectiveness of noise margins deteriorates as the dc level shifts, which can happen if the transmission lines are ac coupled. As described previously, ac coupling is used to remove dc offsets from transmitted signals. On an ac coupled transmission line, the configuration of the ac coupling capacitor and the termination resistor forms a first order high-pass filter. This causes the dc level to wander, particularly on signals that are not dc-balanced (i.e. have long sequences of one logic state). Dc wander and crosstalk cause the signal to encroach on the noise margin increasing the probability of receiving a bit error. This can be observed with the aid of an *eye diagram* (Figure 2.7). Using the eye diagram a signal can be observed to study the effects of band-limited channels, distortion, and timing impairments [6]. At the sampling instant, the received signal should be well above, or below, the threshold voltage. This ensures reliable detection of

a binary one or zero. The eye diagram indicates the margin against possible errors caused by the noise or distortion.

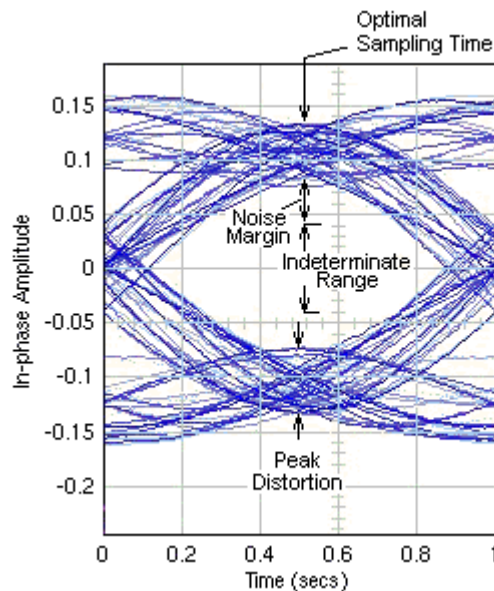


Figure 2.7 Example of an eye diagram.

2.4 Distortion

There are several sources of noise that corrupt transmission line signals.

1. Noise is introduced into a signal line through crosstalk. Crosstalk is generated by signals on adjacent lines through mutual capacitance or mutual inductance. The crosstalk adds to, or subtracts from, a transmitted signal potentially moving the signal closer to the switching threshold thus causing errors.
2. Power supply currents flowing through the ground path can cause the ground (reference) level to change. A signal that is transmitted at a certain voltage level above the ground reference can be received at a different potential if the ground reference changes. Ground voltage potentials can also be altered by fast changing return signal currents due to inductance of the ground path [5].
3. Reflections travel backwards and forwards on improperly terminated lines and can distort signals by way of superposition. The addition or subtraction of the reflected signals with the transmitted signals can cause the signal to infringe on the noise margins.

4. Threshold levels on certain logic families are a function of temperature. Large temperature differences between transmitting and receiving gates may result in reduced margins.

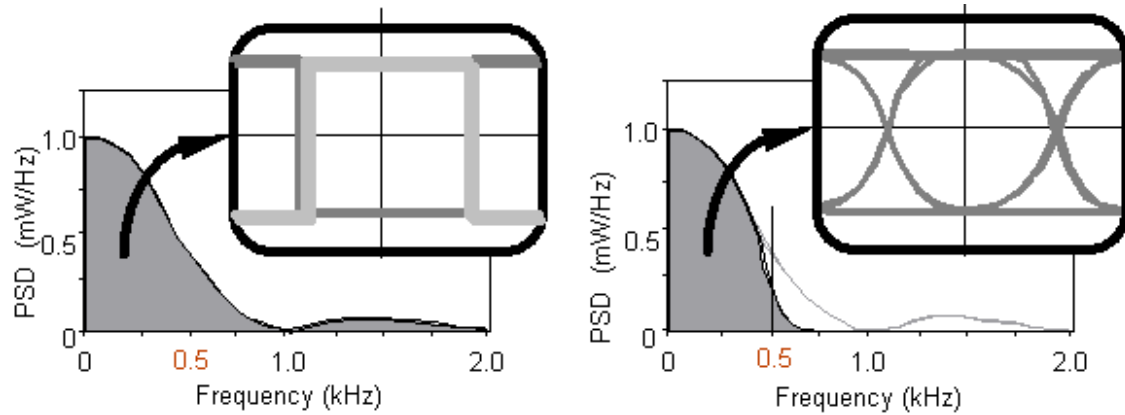


Figure 2.8 (a) Original spectrum of a square pulse and resulting ladder (rectangular eye) diagram and (b) spectrum and eye diagram after loss of high frequency component [10].

5. Losses on a transmission line can be modeled by passing a signal through a lowpass filter. When the high frequencies are lost due to the intrinsic losses in the transmission line and dielectric medium, the original square pulse signal becomes much more rounded and appears much more like an ‘eye’ as shown in Figure 2.8.
6. When ac coupling is used on a transmission line to remove the dc offset, a highpass filter is created which results in the loss of some of the low frequencies in the signal. The consequence of the low frequency loss is shown in Figure 2.9 where dc wander has caused the eye diagram to move up and down, thus the lines appear much thicker.

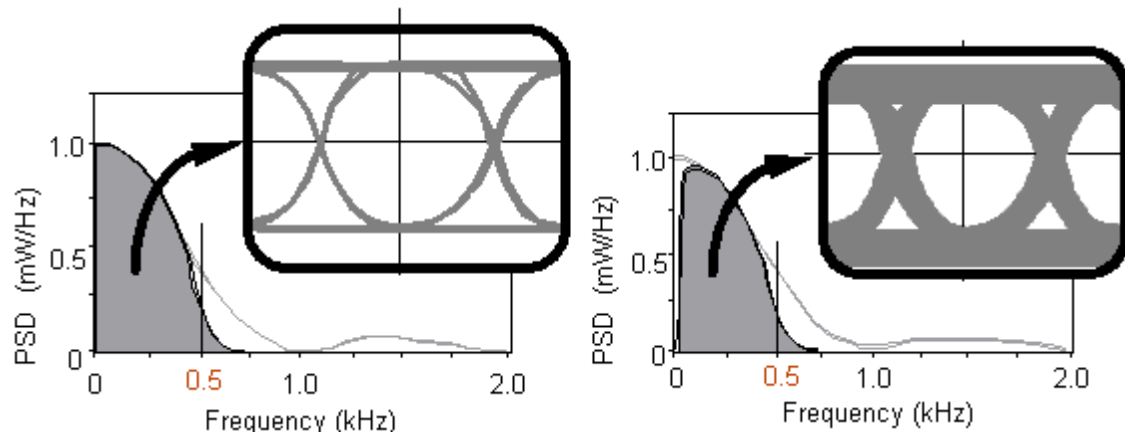


Figure 2.9 Spectrum and eye diagram from a (a) loss of high frequency component due to transmission line effects and (b) loss of low frequency due to ac coupling [10].

Although some of the aforementioned noise sources are present in all electronic systems regardless of operating speed, the others are particular to high speed systems.

2.5 Filters

Filters are electronic components capable of removing unwanted parts of a communications signal. In general, they are frequency selective and are designed to remove unwanted frequencies in a signal. Analog filters are created using basic electrical circuit components (resistors, capacitors, and inductors). Digital filters are created using delay elements and quantized samples of the signal.

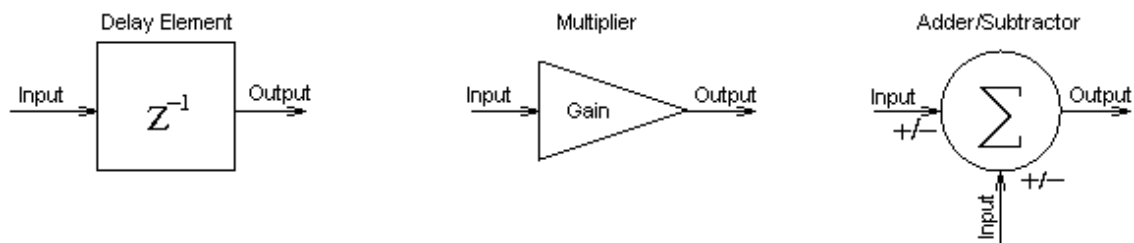


Figure 2.10 Basic digital filter building blocks.

Digital filters require the main blocks shown in Figure 2.10 which perform the following tasks:

- The *delay element* delays the quantized signal sample by one clock cycle by using a register.
- The *multiplier* multiplies the quantized signal sample by a specific value and outputs the result.
- The *adder/subtractor* adds or subtracts two or more quantized signal samples and outputs the result.

A digital filter performs delay, multiply and add operations on digital data during each clock period (signal sample period). This process creates a digitized version of an equivalent analog filter characteristic. Mathematically, digital filters are expressed in the time domain by the following difference equation [12]:

$$y(nT) = \sum_{i=0}^M b_i x((n-i)T) - \sum_{i=1}^N a_i y((n-i)T). \quad (2.12)$$

The first summation in the above difference equation is often referred to as the non-recursive, or feed-forward, part. Similarly, the second summation is referred to as the recursive, or feedback, part. M and N are the *orders* of the non-recursive and recursive parts, respectively.

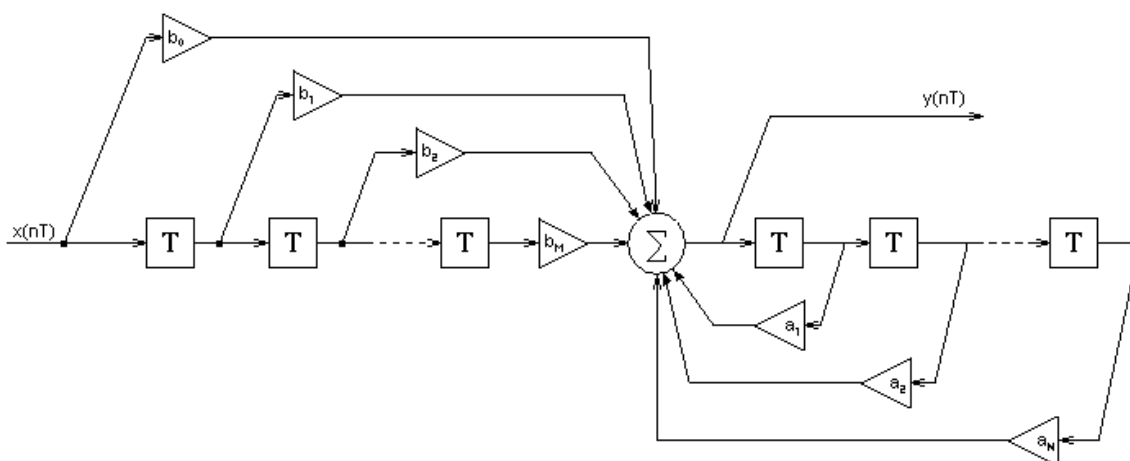


Figure 2.11 Block diagram of a digital filter with recursive and non-recursive parts.

The lower index on the first summation is zero and of the second summation is unity which implies that current and past values of the input and output sequences are used. The block diagram form of the digital filter is shown in Figure 2.11.

It is useful to express Equation 2.14 in another form called the *z-domain transfer function*. This is done using the *z-transform* which results in the following expression:

$$Y(z) = \sum_{i=0}^M b_i X(z) z^{-i} - \sum_{i=1}^N a_i Y(z) z^{-i} . \quad (2.13)$$

Note that in this expression, z^{-i} represents the delay of ‘ i ’ sample times. With some simple algebraic manipulation, the transfer function for a digital filter is obtained and expressed in the *z-domain* as:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^M b_i z^{-i}}{a_0 + \sum_{i=1}^N a_i z^{-i}}, \quad a_0 = 1. \quad (2.14)$$

The coefficients, a_i and b_i , define the multipliers of the filters and z represents the time delay elements. The numerator of the transfer function defines the zeros of the digital filter which are used in finite impulse response (FIR) filters. The denominator of the transfer function defines the poles of the filter and are used in infinite impulse response (IIR) filters. Each of these filter types will be discussed in detail next. A thorough discussion on the properties of digital filters can be found in [12].

2.5.1 Digital IIR Filter

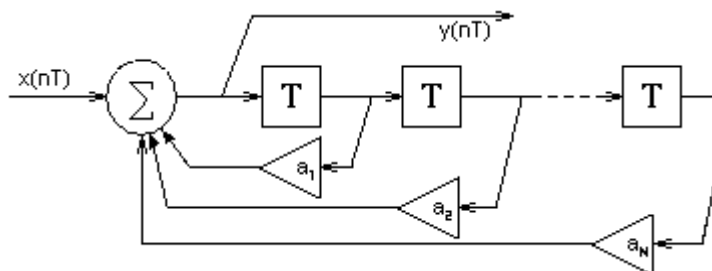


Figure 2.12 Typical form of IIR filter.

An *infinite impulse response* (IIR) filter (Figure 2.12) is also called a *feedback* or *recursive* filter due to the delayed samples being multiplied by coefficients, fed back and added to the incoming signal. Since the influence of previous values is present for long periods of time, it is said to have *memory*. This memory effect can be seen by observing the long decay on the waveform in Figure 2.13(b). It is also seen in the following time domain difference equation

$$y(nT) = \sum_{i=1}^N -a_i y((n-i)T), \quad (2.15)$$

as it always uses the previous value of the output. Each discrete output value, $y(nT)$, of the filter is used N times as it is shifted through the time delay units. A strictly recursive (IIR) filter is represented by the z -domain summation

$$Y(z) = \frac{1}{a_0 + \sum_{i=1}^N -a_i z^{-i}}, \quad \text{where } a_0 = 1. \quad (2.16)$$

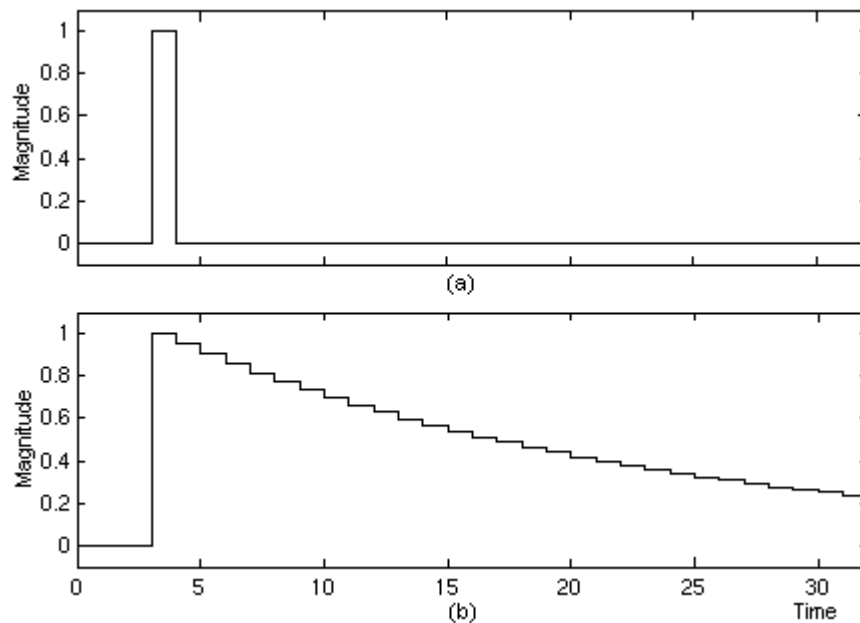


Figure 2.13 (a) Discrete impulse, (b) highpass impulse response for IIR filter.

This filter design is shown in Figure 2.14 and the impulse response is shown in Figure 2.13(b) for $b_1=0.95$. It is very important for the coefficient value b_1 to be less than one or the filter becomes unstable and the output signal will grow quickly without bound. The transfer function for a first order IIR filter is given by the expression

$$H(z) = \frac{1}{1 - a_1 z^{-1}} = \frac{z}{z - a_1}. \quad (2.17)$$

Instability is a possibility when working with feedback filters and care must be taken [12]. Stability issues are also an important consideration because IC's have non-infinite precision. Placing a pole too close to the edge of the unit circle (in the z -domain) may actually result in placing the pole *outside* the unit circle (coefficient would be greater than 1), resulting in an unstable system. Certain software packages like Matlab[®] allow the coefficients to be very close to 1 for simulations because of the almost infinite precision available on a computer. However, it is not feasible to devote large areas of silicon to provide near-infinite precision on an actual IC. This also results in imprecise filter responses and thus imperfect waveforms.

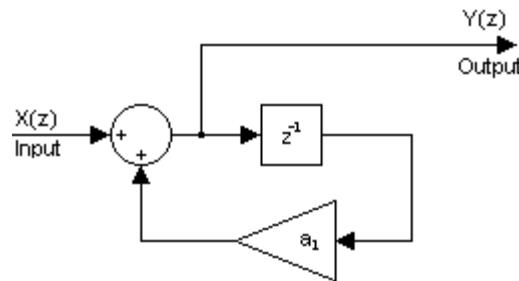


Figure 2.14 First order IIR (recursive) digital filter.

2.5.2 Digital FIR Filter

A non-recursive filter contains only feed-forward sections, and has a *finite impulse response* (FIR). This type of filter is unconditionally stable but also consumes much more silicon area than IIR filters that perform the same function [12]. An FIR filter is described for causal signals by the time domain difference equation

$$y(nT) = \sum_{i=0}^M b_i x((n-i)T), \quad (2.18)$$

and in the z -domain by the transfer function

$$Y(z) = \sum_{i=0}^M b_i z^{-i} \quad (2.19)$$

where M is the order of the filter. This expression defines the zeros of the system. An example of an FIR filter, similar to the one used in this thesis project, is shown in Figure 2.15 and is expressed in the z -domain as

$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M} \quad (2.20)$$

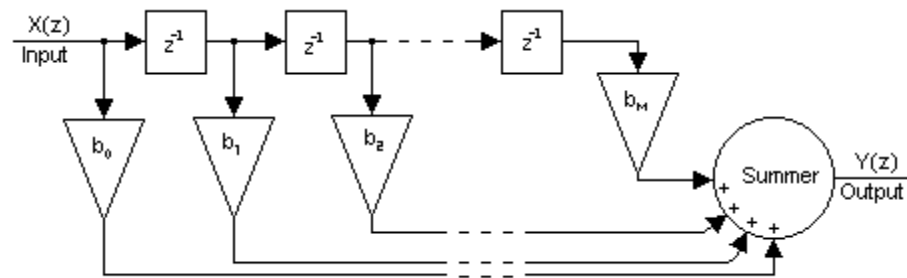


Figure 2.15 Simple FIR (non-recursive) digital filter.

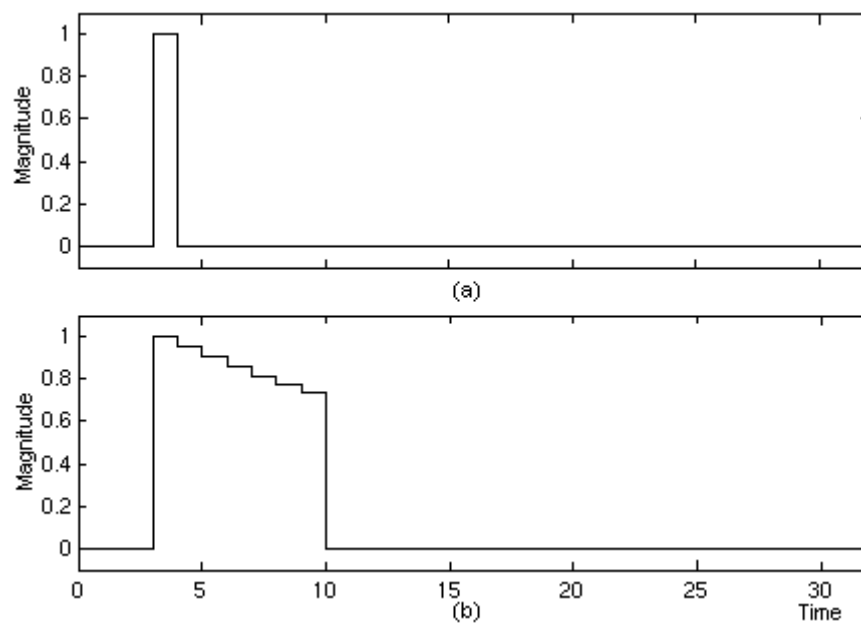


Figure 2.16 (a) Discrete impulse, (b) highpass impulse response for an FIR filter.

The input impulse to this filter is shown in Figure 2.16 (a) and the resulting waveform at the output of the filter is shown in Figure 2.16 (b) with the coefficient values

$$\left. \begin{array}{l} b_0 = 0.95, i = 0 \\ b_i = b_0^{i+1}, i = 1, 2, \dots, 6 \end{array} \right\} \quad (2.21)$$

2.6 Issues with Digital Circuits

One problem with implementing digital components on an integrated circuit (IC) is the issue of silicon area consumed by the compensation circuits. Although shift registers occupy a relatively small area on an IC (depending on the length), the size of digital multipliers and adders are known to be quite large especially if a high degree of accuracy is needed. A multiplier can also be quite complex which raises the issue of signal timing and propagation delay. Complex digital multipliers and adders could delay compensation enough to limit the circuit's usefulness. Also, signal and compensation voltages can be either positive or negative. Multipliers and adders would need circuitry to allow negative numbers.

2.6.1 Digital Multipliers

A solution to the digital multiplier size issue would be to use one with tailored precision thus limiting its complexity. This is possible with block floating point (BFP) multipliers. The accuracy depends on the number of bits the multiplier contains. For example a 5-bit multiplier (4 bits of precision and one sign bit) would have $2^{-4} = 0.0625$ precision with a maximum value of $1 - 0.0625 = 0.9375$. The precision can even be improved without increasing the complexity. For example, if the maximum value of crosstalk is known to be $2^{-2} = 0.25V$ (must be a power of 2, i.e. 2^{-n} , $n = 0, 1, 2, 3, \dots$) then the precision of a 4-bit BFP multiplier could be set to $0.25(0.0625) = 0.015625$ with bit shifting (Figure 2.17). The extra bits resulting from the bit shifting will remain zero and not add to the complexity of the multiplier. The number of bits in a multiplier is therefore chosen based on the absolute maximum value of compensation expected.

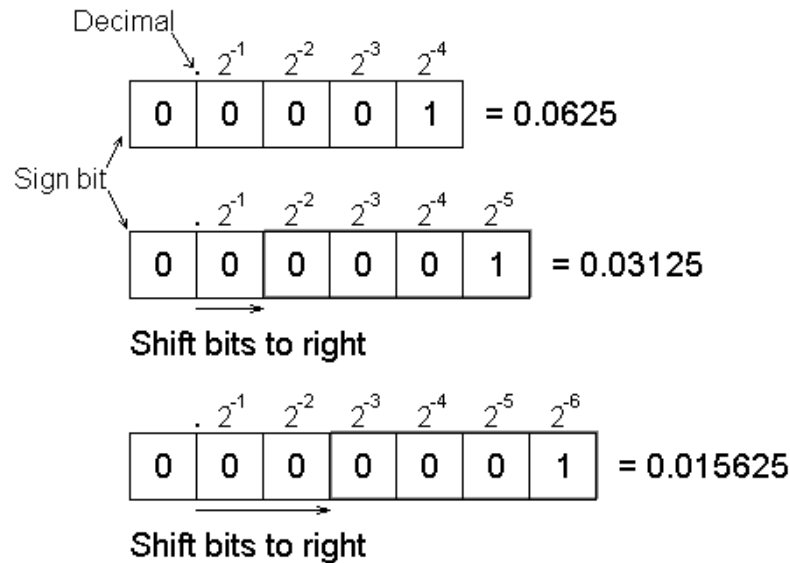


Figure 2.17 Increasing accuracy of BFP multiplier using the concept of *bit shifting*.

In addition to the size and accuracy benefits of the BFP multipliers, they also have the capability of being programmed on-chip, provided there are pins available. This addresses the problem of knowing the PCB layout parameters before the IC is produced, in two ways. As mentioned, the value of the multipliers can be programmed after the chip is manufactured, which also means the effective length of a shift register can be programmed as well. This is done by first calculating the equalizer's desired length, setting the values of the multipliers for this length, then setting all subsequent multiplier coefficients to zero thus effectively truncating the shift register. As an alternative, the length of the shift register can also be programmed without using multiplier values.

The analog domain may contain the simplest solution to the multiplier problem. Since the compensation signal must be in analog form to be added to the voltage input at the receiver, simple on-chip resistors could also be used to obtain the required compensation signal amplitude.

2.6.2 Digital Adders

Adder size remains a significant problem for digital circuits, particularly for FIR filters and multiple tap equalizers. Adders usually add two values together, so if a shift register is long, many adders are needed to generate the output signal. For example a simple equalizer circuit with 16 taps would require $8 + 4 + 2 + 1 = 15$ adders which may be prohibitively large. This is demonstrated in Figure 2.18. This is particularly important with adders since they are relatively large even if there is only one.

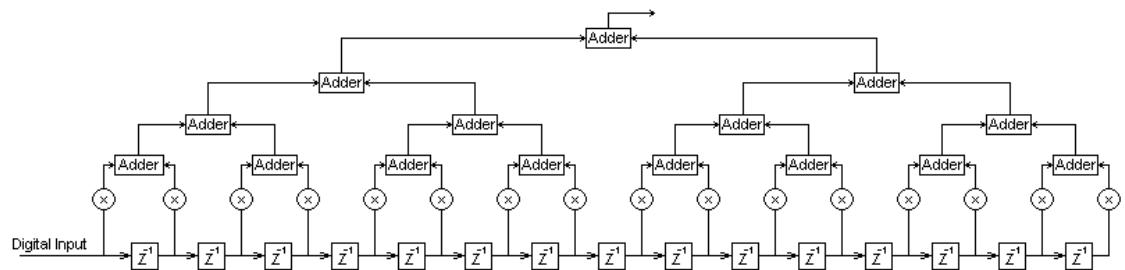


Figure 2.18 A 16-tap equalizer to illustrate the number of adders needed.

3 Dc Wander (Intersymbol Interference)

Ac coupling is effective for removing dc offsets, but it introduces some problems when a signal is not dc balanced (equal time in each logic state). Figure 3.1 illustrates what happens on an ac coupled transmission line when a long string of identical logic 1's is transmitted. The observed voltage decay is exponential and follows:

$$V_C(t) = V_i \cdot e^{-t/RC} = V_i \cdot e^{-t/\tau} \quad (3.1)$$

where V_i is a step input at time $t = 0$, R is the value of the termination resistance in ohms, C is the ac coupling capacitor in Farads and t is time in seconds. Depending on the value of the time constant of the system ($\tau = RC$) and how unbalanced the signal is, signal decay can significantly decrease the noise margins. As the decayed signal approaches the threshold, even a small amount of noise can cause bit errors. Large capacitors for the ac coupling are chosen to ensure the decay is slow.

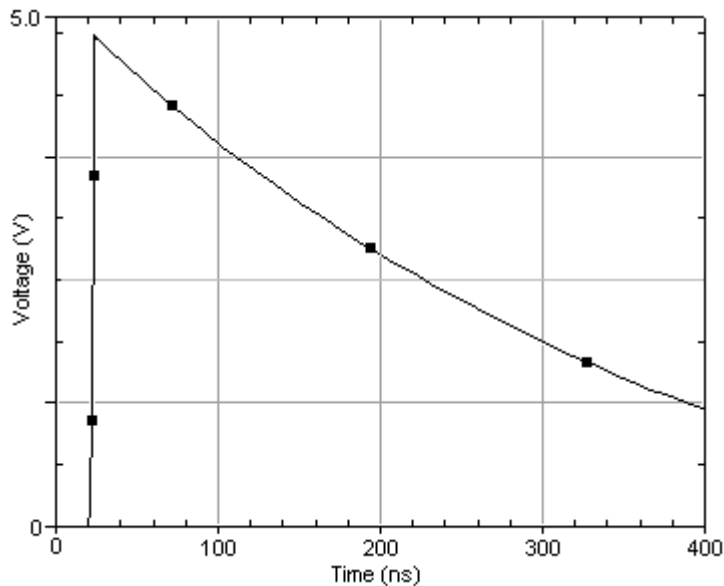


Figure 3.1 Example of a step input response on an ac coupled transmission line.

3.1 Possible Solutions to Dc Wander

There are several solutions to correct for the effects of dc wander all of which have disadvantages. Line codes (such as 4B5B or 8B10B) attempt to transmit the same number of logic 1's and 0's – to dc balance the signal – by adding redundant bits. Another balancing method is to scramble the transmitted signal to eliminate long strings of identical logic values. Special digital codes can also be used that are intrinsically balanced such as Manchester coding [13]. Alternate mark inversion (AMI) or bipolar return-to-zero (BPRZ) are ternary codes used to maintain dc balance in the binary data. Baseline restoration compensates for dc wander by forcing the signal voltage to a threshold level between signal periods. Decision feedback equalization uses filters to restore the signal after dc wander has distorted it. Each of these solutions will be examined next, and a comparison of the methods will be given.

3.1.1 Dc Balanced Signals

A *dc balanced* signal is the key to avoiding the dc wander problem experienced with ac coupled interconnect. It is loosely defined as a binary signal that contains equal numbers of each logic value over a specific time period. It is not sufficient to say that a signal is dc balanced if the average voltage over a long period of time is zero. It must also be balanced in time periods similar to the duration of the time constant ($\tau = RC$) of the ac coupling. This is a very important classification as *unbalanced* signals (where one logic level is much more prevalent than the other) are where ac coupling problems occur.

3.1.2 Dc Balancing Bits

If a dc balanced code is not used, data can be artificially balanced using special bits for balancing. A coding protocol is a set of rules for transmitting data which includes layers for information, error correction, framing bits and/or bits for dc balance.

A popular coding protocol is Integrated Services Digital Network (ISDN). ISDN reserves a certain number of bits specifically for improving signal balance. For example, if the data to be transmitted has a larger number of logic 1's than logic 0's, the

transmitter will even this out by making the appropriate number of dc balance bits equal to zero. Figure 3.2 shows an example of an ISDN frame format using balancing bits [4]. Line codes, like 8B10B, convert 8-bit bytes into 10-bit data characters to maintain (among other things) dc balance by using only the 10-bit characters with similar numbers of ones and zeros [2].

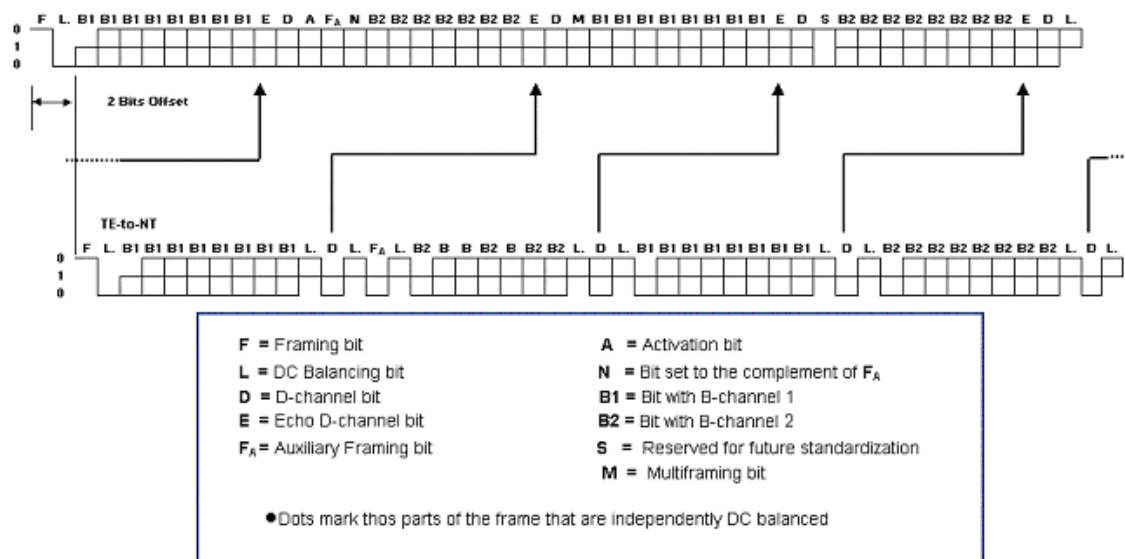


Figure 3.2 ISDN local networking frame format [4].

3.1.3 Dc Balanced Codes

There are several types of binary codes available and are shown in Figure 3.3 [13]. They all have advantages and disadvantages that make them suitable for different situations.

This research is interested in codes that support dc balance. The most desirable codes when concerned with dc wander are ones that contain no dc component. Using this criterion, unipolar NRZ is the least desirable as it has a dc component. However, it has one big advantage of requiring only one voltage supply, which makes it a popular choice for many IC manufacturers. Polar NRZ uses antipodal signaling which facilitates dc balancing provided the data toggles between the different logic values often. Manchester coding is always dc balanced as it satisfies the following equation:

$$\int_{t=0}^T A(t) dt = 0 \quad (3.2)$$

Other codes that also work well at reducing dc wander (without ac coupling) are Bipolar RZ and Bipolar AMI. These are ternary codes used to transmit binary information. A logic 0 is represented by 0V, and $\pm AV$ both represent a logic 1 in Figure 3.3. Each successive logic 1 is the opposite polarity than the previous which promotes dc balance. If there is a long string of identical logic values, the dc component remains at zero.

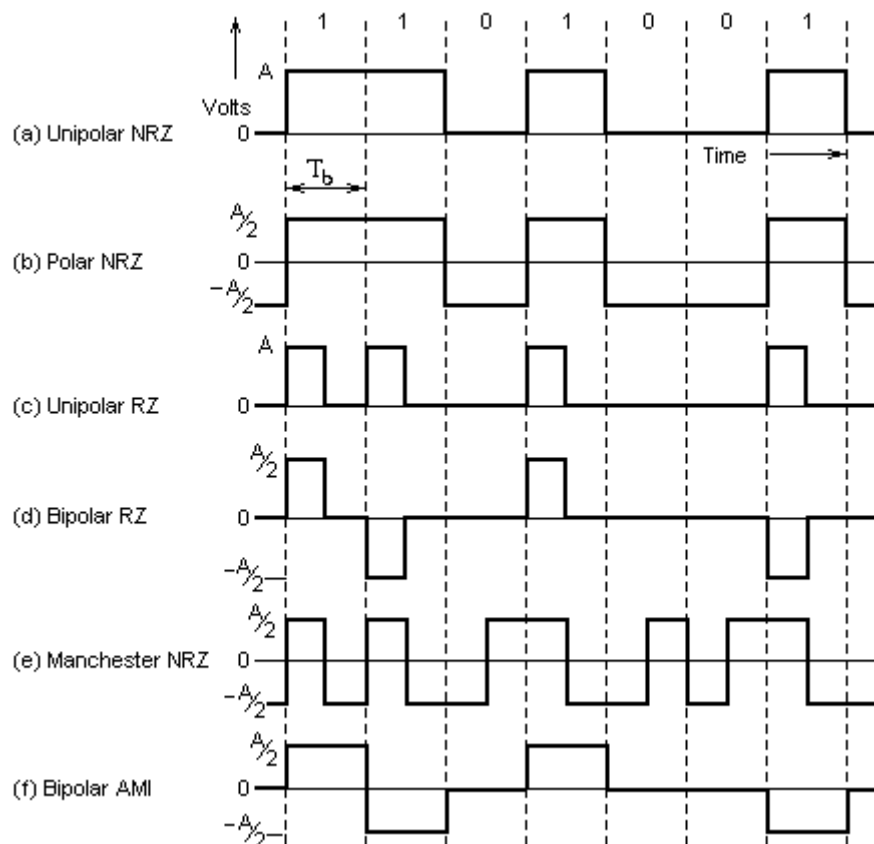


Figure 3.3 Binary signaling formats.

3.1.4 Scrambling Codes

Scrambling is used to randomize the pattern of bits in a data stream. Scrambling the bits breaks up continuous strings of 1's or 0's which would otherwise lead to a

wandering baseline. SONET (Synchronous Optical Network) (a very popular communication scheme) includes a system to scramble digital signals to create dc balanced signals from unbalanced signals. The signal is scrambled at the transmitter and unscrambled at the receiver using a structure known as a PN (pseudo-noise) sequence generator. The International Telecommunications Union (ITU-T) GR-253 standard defines the polynomial $1+x^6+x^7$ algorithm for a SONET scrambler which is shown in Figure 3.4 [2]. The SONET scrambling scheme does not *guarantee* a balanced code; there have been cases of its failure from problem data sequences [14].

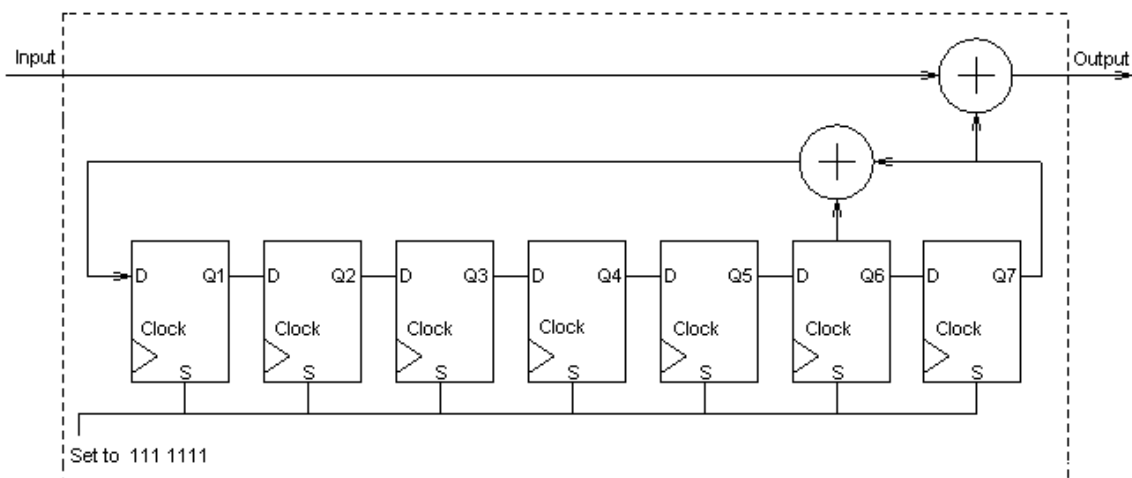


Figure 3.4 SONET scrambler/descrambler.

Killer packets are long lines of code that are designed to make a system fail by *aligning* with the SONET scrambler. When this alignment happens, it creates very long runs of consecutive identical logic levels, which makes the signal unbalanced, and ultimately, the scrambler fails. A data scrambler is designed to change a transmitted data pattern to avoid unbalanced signals. Although the probability of randomly encountering a string of data needed to make a scrambler fail is acceptably remote [14], it can happen. Killer packets were developed (perhaps maliciously) to prove that data scramblers were vulnerable. On an ac coupled transmission line, this will cause the baseline to wander severely and cause transmission errors.

3.1.5 Baseline Restoration and Decision Feedback

Another technique for offsetting dc wander is referred to as dc or baseline restoration [2]. Received pulses are fed through a capacitor and the charge on the capacitor is removed before the next pulse arrives. The capacitor charge is removed by driving the voltage to a specific threshold. Once the threshold is reached, the voltage is removed as shown in Figure 3.5. Since all the charge is removed after each pulse, the baseline or decision reference level is constant at the beginning of each signal interval.

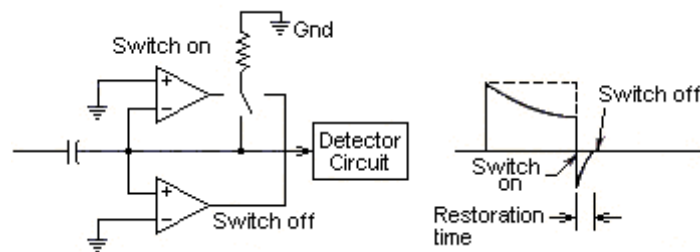


Figure 3.5 Baseline restoration for unipolar codes [2].

Decision feedback, also called *quantized feedback equalization*, is a more useful technique for reducing baseline wander [2]. Figure 3.6 shows the basic idea behind a decision feedback design. The general concept for this type compensation is to create an analog feedback signal to reverse the dc wander effects imposed on the data by the ac coupling.

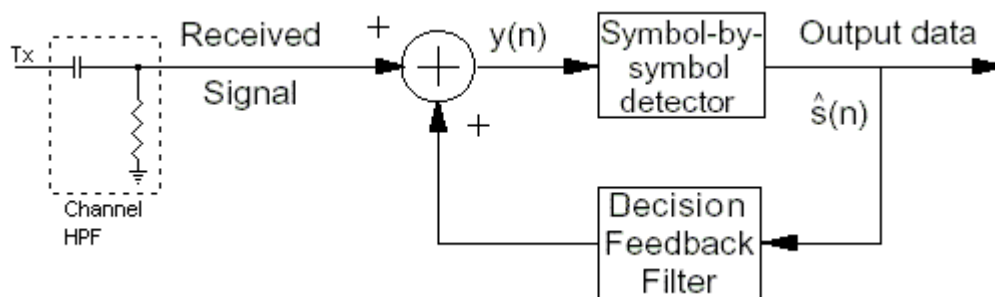


Figure 3.6 Structure of decision feedback equalizer.

The receiver samples the incoming signal and the original data is recovered (hopefully without errors). The reconstructed digital data $\hat{s}(n)$ is then passed through the decision

feedback filter to generate a signal to offset the dc wander distortion. The decisions made in the receiver are binary and thus the values entering the digital filter consist of either a logic 1 or a logic 0. However, signals within the filter have several bits of precision to obtain the correct compensation value. This signal is then added to the received signal, removing the distortion.

3.2 Comparison of Dc Wander Solutions

To better understand the several possible dc wander solutions, it is helpful to compare the advantages and disadvantages of each.

- Regarding data codes, Manchester coding offers the best solution with no dc component. Bipolar RZ and AMI are also viable options with favorable dc balance characteristics. Manufacturers have different reasons for choosing the codes used on their IC's. Selecting the "best" code for a particular situation usually involves balancing a number of mutually exclusive options. Unipolar RZ needs only one power supply voltage which makes it popular with some manufacturers. However, polar NRZ (antipodal) signals offer a superior bit error rate (BER) for a given signal-to-noise ratio (SNR) and a greater possibility of signal dc balance [2]. This is one method used as a viable option for reducing dc wander.
- Signal balancing is another option used to correct the problem of dc wander. This method adds appropriate bits (logic ones or logic zeros) to the outgoing signal to keep the signal balanced (same number of each logic level). This is currently used in certain coding architectures such as ISDN, but relatively few balancing bits per byte limits the effectiveness of this method.
- Scrambling codes use PN generators, called scramblers, to change the code before transmitting. This breaks up long strings of identical logic values that could normally lead to dc wander; SONET was given as an example. Although SONET is a widely used methodology and can be very effective at scrambling the signal, it is susceptible to problem data sequences, called 'killer packets' which could cause the system to fail.

- Obvious disadvantages of baseline restoration are that the signal input must have a zero amplitude level and the system must be disabled during the reset time. These are not feasible options in a high speed system. The bit pulses are usually coded in polar NRZ so there is no time for the charge on the capacitor to be removed.
- In a decision feedback system, the signal loss due to ac coupling is recreated using a digital filter and added back to the incoming signal in real time. It is a simple solution that is scalable and easy to design and implement. Problems with this approach are given below.
 - The decision feedback filter would be located within the IC, so the characteristics of the interconnect channel would either have to be known prior to fabrication of the chip or have the filter parameters programmable. This could be improved if the circuit was adaptive, but would create much more circuitry.
 - Multiplier values for the digital filter have few bits of precision. Attention must be paid to choosing the values to avoid instability in an IIR filter. Adders are also quite large, which could limit the feasibility of this method.

Despite the disadvantages of decision feedback equalization, it is the solution chosen in this research because of the simplicity of the design, and the ability for the feedback filter to closely model a complimentary effect to the ac coupling characteristics.

3.3 Signal Flow in a Decision Feedback System

The discussion of decision feedback begins by assuming the receiver samples the incoming data correctly (i.e. correct sample timing). This is important since a short history of correct decisions is needed for the feedback filter to generate the correct response. From this decision history, a compensation signal is generated by the decision feedback filter that is complimentary to the channel highpass filter (HPF). This compensation signal is then added to the received signal (before the signal is sampled) to restore the signal components that had been removed by the channel HPF (ac coupling). This process effectively reconstructs the original signal and is illustrated in Figure 3.7.

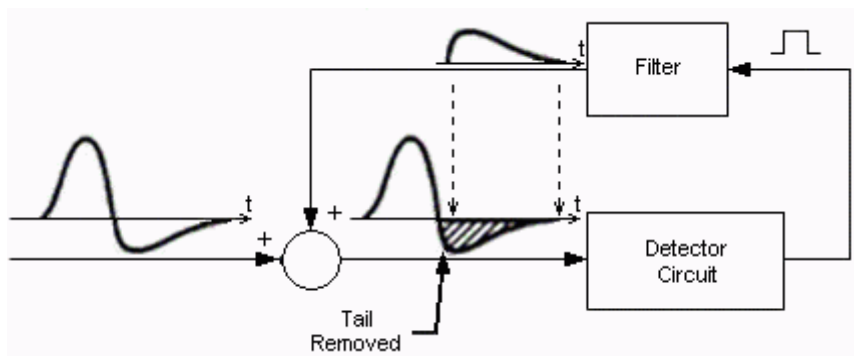


Figure 3.7 Decision feedback equalization [2].

4 Decision Feedback for Dc Wander Compensation

4.1 Digital Filter Structures

Decision feedback has been proposed as an effective method for restoring a digital baseband signal suffering from dc wander. As described in Section 3.3, decision feedback uses a digital filter to create an analog signal that is a close approximation to the portion of the transmitted signal that is lost due to ac coupling. Both IIR and FIR digital filters were discussed in Sections 2.5.1 and 2.5.2 respectively. The dc wander decay continues to reduce to infinite time which can be matched using a filter with an infinite impulse response. What is not known is if an effective FIR filter can be used (given the increase in size it has over the IIR filter). Therefore both configurations will be investigated and the results compared. The filter designs will be covered first, followed by their integration into the decision feedback system.

4.2 Filter Development

Before beginning with the development of a filter, the effect of the ac coupling is studied. The Laplace domain transfer function of an analog first order highpass filter is given by

$$H(s)_{HPF} = \frac{s}{s + \omega_c} \quad (4.1)$$

where ω_c is the cutoff frequency of the filter. The compensating lowpass filter within the decision feedback system must be a corresponding first order system. A compensating analog lowpass filter transfer function is therefore given by

$$H(s)_{LPF} = \frac{\omega_c}{s + \omega_c} \quad (4.2)$$

with the same cutoff frequency, ω_c , as in Equation 4.1.

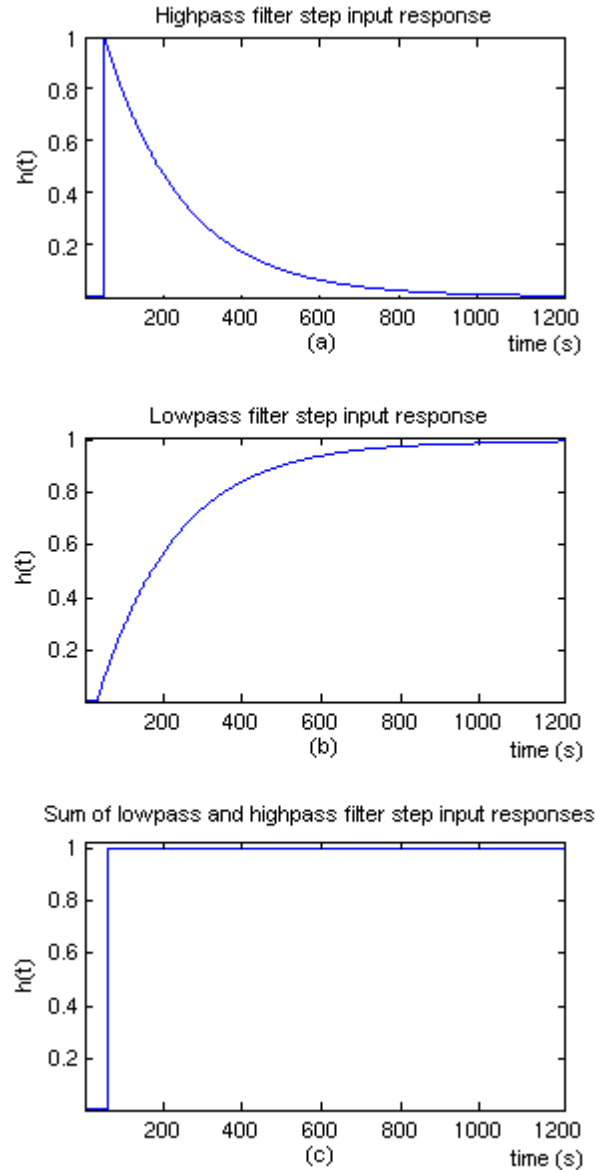


Figure 4.1 Unit step input responses (a) highpass filter, (b) lowpass filter, (c) sum of step input responses of (a) and (b).

Figure 4.1(a) and Figure 4.1(b) show the results of imposing a unit step input on the transfer functions of Equations 4.1 and 4.2, respectively. The step input represents the transition to a logic 1 in a data stream. From Figure 4.1, the highpass filter step response in (a) experiences a decay proportional to $e^{-\omega_c t}$ while the lowpass (b) experiences an exponential increase proportional to $(1 - e^{-\omega_c t})$. When the lowpass filter

is incorporated into the decision feedback system, these responses will be added together, and an *ideal* result would be seen in the frequency domain similar to

$$H(s)_{LPF} + H(s)_{HPF} = \frac{\omega_c}{s + \omega_c} + \frac{s}{s + \omega_c} = \frac{s + \omega_c}{s + \omega_c} = 1.$$

Similarly, this *ideal* compensation in the time domain would be

$$e^{-\omega_c t} + (1 - e^{-\omega_c t}) = 1$$

which is shown graphically in Figure 4.1(c). This is the basis of the development of both the IIR and FIR digital filters.

4.2.1 Infinite Impulse Response (IIR) Filter Design

A first order lowpass IIR digital filter is a very simple design and the block diagram is shown in Figure 4.2. The delay element advances a data value every clock period T_s (T_s is the reciprocal of the sampling frequency f_s).

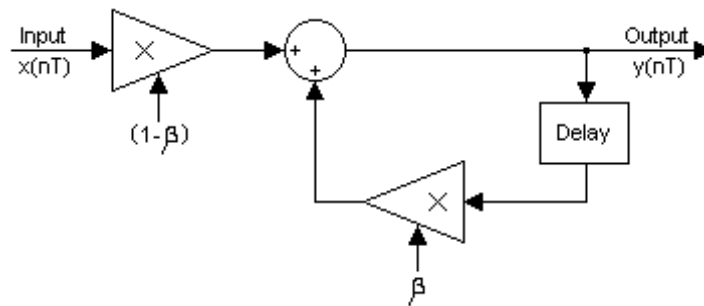


Figure 4.2 First order lowpass IIR digital filter.

The multiplication factor, β , together with the sampling rate f_s , sets the cutoff frequency, ω_c , of the filter; this cutoff frequency must match that of the ac coupling ‘highpass filter’. The multiplier coefficient, β , from Figure 4.2, is derived from the lowpass filter expression (Equation 4.2) using the Bilinear Transform. The details of this derivation are given in Appendix A and results in the difference equation

$$y(nT_s) = (1 - \beta)x(nT_s) + \beta \cdot y((n - 1)T_s).$$

The clock period T_s is the sampling period of the receiver. The value of β is calculated based on an estimate of the time constant τ of a practical system used in industry. Using time constant of $\tau = 200 T_s$ in the calculations results in a multiplier parameter of $\beta = 0.995$.

4.2.2 Finite Impulse Response (FIR) Filter Design

Dc wander can also be compensated using a FIR digital filter. The FIR filter is a tapped delay line that requires a multiplier for every line tap (Figure 4.3). Two different methods can be used to calculate the multiplier coefficients. Emulating the previously designed IIR filter is one method, where the delay period, T_s , is the same for the FIR filter. Each multiplier coefficient for the FIR filter can be calculated using the feedback coefficient of the IIR filter.

$$\text{tap coefficient}(n) = (1 - \beta)\beta^n, \quad n = 1, 2, 3, \dots, M - 1$$

where M is the length of the FIR filter.

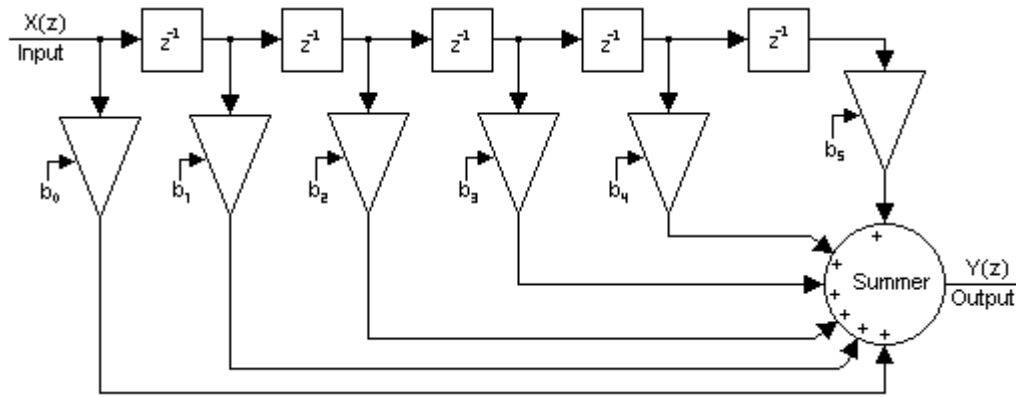


Figure 4.3 FIR digital filter with $M = 6$.

The second method of determining the coefficients is by using the time domain expression representing the decay of the ac coupling $y(t) = e^{-t/\tau}$. Making the equation discrete (replacing t by nT_s), the multipliers are found by calculating the amount of decay that occurs between successive time periods (T_s):

$$\text{tap coefficient}(n) = e^{-nT_s/\tau} - e^{-(n+1)T_s/\tau}, \quad n = 1, 2, 3, \dots, M - 1$$

and the complete filter is shown in Figure 4.3 with $M = 6$.

Clearly as the length of the filter increases, it will more closely emulate the IIR filter and the compensation improves. For long time constants ($\tau \geq 200 T_s$) the filter becomes prohibitively long. For the purpose of completeness, both the IIR and FIR filters will be simulated and compared in the following sections.

4.3 Simulation Strategy

The simulation of the complete decision feedback design is done in Simulink[®]. To do this, a virtual system is needed which includes a

- polar signal transmitter,
- transmission line simulator,
- ac coupling simulator (analog domain),
- analog-to-digital converter,
- additive white gaussian noise (AWGN) channel,
- receiver, and a
- digital filter for decision feedback circuit.

Each of the simulation subsystems listed above, contain one or more Simulink[®] blocks which perform a particular task within the communication system. A description of the signal flow is discussed in the next section. The blocks within each subsystem are discussed in detail in Appendix B to describe their purpose and parameters.

Simulations are performed using each of the FIR and IIR filters in the decision feedback system. Comparing the results will show whether one is more useful than the other. Simulations will also demonstrate the effectiveness of different lengths of FIR filter. In the simulations using Simulink[®], a pseudo-random polar NRZ signal is generated as an input data sequence to the system. The user can change the pseudo-random sequence by entering a different “seed” value to the binary code generator

block. The transmission line effects are incorporated into a Matlab[®] program used to simulate the high frequency roll-off and phase shift that the waveform would experience on an actual transmission line. This is done by convolving the pseudorandom binary data sequence with a time domain representation of a transmission line transfer function. The resulting data sequence is an attenuated version of the original. This new data sequence is used as an input to the remainder of the system, which contains the digital compensation filter.

The ac coupling function is simulated using a Simulink[®] transfer function block. The simulation also provides a user-input bit energy to noise spectral density ratio (E_b/N_0) to observe the effect of varying amounts of noise with dc wander and its compensation. In order to establish the correctness of the simulated system and the noise model, simulations were first run without ac coupling, and the bit-error rates recorded. By varying E_b/N_0 , the bit error rate (BER) can be compared with theoretical values. A chart with theoretic BER values for different coding schemes is shown in Figure 4.4.

Signals that have a dc balanced symbol pattern will minimize dc wander but unfortunately, dc balanced symbol patterns are not always guaranteed to occur. To investigate the effects of unbalanced symbol patterns, noise is added to signals with varying degrees of dc unbalance. As the signal becomes more dc unbalanced, the amount of dc wander compensation increases.

In order to obtain reasonably accurate simulation results, large amounts of data must be used. For a probability of error (or BER) of 10^{-6} , at least 10^6 bits must be fed through the system (enough to get one error). For greater degrees of accuracy many more bits should be used. E_b/N_0 is the ratio of received energy per bit, in joules, to the noise spectral density in watts per hertz. To create a BER graph, the range of E_b/N_0 values is chosen to be 2 dB – 12 dB with much more data used for the cases where the error rate is very small.

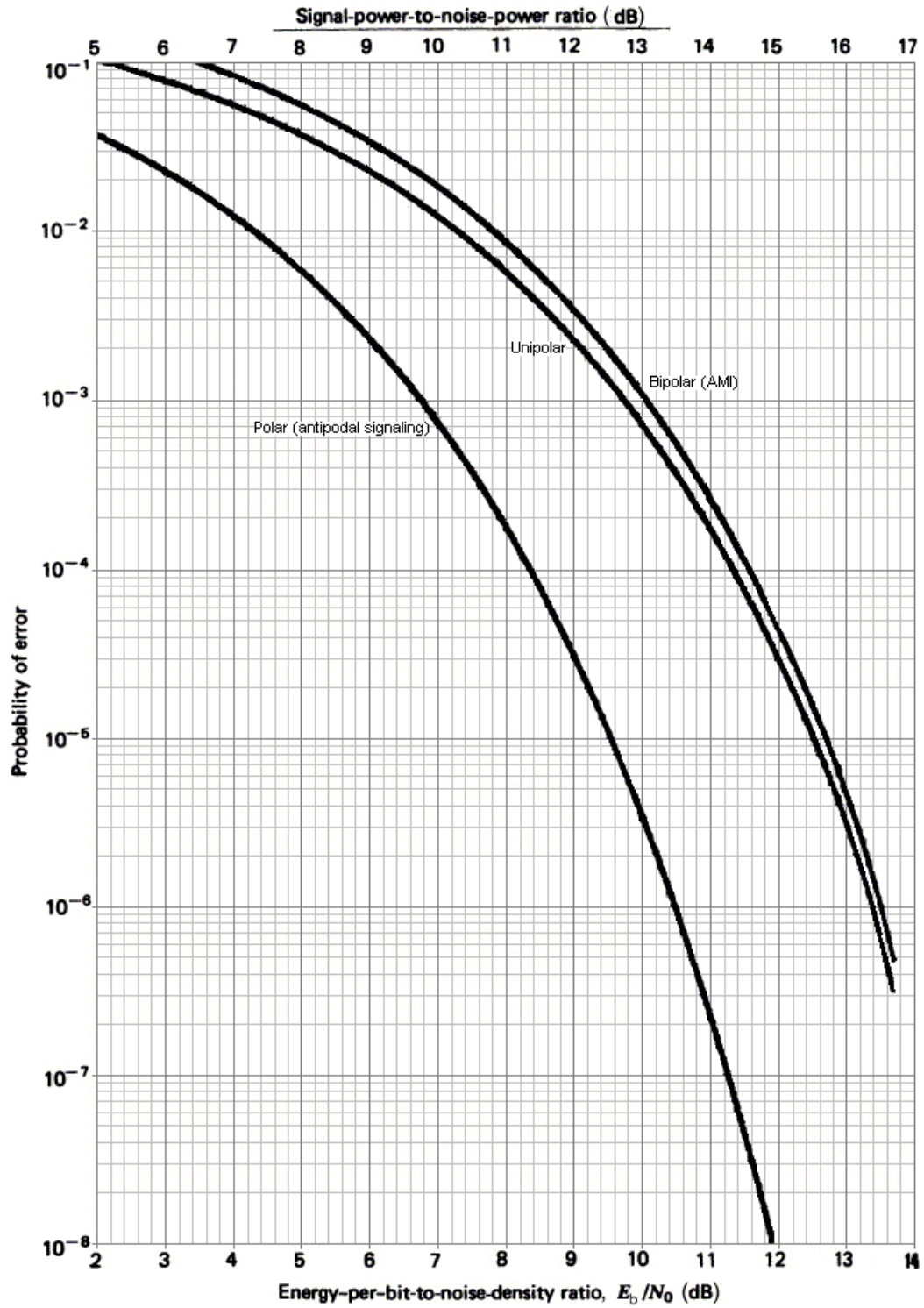


Figure 4.4 Theoretic BER for polar (NRZ), unipolar (NRZ), and bipolar line codes [2].

Table 4.1 Simulation results demonstrating increases in data depending on E_b/N_0 level.

E_b/N_0 (dB)	Data bits (Mbits)	Errors
2	0.5	23613
3	0.5	15069
3.5	0.5	11724
4	0.5	8901
4.5	0.5	6457
5	0.5	4649
5.5	0.5	3211
6	0.5	2113
6.5	0.5	1332
7	0.5	832
7.5	0.5	473
8	0.5	275
8.5	0.5	125
9	1.0	136
9.5	2.0	113
10	4.0	81
11	11.0	24
12	20.0	4

The simulation of a single test vector of length 5×10^5 bits takes 15 minutes to perform. For simulations with low E_b/N_0 and high error rates, one simulation run of 5×10^5 bits is enough, but for low noise levels (9 dB to 12 dB) longer vector lengths are needed as demonstrated in Table 4.1. In the interest of time, the data vector lengths are limited to the minimum length needed to attain reasonable accuracy. Therefore, these graphs are not perfectly accurate due to the relatively small number of samples used to create them. However, they are sufficiently accurate to allow insight into the operation of the system.

4.4 Simulation Results

Simulations were run with 5×10^5 bits using a first order IIR filter and two different FIR filters of length $M = 10$ and 50 . The simulations are repeated for varying degrees of noise ($E_b/N_0 = 2 \text{ dB} - 12 \text{ dB}$, where E_b is the energy per bit which is calculated in Appendix A) and varying degrees of dc signal balance to demonstrate the effectiveness of the filters. The degrees of dc balance used in the simulations were (using probability of zero ' P_0 ' values) $P_0 = 0.5, 0.4$ and 0.3 . It was expected that the IIR filter would recover the signal best and as the length of the FIR filter increased, the closer the BER would approach the values for a dc coupled system. It was also expected that all the filters would be most effective (make the most improvement) when there was a high degree of dc imbalance.

Analyzing the spectrum of the signal at various points in the communication system will show spectral changes before and after the ac coupling, distortion from the transmission line, and dc wander compensation.

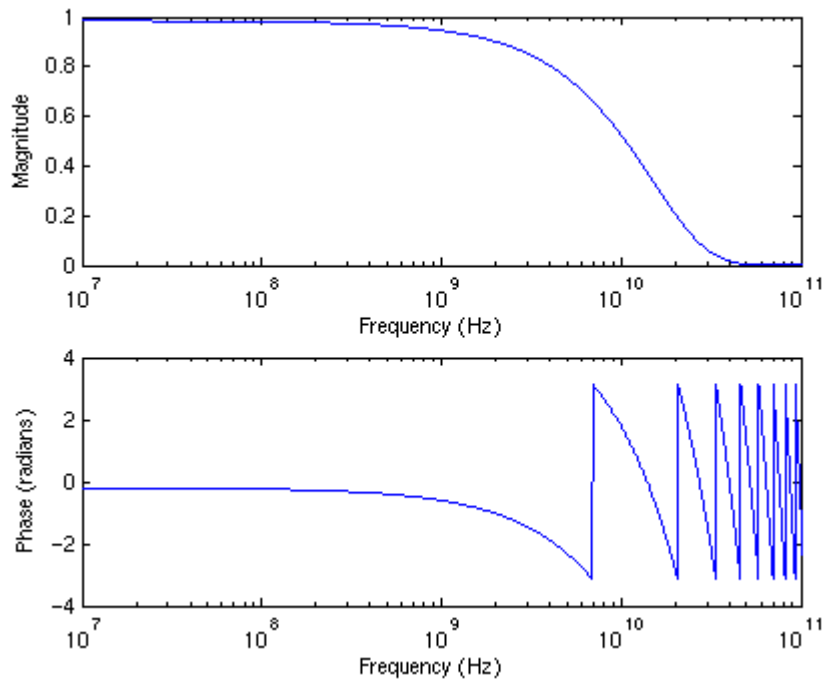


Figure 4.5 Transmission line magnitude and phase response (1 meter).

A transmission line affects the digital signal by removing the high frequencies and by introducing phase distortion. This means that different frequencies are shifted in phase by different amounts. The transmission line magnitude and phase response is shown in Figure 4.5 and the magnitude response is shown again on a linear frequency scale in Figure 4.6(b). The attenuation increases as the frequency increases. When a pulse travels along a transmission line, high frequencies are attenuated. Figure 4.6(a) illustrates the spectrum of a 320 ns pulse while Figure 4.6(c) illustrates the spectrum after the pulse has traveled through 1 meter of transmission line.

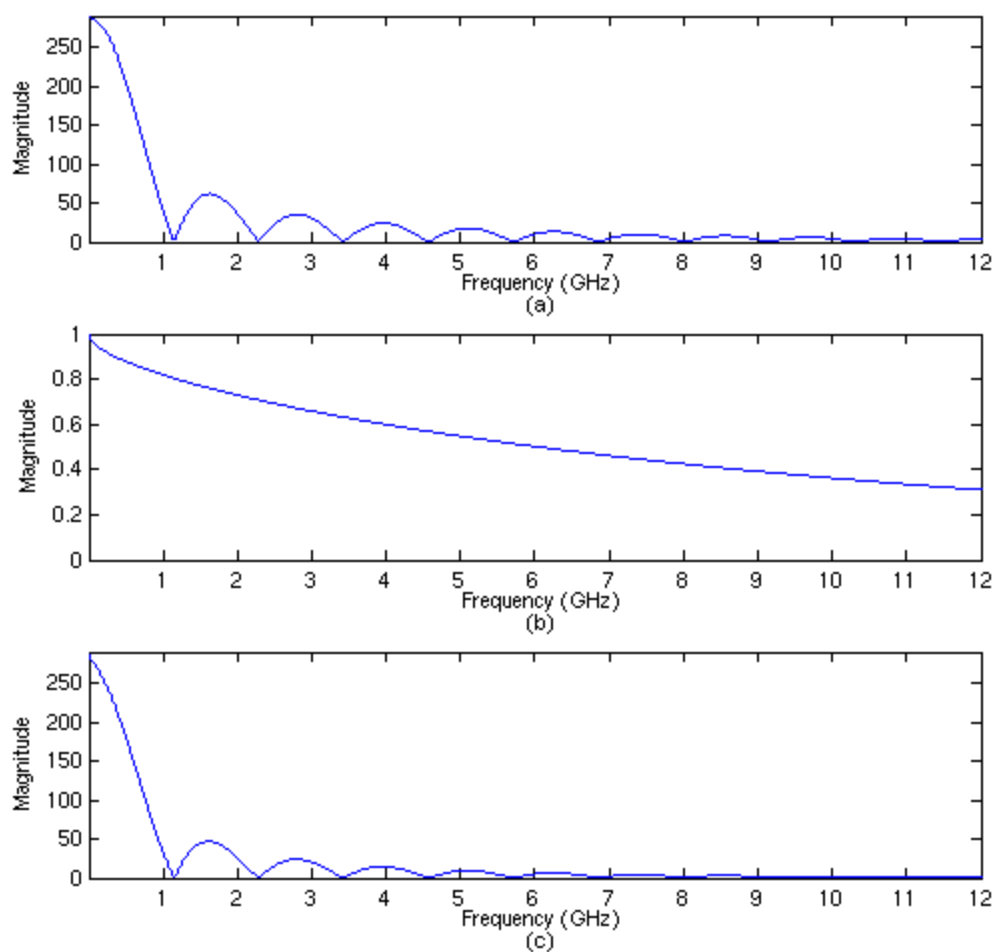


Figure 4.6 (a) Spectrum of 320 ns pulse, (b) magnitude response of 1 meter of stripline transmission line in FR4 circuit board, and (c) spectrum of 320 ns pulse after traveling through 1 meter transmission line.

The removal of the high frequencies in a single ideal pulse has the effect of making the sharp edges of pulse become rounder. The pulse also experiences phase shifts for different frequencies which slightly delay the arrival of the pulse at the receiver. The phase shifts are also a cause of distortion in the pulse waveform. These effects are demonstrated for a single pulse in the time domain in Figure 4.7. The pulse shown in Figure 4.7(a) is a 300 ps pulse with 90 ps rise times. As determined by simulation, the distorted pulse (after 20 cm of transmission line) is shown in Figure 4.7(b). This distortion is particularly dependent on the length of the transmission line. As the length of the line is increased, the distortion becomes worse until a point is reached where the signal is no longer recognizable.

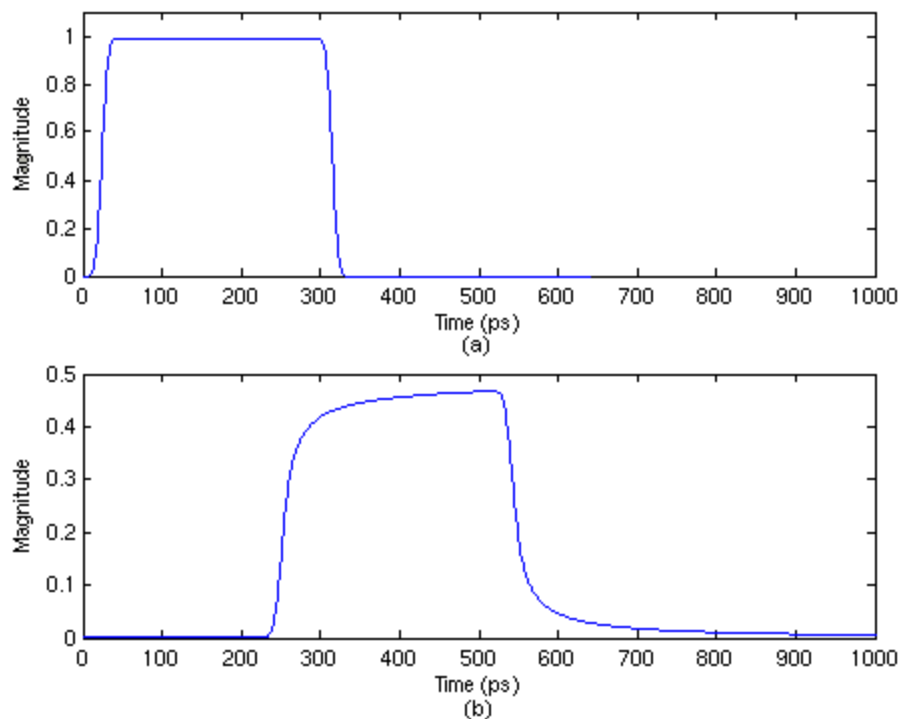


Figure 4.7 Time domain representation of pulse with 90 ns rise time at transmitter and showing attenuation after traveling through a transmission line.

An actual signal (a random sequence of binary 1's and 0's) will have a frequency response similar to that of a pulse which will not have infinitely short rise times (as they were in the earlier simulations used to generate the graphs in Figure 4.6). Longer rise times result in fewer high frequency components in the signal. The

spectrum of a sequence of such bits is shown in Figure 4.8(a). The effect of the transmission line can be observed in Figure 4.8(b). Once the pulse travels through a transmission line (20 cm is used in this simulation) many of the higher frequencies are completely removed and the lower frequencies are attenuated.

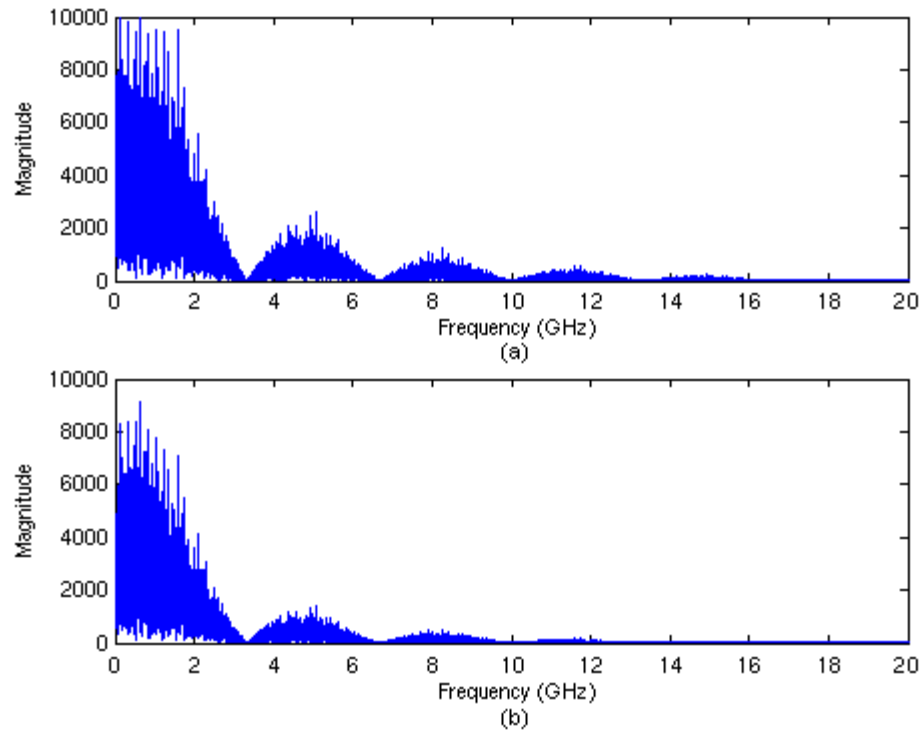


Figure 4.8 Spectrum of a pseudo-random polar NRZ signal (a) with 90 ps rise/fall times and (b) after ac coupling and propagating through 20 cm of (stripline) transmission line.

A Simulink[®] transfer function is used to simulate ac coupling. Since this is a first order highpass filter, the dc component in the spectrum is removed. The spectrum in Figure 4.8(b) includes the simulated ac coupling of a polar NRZ signal and demonstrates the attenuation of low frequencies near zero Hertz and the removal of the dc component.

The most important measurement point in the circuit is just before the receiver. If the signal is not recognizable at the receiver, it is impossible to recover the original data. An eye diagram is created by superimposing multiple delayed bits (of the same signal) on the same axis [15]. The eye diagram is very useful to observe the quality of

the signal as seen by the receiver. It is also helpful to use the eye diagram to view the signal before and after the proposed decision feedback equalization. Figure 4.9 shows an eye diagram affected by dc wander and transmission line effects. As described in Section 2.3, the eye opening is the measure of the quality of a signal. The more the eye is open, the greater the chance of recovering the signal with few or no bit errors. Eye diagrams will be used in several cases during the simulation results to demonstrate the received signal quality.

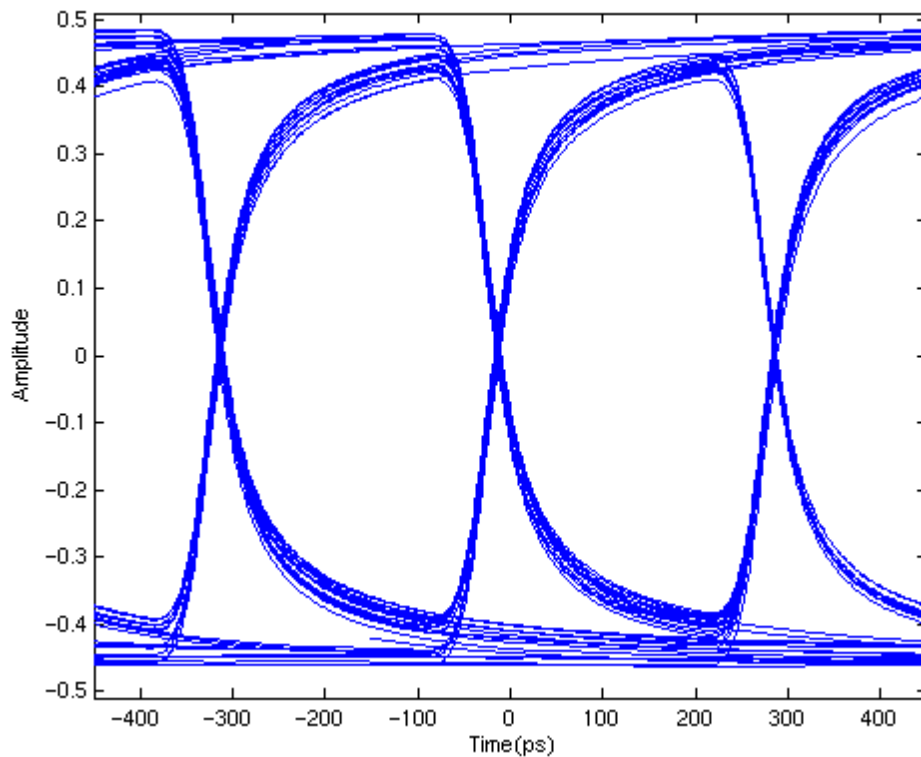


Figure 4.9 Eye diagram from simulation affected by dc wander and transmission line effects.

The first simulation result, shown in Figure 4.10, is not ac coupled and Figure 4.10(a) has a bit energy to noise ratio $E_b/N_0 = 14dB$. Clearly, the eye is open at the sampling point (middle of the eye) and the signal is far away from the threshold range. Figure 4.10(b), (c) and (d) also show balanced signals without ac coupling, but with varying amounts of noise. These plots demonstrate as noise is increased, the eye opening is reduced, and ultimately the bit error rate increases.

The noise margin is reduced as ac coupling is introduced and gets progressively worse as the signal becomes more unbalanced. For the simulations all three parameters (dc balance, noise, and compensation type) are varied to demonstrate the effect each has on signal integrity.

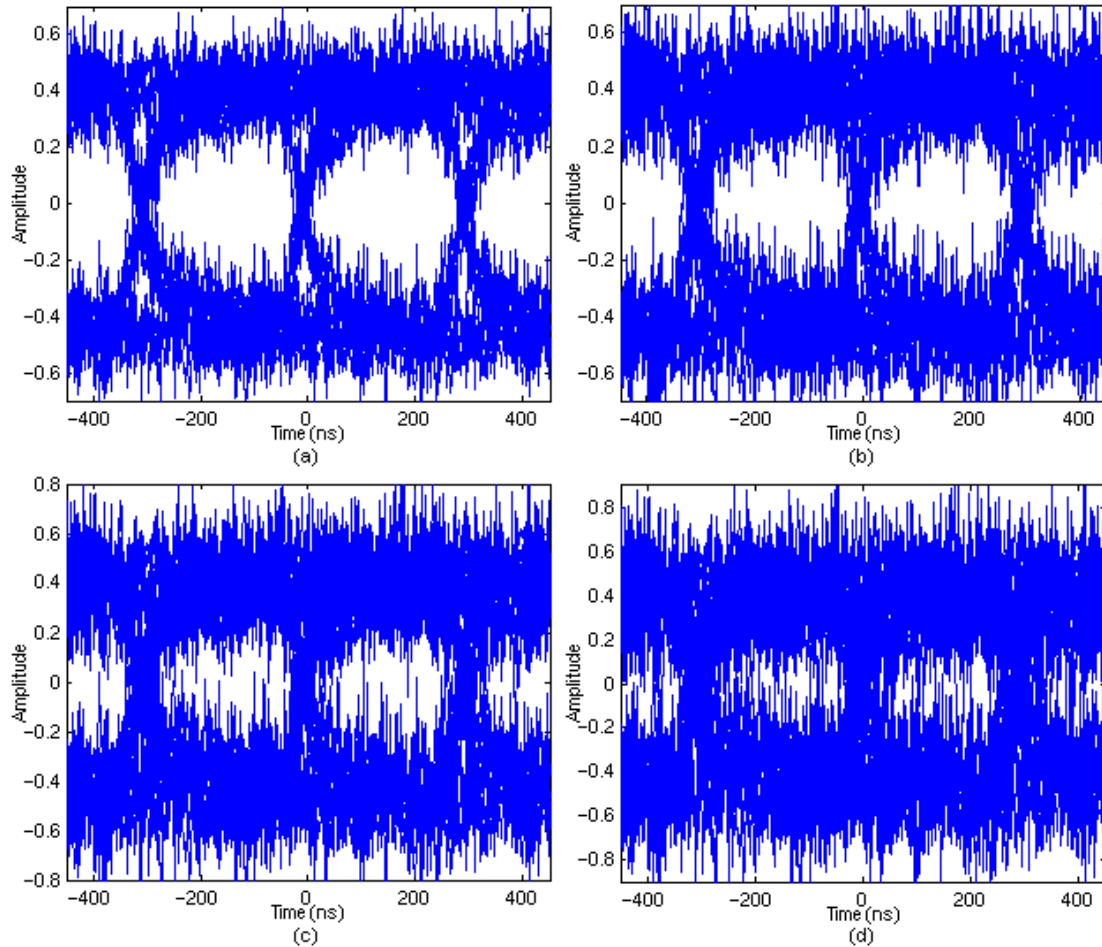


Figure 4.10 Eye diagrams of non ac-coupled signal attenuated by transmission line effects and added AWGN of (a) $E_b/N_0 = 14$ dB, (b) $E_b/N_0 = 12$ dB, (c) $E_b/N_0 = 10$ dB and (d) $E_b/N_0 = 8$ dB.

Figure 4.11 uses eye diagrams to show the effects of ac coupling when the signal is not dc balanced. The probability of zero is set in the simulation to provide an unbalanced signal ($P_{\text{zero}} = 0.3$) and the decision feedback amplitude has been scaled in proportion to the received signal amplitude. In Figure 4.11(a) and Figure 4.11(c) the eye has become somewhat thicker and is positioned off-center in relation to the threshold

(zero volts). The effect of the time constant is evident as the dc wander in Figure 4.11(a) ($\tau = 100T$) is more severe than in Figure 4.11(c) ($\tau = 200T$). Figure 4.11(a) and Figure 4.11(b) are the same signals, but the latter includes dc wander compensation; similarly with Figure 4.11(c) and Figure 4.11(d). The decision feedback equalization for these demonstrations was done using the first order IIR filter designed in Section 4.2.1 with the time constant matched to that of the ac coupling. The compensated eye remains open for both ac coupling time constants and is centered in relation to the threshold.

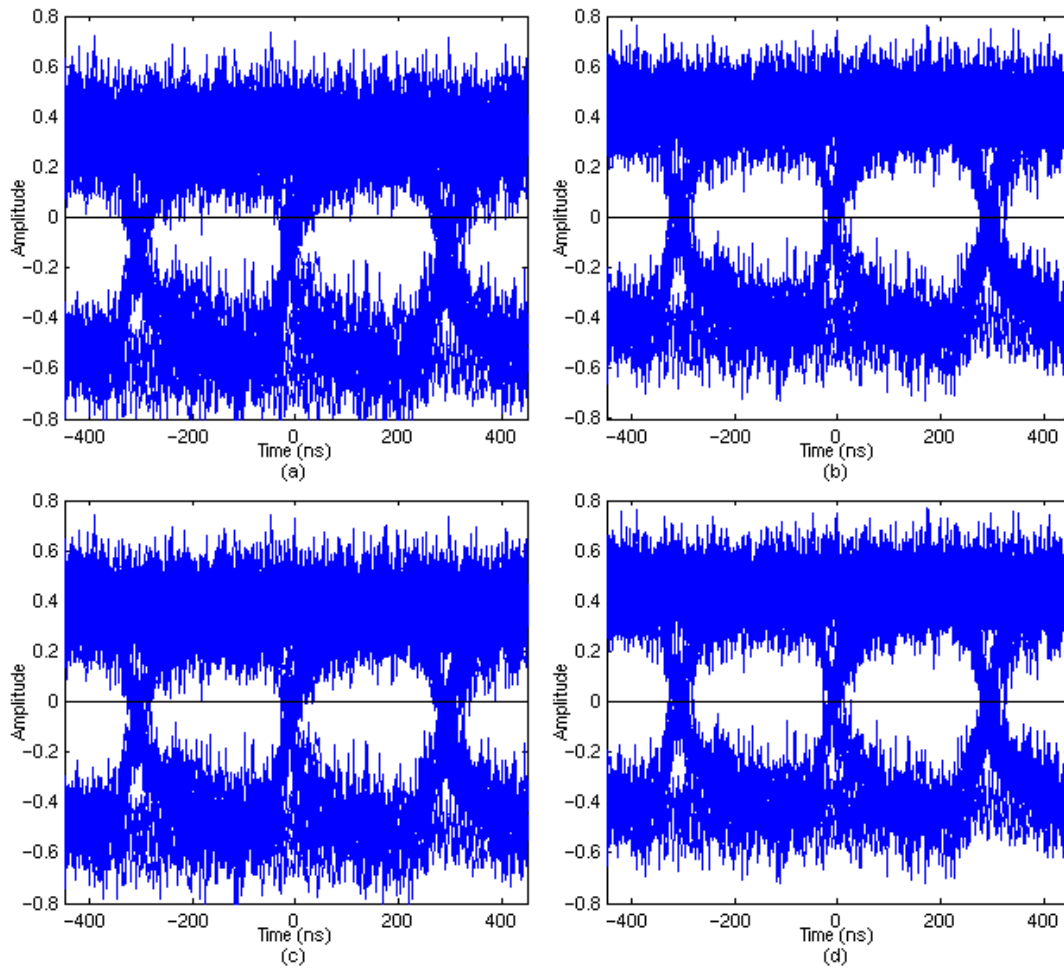


Figure 4.11 Eye diagrams of ac coupled, unbalanced ($P_{\text{zero}} = 0.3$) signal with added noise ($E_b/N_0 = 14$ dB). (a) Time constant ($\tau = 100T$) uncompensated and (b) signal compensated with digital IIR filter. (c) Time constant ($\tau = 200T$) uncompensated and (d) signal compensated with digital IIR filter.

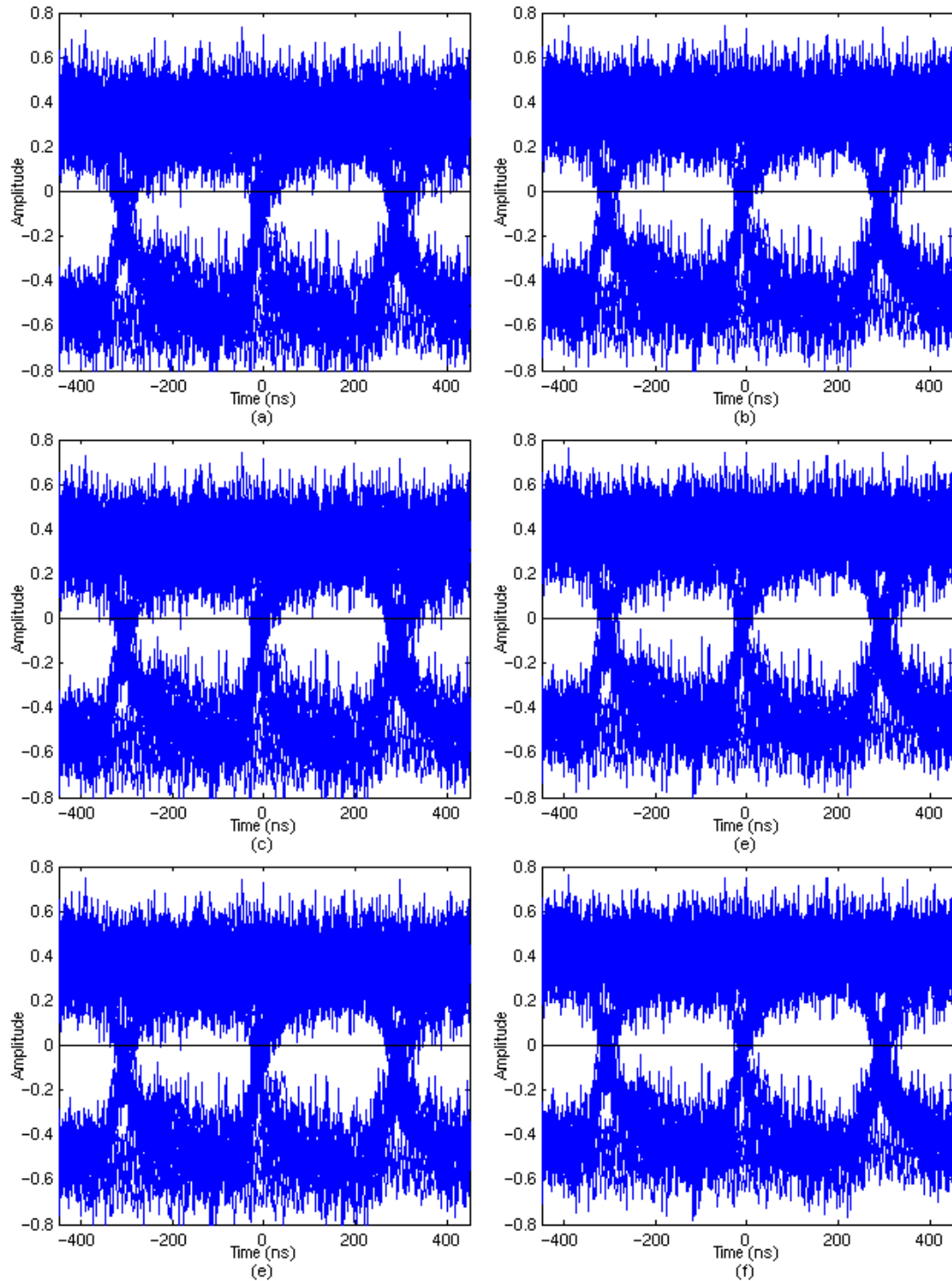


Figure 4.12 Eye diagrams of unbalanced signal ($P_{\text{zero}} = 0.3$). (a) Uncompensated ($\tau = 100T$) and (b) uncompensated ($\tau = 200T$). (c) Signal compensated with 10-tap FIR filter ($\tau = 100T$) and (d) compensated with 10-tap FIR filter ($\tau = 200T$). (e) Signal compensated with 50-tap FIR filter ($\tau = 100T$) and (f) compensated with 50-tap FIR filter ($\tau = 200T$).

The compensation demonstration is repeated for the 10 and 50-tap FIR digital filter cases, and for two different ac coupling time constants. The results are shown in Figure 4.12. The left side column of the illustration shows the uncompensated ac coupled eye, followed by the ac coupled eye compensated with a 10 and 50-tap FIR decision feedback equalizer. It is clear the FIR filter compensation is effective, but not as good as the IIR. Using the results from the eye diagrams, it is expected that the IIR filter will have a lower BER compared to that of the FIR filters which will still have an improved BER over the uncompensated circuit.

The simulation results, as expressed in BER (probability of bit error P_e), are summarized in the next few graphs. Figure 4.13 is the bit error rate graph for a balanced signal (i.e. approximately equal number of logic 1's and logic 0's) with three different forms of compensation. Although it is difficult to distinguish between some of the lines, it can be seen that the filters have little effect in this situation. The filters show their usefulness when the signal becomes unbalanced.

Figure 4.14 shows the simulation results for an unbalanced signal where the degree of imbalance is 40 percent zeros and 60 percent ones ($P_{\text{zero}} = 0.4$). This demonstrates the differences in the compensation filters more clearly. The IIR filter keeps the BER close to theoretical values (without ac coupling), the 50-tap FIR filter performs reasonably well, and the 10-tap FIR filter provides little improvement. In a real system this degree of imbalance may occur infrequently when averaged over a long period of time, but when a smaller sample size is considered, the signal could show high degrees of imbalance.

The third simulation results graph, shown in Figure 4.15, demonstrates an even higher degree of imbalance at $P_{\text{zero}} = 0.3$. The difference in the compensation filters now becomes quite evident. The IIR filter maintains the BER very close to theoretical values while the 10 and 50-tap FIR filters' performance lags far behind. Clearly, in the interests of effective compensation of dc wander and efficient use of silicon real estate, the benefits of the IIR digital filter are far greater than that of both the 10 and 50-tap FIR filters.

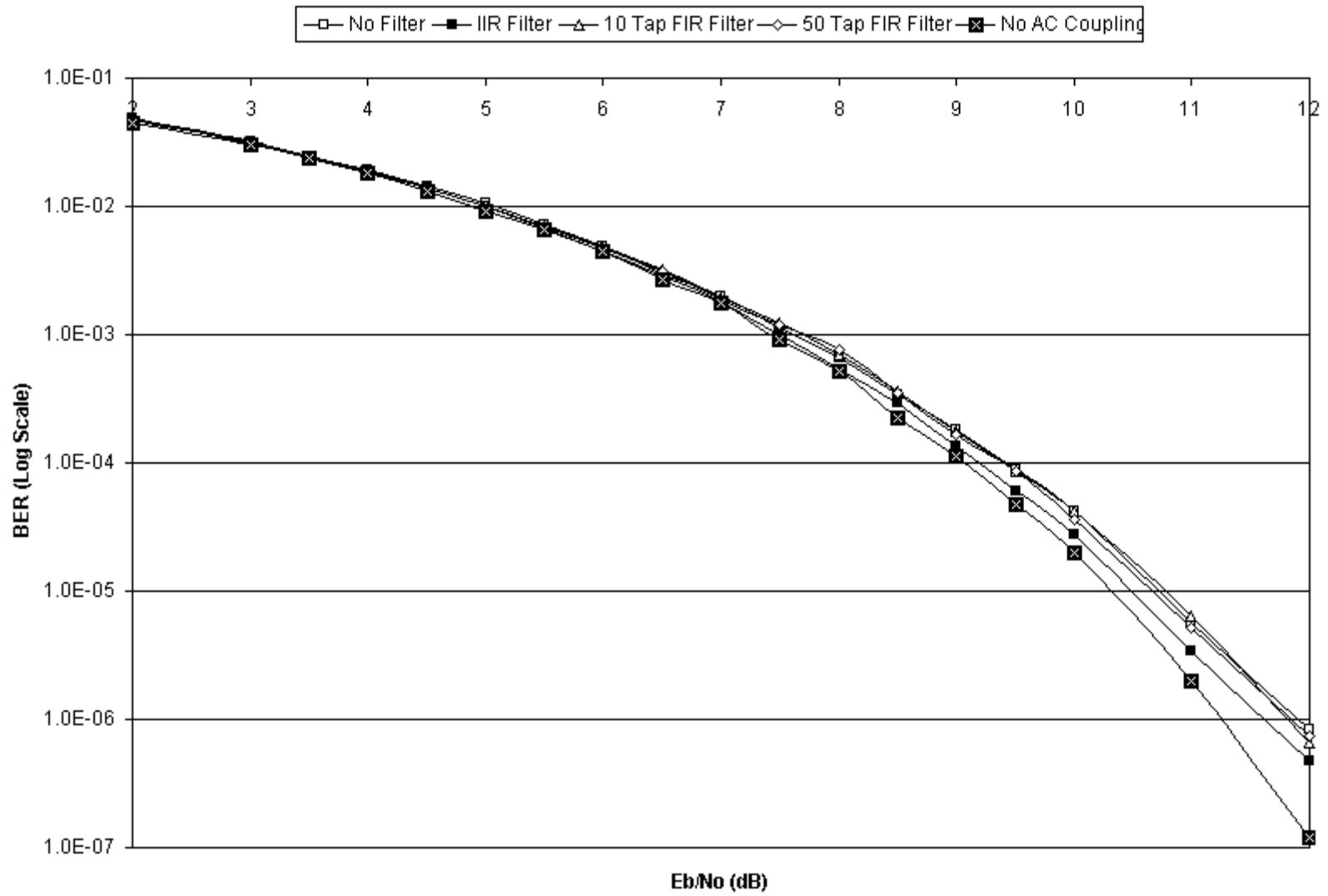


Figure 4.13 Bit error rate (BER) for dc wander compensation with balanced signal ($P_{\text{zero}} = 0.5$).

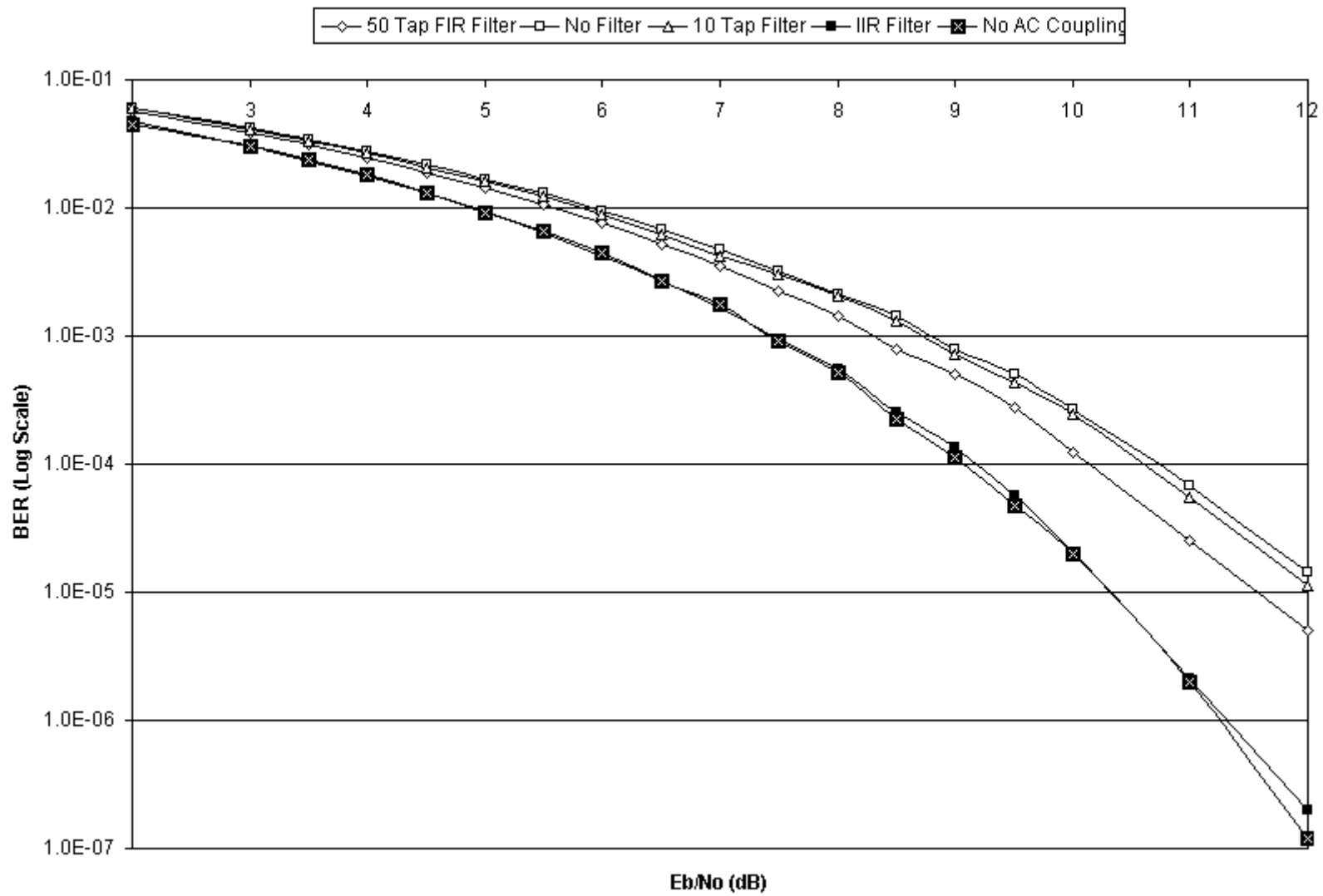


Figure 4.14 Bit error rate graph for dc wander compensation system with unbalanced signal ($P_{zero} = 0.4$).

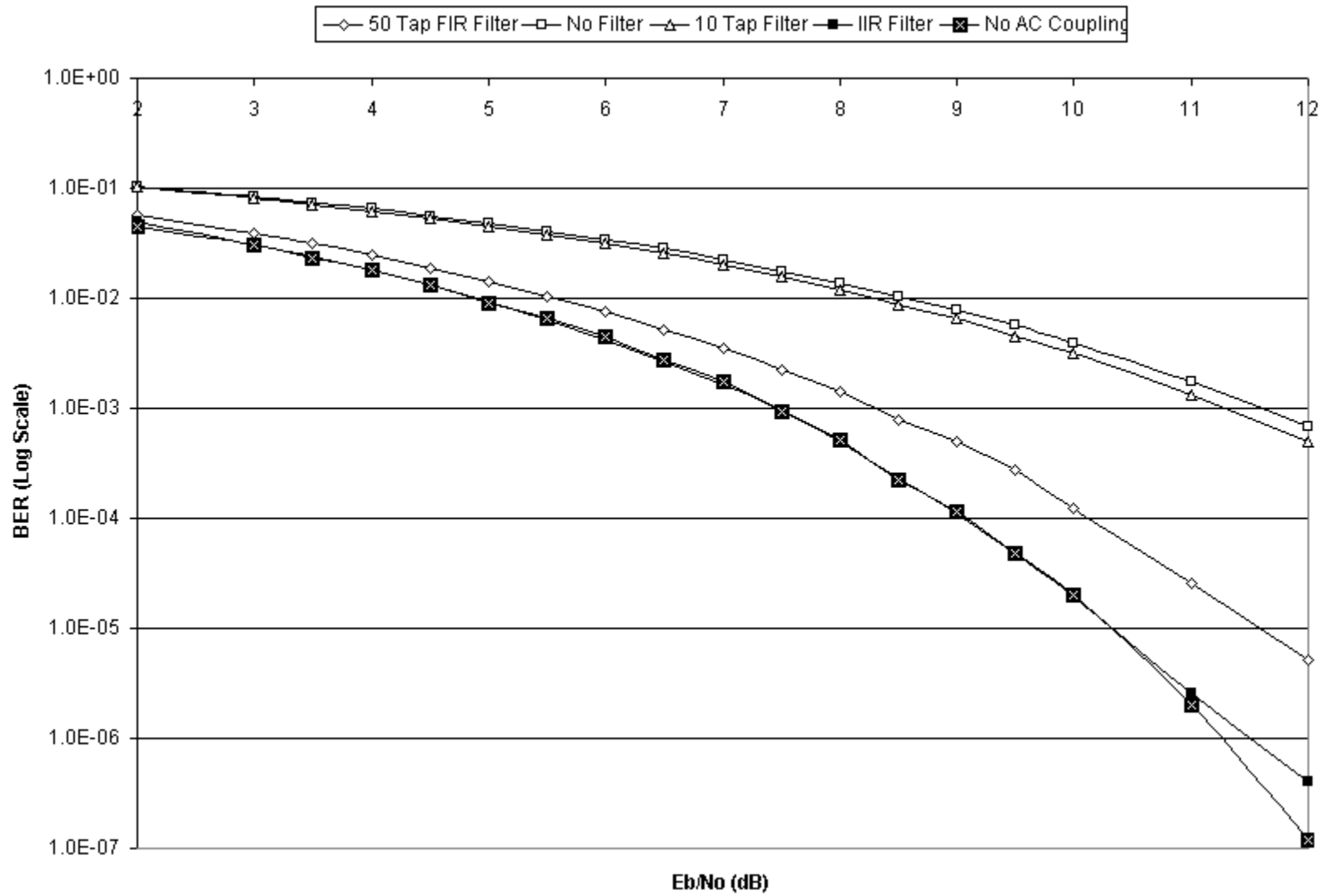


Figure 4.15 Bit error rate graph for dc wander compensation with unbalanced signal ($P_{zero} = 0.3$).

5 Signal Interference (Crosstalk)

When an electrical signal propagates on a transmission line, electric and magnetic fields are created around the line. The magnitude of the fields depends on the geometry and topology of the line, and the voltage and current of the electric signal. As technology progresses, logic switching times get increasingly fast. As the rate of change of voltage increases, there is an increase in displacement currents due to the capacitances associated between coupled transmission lines. The increased rate of change of driving current in the line corresponds to an increased rate of change of magnetic flux around the interconnect which induces voltages in coupled transmission lines. These factors are becoming increasingly significant as switching times decrease, printed circuit boards shrink, and trace lines get closer together. The voltage and current induced on a neighbouring interconnect line (due to changes in magnetic and electric fields) is referred to as electromagnetic interference (EMI). There are two types of EMI coupling that occur on PCB traces, capacitive and inductive, both of which make up “crosstalk”.

5.1 Capacitive Coupling

Capacitive coupling occurs between closely spaced current carrying parallel lines. On a PCB, interconnect lines run very close to one another and digital signals can couple from one line to another. This type of coupling is called mutual capacitance. Mutual capacitance between the neighbouring traces can be modeled by placing an ‘imaginary’ capacitor (C_M) between the lines. The value of C_M is highly dependent on the separation of the two lines, having an inverse relationship to the separation distance. Through this capacitance, a change in voltage in circuit “A” (aggressor circuit) injects a current I_M into circuit “B” (victim circuit) according to the simplified approximation

$$I_M = C_M \frac{dV_A}{dt} \quad (5.1)$$

where V_A is the voltage in trace A. The capacitance C_M is small enough that it does not load circuit A and therefore does not change the circuit's impedance. The injected current causes a voltage pulse to appear on the victim line which, by symmetry, splits evenly and travels down the line in both reverse and forward directions. This is the resulting near end crosstalk (NEXT) and far end crosstalk (FEXT) from the mutual capacitance, respectively. The magnitude of the crosstalk voltage depends on the rise time of the digital signal T_r , the effective resistance of the victim trace R_B , and the mutual capacitance. This assumes Z_0 is essentially real and the line is terminated in $Z_L = Z_0 \approx R_B$. Using some simple algebra [5], the magnitude of capacitive crosstalk is approximated in the following formula as a fraction of the driving waveform:

$$\text{Crosstalk}(\text{capacitive}) = \frac{R_B C_M}{T_r} \quad (5.2)$$

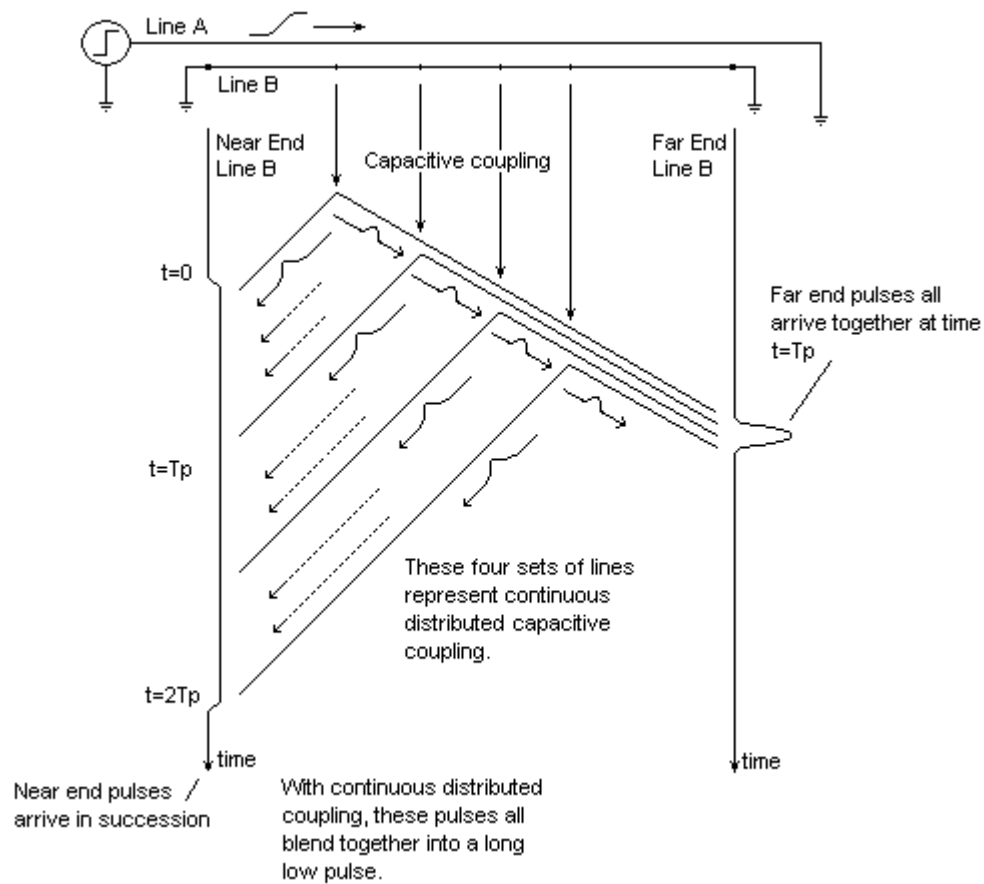


Figure 5.1 Behaviour of NEXT and FEXT generated on a transmission line from capacitive coupling [5].

Because the PCB trace is essentially a transmission line, this crosstalk is created constantly as the rising edge travels along the line. If the propagation time for the rising edge on the aggressor line is T_p , the FEXT component (traveling at the same speed and having the same polarity as the signal on the aggressor line) builds on itself and appears as one large pulse at the receiver at $t = T_p$. The NEXT component also travels at the same speed (and has the same polarity) as the pulse on the aggressor line, but in the opposite direction. This results in a long, low pulse that stays on the line for twice the propagation time, or $2T_p$. Crosstalk is similar for a transmitted falling edge. The concepts of NEXT and FEXT are shown graphically in Figure 5.1. The areas under the NEXT and FEXT pulses are the same, only NEXT is spread out over $2T_p$.

5.2 Inductive Coupling

Inductive coupling, similar to capacitive coupling, also occurs between closely spaced current carrying conductors. This coupling can be thought of as the mutual inductance, L_M , between adjacent circuit loops on the primary and secondary windings in a transformer. The value of L_M has the same dependence on the line separation as with mutual capacitance and induces a voltage, Y_M , in the victim circuit proportional to the rate of change of current in the aggressor circuit according to:

$$Y_M = L_M \frac{dI_A}{dt} \quad (5.3)$$

The victim circuit behaves like the secondary winding of a transformer, so as one end becomes more positive (due to the inductive coupling), the other end becomes more negative. This is evident when the polarities of the FEXT and NEXT are presented. Inductive FEXT behaves similar to its capacitive counterpart, only the induced pulse on line “B” is the opposite polarity to that of the driving signal of line “A” (Figure 5.2). Inductive NEXT behaves exactly like its capacitive counterpart. It manifests as a long low pulse on line “B” (same polarity as the signal on the aggressor line) and travels towards the near end in the opposite direction as FEXT. This phenomenon is shown graphically in Figure 5.2.

The magnitude of crosstalk from mutual inductance depends on the rise time of the digital signal, T_r , the line and termination resistances of the driving circuit, R_A , and

the mutual inductance, L_M . This assumes Z_0 is essentially real and the line is terminated in $Z_L = Z_0 \approx R_A$. This has been shown [5] and results in an approximation for the magnitude of inductive crosstalk as a fraction of the driving signal:

$$\text{Crosstalk}(\text{inductive}) = \frac{L_M}{R_A T_r} \quad (5.4)$$

Since the current in digital circuits on a PCB run in loops with close proximity to one another, mutual inductance is a very significant problem. However, for the most popular configurations of PCB traces (stripline and microstrip) inductive FEXT adds *destructively* to the capacitive FEXT and virtually cancel the effects of one another. This is particularly true for interconnect lines surrounded by a homogeneous medium, as in stripline or embedded microstrip [16].

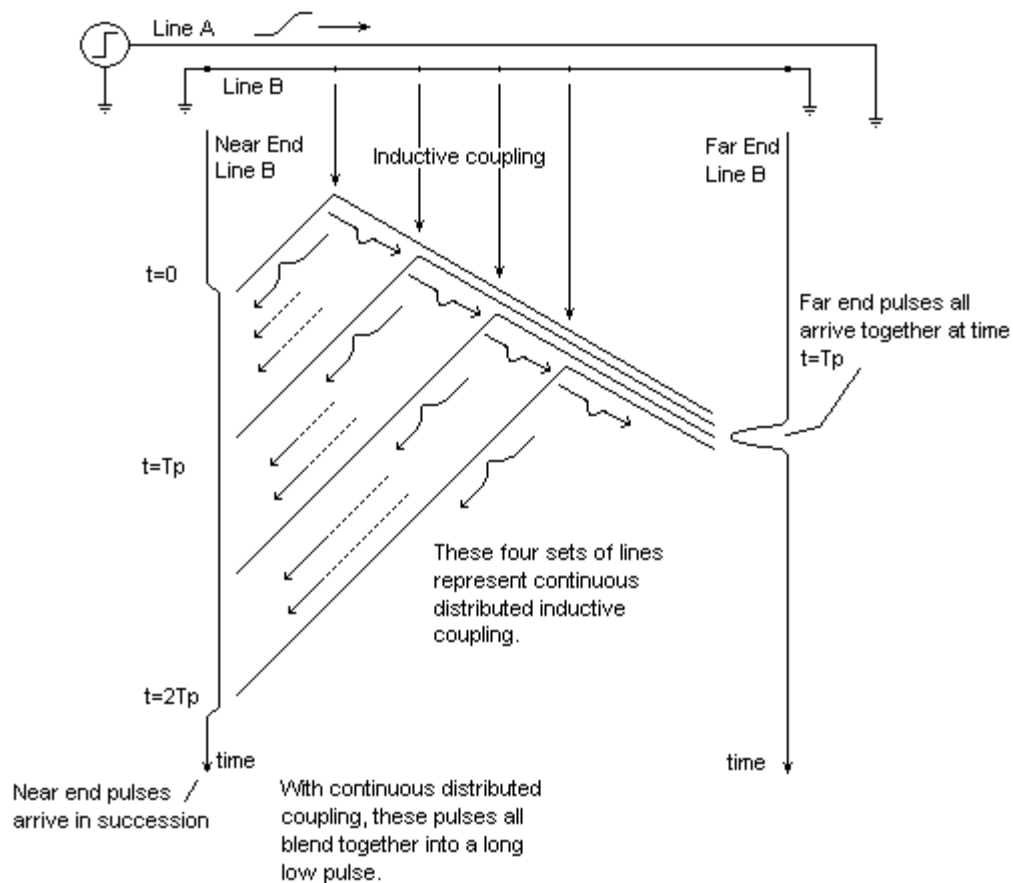


Figure 5.2 Behaviour of NEXT and FEXT generated on a transmission line from inductive coupling [5].

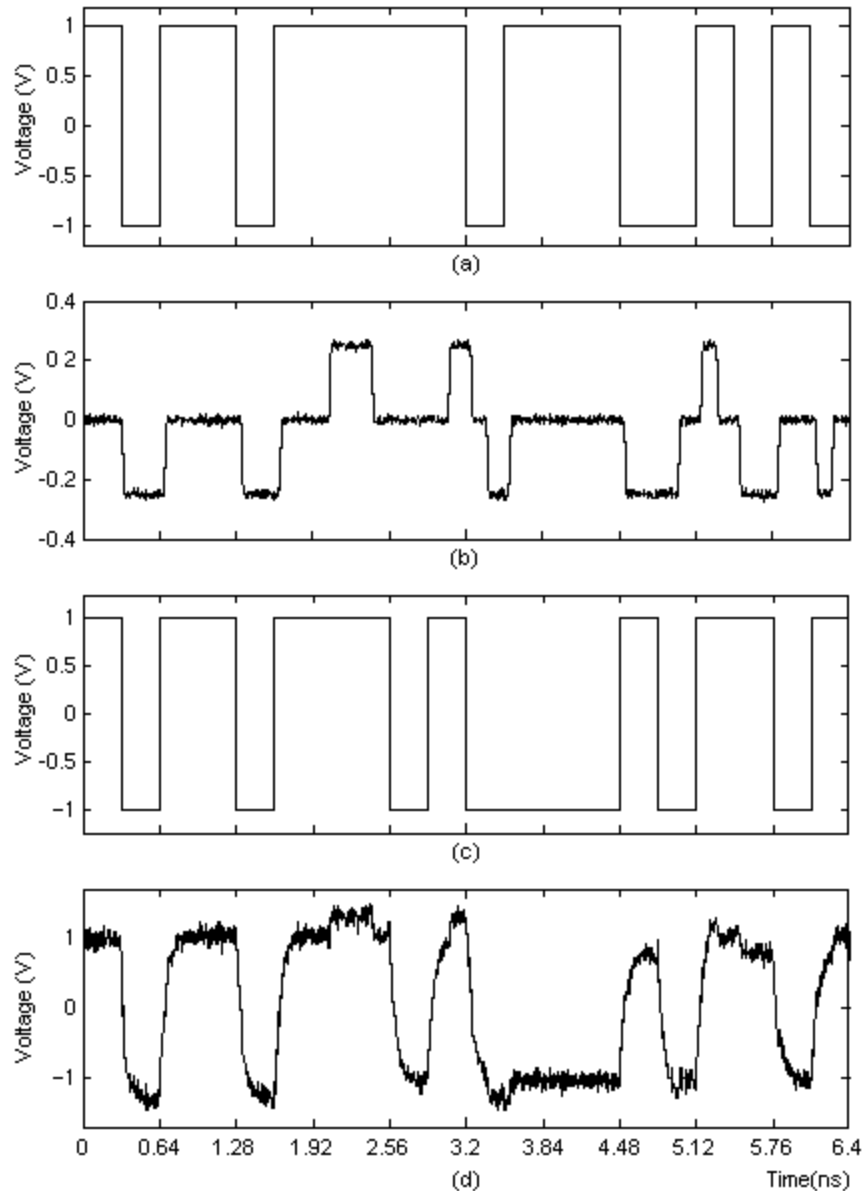


Figure 5.3 (a) Transmitted signal generates (b) NEXT on neighbouring line. (c) Receive signal is (d) affected by crosstalk.

Since inductive NEXT behaves exactly the same way, with the same polarity, as capacitive NEXT, it *constructively* adds to the capacitive NEXT and together they travel toward the near end of the victim trace. NEXT can be a problem if the near end of the victim trace is either a receiving pin, or an improperly terminated transmitter. In most modern designs, the transmitter is properly terminated to prevent reflections and is therefore not an issue. If the near end of the victim trace is a receive pin, NEXT can

affect the voltage level of the signal potentially causing bit errors. A demonstration of microstrip crosstalk and how it affects a neighbouring line is shown in Figure 5.3.

5.3 Solutions for Signal Interference (Crosstalk) Reduction

Crosstalk, as defined in the previous section, has many possible solutions. Many of these solutions are very effective, but have trade-offs which make them undesirable. Some of these solutions also affect the characteristic impedance (Z_0) of the interconnect lines so changes in one parameter requires compensation from another to keep Z_0 at the required value.

One of the simplest methods for reducing the coupling between adjacent traces, and thus reducing crosstalk, is to increase the separation of the lines. This is very effective since the coupling varies with the separation distance by an inverse quadratic relationship [5]. Unfortunately, size is an important issue in PCB design; as board size decreases, so does the cost per unit. Circuit board component and wiring densities increase as board sizes decrease. This prevents designers from increasing spacing between traces in order to reduce crosstalk.

The topology of the trace is another method to keep crosstalk at bay. Since the traces are edge coupled, increasing the trace height greatly increases crosstalk as does increasing the distance a trace is away from a solid ground plane. Decreasing the voltage level of the transmitted signal also reduces the amount of radiated EM waves but since crosstalk is always a percentage of the transmitted signal, the effect is the same.

As explained in [5], solid ground planes offer much better crosstalk immunity compared to 'slotted' ground planes, 'crosshatched' ground planes, or power and ground 'fingers'. Solid ground planes are used extensively in digital systems and help reduce induced crosstalk voltages on neighbouring interconnect tracks. Ground traces offer additional reduction of crosstalk but little more than a solid ground plane provides. In fact, in a digital system, if tracks are positioned far enough apart to allow a ground trace to fit in between, the separation itself has provided ample crosstalk reduction and the ground trace adds little more. Over a solid ground plane the voltages for inductive

and capacitive crosstalk are roughly equal. This means the forward crosstalk (FEXT) voltages cancel while the reverse crosstalk (NEXT) voltages reinforce. This is particularly true with stripline where the balance between the inductive and capacitive coupling is very good resulting in very small forward coupling coefficients. According to [5]:

“Microstrips, for which the electric field lines responsible for crosstalk travel mostly through air instead of through the dielectric, have somewhat less capacitive crosstalk than inductive, leading to a small negative forward-coupling coefficient.”

Improved PCB board material (with a lower dielectric constant) lowers the capacitance between lines and thus inhibits the propagation of EM radiation to neighbouring traces. Since crosstalk is directly related to the mutual capacitance between adjacent traces, the reduction in capacitance offered by the improved dielectric is very desirable. Materials with lower dielectric constants are currently available but are prohibitively expensive. Hence most board manufacturers and electronic design companies have opted not to use them.

Signal transition time and line length are major contributing factors in the severity of crosstalk. Since crosstalk is proportional to the rate of change of a signal, it makes sense that if a signal has a longer rise time (a slower rate of change) crosstalk will be less severe. Currently signal transition times can be 80 ps or faster. In high speed applications these fast transitions are necessary when chip manufacturers want to transmit and receive data above 3 Gb/s. Therefore longer transition times is not a viable solution for crosstalk reduction. Shorter line lengths can also reduce crosstalk. If a pair of adjacent traces are short enough to satisfy Equation 5.5, NEXT never reaches its maximum value.

$$length < \frac{T_r}{6T_d} \quad (5.5)$$

Using the formula given above (T_r = rise time, T_d = propagation delay and $length$ is the line length) it can be shown that the length of a signal's rising edge (for example 80 ps rise time) is

$$\frac{T_r}{T_d} = \frac{80ps}{70.87 \frac{ps}{cm}} = 1.1289 \text{ cm} \quad (5.6)$$

(using propagation delay of a signal in FR4 material, Table 1.1 of [5]). It is interesting to note from the result in Equation 5.6 that if a typical length of interconnect on a PCB is 20 cm, at a bit rate of 3.125 Gb/s there could be almost 5 bits of information on the interconnect at any given time (bit period is 320 ps). Using the result in Equation 5.6 and applying it to Equation 5.5 gives the maximum length of interconnect line needed to ensure crosstalk does not reach its maximum level.

$$\frac{T_r}{6T_d} = \frac{1.1289cm}{6} = 0.18815 \text{ cm} \quad (5.7)$$

This is a very short line, and positioning ICs this close together is not a practical solution. It is important to note that even if crosstalk does not reach its maximum level, it is still present and can be a problem.

The crosstalk reduction methods described here have been thoroughly investigated and verified [5], [6], and [8], and many are already in place. For example, line spacing and topology have been optimized given the dielectric constant of FR4 board material and a characteristic differential impedance of approximately 100Ω. Also, solid ground planes are used to reduce EM waves from being picked up by interconnect, and stripline style interconnect are used because they have less dispersion than microstrip [8]. *Low voltage differential signaling* (LVDS) is a signal transmission method that uses lower voltage levels to reduce the EM radiation created while its differential signaling provides noise immunity and reduces the coupling between the differential line pair. The reduced coupling is seen with the magnetic field in Figure 5.4 for odd mode impedance. It has the effect of inserting a ground plane between the differential line pair [6]. However, this does not prevent crosstalk between neighbouring sets of differential lines.

Very short line lengths could be used to prevent the crosstalk from reaching its maximum level, but as mentioned earlier, the lines would have to be less than a couple millimeters. This is, however, what Sun Microsystems is proposing to do as mentioned

in a recent press release [17]. They wish to remove the interconnect line altogether by mating the communicating ICs eliminating the need for pins, and thus interconnect. They also agree that this technology is a few years from production, and also has some restrictions.

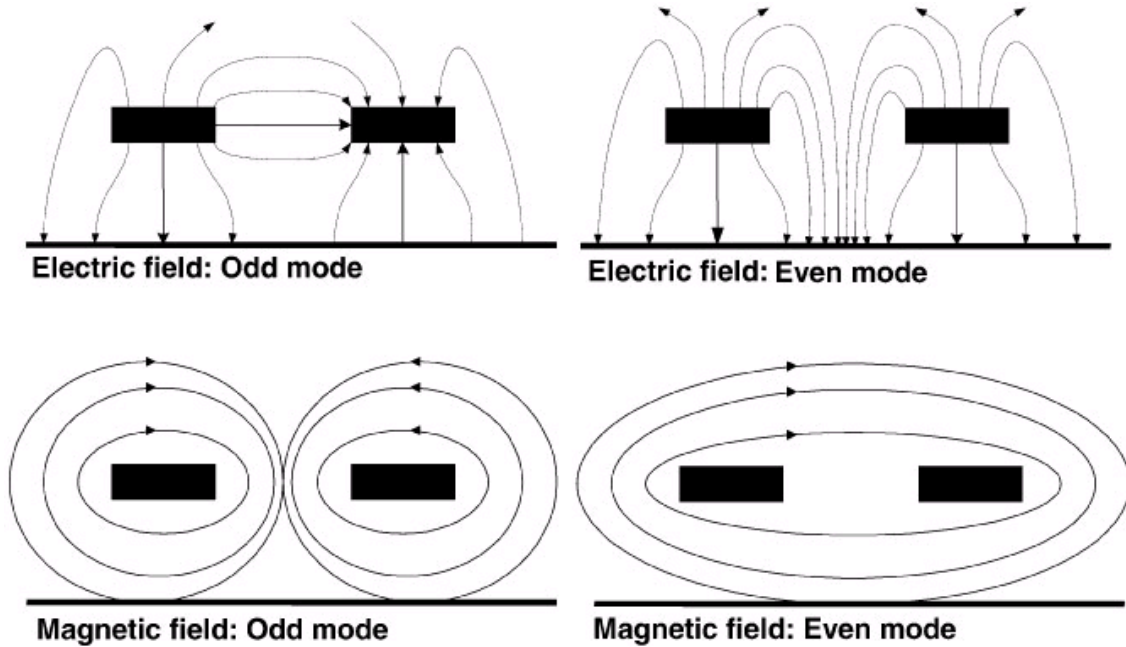


Figure 5.4 Odd and even mode electric and magnetic field patterns for a simple two-conductor system [6].

Even with most of the current technology measures in place, near end crosstalk (NEXT) continues to be a problem. Two separate digital circuits are proposed to compensate for it. Although the implementation of these circuits is beyond the scope of this project and thesis, obvious limitations in the designs will be addressed given the nature (speed) of the systems involved.

5.4 Compensating Near End Crosstalk using a Digital Equalizer

If a single rising or falling edge is transmitted, the near-end crosstalk (NEXT) will be present for a period of $2T_p$, or twice the propagation time of the transmitted signal. From Equation 5.5, if $T_r < 6T_p$, where $T_p = T_d \cdot length$, is the propagation time on a line, the NEXT voltage will reach its maximum height (a_0). The circuit will then

need to compensate for the same amount of time ($2T_p$) and supply a maximum voltage (equal and opposite to the NEXT crosstalk) of a_0 . The digital equalizer structure shown in Figure 5.5 is one proposed solution to offset the amount of crosstalk present at the receiving node caused by the transmitted signal.

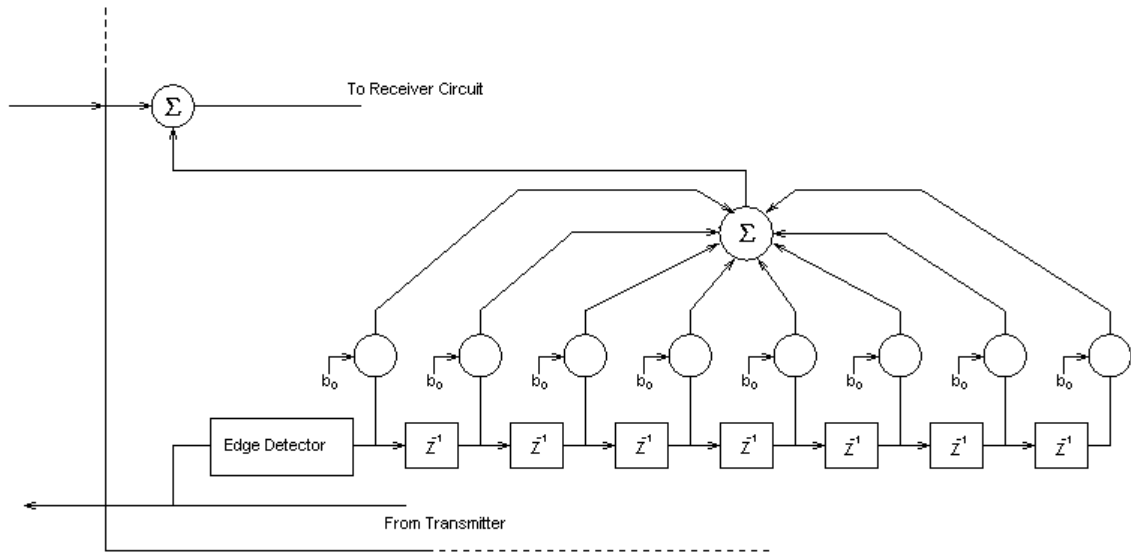


Figure 5.5 Crosstalk compensation using digital feed-forward circuit.

An edge or level detector is used to sense whether a transition has taken place. If a rising edge is detected (which creates a positive NEXT pulse on the victim line) the detector outputs a “-1”. This value enters the shift register and is multiplied using a gain element which creates a compensation signal equal and opposite to the imposed NEXT on the neighbouring line. This signal is then added to the incoming signal on the victim line where it cancels the crosstalk. The value from the edge detector propagates in the shift register creating the same compensation signal at each point. Similarly, if a falling edge is detected, a +1 is output from the edge/level detector and a signal (equal and opposite magnitude of the crosstalk) is generated to cancel the crosstalk generated by the logic 1 to logic 0 transition. If no transitions occur, no crosstalk is created so a value of zero is output from the edge/level detector so no compensation is generated.

Although the circuit in Figure 5.5 looks simple and small in a simulation, the physical implementation would require a large amount of silicon area due to the number of adders needed. Also since the circuit must operate at very high bit rates,

compensation must begin very quickly. This is not possible with the amount of circuitry involved in the many addition operations required for this system. Even with these limitations, the system will be investigated and simulated to demonstrate its operation. The second circuit proposed to compensate for NEXT was designed with the limitations of the equalizer circuit in mind (i.e. too large and too slow). This second compensation method is outlined next.

5.5 Compensating Near End Crosstalk using 2-Bit Selection Circuit

A second proposed solution is similar to the previous one as it uses an edge/level detector and a delay system. The difference lies in the method it uses to generate the compensation signal. It works on a principle that induced crosstalk changes every time one of four things happen. Either a rising or falling edge is transmitted on the aggressor line, or the rising/falling edge reaches a point where it no longer imposes crosstalk on the neighbouring line. This happens when the signal either reaches the destination IC or the traces get far enough apart so that the crosstalk is no longer a problem. Since there can never be two or more consecutive rising (or falling) edges and assuming the crosstalk reaches its maximum level for every transition, the crosstalk has only three possible states with voltages equaling either $+a_0$, $-a_0$, or zero.

Table 5.1 Digital 2-bit values representing levels of NEXT.

NEXT Voltage (V)	MSB	LSB
0	0	0
+ a_0	0	1
- a_0	1	0
N/A	1	1

From a digital perspective, these three states can be represented as a 2-bit value as shown in Table 5.1. The proposed circuit uses a system similar to this, which calculates a 2-bit value and generates the compensation signal that corresponds to it. It does this by receiving signals from the edge detector circuit before they are put in the delay line,

which signals the beginning of the NEXT, and after they come out the end, signaling the end of the NEXT.

The general block diagram of this system is shown in Figure 5.6. The input signals to the 2-bit adder/subtractor can be thought of as ‘adds’ or ‘subtracts’. Since it is possible that the ‘add’ and ‘subtract’ signals occur simultaneously, there are five possible operations, single add, single subtract, double add, double subtract and no-operation. However, if attention is paid to the timing during the design of the system, it may be possible to ensure all adds and subtracts occur at different times, eliminating the double add and double subtract operations.

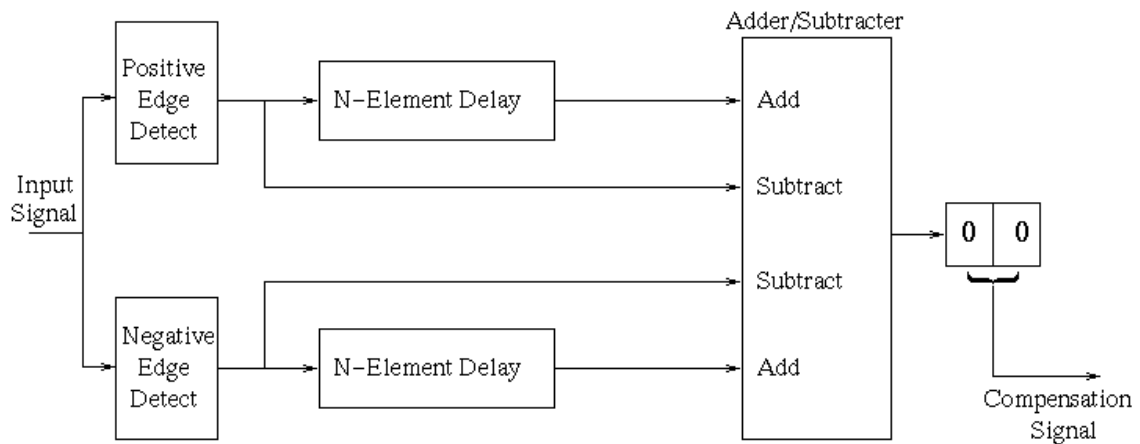


Figure 5.6 Block diagram for proposed NEXT compensation circuit.

5.6 Compensation Circuit Design Considerations

To compensate for the affects of crosstalk, two circuits have been proposed that are quite different in how they generate a compensation signal. However, they share some common components that need further attention. The two proposed designs use a delay element whose length depends on

- propagation delay of the transmitted signal and
- distance the aggressor and victim lines run close and parallel,

and multiplier coefficients whose value depends on

- geometry / topology of the interconnect lines,
- separation distance of the aggressor and victim lines,

- dielectric constant of PCB material, and
- transition times of aggressor signal.

If there are no provisions on the integrated circuit (IC) for changing the length of the delay line or multiplier coefficients after the IC has been manufactured, one of two things could be done. Either the layout and component information would be needed prior to the IC's production, or designers would have to lay their PCB out as per the specifications of the compensation circuit. This could limit the circuit's use or make the IC restrictive in designs. Some IC's actually have pins that are used for setting on-chip parameters which would avoid this problem.

As mentioned in Section 2.6 multipliers and adders continue to be a problem when implementing circuits due to the area they consume on an IC and the high bit rate of the system. This is why two different circuits have been considered. The second proposed crosstalk solution, the 2-bit selection method uses only one adder and one multiplier which could prove to be the key difference that decides which method is more desirable. It provides decreased complexity and propagation delay through the circuit, and directly addresses silicon area issues.

6 Digital Circuits for Crosstalk Compensation

Digital circuits have been proposed as a solution to near end crosstalk (NEXT) on PCB interconnect lines. Using a digital circuit, a signal can be generated that matches the characteristic crosstalk that appears on neighbouring lines caused by fast signal transitions. Once this compensating signal has been generated it can be used to offset the crosstalk and restore the original signal. This can be done because, if the configuration of the PCB is known, the induced crosstalk can be estimated and a circuit can be designed to match the characteristics of the crosstalk expected.

6.1 System Parameters

Several parameters are required to develop the equalizer. Clock frequency is an important factor in designing the length of the equalizer shift register since time one single value stays in the shift register must equal $2T_p$, or twice the propagation delay of the transmitted signal. However, a shift register is not the only way to implement a delay line. If the delay line can be asynchronous, the clock frequency is not important, as in the 2-bit selection circuit.

Propagation time of a signal is found by calculating the propagation delay T_d of a particular configuration of line, and multiplying by the line length. However, for the line delay it is important to only calculate for the length where the coupled interconnect lines run close together and parallel, where NEXT actually occurs.

Topology, dielectric constant, and separation of the interconnect lines play two roles in the development circuit. First, all these parameters directly affect the speed at which the signal propagates from transmitter to receiver. They are therefore used in the propagation delay calculations, which affect the length of the delay line. Secondly, topology, separation of lines, dielectric constant, and signal transition times determine the magnitude of the crosstalk voltage. Thus, these parameters are used to calculate the

multiplier coefficients in the circuit. The values of these parameters will mostly be borrowed from technical papers or textbooks to obtain realistic values for simulation.

6.2 Edge Detection

Detecting a rising or falling edge of an outgoing signal is crucial for providing the appropriate compensation for the resulting crosstalk interference. Both proposed designs (equalizer and 2-bit selection) require a method for detecting this transition

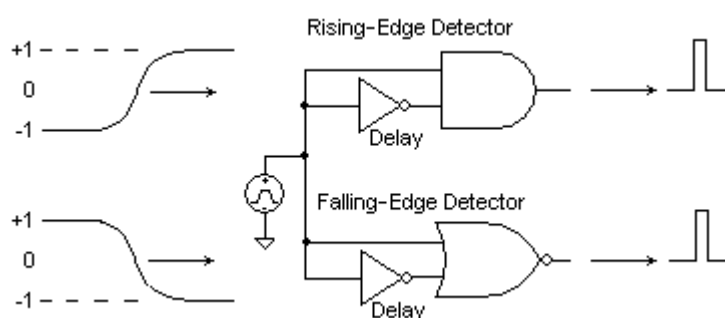


Figure 6.1 Edge detection circuit for proposed compensation schemes.

Since timing is crucial in crosstalk compensation, the detection of rising and falling edges needs to be as fast and simple as possible. Figure 6.1 shows a very simple edge detection circuit which uses a delayed version of the input signal (using an inverter) to generate a short pulse for every rising edge or falling edge. The only requirement of these pulses is that they need to be able to trigger a flip-flop in the next stage. Each pulse will be used twice, once to begin compensation of the crosstalk, and a delayed version of the same pulse will signify when the compensation for that edge is to stop. This sequence of events will be described in Sections 6.3 and 6.4.

6.3 Digital Equalizer Design

The design of the equalizer begins by determining the parameters of the system. Since this equalizer is a very simple design, the length of the shift register and the tap multiplier coefficients (which all have the same value) are the only parameters that need to be calculated. A stripline is assumed for the interconnect configuration with a

commonly used dielectric material (FR4). The parameters below demonstrate the calculation for the length of the equalizer.

- Interconnect line length $len = 30$ cm (the length of line that runs closely parallel to an adjacent “victim” line where the NEXT will be observed).
- Bit rate (f_{bit}) = 3.125 Gbps is the data transmission rate.
- Dielectric constant for FR4 (ϵ_r) = 4.5.
- Clock rate (f_{clk}) = $2(f_{bit})$ is the rate at which the delay units are clocked in the equalizer. Could also be $4(f_{bit})$ but no less than (f_{bit}).
- Speed of light in a vacuum (c) = 2.998×10^8 m/s.

The equalizer length is then calculated using the following formula

$$Length = 2(len \cdot T_d \cdot f_{clk}) \text{ delay units}, \quad (6.1)$$

where $T_d = \frac{\sqrt{\epsilon_r}}{c}$ is the propagation delay and is measured in seconds per meter.

Obviously the value of ‘Length’ must be an integer and in this case the length of the shift register would be

$$Length = 2(0.3) \left(\frac{\sqrt{4.5}}{2.998 \times 10^8} \right) \cdot 6.25 \times 10^9 = 27 \text{ delay units}.$$

The calculation for the multiplier coefficients is much more involved. Some textbooks and papers have approximations for the calculations for reverse voltage crosstalk (NEXT) coefficients [5], [18]–[20]. The formula in [18] uses even and odd mode impedance values of stripline to calculate the NEXT coefficient which is given by

$$X = \frac{Z_{oe} - Z_o}{2(Z_{oe} + Z_o)} - \frac{Z_{oo} - Z_o}{2(Z_{oo} + Z_o)} \quad (6.2)$$

- X is the reverse crosstalk coupling coefficient,
- Z_o is the stripline’s characteristic impedance,
- Z_{oe} is the even mode impedance, and

- Z_{oo} is the odd mode impedance.

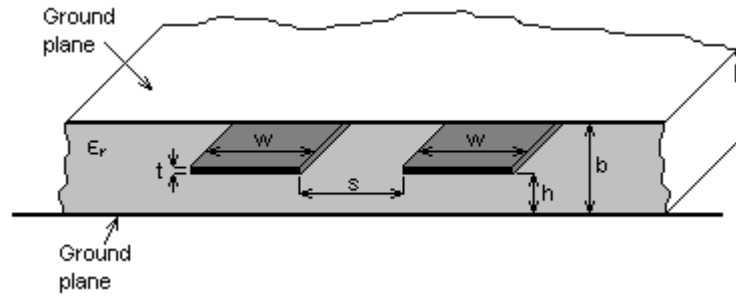


Figure 6.2 Geometry and topology of dual stripline configuration.

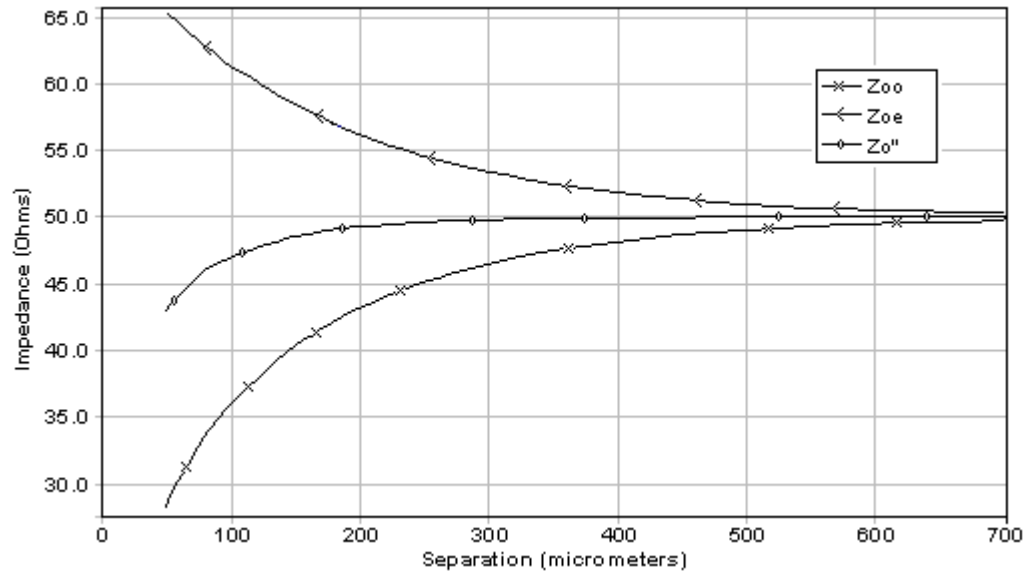


Figure 6.3 Impedance values versus trace separation for odd (Z_{oo}), even (Z_{oe}), and single line characteristic impedance (Z_o'') of stripline interconnect [18].

Since the value of the NEXT coefficient is dependent on the geometry and topology of the interconnect, dimensions are chosen for 100Ω differential stripline and listed below.

- Height of substrate (b) = $500\mu m$ or $19.68mil$
- Trace thickness (t) = $17\mu m$ or $0.7mil$
- Trace separation (s) = $300\mu m$ or $11.8mil$
- Trace width (w) = $176.56\mu m$ or $6.95mil$

- Dielectric constant (ϵ_r) = 4.5

The traces are midway between the ground planes as shown in Figure 6.2. Equation 6.2 uses values from the graph in Figure 6.3 to generate the NEXT coefficients.

The crosstalk coefficient, α_R , represents a percentage of the driving waveform's amplitude. The calculated crosstalk coefficient values are plotted versus stripline trace separation in Figure 6.4. This shows how crosstalk can be reduced if the distance between traces is increased. Using the graph below, a separation value can be chosen to calculate the multiplier coefficient.

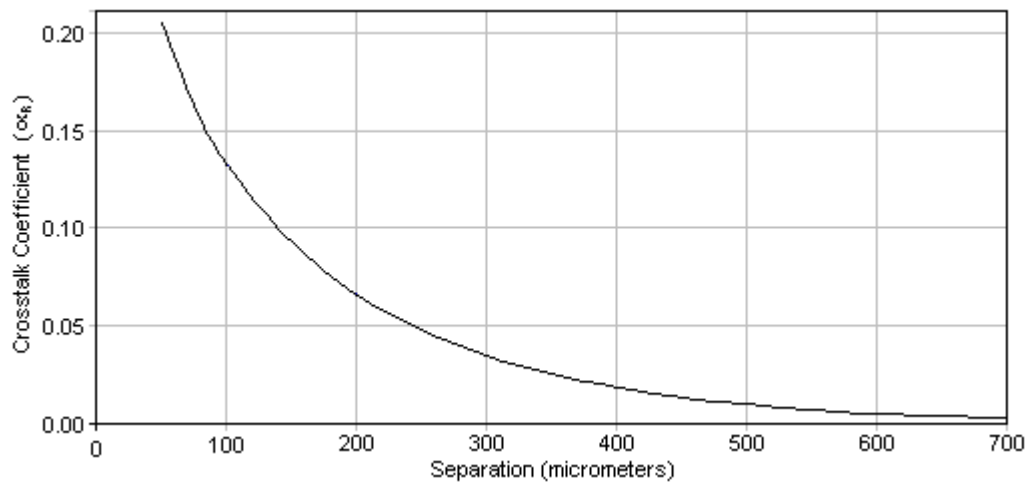


Figure 6.4 Reverse voltage crosstalk coefficient α_R as a function of stripline separation [18].

Since the single line characteristic impedance (Z_0) for a balanced pair of coupled transmission lines is commonly around 50Ω (100Ω differential), a separation value of $200\ \mu\text{m}$ is chosen which results in a characteristic impedance of approximately 49Ω and a NEXT coefficient of approximately $\alpha_R = 0.065$. The NEXT coefficient translates to a percentage of the aggressor signal voltage so the magnitude of reverse crosstalk is found [5] with

$$NEXT(t) = \alpha_R (V(t) - V(t - 2T_p)) \quad (6.3)$$

where t = time in seconds

$V(t)$ = driving (aggressor) waveform in Volts,

α_R = reverse coupling coefficient, and

T_p = propagation time of signal on line in seconds.

Although a NEXT coefficient of $\alpha_R = 0.065$ seems quite low (only 6.5 percent), the important thing to note is that this value is 6.5 percent of the *transmitted* (aggressor) signal amplitude. The transmitted signal on the victim line is attenuated by transmission line effects and has an amplitude less than half (that of the aggressor signal) at the receiver. Thus the 6.5 percent is a much higher percentage of the *received* signal amplitude, and a much bigger problem.

The shift register length and multiplier coefficient must be calculated for each individual system based on the layout and topology of the circuit board and the amplitude of the driving waveform. The ensuing simulation system and results will be based on the parameter values which were chosen and calculated earlier in this section.

6.4 2-Bit Selection (NEXT Compensation) Design

The design for the 2-bit selection circuit contains much of the same information as was given in Section 6.3. The calculations for delay line (NEXT duration) and the crosstalk coefficient are identical for this section; the way this compensation is carried out is what differs between the two models.

The reason for this second method of compensation is to address the size and propagation delay issues that are present in the equalizer design method. The 2-bit selection compensation technique is a simpler design and uses fewer components, which makes it inherently faster. The goal is to show that this circuit works as well as the equalizer circuit at compensating NEXT, and is better with regards to speed and size.

As mentioned in Section 5.5 this NEXT compensation technique involves three states. It is assumed the crosstalk will always reach its highest level, and it will be either positive, negative or zero. A simple idea to generate this type of result is to use a circuit as shown in Figure 6.5. The state machine and state table for this circuit, shown in

Figure 6.5(b), demonstrate how the circuit changes states, and their corresponding outputs, respectively. The states and outputs are summarized in Table 6.1 below.

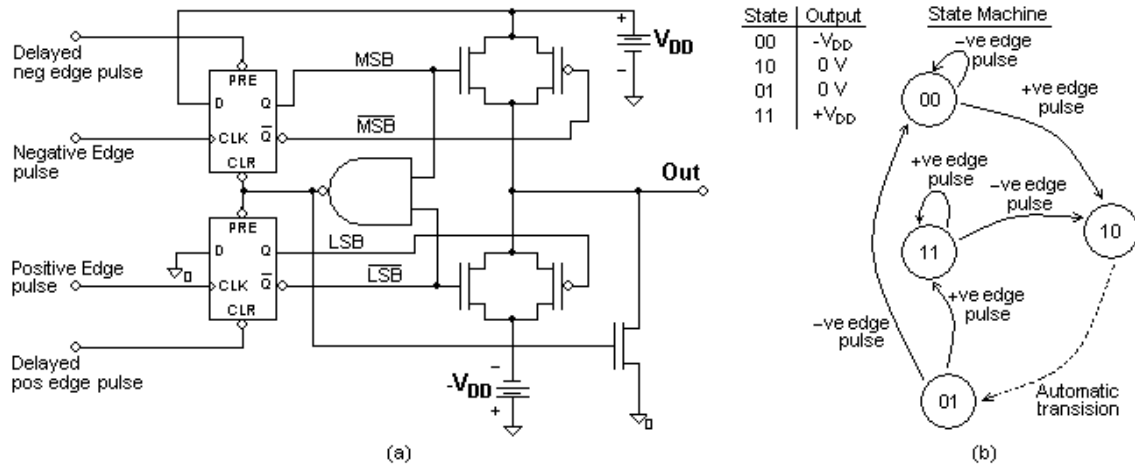


Figure 6.5 (a) Near end crosstalk (NEXT) tri-state circuit and (b) state machine.

The circuit will never stay in state ‘2b’; the logic always forces it back to state ‘2a’. The voltages output from this system are equal to the positive and negative supply voltages, which can then be scaled to attain the correct voltage value needed to compensate the crosstalk. Since the magnitude of this voltage is the same for both positive and negative values, the circuitry can be kept simple.

Table 6.1 State table for 2-bit selection circuit.

State	MSB	LSB	Output Voltage (V)
1	0	0	$-V_{DD}$
2a	0	1	0
2b	1	0	0
3	1	1	$+V_{DD}$

Once the edges of the outgoing signal have been detected, compensation of the crosstalk can begin. This is done using the short pulse created by the edge detection circuit described in Section 6.2. Since the crosstalk ends after a period of $2T_p$, the same

pulse can be used to stop the compensation $2T_p$ later. This is done by sending a copy of the pulse through a delay line (for example a series of buffers). Short pulses are required so as to minimize interaction between pulses (i.e. pulses should arrive at different times and be gone before the next one arrives). All the pulses are then fed into the tri-state circuit.

The inputs to the tri-state circuit thought of as ‘adds’ and ‘subtracts’. An ‘add’ brings the circuit from a “00” state to the “01/10” state, and another ‘add’ brings the state to “11”. ‘Adds’ can also be thought of as bringing the compensation from negative voltage to zero volts, then to positive voltage. ‘Subtracts’ do the opposite.

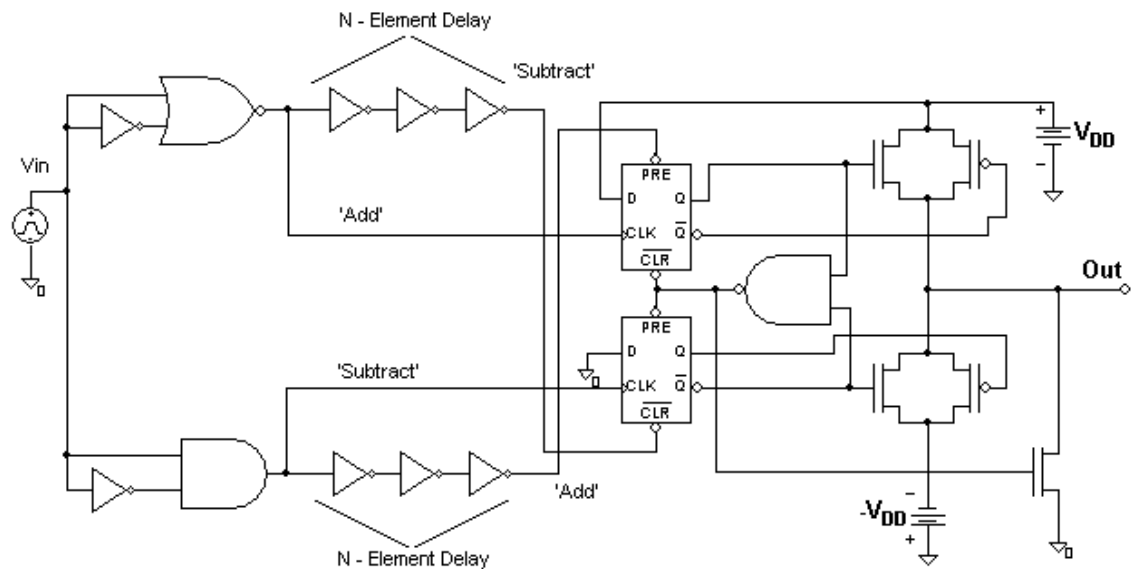


Figure 6.6 NEXT compensation circuit (I).

A rising edge causes positive NEXT to be imposed on the neighbouring line; therefore a negative compensation signal (subtract) needs to be generated. Then intuitively, a falling edge causing negative NEXT needs a positive compensation signal (add). To stop the compensation generated by a certain ‘add’ (subtract), the pulse that emerges from the delay line is then treated as a ‘subtract’ (add). Two tri-state circuits have been designed and investigated to determine which is more effective. The criterion to determine effectiveness is simply speed since each circuit generates the exact same output.

To demonstration of flow for the circuit in Figure 6.6, an initial transition pulse is applied directly to the flip-flop of the tri-state circuit, avoiding any extra gate delays, to change states. The signal is then fed through a series of inverters to attain the desired asynchronous delay. After the delayed signal emerges it is used to ‘set’ or ‘reset’ the appropriate flip-flop to return the circuit back to its original state (NEXT stops). The top and bottom flip-flops in both designs are used for add and subtract signals, respectively. This is done by using the pulses as either a clock signal (Figure 6.6 and Figure 6.7) or an asynchronous ‘set’ or ‘reset’ (Figure 6.6). The circuit in Figure 6.7 uses exclusive-or (XOR) gates to integrate the ‘add’ and ‘subtract’ pulses on to the same line, which adds delay, and does not use the asynchronous set and reset.

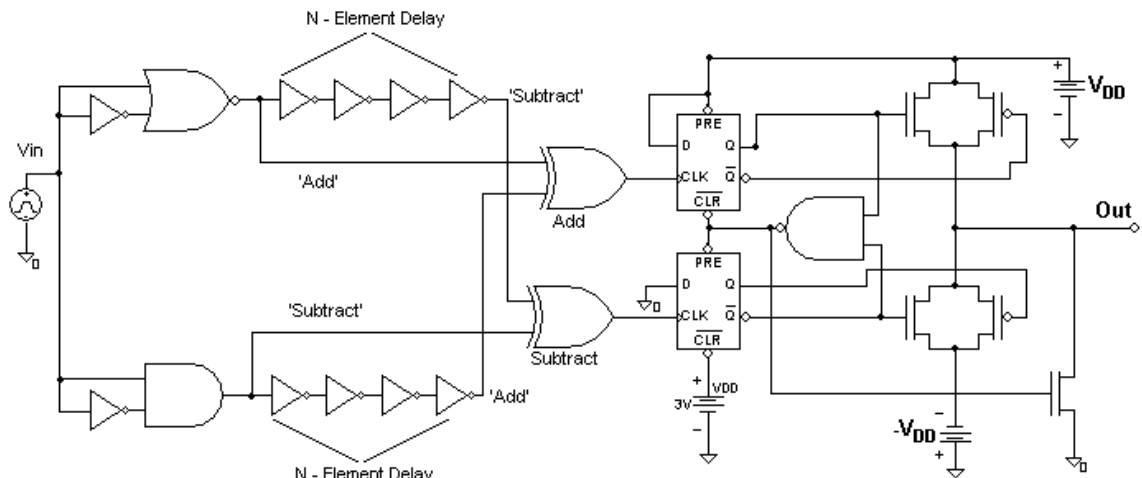


Figure 6.7 NEXT compensation circuit (II).

The simulation of these circuits was done using Altera® Quartus II® software to verify their operation and analyze the time it takes to generate a compensation signal. Since very fast technology devices were not available for the simulations, the resulting delays of each circuit were compared relative to one another. It is believed that the faster circuit in the available technology will also be the faster one in a more modern (faster) technology. The faster circuit is chosen for the 2-bit selection (NEXT compensation) technique in this thesis and would be the one recommended to use in the actual circuits if they are implemented on an IC. The results of the timing simulation for the circuits in Figure 6.6 and Figure 6.7 are given in the simulation results (Section 6.6).

The following complete circuit simulations are performed in Simulink® assuming two scenarios: first when the compensation is fast enough and then when the compensation is too slow to provide the required NEXT correction.

6.5 Simulation Strategy

Simulations are performed using the two different proposed compensation circuit solutions, the equalizer and the 2-bit selection circuit. Comparing the results will show their effectiveness, but not necessarily their practicality. If a system is very effective at counter-acting crosstalk in a simulated environment but is too large or too slow when implemented, it is not practical. Since the systems are designed to generate the same output, the reason for simulating both is to show that the simpler (theoretically faster) 2-bit selection scheme can compensate as well as the equalizer compensation circuit.

In the simulations, unique pseudo-random polar NRZ signals are generated as transmitted data sequences. The user can change the pseudo-random sequence by entering a different “seed” value to the random binary code generator block. The transmitted signal will be used to help generate the estimated near end crosstalk (NEXT). The received signal (a different random sequence) will have the generated crosstalk added to it, simulating the effect on a ‘random’ input. The received signal will also have the effects of transmission line loss imposed on it. This attenuation is so significant, the received signal amplitude becomes much closer the crosstalk amplitude. The received signal data was generated exactly the same as it was for the dc wander section (4.3).

It is assumed that the crosstalk voltage caused by a single transition edge remains constant during the travel time on the victim line. This is to reduce the complexity of the simulation system and the compensation technique. In reality, the crosstalk signal will also experience decay due to the transmission line effects as it travels the length of the interconnect line. Also, since ac coupling is almost standard in the type of application being researched, the simulations include having the incoming signal influenced by ac coupling as well as without. Simulations also include one set of

data demonstrating the bit error rate (BER) of the system with uncompensated near end crosstalk (NEXT).

Table 6.2 Crosstalk simulation results demonstrating increases in data used depending on E_b/N_0 level.

E_b/N_0 (dB)	Data bits (Mbits)	Errors
2	0.5	24012
3	0.5	15748
3.5	0.5	12245
4	0.5	9430
4.5	0.5	7119
5	0.5	5184
5.5	0.5	3748
6	0.5	2394
6.5	0.5	1548
7	0.5	1016
7.5	0.5	618
8	0.5	341
8.5	0.5	178
9	1.0	180
9.5	2.0	185
10	4.0	161
11	11.5	74
12	20.0	20

The simulation provides a user-input, bit energy to noise spectral density (E_b/N_0) to observe the effect that crosstalk and compensation have on the bit error rates (BER). In order to establish the correctness of the simulated system and the noise model, simulations were first run without crosstalk interference. After varying E_b/N_0 , the BER is compared with theoretical values. A chart with theoretical BER values for different

coding schemes is shown in Section 4.3, Figure 4.4. The important Simulink® blocks used in this simulation are outlined in Appendix D.

In order to obtain reasonably accurate simulation results, large amounts of data must be used. To obtain a probability of error (or BER) in the 10^{-6} range, at least one error in 10^6 bits must be recorded. Greater accuracy requires a much greater number of bits be used. Therefore, much more data is used for the cases where the error rate is very small. An example of this is outlined in Table 6.2 for one of the plots; the rest of the data is included in Appendix C. To create a BER graph, a range of E_b/N_0 values is chosen from 2 dB – 12 dB.

The simulation of a single test vector of length 5×10^5 bits in this NEXT compensation circuit takes 20 minutes to perform. In the interest of time, the data vector lengths and number of E_b/N_0 levels are limited to the values given above. These graphs are not perfectly accurate due to the relatively small number of bits used to create them. However, they are sufficiently accurate to allow insight into the operation of the system.

6.6 Simulation Results

The simulation of the crosstalk circuits in Quartus II® was needed to verify the functionality and timing capabilities of the circuit. Several events were chosen to compare the two circuits to determine which was faster. The events (variables $t_a - t_e$) are outlined in the following list and refer to the waveform graphs in Figure 6.8 and Figure 6.9:

- t_a is the time from a falling edge on the aggressor signal (Signal_In), to the beginning of the resultant ‘add’ pulse.
- t_b is the time from a falling edge on the aggressor signal, to the change in state from a 00 to a 01. This includes the change from state 00 to state 10, then to 01.
- t_c is the time for the delayed ‘subtract’ pulse (after the delay it becomes an ‘add’ signal) to change the state from a 01 to a 11.
- t_d is the time for the delayed ‘add’ pulse (becomes a ‘subtract’ signal after the delay) to change the state from a 01 to a 00.

- t_e is the time for an ‘add’ pulse (generated by a falling edge of the aggressor signal) to change the state from a 01 to a 11.

Table 6.3 Propagation delays through 2-bit selection circuit.

Circuit Number	Changing state speed (ns)				
	t_a	t_b	t_c	t_d	t_e
Figure 6.6	6.92	12.94	3.01	0.80	2.24
Figure 6.7	6.89	13.82	4.00	3.32	3.00

Since the aggressor signal was used for both circuits, the simulations can be compared with meaningful results. It can be verified in Table 6.3, that the circuit in Figure 6.6 exhibits faster switching times than the alternative circuit in Figure 6.7. Therefore the former circuit is chosen as the primary design for this crosstalk correction technique.

Each simulation in Simulink[®] was run with at least 5×10^5 bits for each system (equalizer and 2-bit selection circuit). The simulations are repeated for varying degrees of noise ($E_b/N_0 = 2 \text{ dB} - 12 \text{ dB}$). It was expected that both compensation circuits would perform the same, and the advantage one circuit had over the other was the size, and speed of the circuit when simulated with the Quartus II[®] software. The point of interest for this simulation is at the receiver, where the crosstalk becomes a problem.

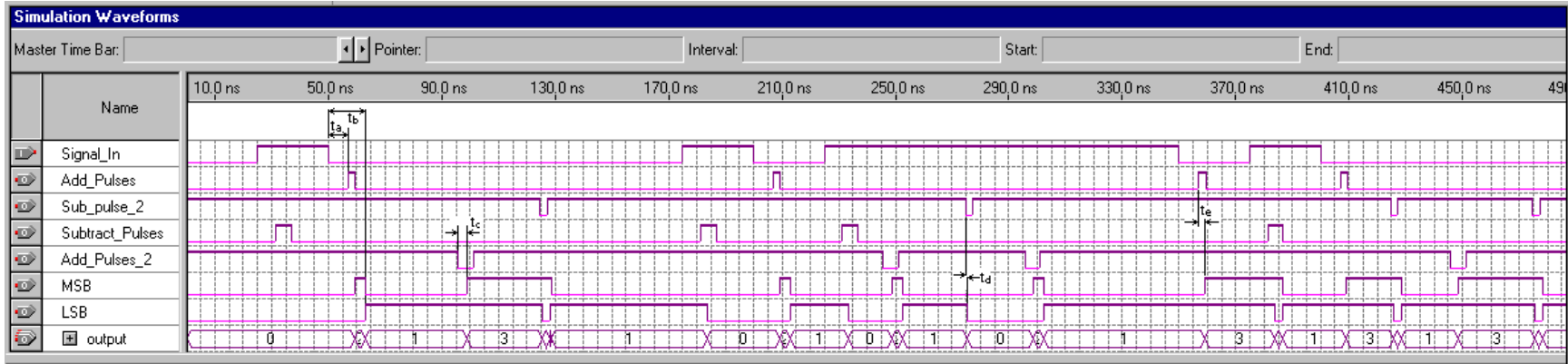


Figure 6.8 Timing diagram for crosstalk compensation circuit in Figure 6.6.

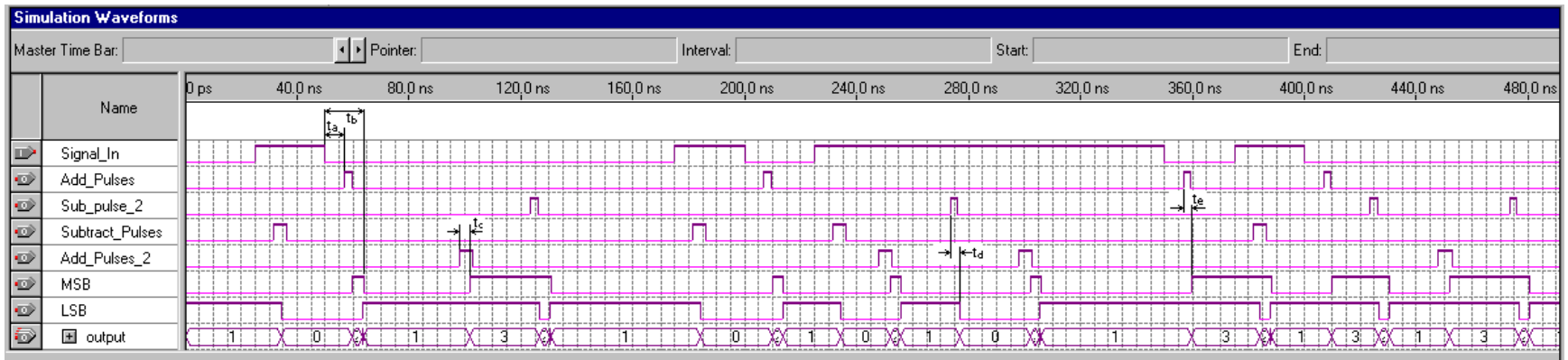


Figure 6.9 Timing diagram for crosstalk compensation circuit in Figure 6.7.

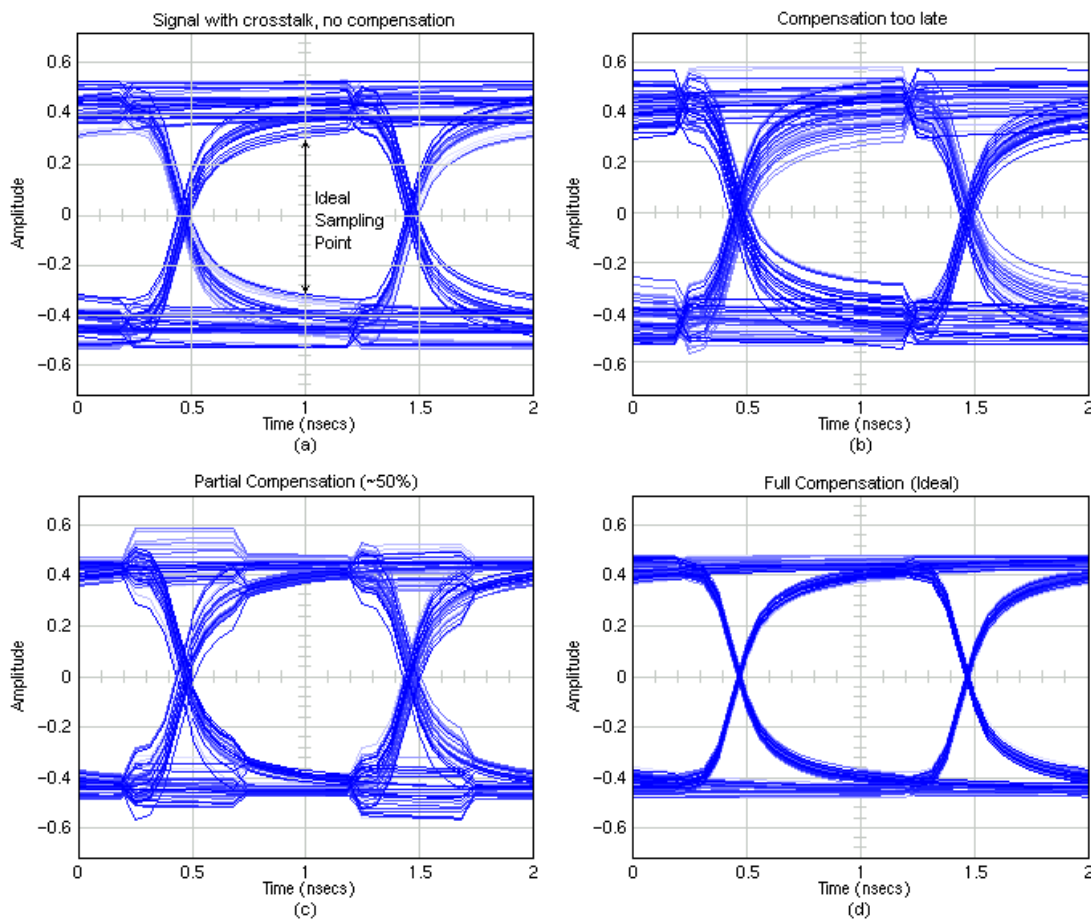


Figure 6.10 Eye diagrams without AWGN showing (a) signal affected by crosstalk, (b) unsuccessful compensation of crosstalk, (c) partial compensation, and (d) full compensation.

The results of the simulations in Simulink[®] are encouraging. The eye diagrams in Figure 6.10 show the received signal influenced by crosstalk, and the result of a successful compensation. Clearly, the difference between the signal without compensation, and the fully compensated signal is quite significant. This is quantified by observing the distance between zero volts (approximate threshold) and the nearest part of the signal at the sampling point. Another important point to note is the figure showing unsuccessful compensation of the crosstalk in Figure 6.10(b). Clearly, the eye diagram is worse than the uncompensated signal in Figure 6.10(a). The partial compensation shown at 50 percent can be any percentage depending on when the crosstalk begins, and when the compensation signal arrives.

The results from the Simulink[®] simulations are expressed in BER (probability of error P_e) and summarized in the following graphs. Figure 6.11 shows the bit error rate graphs for a balanced ac coupled signal affected by crosstalk with and without compensation. Many of the systems of interest in this research involved ac coupled lines; therefore a graph showing the crosstalk compensation in this environment was deemed necessary. Although it is difficult to distinguish many of the lines, it can be seen that the equalizer compensation and the 2-bit selection compensation methods perform almost identically. This is an important conclusion since it was proposed earlier that the 2-bit selection circuit could perform as well as the equalizer circuit with less circuitry.

Figure 6.12 shows the simulation results for a non ac coupled signal. The results are much the same as in Figure 6.11 where it can be seen that both compensation techniques are effective in reducing the BER compared with no compensation. It is also important to notice that for the case when the compensation was too slow (added after the crosstalk infected signal was sampled) the BER actually went up. This shows the importance of successfully compensating the signal in time and how harmful it can be if the compensation is too late.

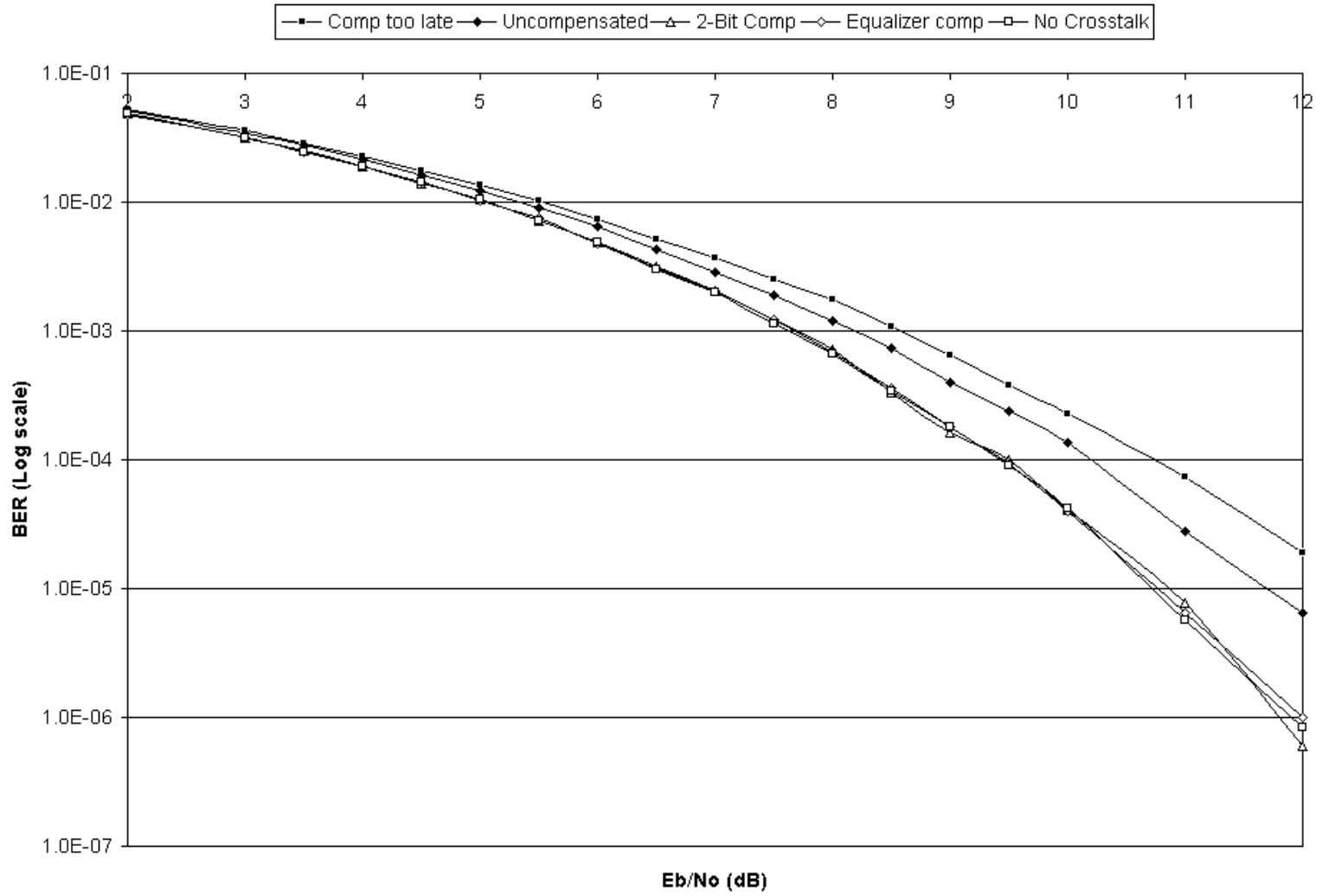


Figure 6.11 Bit Error Rate (BER) graph showing incoming ac coupled signal with crosstalk, with and without compensation.

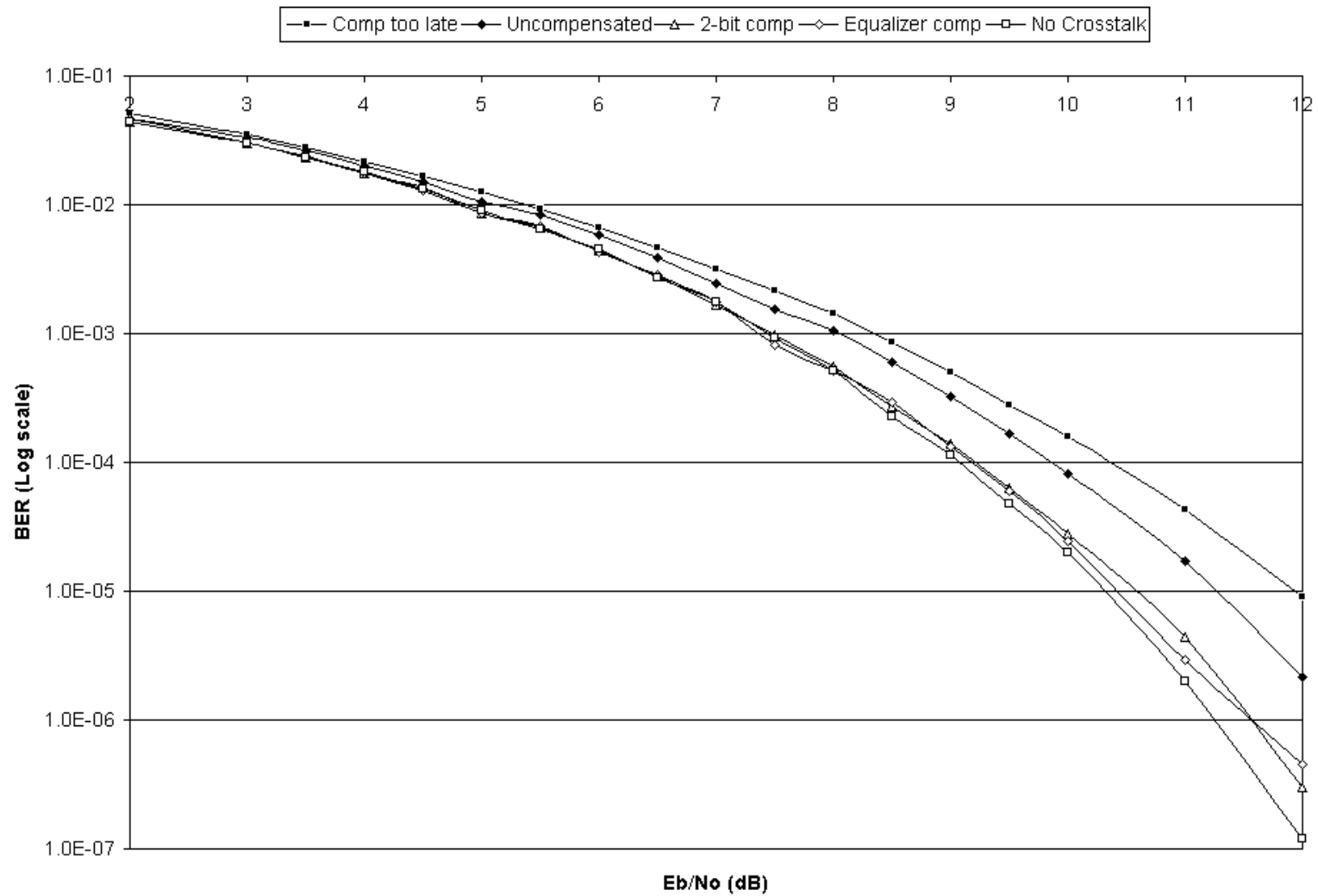


Figure 6.12 Bit Error Rate (BER) graph for non ac coupled signal showing crosstalk with and without compensation.

7 Conclusions and Future Work

7.1 Summary

Speed is a constant concern in almost every aspect of electronics. As data rates on printed circuit board (PCB) backplanes increase, they invariably reach limits that challenge engineers to overcome. At current data rates and logic switching times, there are several factors impeding the upward progression of bit error rates (BER) including dielectric losses, interference (crosstalk), and dc wander.

The objective of this thesis was to validate proposed compensation schemes used to alleviate the effects of dc wander and near end crosstalk (NEXT). Investigating the effects of these phenomena enabled the author to develop circuits to perform the needed compensation. These techniques were validated by first developing a theoretical framework of the problem and solution, then verifying the theory with simulation. After establishing reference bit error rate (BER) graphs for the polar NRZ code used in the research, numerous simulations were performed to verify the functionality of the designed circuits.

7.2 Results

In all the results graphs for the dc wander simulations, the 10-tap finite impulse response (FIR) filter showed a modest improvement in BER. This was expected as the time constant for the simulated system was much longer than the filter could compensate for. The 10-tap filter is also relatively large in size which is also a disadvantage. The same is true for the 50-tap FIR filter. Again, the BER improvement for the 50-tap FIR filter was fairly modest, although much better than the 10-tap FIR, but the size of the 50-tap filter is five times larger. It can be concluded that as the length of the FIR filter increases ('length' approaches that of the time constant of $\tau = 200T_s$)

the compensation graph would approach the ideal case, which is when there is no ac coupling. Obviously implementing a filter of this length ($200T_s = 200\text{-tap}$) for almost perfect compensation is not desirable considering the infinite impulse response (IIR) filter can perform the same (almost perfect) compensation with much less circuitry. The first order IIR digital filter proved to be the best design for lessening the effects of dc wander. As mentioned, its advantages over the FIR digital filters were its smaller size, less complexity and much better compensation ability. Although the 10-tap and 50-tap FIR filters offered improvements over no compensation, the improvement was small, and the large areas they would consume on an integrated circuit eclipsed any BER improvement.

The results for the crosstalk compensation show that the 2-bit selection compensation technique performed just as well as the original equalizer design, and has two major advantages over the equalizer method. Firstly, the size of the circuit would be much smaller than the equalizer method if implemented on an integrated circuit. This is very significant since silicon area can be very expensive. The second advantage is speed. The 2-bit selection method allowed the state of the tri-state circuit to be changed with much less delay, thus providing a much quicker compensation signal relative to the equalizer.

The important reality of this system, which was discovered after running several simulations with different timing parameters, is that timing is crucial. If the correctly generated compensation signal corrects the NEXT before the signal is sampled, the compensation technique works very well. Unfortunately, if the compensation circuit is too slow, it could potentially cause a *higher* BER than without the compensation. Successful compensation depends on the correcting signal being applied in the time range between a transmitted edge (beginning of crosstalk) and the sampling of the received signal. If there is enough time for the signal to be compensated prior to sampling, there is no problem. This could be made easier, for example, with integrated circuits containing a SERDES (SERializer / DESerializer) system [21]. In these systems the signal is 'known' in a parallel circuit prior to serialization (transmission). If the information is used by the compensation circuit at this point, there may be enough time

for the circuit to generate the appropriate correcting signal and apply it before the affected signal is sampled.

7.3 Future Work

Future work includes designing an amplitude detection circuit for the decision feedback system so as not to over or under compensate the incoming signal. Decision feedback, as the name implies, uses the ‘decision’ of a logic 1 or a logic 0 in the received signal to generate the feedback signal. The problem with this is that the logic representation of the bit does not describe the amplitude of the voltage of the incoming signal. An adaptive circuit is needed to detect the amplitude of the incoming signal that will adjust a gain element to provide the appropriate compensation signal level.

REFERENCES

- [1] Pauline Rigby, *Velio Cleans up*, Light Reading, <http://www.lightreading.com>
http://www.lightreading.com/document.asp?doc_id=3314. January 22, 2001
- [2] J. Bellamy, *Digital Telephony, 3rd Edition*, John Wiley & Sons, Inc., New York, NY, 2000.
- [3] M.W. Haney, H. Thienpont, T. Yoshimura, *Introduction to the Issue on Optical Interconnects*, IEEE Journal of Selected Topics in Quantum Electronics, vol. 9, no. 2, pp. 347-349, March/April 2003.
- [4] J. McLaren, *Integrated Services Digital Network (ISDN)*, Murray State University.
- [5] H.W. Johnson and M. Graham, *High-Speed Digital Design: A Handbook of Black Magic*, Prentice-Hall PTR, Upper Saddle River, New Jersey, 1993.
- [6] S.H. Hall, G.W. Hall and J.A. McCall, *High Speed Digital System Design: A Handbook of Interconnect Theory and Design Practices*, John Wiley & Sons, New York, NY, 2000.
- [7] D.M. Pozar, *Microwave Engineering, 2nd Edition*, John Wiley & Sons, Inc., New York, NY, 1998.
- [8] H.W. Johnson and M. Graham, *High-Speed Signal Propagation: Advanced Black Magic*, Prentice-Hall PTR, Upper Saddle River, New Jersey, 2003.
- [9] C. Schuster and W. Fichtner, *Parasitic Modes on Printed Circuit Boards and Their Effects on EMC and Signal Integrity*, IEEE Transactions on Electromagnetic Compatibility, vol. 43, no. 4, pp. 416 – 425, November 2001.
- [10] D.E. Dodds, *Baseband Digital Transmission*, EE352 Course notes, page 134 University of Saskatchewan, Saskatoon, Saskatchewan, Canada, September 2002, <http://dragon.usask.ca/Web-EE352/Chapters/Ch7-NRZ-03a.pdf>
- [11] W.H. Hayt, Jr., *Engineering Electromagnetics, 5th Edition*, McGraw-Hill, Inc., 1989.

- [12] E.P. Cunningham, *Digital Filtering: An Introduction*, Houghton Mifflin Company, 1992.
- [13] L.W. Couch II, *Digital and Analog Communication Systems, 5th Edition*, Prentice-Hall, Upper Saddle River, NJ, 1997.
- [14] H.W. Johnson, *Killer Packets*, High Speed Digital Design Newsletter, Volume 5 Issue 7.
- [15] M.C. Jeruchim, P. Balaban, and K.S. Shanmugan, *Simulation of Communication Systems*, Plenum Press, 233 Spring Street, New York, NY, 1992.
- [16] Y.S. Sohn, J.C. Lee, H.J. Park, S.I. Cho, *Empirical Equations on Electrical Parameters of Coupled Microstrip Lines for Crosstalk Estimation in Printed Circuit Board*, IEEE Transactions on Advanced Packaging, vol. 24, no. 4, pp. 521-527, November 2001.
- [17] Reuters, *Chip-Design breakthrough from Sun*, September 22, 2003. MSNBC website: <http://stacks.msnbc.com/news/970349.asp>
- [18] A. Staniforth, *Reverse Voltage Crosstalk (A discussion of Exact and Approximate Equations)*, Application Note 162, polarinstruments.com, 2002.
- [19] A.J. Rainal, *Impedance and Crosstalk of Stripline and Microstrip Transmission Lines*, IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part B, vol. 20, no. 3, August 1997.
- [20] M. Sung, W. Ryu, H. Kim, J. Kim, and J. Kim, *An Efficient Crosstalk Parameter Extraction Method for High-Speed Interconnection Lines*, IEEE Trans. on Advanced Packaging, vol. 23, no. 2, pp. 148-155, May 2000.
- [21] Pauline Rigby, *Mindspeed Claims SerDes Superiority*, Light Reading, <http://www.lightreading.com> November 07, 2001
http://www.lightreading.com/document.asp?doc_id=9338
- [22] J.G. Proakis, *Digital Communications, 4th Edition*, McGraw-Hill Inc., 1221 Avenue of the Americas, New York, NY, 2001.

A. Calculations

A.1 Digital IIR Filter

Using the Bilinear Transform (BLT), the z-domain transfer function of a lowpass filter with a cutoff frequency, ω_c (Equation 4.2) is found to be

$$H(z)_{LPF} = \left(\frac{\omega_c}{2/T + \omega_c} \right) \cdot \left[\frac{z}{z - \beta} + \frac{1}{z - \beta} \right] \quad (\text{A.1})$$

where $\beta = \frac{2/T - \omega_c}{2/T + \omega_c}$. Taking the inverse z-transform of Equation 4.2 results in the time domain expression

$$h(t)_{LPF} = \frac{\omega_c}{2/T + \omega_c} (\beta^t + \beta^{t-1}) \cdot u(t) = \left(\frac{4\omega_c T}{4 - (\omega_c T)^2} \right) \cdot \beta^t u(t). \quad (\text{A.2})$$

If $\omega_c \ll 1$, which is expected, then $\left(\frac{4\omega_c T}{4 - (\omega_c T)^2} \right) \approx (1 - \beta)$ and the final time domain expression is given by

$$y(t) = (1 - \beta) \cdot \beta^t \cdot u(t). \quad (\text{A.3})$$

Analyzing the block diagram results in the following difference equation

$$y(nT) = (1 - \beta) \cdot x(nT) + \beta \cdot y((n-1)T) \quad (\text{A.4})$$

and the z-transform version of Equation A.4 is

$$Y(z) = (1 - \beta) \cdot X(z) + \beta \cdot Y(z)z^{-1}. \quad (\text{A.5})$$

With some simple algebraic manipulation, the z-domain transfer function is found to be

$$H(z) = \frac{Y(z)}{X(z)} = (1 - \beta) \left(\frac{z}{z - \beta} \right) \quad (\text{A.6})$$

and the time domain expression is obtained by taking the inverse z-transform of Equation A.6

$$y(t) = (1 - \beta) \cdot \beta^t u(t) \quad (\text{A.7})$$

which matches the earlier Equation A.3 which was calculated using the Bilinear Transform.

A.2 Digital FIR Filter

Using Equation (A.7) derived previously with discrete values results in

$$y(nT) = (1 - \beta) \cdot \beta^{nT} u(nT), \quad n = 1, 2, \dots, (N - 1) \quad (\text{A.8})$$

where $T = 1$ and N is the length of the filter. With $\beta = \frac{2/T - \omega_c}{2/T + \omega_c}$ and $\omega_c = 1/\tau$, the tap values can be calculated as

$$y(0) = (1 - \beta) \cdot \beta^0$$

$$y(1) = (1 - \beta) \cdot \beta^1$$

$$\vdots$$

$$y(N - 1) = (1 - \beta) \cdot \beta^{N-1}.$$

Alternatively, the design of the digital FIR filters can be derived from the time domain expression of the analog exponential decay of the ac-coupling signal which is

$$h(t) = e^{-t/\tau} \quad (\text{A.9})$$

where the time constant of the system was chosen earlier to be $\tau = 200T$. In the digital domain, Equation (A.8) is represented as

$$h(nT) = e^{-nT/\tau}. \quad (\text{A.10})$$

The desired digital filter will have to output a behaviour opposite to the decay exhibited by the ac coupling circuit. This is done with the following formula

$$G(n) = e^{-nT/\tau} - e^{-(n+1)T/\tau} \quad n = 0, 1, \dots, (N-1) \quad (\text{A.11})$$

where $G(n)$ is the gain value of multiplier 'n' and N is the length of the filter.

A.3 Energy per Bit to Noise Spectral Density Ratio (E_b/N_0)

The calculation of bit energy is required to operate the Simulink[®] AWGN block. To calculate the energy per bit the average power per bit must first be calculated. Average bit power for a discrete time signal is calculated [22] by

$$P = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 \quad (\text{A.12})$$

where $x(n)$ is the signal, and N is the number of samples in one bit period. The number of samples per bit for the simulations was chosen to be $N = 16$, thus the sample time (T_s) is one-sixteenth the bit period (T_b in seconds). Once the average power per bit is calculated (P has units of watts, or joules/ T_b), the energy per sample is calculated. The units of which are determined in Equation A.13.

$$E_s = P \cdot T_s = \left(\frac{\text{joules}}{\text{sec}} \right) \cdot \left(\frac{\text{sec}}{\text{sample}} \right) = \frac{\text{joules}}{\text{sample}} \quad (\text{A.13})$$

The Simulink[®] AWGN block requires the input signal power (watts) and the symbol period to generate the user defined value for E_b/N_0 . In this case the symbol period is the sample period, T_s , which is a fraction (1/16) of T_b .

B. Simulink[®] Simulation Blocks and Parameters

B.1 Bipolar Signal Transmitter

This is a user-defined subsystem (Figure B.1) that contains a Bernoulli Binary Generator (BBG) and a Look-up Table (LUT). The BBG simply generates a binary sequence (unipolar NRZ) with a user-specified ‘probability of zero’. This is particularly useful for generating an unbalanced signal. The binary sequence can also be changed by the user by specifying an ‘initial seed’. For convenience, the sample time is set to one second. The LUT converts the unipolar NRZ signal to polar NRZ.

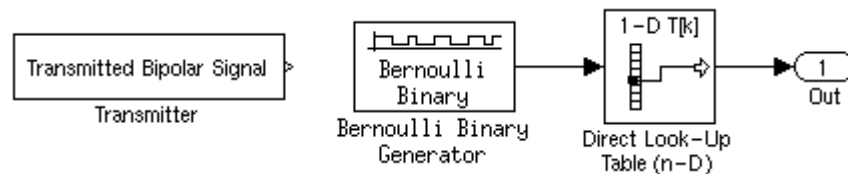


Figure B.1 Transmitter block and contents in simulations.

B.2 Ac Coupling

Although the ac coupling subsystem is user defined, it contains only one Simulink[®] block, a continuous-time transfer function as shown in Figure B.2. The values of the transfer function are determined by using Laplace domain circuit analysis on a first order (highpass filter) series RC circuit.

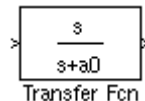


Figure B.2 Ac coupling block for simulations.

This results in the transfer function in Equation 4.1 which can be entered into the transfer function block in Simulink[®].

B.3 Analog to Digital Converter

This subsystem contains only one important Simulink[®] block, *Sampled Quantizer Encode*, which digitizes the signal (Figure B.3). This is done to facilitate the addition of noise to the system and the recovery of the signal at the receiver. An accurate digitization of the signal is needed, so a sample time of $t=0.0625T_b$ is chosen with a bit rate of $T_b=1\text{second}$ (for convenience). Assuming bit voltage levels are 1 volt (logic 1) and -1 volt (logic 0), the quantization will then have to span at least 2 volts. Dc wander will also cause the signal to move up and down to a maximum of 2 volts (or minimum of -2 volts), therefore the limits of the quantization are set to 2.0 volts and -2.0 volts in intervals of 0.0625 Volts.

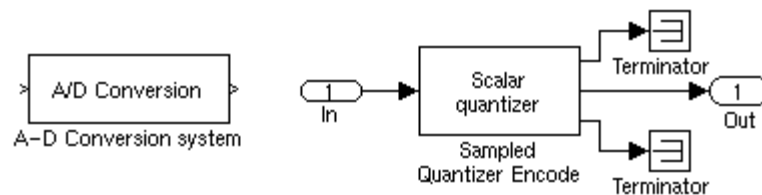


Figure B.3 The A-D conversion block (left) contains a sampled quantizer encode block (right).

B.4 Transmission Line Simulation

Transmission line simulation is done in Matlab[®]. Once the polar signal has been generated and passed through the ac coupling block, then digitized, the result is output to the Matlab[®] workspace. There it is run through the program 'tlinemodel.m' which shapes the signal by convolving it with a gaussian waveform. The transmission line formulas were used from [7] and the step input response was generated. The signal was then convolved with the transmission line step response and saved to a file. This file was the input to the second stage of the dc wander compensation circuit. The file 'tlinemodel.m' is given in Appendix D.

B.5 Additive White Gaussian Noise (AWGN) Channel

This Simulink[®] block receives a discrete-time signal and adds a pseudo-random noise to it (Figure B.4). Changing the *initial seed* variable in the *Properties* of this block can change the noise sequence from one simulation to the next. The bit time remains at $T=1\text{second}$ which was defined earlier, and in accordance with the rest of the system. The signal-to-noise ratio, E_b/N_0 , is varied by entering a value (in decibels) to the block properties. The input signal power is also required (in watts) which is calculated for a polar signal with a rectangular pulse shape in [13] to be

$$P(f) = T_b \int_{-\infty}^{\infty} \left(\frac{\sin(\pi f T_b)}{\pi f T_b} \right)^2 df = 1 \text{ Watt.} \quad (\text{B.1})$$

However, since the previous subsystem (quantizer) used a sampling frequency of $t=0.0625T_b$, this must be used to calculate the input power for each sample. This becomes a value entered into the AWGN block for sample time. To determine the input signal power a sample waveform was chosen and integrated as per formula B.1 which results in $P(f) = 0.2 \text{ Watts/symbol}$. Also see Section A.3

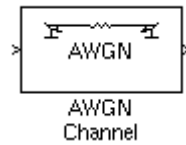


Figure B.4 Additive white gaussian noise block for simulations.

B.6 Receiver

The downsampling block in Simulink[®] (Figure B.5(a)) uses one in every N samples as the value of the bit. Since the quantizer samples the analog signal $N=16$ times, the downsampler takes every 16th sample as the bit value. Since the optimum sample spot is in the middle of the “eye”, the signal is delayed using an *integer delay* block (Figure B.5(b)). The integer delay block is set to delay the signal by 10 samples, or one-tenth of a symbol period. The delay is determined by examining the output of the system and adjusting manually. Once the downsampling is complete, a decision is made whether the symbol is represents a logic 1 or a logic 0. This is done with the *Signum*

block (Figure B.5(c)). It uses the value zero as the threshold value and sets the bit according to

$$y(nT) = \begin{cases} +1, & x(nT) \geq 0 \\ -1, & x(nT) < 0 \end{cases}$$

where $x(nT)$ and $y(nT)$ are the input and output to the Signum block, respectively. The *Complex to Real* block is used to convert the signal from a complex value to a real value (Figure B.5(d)).

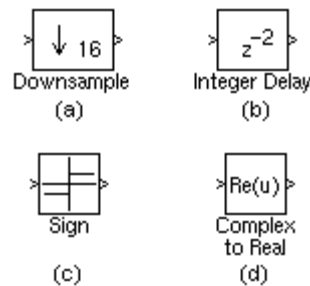


Figure B.5 Simulink[®] blocks (a) downsample, (b) integer delay, (c) signum, and (d) complex to real conversion.

B.7 Digital Filter for Decision Feedback

The digital filter subsystem contains several basic Simulink[®] blocks including *unit delay*, *gain*, and *sum* blocks. The unit delay block simply delays its input by one clock period. The gain block can take on any value which is set by the user, and simply multiplies its input by its gain value. The sum block can be set to add or subtract multiple inputs by adding "+" or "-" signs in the properties box. As the data propagates through the filter, a signal is generated, fed back, and added to the incoming signal (before it reaches the receiver) to offset the effects of the ac coupling. The digital IIR and FIR filters used in this research were designed in previous sections.

B.8 Discrete-Time Eye Diagram Scope

This block, shown in Figure B.6, is used to determine the quality of the transmitted signal as it reaches the receiver. The sampling frequency was 16 samples/bit and a useful eye diagram shows a complete eye, with about half an eye on either side.

Therefore in the ‘samples per symbol’ field a 32 is entered. The rest of the parameters are adjusted as the user sees fit to get a proper eye.

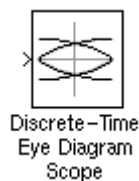


Figure B.6 Eye diagram scope block used in simulations.

B.9 Error Rate Calculation and Display

The ‘error rate calculation’ block compares the input bit-stream with the output bit-stream. The bit-streams should be the same if the data is properly received. If a bit is found to be wrong the ‘bit error rate’ is calculated and output to the display block. The display block shows the number of bits transmitted, number of errors, and the bit error rate. Both blocks are shown in Figure B.7.

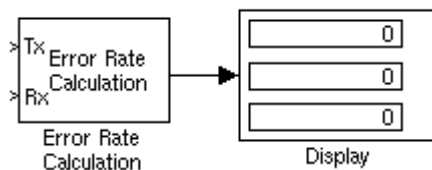


Figure B.7 Error rate calculation and display blocks.

B.10 Edge Detector Block

A ‘Detect Rise Positive’ block, shown in Figure B.8, performs the edge detection needed for the crosstalk compensation circuit. This block produces a short pulse every time a rising edge is detected.

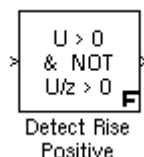


Figure B.8 Edge detection block.

B.11 Crosstalk Equalizer

This block was designed as a crosstalk correction circuit using the ‘detect rise positive’ block as the edge detector that generates the pulses needed. The pulses are then fed through the equalizer which contain delay elements, multipliers and an adder as shown in Figure B.9. When the values from the multipliers are added, the result is output to the main simulation program.

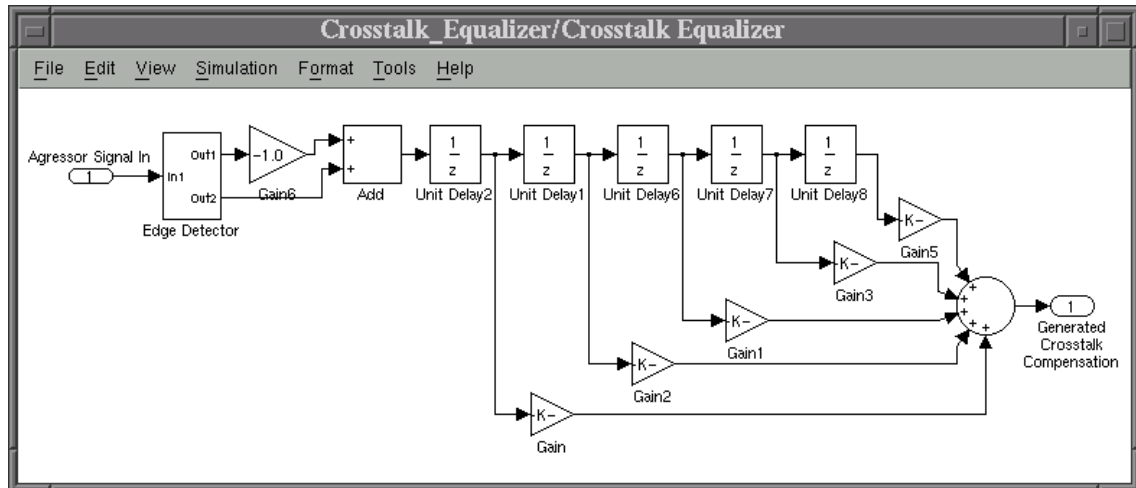


Figure B.9 Crosstalk equalizer block diagram for simulation in Simulink®.

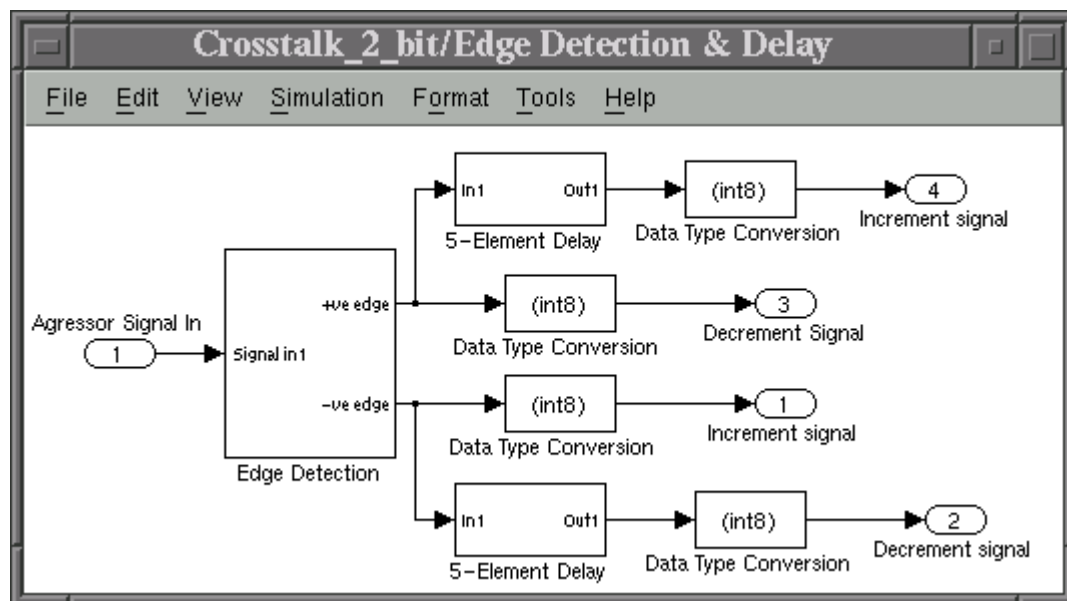


Figure B.10 Edge detection and delay simulation circuit for 2-bit adder.

B.12 2-bit Compensation Circuit

The 2-bit compensation circuit simulation block was designed to mirror the behaviour of the proposed circuit as closely as possible. Although the blocks do not look exactly as they do in the circuit diagrams in Section 6.4, the behaviour is almost exactly the same. The simulation circuit for edge detection and pulse delay is shown in Figure B.10 which contains ‘Detect Rise Positive’ blocks as edge detectors. The tri-state circuit is simulated with an “adder” circuit shown in Figure B.11.

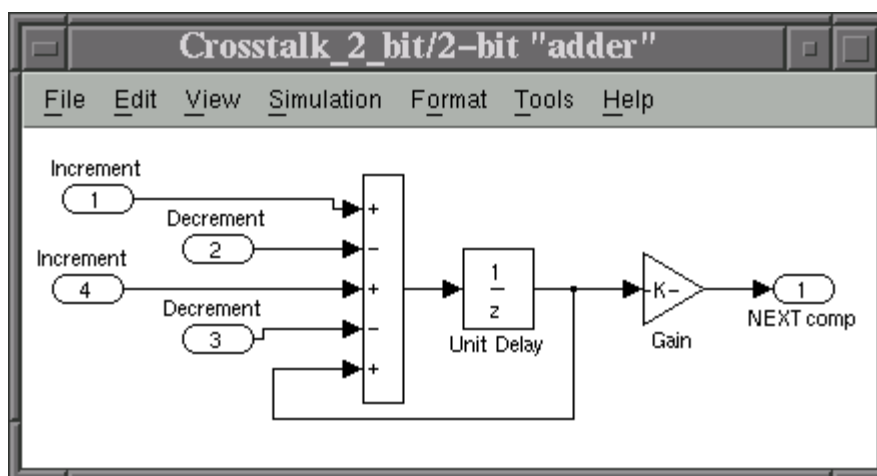


Figure B.11 2-bit crosstalk "adder" simulation circuit.

B.13 Generation of Crosstalk Interference

The generation of the crosstalk, which is added to the incoming signal, is done much the same way the compensation signal is generated for the equalizer method. A single diagram was made to show the different levels of the circuit that was designed in Simulink®. This circuit is shown in Figure B.12 below. The highest block in the hierarchy is the ‘Add crosstalk to signal’ block in the top left of the figure. Inside this block is the circuit shown directly to its right. The block outlined in bold black line is expanded in the middle circuit (follow the arrow). If the bold outlined block in this diagram is now opened, the circuit in the bottom left emerges. Finally, the circuit on the bottom right is present twice (see arrows) in the circuit on the bottom left.

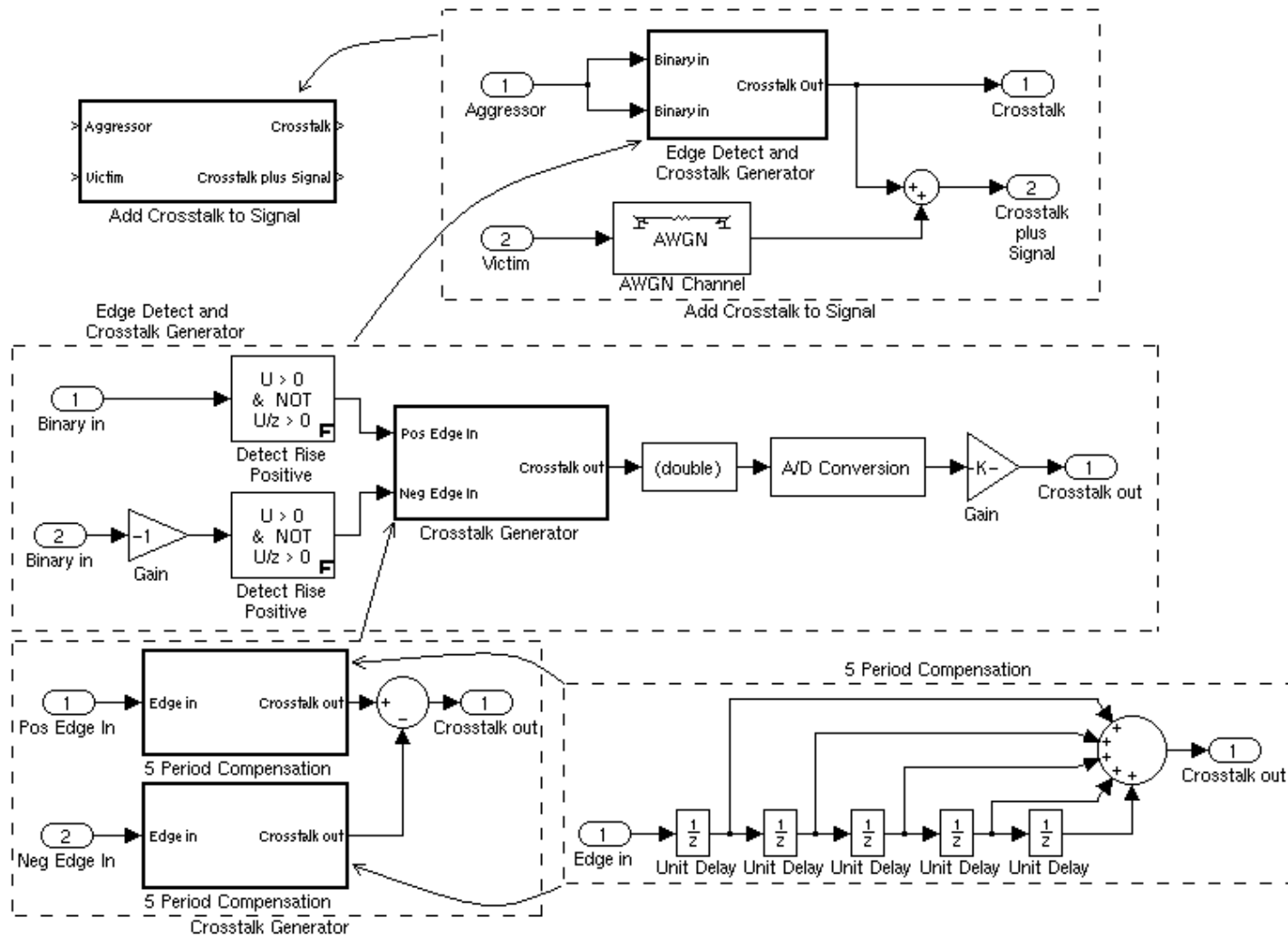


Figure B.12 Crosstalk generator for Simulink® simulations.

C. Results Tables

C.1 Decision Feedback Results Tables

Table C.1 Dc wander compensation with balanced signal ($P_z = 0.5$).

E_s/N_0	Baseline		No Filter		10-tap FIR filter		50-tap FIR filter		IIR filter	
(dB)	Bits	Errors	Bits	Errors	Bits	Errors	Bits	Errors	Bits	Errors
	($\times 10^6$)		($\times 10^6$)		($\times 10^6$)		($\times 10^6$)		($\times 10^6$)	
2	0.5	22125	0.5	24193	0.5	23989	0.5	24103	0.5	24207
3	0.5	15112	0.5	15886	0.5	15801	0.5	15740	0.5	15714
3.5	0.5	11735	0.5	12312	0.5	12349	0.5	12130	0.5	12153
4	0.5	9021	0.5	9554	0.5	9427	0.5	9409	0.5	9283
4.5	0.5	6607	0.5	7125	0.5	6922	0.5	6882	0.5	6906
5	0.5	4530	0.5	5237	0.5	5071	0.5	4928	0.5	4933
5.5	0.5	3257	0.5	3582	0.5	3468	0.5	3505	0.5	3412
6	0.5	2251	0.5	2437	0.5	2417	0.5	2346	0.5	2258
6.5	0.5	1355	0.5	1484	0.5	1610	0.5	1579	0.5	1442
7	0.5	877	0.5	1002	0.5	964	0.5	930	0.5	906
7.5	0.5	463	0.5	563	0.5	612	0.5	598	0.5	501
8	0.5	258	0.5	333	0.5	354	0.5	383	0.5	267
8.5	0.5	113	0.5	172	0.5	180	0.5	173	0.5	146
9	1.0	113	0.5	91	1.0	179	1.0	168	1.0	135
9.5	2.0	96	1.0	90	2.0	174	2.0	171	2.0	119
10	3.0	60	2.5	104	4.0	166	4.0	145	4.0	112
11	7.5	15	9.0	51	11.5	73	11.5	59	9.0	31
12	25.0	3	12.0	10	20.0	13	20.0	15	15.0	7

Table C.2 Dc wander compensation with unbalanced signal ($P_z = 0.4$).

E_s/N_0	Baseline		No Filter		10-tap FIR filter		50-tap FIR filter		IIR filter	
	(dB)	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)
2	0.5	22125	0.5	29790	0.5	29652	0.5	28381	0.5	23613
3	0.5	15112	0.5	20924	0.5	20662	0.5	19457	0.5	15069
3.5	0.5	11735	0.5	17271	0.5	16782	0.5	15639	0.5	11724
4	0.5	9021	0.5	13766	0.5	13591	0.5	12350	0.5	8901
4.5	0.5	6607	0.5	10774	0.5	10364	0.5	9374	0.5	6457
5	0.5	4530	0.5	8422	0.5	8014	0.5	7151	0.5	4649
5.5	0.5	3257	0.5	6465	0.5	6079	0.5	5260	0.5	3211
6	0.5	2251	0.5	4672	0.5	4428	0.5	3798	0.5	2113
6.5	0.5	1355	0.5	3400	0.5	3135	0.5	2596	0.5	1332
7	0.5	877	0.5	2389	0.5	2118	0.5	1765	0.5	832
7.5	0.5	463	0.5	1629	0.5	1510	0.5	1108	0.5	473
8	0.5	258	0.5	1064	0.5	1018	0.5	706	0.5	275
8.5	0.5	113	0.5	714	0.5	647	0.5	397	0.5	125
9	1.0	113	1.0	797	1.0	713	1.0	503	1.0	136
9.5	2.0	96	2.0	997	2.0	874	2.0	547	2.0	113
10	3.0	60	4.0	1086	4.0	970	4.0	492	4.0	81
11	7.5	15	11.5	779	11.5	632	11.5	290	11.5	24
12	25.0	3	20.0	290	20.0	228	20.0	102	20.0	4

Table C.3 Dc wander compensation with unbalanced signal ($P_z = 0.3$).

E_s/N_0 (dB)	Baseline		No Filter		10-tap FIR filter		50-tap FIR filter		IIR filter	
	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors
2	0.5	22125	0.5	52412	0.5	51016	0.5	28381	0.5	24761
3	0.5	15112	0.5	42080	0.5	40417	0.5	19457	0.5	15510
3.5	0.5	11735	0.5	36665	0.5	35781	0.5	15639	0.5	11936
4	0.5	9021	0.5	32480	0.5	31068	0.5	12350	0.5	8900
4.5	0.5	6607	0.5	28095	0.5	26700	0.5	9374	0.5	6528
5	0.5	4530	0.5	23916	0.5	22381	0.5	7151	0.5	4740
5.5	0.5	3257	0.5	20302	0.5	19124	0.5	5260	0.5	3198
6	0.5	2251	0.5	16904	0.5	15678	0.5	3798	0.5	2083
6.5	0.5	1355	0.5	14244	0.5	12893	0.5	2596	0.5	1311
7	0.5	877	0.5	11189	0.5	10195	0.5	1765	0.5	828
7.5	0.5	463	0.5	8863	0.5	7844	0.5	1108	0.5	481
8	0.5	258	0.5	6866	0.5	6056	0.5	706	0.5	250
8.5	0.5	113	0.5	5194	0.5	4385	0.5	397	0.5	114
9	1.0	113	1.0	7746	1.0	6683	1.0	503	1.0	110
9.5	2.0	96	2.0	11259	2.0	9107	2.0	547	2.0	96
10	3.0	60	4.0	15644	4.0	12746	4.0	492	4.0	78
11	7.5	15	11.5	19905	11.5	15312	11.5	290	11.5	30
12	25.0	3	20.0	13800	20.0	9945	20.0	102	20.0	8

C.2 Crosstalk Compensation Results Tables

Table C.4 Crosstalk compensation circuit with ac coupled data.

E_S/N_0	Baseline		No Comp		Comp too late		Equalizer		2-Bit Selection	
(dB)	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors
2	0.5	22125	0.5	25641	0.5	26170	0.5	24012	0.5	24204
3	0.5	15112	0.5	16955	0.5	18001	0.5	15748	0.5	15618
3.5	0.5	11735	0.5	13759	0.5	14354	0.5	12245	0.5	12474
4	0.5	9021	0.5	10728	0.5	11443	0.5	9430	0.5	9488
4.5	0.5	6607	0.5	8089	0.5	8784	0.5	7119	0.5	7060
5	0.5	4530	0.4	4920	0.5	6761	0.5	5184	0.5	5238
5.5	0.5	3257	0.4	3569	0.5	5077	0.5	3748	0.5	3600
6	0.5	2251	0.4	2394	0.5	3683	0.5	2394	0.5	2451
6.5	0.5	1355	0.4	1737	0.5	2583	0.5	1548	0.5	1599
7	0.5	877	0.4	1129	0.5	1841	0.5	1016	0.5	1032
7.5	0.5	463	0.5	942	0.5	1252	0.5	618	0.5	606
8	0.5	258	0.5	569	0.5	876	0.5	341	0.5	358
8.5	0.5	113	0.5	365	0.5	539	0.5	178	0.5	165
9	1.0	113	0.5	198	1.0	643	1.0	180	1.0	163
9.5	2.0	96	2.5	597	2.0	763	2.0	185	2.0	199
10	3.0	60	4.0	543	4.0	899	4.0	161	4.0	165
11	7.5	15	11.5	322	11.5	839	11.5	74	11.5	89
12	25.0	3	20.0	128	20.0	383	20.0	20	20.0	12

Table C.5 Crosstalk compensation circuit with non ac coupled data.

E_s/N_0	Baseline		No Comp		Comp too late		Equalizer		2-Bit Selection	
(dB)	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors	Bits ($\times 10^6$)	Errors
2	0.5	22125	0.5	23231	0.5	25624	0.5	23256	0.5	22102
3	0.5	15112	0.5	16474	0.5	17404	0.5	15213	0.5	15216
3.5	0.5	11735	0.5	13146	0.5	13907	0.5	11849	0.5	11764
4	0.5	9021	0.5	10048	0.5	10757	0.5	8842	0.5	8826
4.5	0.5	6607	0.5	7618	0.5	8357	0.5	6463	0.5	6772
5	0.5	4530	0.5	5206	0.5	6355	0.5	4298	0.5	4340
5.5	0.5	3257	0.5	4208	0.5	4612	0.5	3389	0.5	3290
6	0.5	2251	0.5	2891	0.5	3335	0.5	2163	0.5	2205
6.5	0.5	1355	0.5	1945	0.5	2340	0.5	1442	0.5	1391
7	0.5	877	0.5	1222	0.5	1597	0.5	885	0.5	840
7.5	0.5	463	0.4	616	0.5	1081	0.5	412	0.5	492
8	0.5	258	0.4	416	0.5	707	0.5	254	0.5	281
8.5	0.5	113	0.4	242	0.5	426	0.5	147	0.5	137
9	1.0	113	0.8	261	1.0	503	1.0	132	1.0	139
9.5	2.0	96	2.0	331	2.0	551	2.0	119	2.0	125
10	3.0	60	4.0	322	4.0	641	4.0	98	4.0	111
11	7.5	15	11.0	190	11.5	500	11.5	34	11.5	51
12	25.0	3	20.0	43	20.0	182	20.0	9	20.0	6

D. Program Code and Block Diagrams

D.1 Matlab[®] Code

There were two versions of transmission line simulation files. They are used in the same manner but use different sets of formulae. There is also a file to be run before the dc wander correction and crosstalk correction simulations that will calculate the multiplier values used during the simulations.

- tlinemodel.m
- tapmultipliers.m
- crosstalkparams.m

These files require a set of data created in the Simulink[®] file AAWGN which is 5×10^5 bits long sampled 16 times per bit. The resultant waveform (with transmission line deterioration) is saved to a file. The file is then ready to be the input to either the dc wander correction system or the crosstalk correction system.

D.1.1 Transmission Line Modeling (tlinemodel.m)

```

% Transmission line modeling
% June 20th, 2003
% Bernie Boos
% File Name: tlinemodel.m
% This file takes an input waveform generated in Simulink
% and applies transmission line effects on it.

% ----- Constants -----

dt=20e-12;    % Time division (seconds)
N=2^19;      % Length of sample vector

k=0:N/2;     % Index to frequency points
fk=k./(N*dt); % Frequencies used to sample Fourier transform functions (Hertz)

wk=2*pi.*fk; % Same as fk, but in radians
wk(1)=wk(2)/1e6;% Offset zeroth frequency slightly from zero to avoid divide by zero error

x=0.2;      % Length of transmission line (meters)
vo=1.4448e8; % Velocity of propagation on stripline (m/s)
Zo=50;      % Characteristic impedance (ohms)
thetao=0.027; % Angle formed by real and imaginary parts
            % of complex electric permittivity at frequency wo (radians)
Rdc=5.6;    % dc resistance
wo=1*10^9; % Frequency at which ac line parameters are specified
Ro=40.7;    % Real part of ac resistance at frequency wo

% ----- Formulas -----

Co=1/(Zo*vo);
L=Zo/vo;
C=(Co.*((j.*wk)./wo).^((-2*thetao)/pi));
Rac=(Ro*(1+j)).*(sqrt(wk./wo));
R=sqrt((Rdc^2)+(Rac.^2));

```



```

% ----- Pulse Shaping -----
points=length(fk);      % Number of points for time and frequency scales
fs=1/(dt);             % Sampling frequency

freqscale=(0:(fs/(points)):(fs-(fs/(points))));
timescale=(0:dt:dt*(points-1));

HW=4;                  % Gaussian window for pulse shaping
hw=gausswin(HW);      % gaussian waveshaper

% Shaping input waveform
waveshaping=conv(sampleddata.signals.values,hw)./(HW/2);

% Propagation constant for transmission line
gamma=sqrt((R+(j*L.*wk)).*(j.*C.*wk));

% Frequency representation of transmission line
TLINETF=exp(-x.*gamma);

% Inverse fourier transform to get time domain
tlinetf=ifft(TLINETF);

% Shorten time domain representation of t-line
% so convolution does not take too long
m=6;
len=1500;
shorth=tlinetf(1:len);

% Clear up some memory for producing large files

clear tlinetf; clear TLINETF; clear gamma;
clear sampleddata; clear hw; clear n; clear tn; clear k; clear fk; clear wk;
clear C; clear Rac; clear R; clear Zs; clear Zl; clear Zc;
clear freqscale; clear timescale;

pack; % Memory management

```

```
% end clear up memory

% Convolving input (shaped) with time domain t-line
distorted=conv(shorth,waveshaping);

save('distorted.mat', 'distorted');
% End of program tlinemodel.m
```

D.1.2 Calculating Filter Tap Multipliers (tapmultipliers.m)

```
% Bernie Boos
% Matlab program
% File name: tapmultipliers.m
% Used to calculate tap weights for FIR and IIR filters
% June 13th, 2003

clear all; close all;
time_values=[0:1:50]; % Time values
tau=200; % Time constant
wc=1/tau;

decay_values=exp(-time_values/tau); % Calculate decay

% Calculate the multiplier values for filters, need 50 values for the 50-tap filter

for i=[1:(length(time_values)-1)]
    tap_values(i)=decay_values(i)-decay_values(i+1);
end;

beta=(2-wc)/(2+wc); % Calculate values for IIR filter
inbeta=(1-beta);

% Put the multiplier values in variables to use in Simulink
a0=tap_values(1);
a1=tap_values(2);
a2=tap_values(3);
a3=tap_values(4);
a4=tap_values(5);
a5=tap_values(6);
a6=tap_values(7);
a7=tap_values(8);
a8=tap_values(9);
a9=tap_values(10);
a10=tap_values(11);
a11=tap_values(12);
```

```
a12=tap_values(13);
a13=tap_values(14);
a14=tap_values(15);
a15=tap_values(16);
a16=tap_values(17);
a17=tap_values(18);
a18=tap_values(19);
a19=tap_values(20);
a20=tap_values(21);
a21=tap_values(22);
a22=tap_values(23);
a23=tap_values(24);
a24=tap_values(25);
a25=tap_values(26);
a26=tap_values(27);
a27=tap_values(28);
a28=tap_values(29);
a29=tap_values(30);
a30=tap_values(31);
a31=tap_values(32);
a32=tap_values(33);
a33=tap_values(34);
a34=tap_values(35);
a35=tap_values(36);
a36=tap_values(37);
a37=tap_values(38);
a38=tap_values(39);
a39=tap_values(40);
a40=tap_values(41);
a41=tap_values(42);
a42=tap_values(43);
a43=tap_values(44);
a44=tap_values(45);
a45=tap_values(46);
a46=tap_values(47);
a47=tap_values(48);
a48=tap_values(49);
```

```
a49=tap_values(50);

load('datadisc1/thirty_10220'); % Load input data file (change file to load different data)

% Put data into form (structure) that can be loaded by Simulink
distortedsignal.time=[];
distortedsignal.signals.values=distorted;
distortedsignal.signals.dimensions=[1];

% This is to set the first few bits to a known number , delay elements need an initial condition
distortedsignal.signals.values(1:48)=-1;

clear distorted; % Free up some memory
% End of tapmultipliers.m
```

D.1.3 Calculating Crosstalk Parameters (crosstalkparams.m)

```
% Bernie Boos
% Matlab program
% File name: crosstalkparams.m
% Used to calculate multipliers for crosstalk circuits
% Can be used for both stripline and microstrip
% August 15th, 2003

stripline=1;
if stripline
    microstrip=0;
else
    microstrip=1;
end

if stripline
    amax=0.065; % Maximum crosstalk as calculated for specific stripline configuration
    mult=0.05;  % multiplier for compensation
end

if microstrip
    amax=0.25; % Maximum crosstalk as calculated for microstrip configuration end

% Load data from file for input (change file to load different data)
load('matlab code/datadisc1/NACC4');

% Put loaded data into structure recognized by Simulink
distortedsignal.time=[];
distortedsignal.signals.values=distorted;
distortedsignal.signals.dimensions=[1];

% Set first few bits to -1, for delay element initial conditions
distortedsignal.signals.values(1:48)=-1;

clear distorted; % Clear up some memory
% End of crosstalkparams.m
```

D.2 Simulink® Block Diagrams of Complete Systems

Below are the main Simulink® programs used to simulate the systems in this thesis. The block diagram in Figure D.1 generates the polar NRZ waveform, performs ac coupling and an A-D conversion on the signal and sends the resultant 500k bit waveform to the Matlab® workspace.

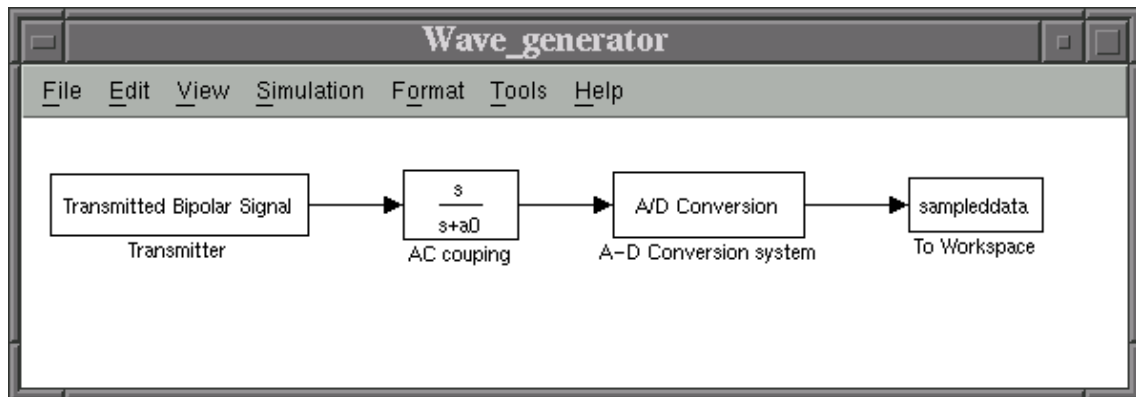


Figure D.1 Waveform generator, with ac coupling block and A-D conversion.

Once the data is on the workspace, a transmission line simulation will be performed by running a Matlab® program called ‘tlinemodel.m’. This program will generate data that will be the input to the systems in Figure D.2, Figure D.3, and Figure D.4. There were 25 different files created for each of the following types of signal:

- Non ac coupled signal.
- Balanced ac coupled signal ($P_z = 0.5$).
- Unbalanced ac coupled signal ($P_z = 0.4$).
- Unbalanced ac coupled signal ($P_z = 0.3$).

The diagram in Figure D.2 is the complete circuit used to simulate the compensation of dc wander. This diagram has the IIR filter included, but is the same file used to simulate the FIR filters.

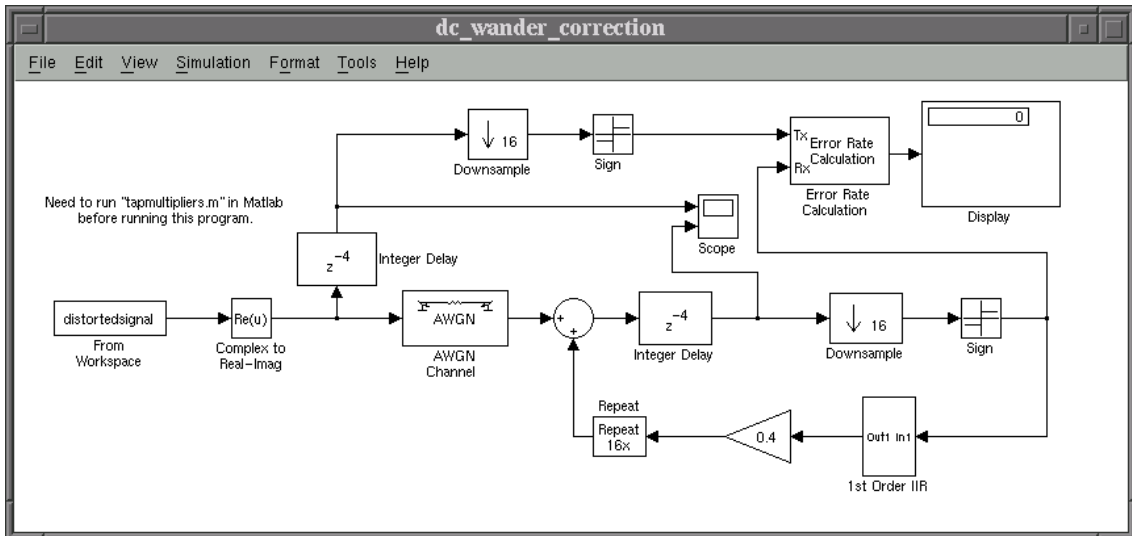


Figure D.2 Complete circuit for dc wander compensation with decision feedback.

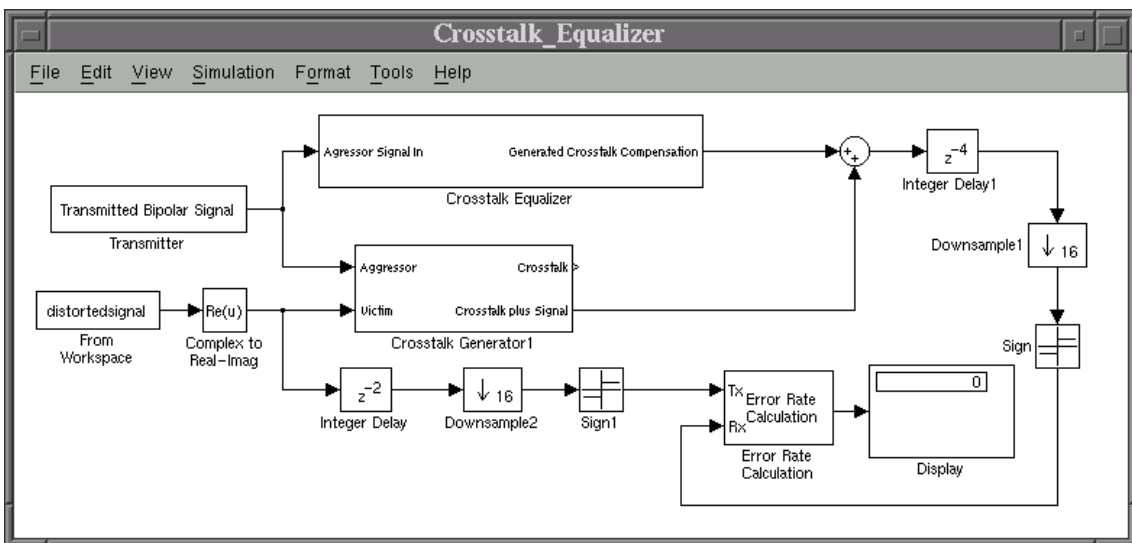


Figure D.3 Crosstalk compensation circuit with equalizer technique.

The diagram in Figure D.3 shows the simulation setup to demonstrate the crosstalk compensation circuit for the equalizer configuration. The diagram showing inside the equalizer block is given in Appendix C. The 2-bit selection compensation technique is shown in Figure D.4 which includes the complete simulation setup. The outgoing signal (that is used to create the crosstalk interference) is generated as the simulation runs while the ‘received’ signal is imported from the Matlab® workspace.

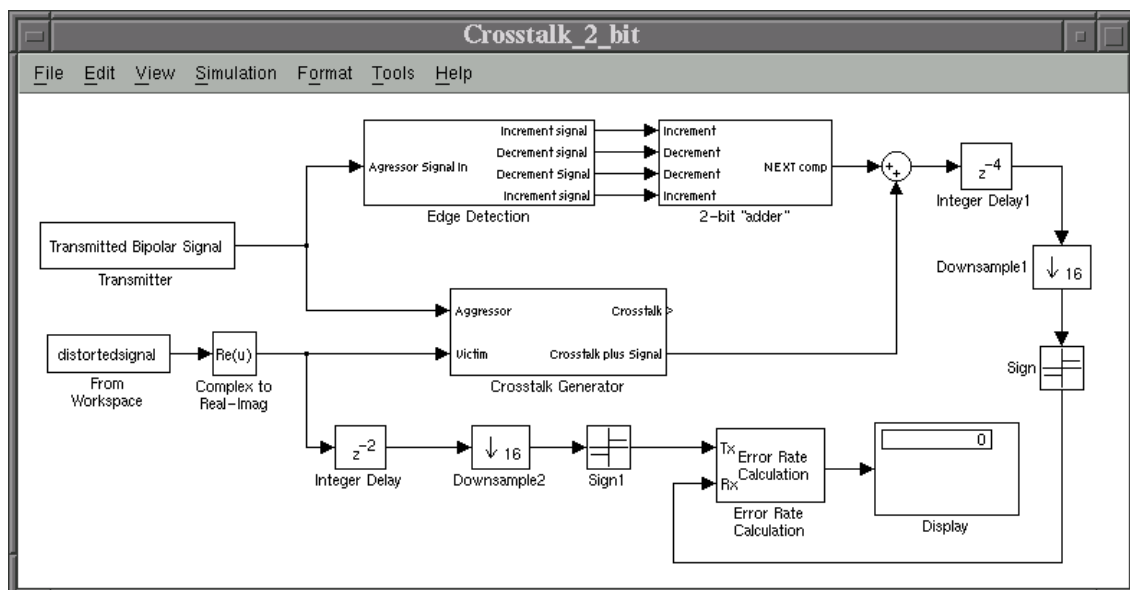


Figure D.4 Crosstalk correction using 2-bit selection compensation technique.

E. Circuit Diagrams from Quartus II®

The following block diagrams are screen captures from the Quartus II® software showing the blocks used to design the crosstalk compensation circuits for the 2-bit selection scheme. The blocks (logic gates and flip-flops) are standard and were arbitrarily chosen from existing Quartus® libraries. Quartus II® is designed to optimize out any circuitry it deems redundant. Thus, when building delay lines, long strings of inverters were optimized out. To ‘trick’ the software into leaving the inverters in, a block called an ‘lcell’ is used. By placing an *lcell* on either side of each inverter, an asynchronous delay line could be created to generate the desired effect.

In order to get the simulation to work at reasonably fast speeds (nanoseconds) the Stratix® chip set was chosen in the simulator which is the fastest Altera® FPGA devices currently available. The chip number for these simulations is EP1S10F780C6 which proved to be fast enough to generate the correct output at the chosen 50 ns bit period.

Designing the length of the delay lines required some trial and error to get proper timing. The criteria was to get the pulses to occur at different times within the bit period so each pulse could change the state of the tri-state circuit before the next pulse arrived. In the interest of space and detail, the delay lines in the following figures are much shorter than those used in the actual simulator. The simulations contained 95 and 96 *lcell-inverter-lcell* units for Figure E.1 and Figure E.2, respectively, to ensure the system works properly. Each unit averaged ~700 ps of delay and the complete lines had between 64 and 67 ns of delay. The circuit in Figure E.1 needed an odd number of inverters since its output was normally high, whereas the circuit in Figure E.2 needed an even number of inverters.

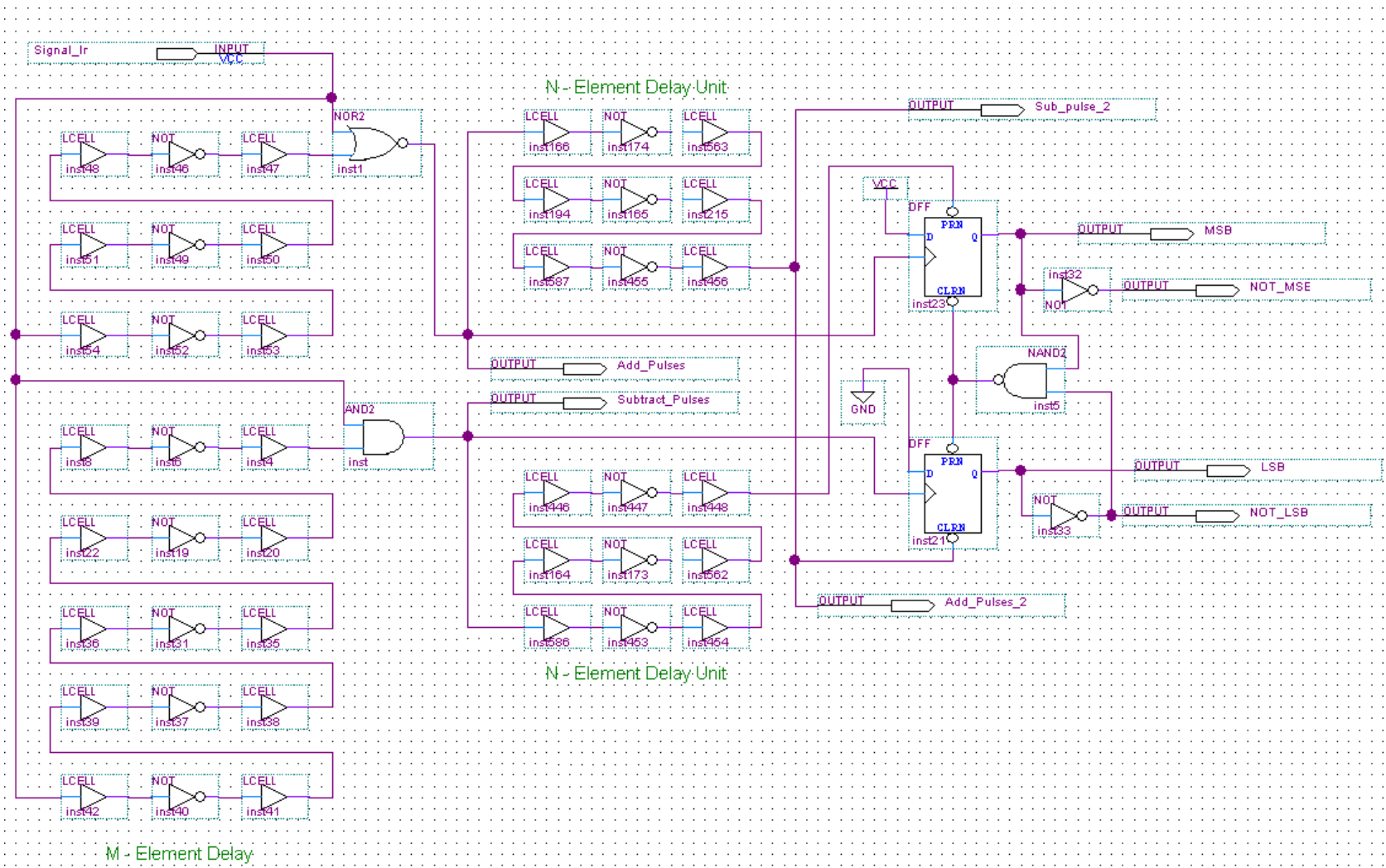


Figure E.1 Crosstalk compensation simulation for functionality and timing (I).

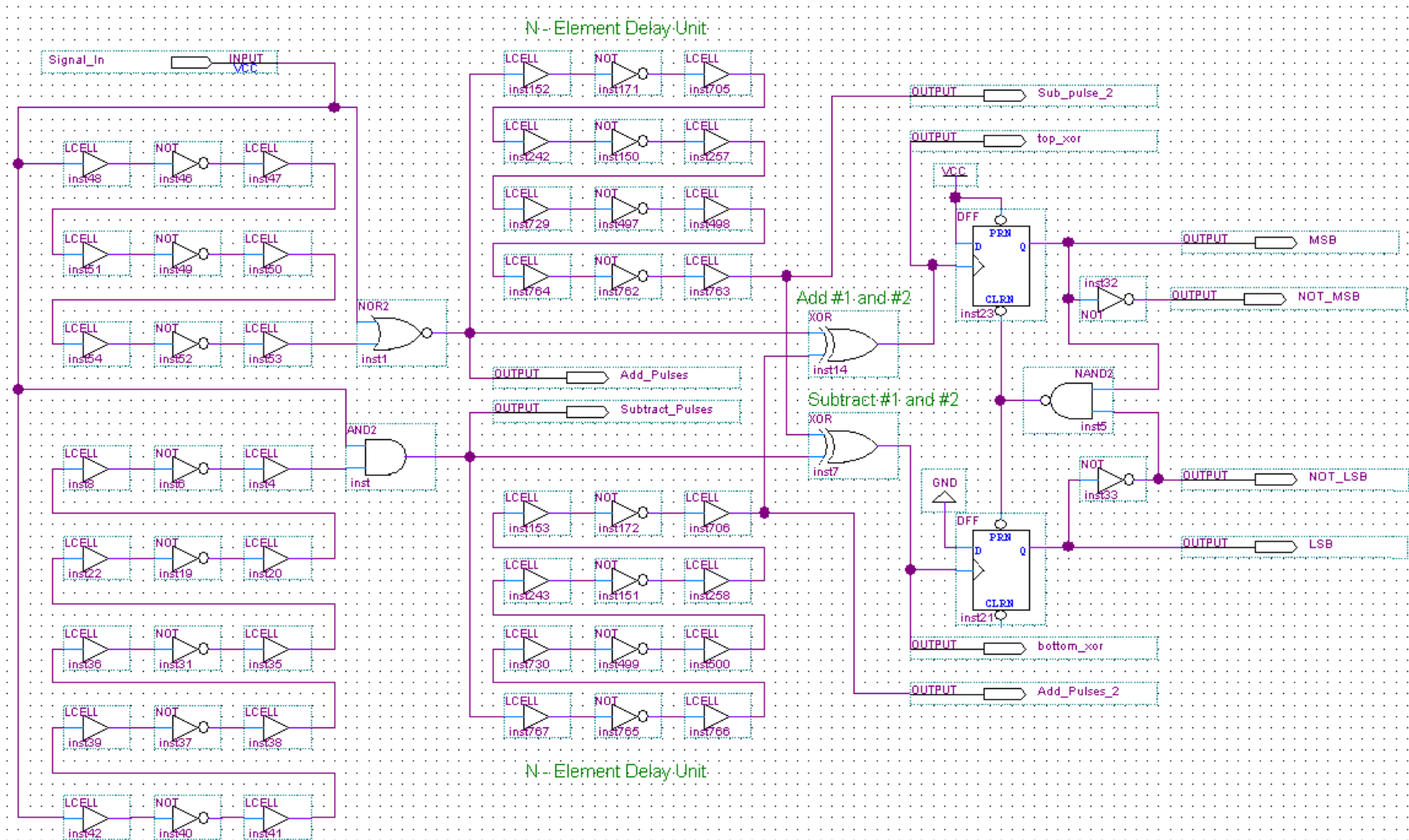


Figure E.2 Crosstalk compensation simulation for functionality and timing (II).