

CROSSTALK-AWARE TRANSMITTER PULSE-SHAPING FOR PARALLEL
CHIP-TO-CHIP LINKS

by

Mike Bichan

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

© Copyright by Mike Bichan 2006

CROSSTALK-AWARE TRANSMITTER PULSE-SHAPING FOR PARALLEL CHIP-TO-CHIP LINKS

Mike Bichan

Master of Applied Science, 2006
Graduate Department of Electrical and Computer Engineering
University of Toronto

Abstract

This thesis examines the difficulties involved in transmitting data over chip-to-chip links. Links for which crosstalk from adjacent parallel channels dominates the receiver noise are given particular attention. The idea of using a crosstalk-aware pulse shape to mitigate crosstalk is explored. This method stands in contrast to the traditional method of explicit crosstalk cancellation in which each parallel transmitter takes the bit streams of its two neighbours as input in order to cancel the crosstalk from those bit streams. Some measurements are performed on a board-to-board channel which are then used to find the optimal transmitter pulse shape for that channel. Finally, a 5-Gb/s chip-to-chip transmitter is designed in 0.13- μm CMOS based on information from the simulations and measurements performed.

Acknowledgments

Rationality is the recognition of the fact that nothing can alter the truth and nothing can take precedence over the act of perceiving it.

Ayn Rand, *Atlas Shrugged*

I am grateful to my supervisor, Professor Tony Chan Carusone. This thesis would not have been possible without his guidance. His enthusiasm is contagious and makes it easy to stay motivated.

I would also like to thank Professor Johns, Professor Gulak, and Professor Yu for serving on my thesis defense committee and providing valuable feedback that helped improve this thesis.

Many thanks go to CMC, MOSIS, and IBM for allowing me access to the 0.13- μm CMOS design kit and especially CMC for allocating me area on a fabrication run.

Thank you to my fellow graduate students in BA5158 who were always ready and willing to share their knowledge and expertise, and especially Raf Karakiewicz for many fruitful discussions.

I am indebted to my parents for their constant love and support throughout my life. I would not be where I am today without them.

Finally, I would like to thank Danielle Tchao for being a wonderful companion on the journey.

Contents

List of Figures	vi
List of Tables	ix
1 Introduction	1
1.1 Motivation	1
1.2 Background Information	2
1.3 The State of the Art	4
1.3.1 Chip-to-Chip Transceivers	4
1.3.2 Fractionally-Spaced Equalizers	6
1.3.3 Filter Tap Weight Selection	7
1.3.4 Insights from DSL	8
1.4 Thesis Organization	9
2 Chip-to-Chip Channel Impairments	10
2.1 Introduction	10
2.2 Chip-to-Chip Channel Modelling	12
2.3 Summary	15
3 Optimal Pulse Shape	17
3.1 Introduction	17
3.2 Pulse Shape Search Methodology	18
3.2.1 Figure of Merit	22
3.2.2 Searching the Space of Candidate Pulse Shapes	23
3.3 Results of the Exhaustive Search	23
3.3.1 PCB Channel with No Crosstalk	23
3.3.2 PCB Channel with Crosstalk	24
3.4 Guideline for Equalizer Specification: Optimal Pulse Shapes above 2.7 Gb/s	28
3.5 Summary	28

4	Measurement Results	31
4.1	Introduction	31
4.2	Time Domain Reflectometry	31
4.3	Output Eye Diagrams	32
4.4	Bit Error Rate Testing	36
4.5	Summary	38
5	Transmitter Design	40
5.1	Introduction	40
5.2	Transmitter Architecture	40
	5.2.1 Output Driver	41
	5.2.2 Delay Cell	43
5.3	Simulation Results	52
5.4	Test Chip Results	57
5.5	Summary	57
6	Conclusion	61
6.1	Conclusion	61
6.2	Suggestions for Future Work	62
	Appendix A Aggregate Data Rate Derivations	64
	Appendix B Simulation Data	65
	References	70

List of Figures

2.1	General chip-to-chip channel.	11
2.2	(a) Physical configuration and (b) frequency response of a simple PCB channel and (c) a more complex PCB channel.	11
2.3	Two parallel microstrip lines.	13
2.4	Through and crosstalk responses of the channel.	14
2.5	Effect of L variation on f_{3dB}	14
2.6	Effect of s variation on f_{3dB}	15
2.7	Effect of L variation on $ G(f) _{max}$	16
2.8	Effect of s variation on $ G(f) _{max}$	16
3.1	Measured step response of the board-to-board channel shown in Figure 3.2.	19
3.2	Board-to-board communication link.	20
3.3	Close-up of six adjacent chip-to-chip links.	20
3.4	Contour plot of simulated crosstalk-free eye opening.	25
3.5	Contour plot of simulated E2C against number of taps and taps per UI.	26
3.6	Comparison of optimal and regular NRZ pulses.	26
3.7	Simulated E2C vs. $Taps_{total}$ for a filter with $Taps_{perUI} = 2$	27
3.8	Plot of simulated E2C showing effect of time granularity.	27
3.9	Contour plot of E2C for a data rate of 5Gb/s.	28
3.10	Plot of simulated E2C showing effect of time granularity at 5Gb/s.	29
3.11	Contour plot of E2C for a data rate of 7.5Gb/s.	29
3.12	Plot of simulated E2C showing effect of time granularity at 7.5Gb/s.	30
4.1	Equalizer proof-of-concept test setup.	32
4.2	Impulse response of the channel, computed from the step response (time step = 10 ps).	33
4.3	Frequency response of the through channel (\odot), first crosstalk channel (\triangle), and second crosstalk channel (\square) computed from the step response (time step = 50 ps).	33
4.4	Measured eye diagram at 2.7 Gb/s with a PRBS sequence of length $2^{31} - 1$. This figure shows the channel input with no aggressors.	34

List of Figures

4.5	Measured eye diagram at 2.7 Gb/s with a PRBS sequence of length $2^{31} - 1$. This figure shows the channel output corresponding to Figure 4.4 with no aggressors.	35
4.6	Impact of adjacent aggressor signals on desired signal.	35
4.7	Measured eye diagram at 2.7 Gb/s with a PRBS sequence of length $2^{31} - 1$. This figure shows the output of the chip-to-chip channel for square pulse input with two aggressors.	36
4.8	Measured eye diagram at 2.7 Gb/s with a PRBS sequence of length $2^{31} - 1$. This figure shows the output of the chip-to-chip channel for crosstalk-aware pulse input with two aggressors.	37
4.9	Bathtub plot comparing crosstalk-aware and square pulses.	38
4.10	Bathtub plot comparing crosstalk-aware and pre-emphasis pulses. BER is higher than in Figure 4.9 because a smaller signal swing was used in this measurement for both pulse shapes.	39
5.1	Block diagram of the proposed transmitter.	41
5.2	Schematic diagram of the output driver cell. Gate lengths are $0.12 \mu\text{m}$. Gate widths of M_1 and M_2 are $80 \mu\text{m}$	42
5.3	Schematic diagram of the crossbar switch. Gate lengths are $0.12 \mu\text{m}$ and gate widths are $10 \mu\text{m}$. The pass transistors M_3 – M_6 have $R_{on} = 285 \Omega$	43
5.4	Schematic diagram of the improved output driver cell.	44
5.5	Schematic diagram of the starved inverter delay cell. Gate lengths are $0.12 \mu\text{m}$	46
5.6	Schematic diagram of the common source delay cell. Gate lengths are $0.12 \mu\text{m}$ and gate widths are $10 \mu\text{m}$ unless otherwise stated.	47
5.7	(a) Single transistor load, (b) I-V curve comparison, (c) symmetric load.	47
5.8	Schematic diagram of the diode-connected delay cell. Gate lengths are $0.12 \mu\text{m}$ and gate widths are $10 \mu\text{m}$ unless otherwise stated.	48
5.9	Schematic diagram of the self-biased symmetric-load delay cell. Gate lengths are $0.12 \mu\text{m}$ and gate widths are $10 \mu\text{m}$ unless otherwise stated.	49
5.10	Schematic diagram of the low voltage delay cell. Gate lengths are $0.12 \mu\text{m}$ and gate widths are $10 \mu\text{m}$	49
5.11	Schematic diagram of the low voltage delay cell with cross-coupled inverters. Gate lengths are $0.12 \mu\text{m}$ and gate widths are $10 \mu\text{m}$	50
5.12	Delay cell comparison. Note: simulations conducted at 1 Gb/s. Power dissipation of the starved inverter cell would be higher at the targeted bit rate of 5 Gb/s.	51
5.13	Supply voltage scaling: (a) low voltage cell, (b) self-biased, symmetric-load cell.	52

List of Figures

5.14	Spectre simulation of (a) regular NRZ and (b) pre-emphasis pulse shapes at 5 Gb/s with a PRBS sequence of length $2^7 - 1$. The corresponding signal and crosstalk outputs are shown in (c)-(f) for the chip-to-chip channel. $Taps_{perUI} = 3$	54
5.15	Spectre simulation of (a) slew-rate limited and (b) square pulse shapes at 5 Gb/s with a PRBS sequence of length $2^7 - 1$. The corresponding signal and crosstalk outputs are shown in (c)-(f) for the chip-to-chip channel. $Taps_{perUI} = 3$	55
5.16	Spectre simulation of (a) the optimal pulse shape at 5 Gb/s with a PRBS sequence of length $2^7 - 1$. Also shown are (b) the corresponding signal output and (c) the crosstalk output for the chip-to-chip channel. $Taps_{perUI} = 3$	56
5.17	Spectre simulation at 5 Gb/s with a PRBS sequence of length $2^7 - 1$. When longer delays are used the bandwidth of the delay cell decreases, increasing jitter in the delay cells farther down the chain.	58
5.18	Photomicrograph of the test chip in 0.13- μ m CMOS. The die dimensions are 1.5 mm \times 1.5 mm.	59

List of Tables

1.1	Comparison of several signalling schemes.	3
1.2	State of the art chip-to-chip communication circuits.	5
1.3	Circuits using fractionally-spaced equalizers.	7
5.1	Summary of delay cell characteristics.	51
5.2	Simulated transmitter characteristics.	60
B.1	Optimization data for a chip-to-chip link with no crosstalk	66
B.2	Optimization data for a chip-to-chip link including crosstalk	67
B.3	Optimization data for a chip-to-chip link with crosstalk at 5Gb/s	68
B.4	Optimization data for a chip-to-chip link with crosstalk at 7.5Gb/s	69

List of Acronyms

2-PAM two-level pulse amplitude modulation

4-PAM four-level pulse amplitude modulation

BER bit error rate

DAC digital-to-analog converter

DLL delay-locked loop

DSL digital subscriber line

DUT device under test

DSM dynamic spectrum management

E2C eye-to-crosstalk ratio

FEXT far-end crosstalk

FIR finite impulse response

IC integrated circuit

I/O input/output

ISI inter-symbol interference

MIMO multiple-input multiple-output

NEXT near-end crosstalk

NRZ nonreturn-to-zero

ParBERT parallel bit error ratio tester

PCB printed circuit board

List of Acronyms

PD phase detector

PSD power spectral density

PRBS pseudo-random bit stream

$Taps_{total}$ total number of taps

$Taps_{perUI}$ taps per UI

UI unit interval

1 Introduction

1.1 Motivation

CROSSTALK between adjacent channels is a severe problem in chip-to-chip communication links. It exists as a result of parasitic capacitance and inductance on printed circuit boards and it is a barrier preventing bit rates for parallel chip-to-chip links from increasing past 5 Gb/s/pin. Even more dramatic are the effects of crosstalk on board-to-board channels and multidrop busses. To extend the useful bandwidth of these channels, it is possible to use a transmitted pulse shape that minimizes crosstalk while also equalizing inter-symbol interference (ISI).

The desire for higher chip-to-chip bit rates stems from the computer industry. For most of the history of the computer, system performance has been limited by the maximum clock frequency of the CPU. In recent years, improvements in integrated circuit (IC) fabrication technology have led to computer chips running at speeds approaching 4 GHz. This frequency is approximately equal to the bandwidth of a typical chip-to-chip channel on a printed circuit board (PCB). Now an important performance-limiting factor is the speed at which data can be sent between different chips in the same system.

As chip speeds increased over the past two decades, the aggregate chip-to-chip bit rate was typically increased by increasing the number of input/output (I/O) pins devoted to high-speed off-chip communication and scaling the bit rate of each pin in step with the clock speed, as shown in (1.1).

$$\text{aggregate bit rate} = \left(\frac{\text{bit rate}}{\text{channel}} \right) \times \left(\frac{\text{channels}}{\text{chip}} \right) \quad (1.1)$$

Each channel uses one or two package pins. The number of pins on a typical package, however, has not kept pace with the growth in chip speed. And as mentioned above, the per pin bit rate can no longer easily scale higher as it is already bumping into the limited bandwidth of the channel. To make matters worse, the trend in computer architecture is towards multiple CPU cores per chip, which require a higher system bandwidth to keep them supplied with data. This disparity shifts the focus to improving the performance of the I/O transceiver so that it can function at higher speeds in the hostile chip-to-chip environment.

1.2 Background Information

Chip-to-chip signaling presents the eager circuit designer with numerous options to accomplish the task of moving data from point A to point B. Without considering specific channel characteristics for the moment, the following options are possible at the *system level*:

- single-ended vs. differential
- serial vs. parallel
- unidirectional vs. bidirectional
- two-level pulse amplitude modulation (2-PAM) vs. four-level pulse amplitude modulation (4-PAM)

We want to be able to compare these different techniques on an equal footing from a system-level perspective. For example, although a differential signalling scheme can usually transmit faster than a single-ended scheme, it also uses twice as many pins. To achieve the same aggregate rate as a single-ended scheme, the differential scheme must transmit twice as fast. Table 1.1 shows the bit rate that each scheme requires to achieve an aggregate bit rate equivalent to a differential, unidirectional, 2-PAM system. A brief derivation of these values can be found in Appendix A.

To find the optimal signalling scheme, we need to find the maximum achievable bit rate for each of these approaches and compare them according to Table 1.1. This table is simplified. With single-ended signalling the issue of a signal reference must be considered, which will increase the value of 0.5 in Table 1.1 somewhat. In order to find

Signalling Scheme	Required Symbol Rate [†]
Differential, unidirectional, 2-PAM	1
<i>Single-ended</i> , unidirectional, 2-PAM	0.5
Differential, <i>bidirectional</i> , 2-PAM	0.5
Differential, unidirectional, <i>4-PAM</i>	0.5

[†]Relative to differential, unidirectional, 2-PAM scheme

Table 1.1: Comparison of several signalling schemes.

the maximum achievable bit rates we must consider the various channel impairments. When ISI must be dealt with, there are several ways to design the required equalizer at the *circuit level*:

- transmit-side vs. receive-side
- continuous time vs. discrete time
- baud-spaced vs. fractionally-spaced
- analog vs. digital

On some particularly inhospitable channels, parasitic capacitance and inductance lead to crosstalk between adjacent channels. Adding crosstalk to the mix forces us to choose one of the following methods:

- tolerance (i.e. no method)
- simple slew rate limiting
- crosstalk cancellation
- pulse shaping

Each channel has unique characteristics that will require a particular combination of the methods outlined above. Each situation will also have certain power, area, packaging, and bit rate criteria that will further define the requirements. It is up to the circuit designer to strike the right balance in order to produce the most effective circuit.

This thesis aims to maximize the aggregate bit rate of a parallel, unidirectional, 2-PAM link using crosstalk-aware transmit-side equalization.

1.3 The State of the Art

1.3.1 Chip-to-Chip Transceivers

Following the wealth of chip-to-chip transceiver techniques presented in Section 1.2, recently reported circuits can be similarly classified. For a summary of recent circuits, see Table 1.2.

The chip-to-chip circuits that achieve the highest per-channel bit rate fall into the category of unidirectional, differential, serial transceivers. This category of transceivers suffers from the least amount of unwanted noise. There are no adjacent channels injecting noise onto the desired channel, and any noise that affects both channels equally is partially cancelled by the differential nature of the link [17]. In addition, data is sent in only one direction so there is no echo signal to cancel. The fastest of this type of circuit currently achieves a bit rate of 20 Gb/s over short backplane and coaxial cable channels [1, 2]. These circuits have large overhead in terms of power and area, with on-chip inductors being often used.

When longer backplane and coaxial cable channels are considered, the maximum achievable bit rate drops to 10 Gb/s [3, 4]. Longer channels have more severe ISI and reflections that limit the speed. Again, this high-speed signaling requires a disproportionate amount of power in order to function. To be suitable as an I/O cell on a large chip, a chip-to-chip transceiver must be power- and area-efficient. Transceivers optimized for power are able to achieve 8 Gb/s [5, 6].

Large digital systems often require a large number of I/O cells transmitting and receiving data in parallel. This requirement puts additional strain on the chip-to-chip transceiver because of the interference caused by adjacent channels. The highest reported bit rate for a transceiver with multiple parallel channels is 6.4 Gb/s/channel [8, 9]. Several other parallel chip-to-chip transceivers have also been reported [10, 11].

Once a designer forgoes the use of differential signaling, the challenge becomes still more difficult. Especially in parallel links there is a lot of noise that is common to both inputs in a differential link, and so can be cancelled out. In a single-ended system that noise is felt at the receiver as a reduced eye opening. Another problem with single-ended signaling is the need for a reference to compare to the received signal.

Reference	Bit Rate (Gb/s)	Channel	single-ended	differential	serial	parallel	unidirectional	bidirectional	No. of PAM levels	Technology	Year
[1]	20	16" backplane	X	X		X		4	0.18 μm		2005
[2]	20	1m coaxial cable	X	X		X		2	0.13 μm		2005
[3]	10	30" backplane	X	X		X		2	0.13 μm		2005
[4]	10	16.3m AWG28	X	X		X		2	0.11 μm		2005
[5]	8	50 cm	X	X		X		2	0.13 μm		2003
[6]	8	5 cm FR4	X	X		X		2	0.13 μm		2005
[7]	5	26" FR4	X	X		X		2/4	0.13 μm		2005
[8]	6.4	none mentioned	X			X	X	2	90 nm SOI		2005
[9]	6.4	18 cm FR4	X			X	X	2	0.11 μm		2005
[10]	3.2	40" backplane	X			X	X	2	0.13 μm		2003
[11]	3.2	none mentioned	X			X	X	2	0.16 μm		2002
[12]	3	multidrop bus	X			X	X	2	0.25 μm		2005
[13]	3.6	8 cm FR4 + coax	X		X	X		2	0.18 μm		2004
[14]	8	4.6" FR4	X		X		X	2	0.35 μm		2004
[15]	4	multidrop bus	X		X		X	4	0.10 μm		2005
[16]	6.4	18 cm FR4	X	X			X	2	0.18 μm		2003

Table 1.2: State of the art chip-to-chip communication circuits.

The reference should track the same noise sources that are seen on the signal. This noise can come from the power supply and ground, and is injected on-chip, in the pad frame, in the package, and on the board. Using more pins for the reference enhances its noise-cancellation ability. The best reference would include one reference pin for every signal pin, but this uses the same total number of pins as differential signaling with half the signal swing. Using fewer reference pins increases the received noise. For single-ended, parallel links the maximum reported bit rate is 3 Gb/s [12]. The highest reported speed for a single-ended, serial transceiver is 3.6 Gb/s [13].

Finally, links that attempt to signal in both directions at once on the same line face a still bigger challenge. However, in this case the compensation is that twice as much data can be sent over the same pins. For bidirectional signaling schemes, the highest bit rates achieved have been 8 Gb/s and 6.4 Gb/s in [14] and [16], respectively. The circuit described in [15] combines 4-PAM and single-ended signalling with bidirectional operation, and achieves a bit rate of 5 Gb/s/pin.

1.3.2 Fractionally-Spaced Equalizers

While each of the above transceivers must operate within the constraints of the given channel, the equalizers used can be designed independently of the system level configuration. It turns out that most high-speed chip-to-chip transceivers use a baud-spaced equalizer. This type of equalizer is easier to implement at high speeds because the delay between taps need only be as low as one unit interval (UI). One UI is equal to one symbol period. For example, a 1-Gb/s 2-PAM signal has a UI of 1 ns. Fractionally-spaced equalizers become more difficult to implement with every added filter tap per UI.

This difficulty has not completely prevented the fastest circuits from making use of this technique, as seen in [1] and [18] which use four-tap equalizers with a tap spacing of $1/3$ UI. These circuits use a tapped LC ladder to implement the delay line. While LC ladder-based delay lines can achieve the smallest delay times, there is also a large overhead in terms of area. The circuit in [19] operates at 10 Gb/s but uses 87 spiral inductors.

A six-tap, $T/8$ -spaced equalizer running at a bit rate of 1 Gb/s is presented in [20].

Reference	Bit Rate (Gb/s)	Channel	No. of Taps	Taps per Bit	Delay Cell	No. of PAM levels	Technology	Year
[1]	20	16" backplane	7+4	3	LC ladder	4	0.18 μm	2005
[18]	10	26" backplane	4	3	LC ladder	2	0.18 μm	2004
[21]	2.5	120" FR4	4	4	active-inductor load	2	0.25 μm	2005
[20]	1	220" FR4	6	8	common-source amplifier	2	0.25 μm	2005
[19]	10	600 m MMF	7	$1\frac{1}{3}$	LC ladder	2	0.13 μm	2005

Table 1.3: Circuits using fractionally-spaced equalizers.

A four-tap $T/4$ -spaced equalizer at 2.5 Gb/s is presented in [21]. These circuits use active elements instead of inductors in the delay cells. These occupy less area than the inductor-based delay lines, but consume more power in general. All of these fractionally-spaced equalizer circuits are summarized in Table 1.3.

1.3.3 Filter Tap Weight Selection

Determining the optimal filter tap weights is crucial if the finite impulse response (FIR) filter is to properly equalize the channel. In general, tap weights are chosen to cancel ISI introduced by the channel. In [1], tap weights are derived from measured impulse response data. Simulations are then performed with different filter configurations (total number of taps and taps per bit) to find the best one. Crosstalk is not taken into account because the circuit is operating as a serial transceiver. Tap weights for the other serial transceivers in Table 1.2 are chosen in a similar manner.

Even the parallel transceivers surveyed do not pay attention to crosstalk when choosing tap weights. In [9], “the pre-emphasis weights are adjusted manually” but no mention is made of crosstalk. The circuit in [10] uses one-tap pre-emphasis where the maximum pre-emphasis is 15% of the normal square pulse amplitude. In [12], the ISI contribution of one post-cursor sample is measured and that value is used for the single post-cursor equalizer tap.

There is an opportunity here to make the filter tap weight selection “crosstalk-aware”. If the effect of one channel’s signal on the two adjacent channels is taken into account when choosing tap weights, there is the potential to minimize crosstalk noise without costly cancellation circuits. This concept is explored further in Chapter 3.

1.3.4 Insights from DSL

The same issues now being encountered in parallel chip-to-chip interconnect have already been seen in digital subscriber line (DSL) systems. In these systems, signals are sent on twisted-pair copper wires between the DSL customer’s modem and the central office. Because these copper wires are from the existing phone lines, multiple twisted pairs are bundled together in the same phone cable. This bundling results in crosstalk between the wires in the bundle.

The conventional way of solving the crosstalk problem is to enforce a power spectral density (PSD) mask on the outputs of both the central office and the modem [22]. This mask ensures that less power is transmitted in the frequency range that causes the most harm to adjacent wires. Although this solution is effective in reducing crosstalk, it is also overly conservative and results in a suboptimal bit rate. PSD masking in DSL systems can be compared to slew rate limiting in chip-to-chip signaling. Slew rate limiting is simple to implement and it reduces the high-frequency content of the nonreturn-to-zero (NRZ) pulse shape. It is also not the optimal solution.

In DSL systems, one way to improve upon the performance of a basic PSD mask is with the idea of dynamic spectrum management (DSM). DSM allows each transceiver to optimize its spectrum usage with respect to both its own channel and the power spectra of adjacent transceivers. In [23], a method is proposed that maximizes the aggregate bit rate of all wires in a bundle even when the transceivers associated with each wire cannot share information. This method of iterative water-filling simply has each transceiver putting more power into frequency bands where it sees a higher SNR, determined both by the channel and by crosstalk. After several iterations, each transceiver finds the most power-efficient way to transmit information.

Some gains are possible if the transceivers operate separately, but there is more scope for performance gain if the transceivers can cooperate. In this case, the bundle

of twisted pairs can be treated as a multiple-input multiple-output (MIMO) channel where the outputs of the subchannels are processed together in order to remove any mutual interference [23]. When the number of subchannels is large, however, this signal processing can become quite complicated. This technique would seem to be applicable to chip-to-chip signaling because all adjacent transceivers reside on the same chip, but at high bit rate the amount of signal processing required might be excessive.

While not all techniques used in DSL systems work well in the chip-to-chip environment, the idea of managing the spectrum usage of each channel to minimize its negative impact on other channels has merit. Chapter 3 will be devoted to finding a transmitted pulse shape that maximizes aggregate bit rate for a number of parallel channels.

1.4 Thesis Organization

Finite channel bandwidth is not the only obstacle to successful transmission of data across a chip-to-chip channel. In systems with many parallel high-speed channels, crosstalk between channels is also a problem. In addition, while it may begin and end at an IC, the channel also comprises bond wires, chip packages, PCB traces, and vias. This heterogeneity leads to impedance discontinuities along the length of the channel, which result in reflections at the receiver. Chapter 2 is a discussion of the effect of microstrip channel geometry on frequency response.

Chapter 3 presents a method to find the optimal pulse shape for a given chip-to-chip channel. Pulse shapes for a measured board-to-board channel are discussed. Chapter 4 presents the rest of the measurements performed on the board-to-board channel as well as a proof-of-concept of a crosstalk-aware transmitter. Conventionally, a one-tap pre-emphasis pulse is used to transmit data across chip-to-chip channels. These measurements show that using a crosstalk-aware pulse shape with the same peak swing reduces the bit error rate (BER) by a factor of 100.

A transmitter has been designed in order to apply the findings of Chapters 3 and 4. This transmitter consists of two main blocks—a delay cell and an output driver—and is discussed in Chapter 5.

2 Chip-to-Chip Channel Impairments

2.1 Introduction

THE design of a transmitter requires knowledge of the target channel. A general block diagram for a chip-to-chip channel is shown in Figure 2.1. The specification of the transmitter depends on the characteristics of the channel. In chip-to-chip communication this dependence is somewhat problematic because the set of all chip-to-chip channels includes a variety of physical configurations that have significantly different characteristics.

In a sample channel such as the one shown in Figure 2.2(a), the two chips are very close together on the same board and are connected with a serial link. This leads to the frequency response shown at the bottom of Figure 2.2(a) in which the gain of the channel decreases monotonically as frequency increases. In addition, the channel bandwidth decreases as the distance separating the two chips increases. Since high frequencies are attenuated most by this channel, to achieve the highest data rate with an open eye at the receiver we need to accentuate the high frequency content of the signal. This can be accomplished with an equalizer either at the receiver or the transmitter. In this thesis, this type of channel will be referred to as a *chip-to-chip channel*.

The more complex channel shown in Figure 2.2(b) is less hospitable because the two chips are on separate boards and are connected by a parallel link. This not only typically increases the distance between the chips, but also substantially increases the crosstalk among parallel paths when the signals travel through a board connector. This kind of physical configuration is common in server backplanes where many daughter-cards are connected to a single motherboard. It is clear from the frequency response

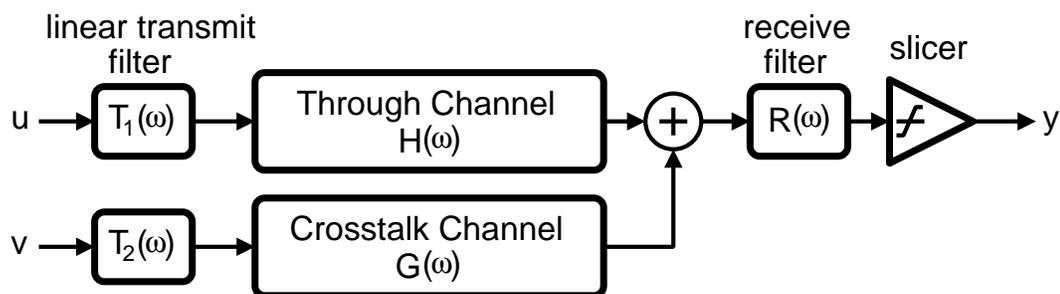


Figure 2.1: General chip-to-chip channel.

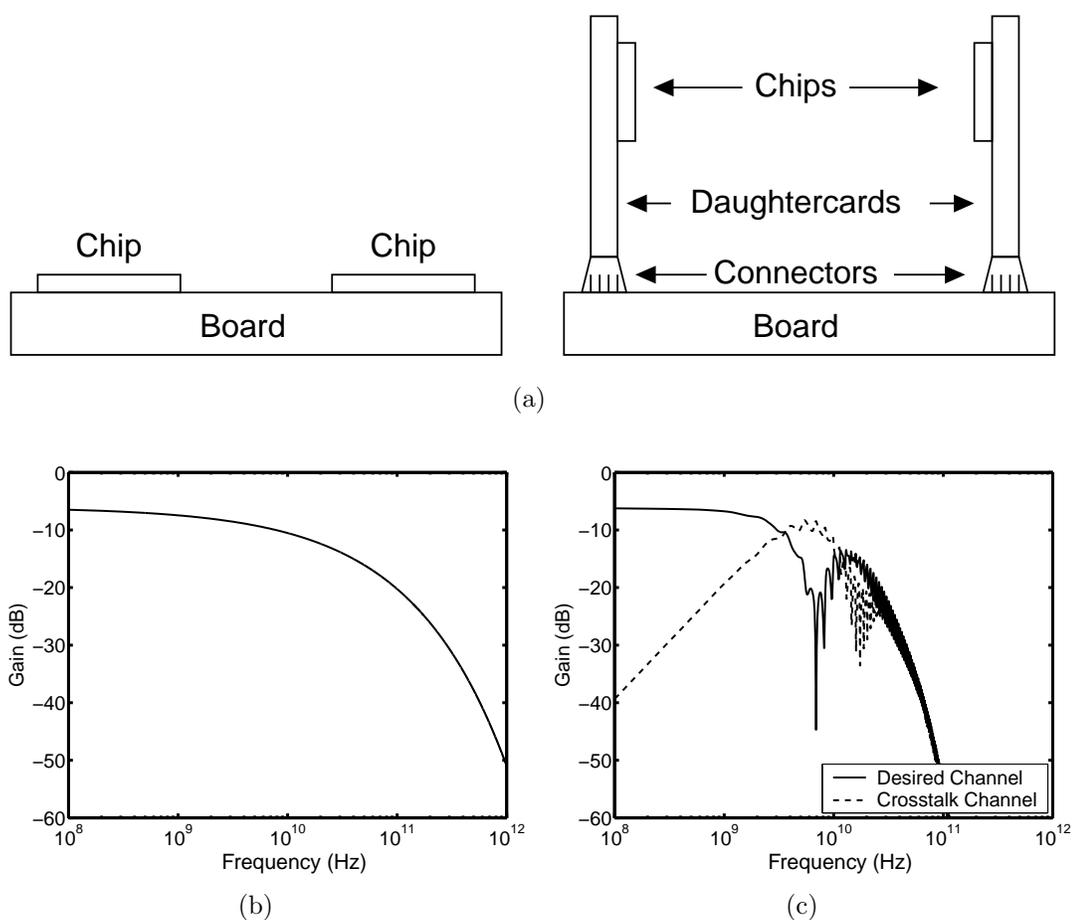


Figure 2.2: (a) Physical configuration and (b) frequency response of a simple PCB channel and (c) a more complex PCB channel.

shown in Figure 2.2(b) that there is a range of frequencies that contribute more to crosstalk on adjacent channels than they do to the signal on the desired channel. In order to use this channel most effectively, the transmitter needs to limit its output to frequencies outside this band. In this thesis, this type of channel will be referred to as a *board-to-board channel*.

Judging from the frequency responses presented in Figure 2.2, the requirements of the two channels are in conflict. One channel requires high-frequency content to be amplified and the other requires certain frequencies to be attenuated. Ideally the spectral content of the modulated output signal would be well suited to the frequency response of the channel.

Even superficially identical channels can have varying characteristics caused by the type of connectors and vias used in the signal path. These components can cause impedance discontinuities that produce reflections as well as influence crosstalk between parallel signal paths.

Not modelled here are the effects of impedance discontinuities along the channel. These can be caused by connectors and vias as well as chip packages and sockets.

2.2 Chip-to-Chip Channel Modelling

A diagram of a chip-to-chip channel consisting of two parallel microstrip lines is shown in Figure 2.3. In general there would be many lines in parallel. The geometry of the microstrip lines is determined by the variables h , t , ϵ_r , w , s , and L . This geometry influences the frequency response of the channel. Typically, however, h , t , and ϵ_r are fixed and the value of w is chosen in order to set the characteristic impedance of the line, Z_0 , equal to 50Ω . This leaves s , the trace separation, and L , the trace length, as the only parameters under the control of the board designer. While s and L are often strongly influenced by board size, it should be noted that they have a large impact on the frequency response of the channel.

An electromagnetic simulator¹ has been used to model the frequency response of the chip-to-chip channel. Figure 2.4 shows the *through response* and *crosstalk response*

¹Advanced Design System by Agilent Technologies

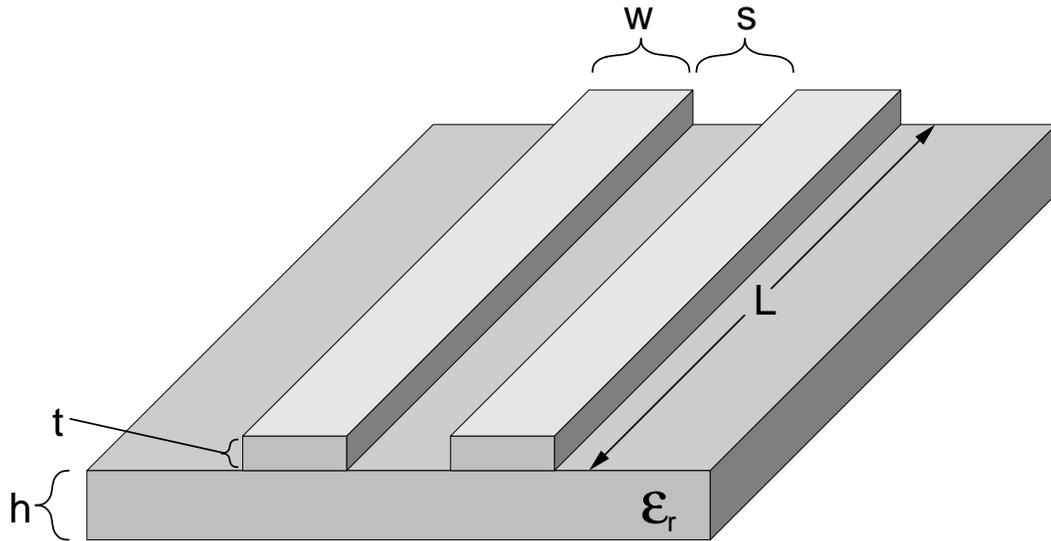


Figure 2.3: Two parallel microstrip lines.

for the channel in Figure 2.3 with $Z_0 = 50 \Omega$, $s = 2 \text{ mm}$, and $L = 30 \text{ cm}$. Notice that there is a range of frequencies for which the crosstalk response is greater than the through response. For a parallel link, minimizing the power transmitted in this band will increase the aggregate bit rate of the entire link.

The 3 dB frequency of the through response (f_{3dB}) and the maximum gain of the crosstalk response ($|G(f)|_{max}$) are marked on Figure 2.4. To gain an intuition about the effect of microstrip geometry on the frequency response, f_{3dB} and $|G(f)|_{max}$ are plotted as s and L change. Figures 2.5 and 2.6 show how the 3 dB frequency of the channel changes as s and L are varied. Figure 2.5 is unsurprising; attenuation increases as trace length increases. However, Figure 2.6 shows that the mere presence of another trace worsens the frequency response of the desired channel. This does not include the additional negative effect of crosstalk noise from the adjacent channel.

Figures 2.7 and 2.8 show how the crosstalk response changes with s and L . Both figures plot the maximum crosstalk gain as well as the frequency at which that gain occurs. Figure 2.7 shows that increasing the trace length has little effect on the maximum crosstalk gain, except to move the maximum to lower frequencies. In Figure 2.8, we see that crosstalk increases exponentially as trace separation decreases. At the same

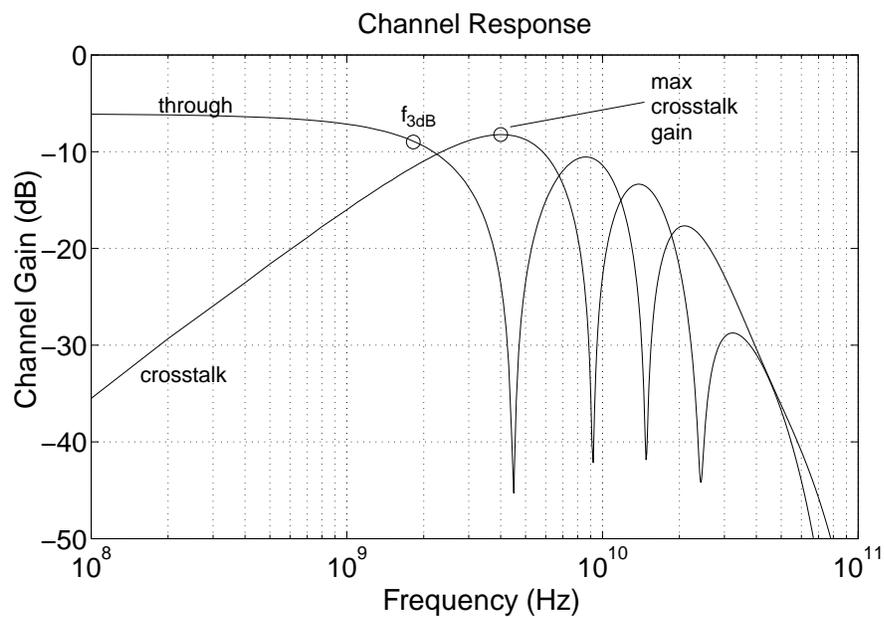


Figure 2.4: Through and crosstalk responses of the channel.

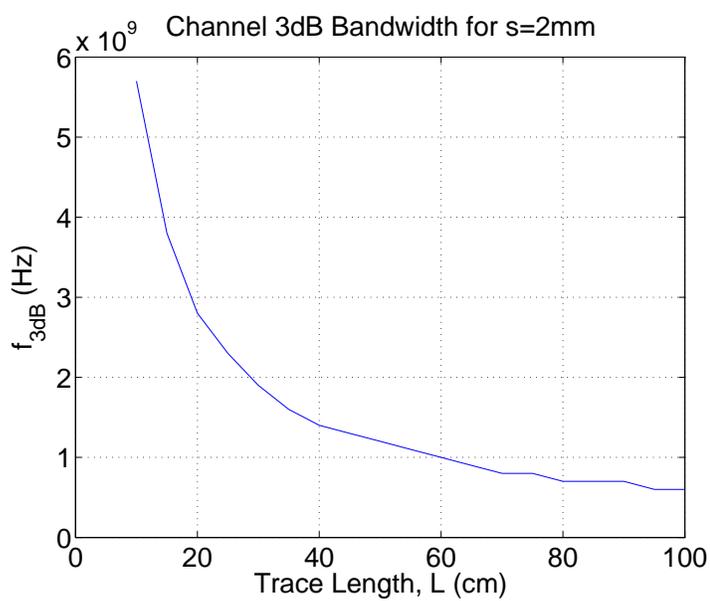


Figure 2.5: Effect of L variation on f_{3dB} .

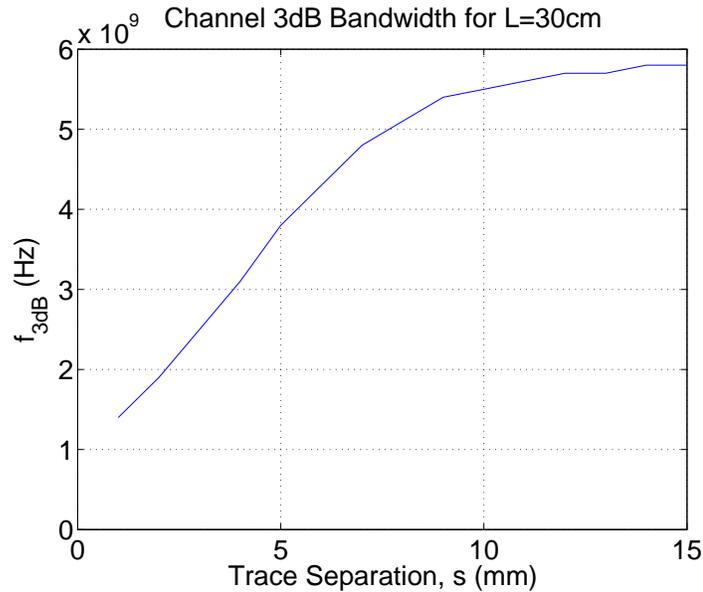


Figure 2.6: Effect of s variation on f_{3dB} .

time, the frequency of that maximum also decreases.

These simulation results present a conflict. Clearly, large separation between traces is desirable. However, in a large PCB with many components, increasing the trace separation may mean having to increase the size of the entire board. Consequently, trace length must increase and that decreases performance. There is undoubtedly an optimal tradeoff to be made, but in any case the negative effects of interference from adjacent signal lines will have to be alleviated.

2.3 Summary

This chapter has presented simulation results showing the effect of geometry on the frequency response of two coupled microstrip lines. These simulations have shown that both decreasing trace separation and increasing trace length lead to a degraded channel frequency response. However, in a practical circuit board design s and L cannot both be minimized at the same time. Therefore, a way must be found to deal with the problem of crosstalk from adjacent lines.

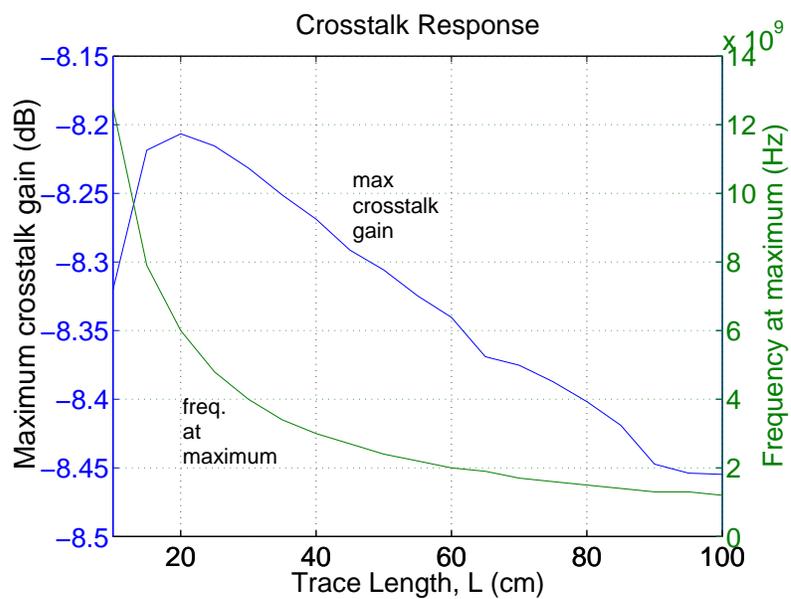


Figure 2.7: Effect of L variation on $|G(f)|_{max}$.

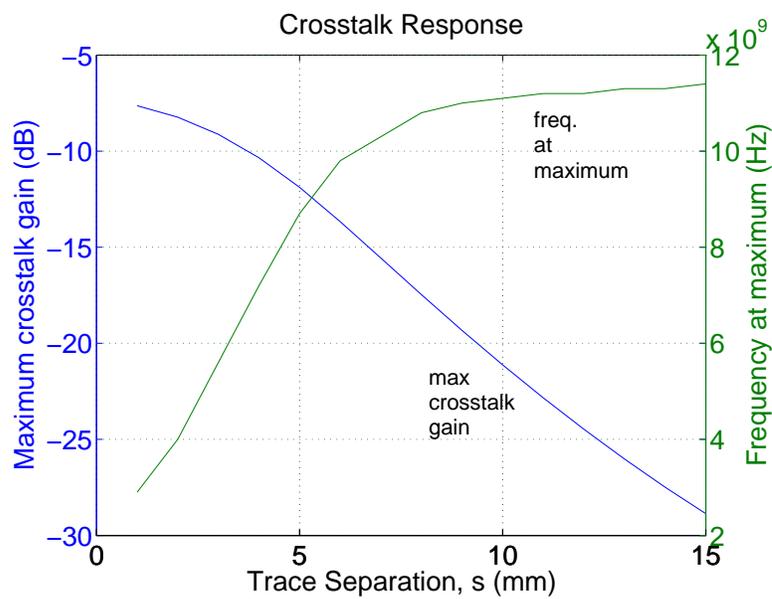


Figure 2.8: Effect of s variation on $|G(f)|_{max}$.

3 Optimal Pulse Shape

3.1 Introduction

WHEN designing an equalizer for a band-limited channel, the parameters of the equalizer are chosen such that a cascade of the channel and the equalizer produces a flat frequency response up to some desired frequency. In this way, more equalization effort can increase the bandwidth of the channel, subject to some restrictions. Transmit-side equalization is limited by the maximum signal level at the output, a limit imposed by the supply voltage and the characteristics of the fabrication technology. Receive-side equalization is limited by the fact that Gaussian channel noise is amplified along with the signal. Since mere amplification cannot improve signal-to-noise ratio at a specific frequency, the existence of random noise at the input of the receiver limits the effectiveness of receive-side equalization.

For a low-pass channel, in which attenuation increases at higher frequencies, this procedure results in an equalizer that amplifies the high-frequency content of the signal. For a transmit-side equalizer, high-frequency boost manifests itself in the time-domain as pre-emphasis. This result is optimal when the channel under consideration is a serial link or when adjacent signal lines are far enough removed from one another so as to cause no interference. However, in the interest of minimizing board area and of maximizing the number of I/O ports per chip, often signal lines end up being packed closely together. This close proximity of signal lines introduces another factor into equalizer design, namely crosstalk.

In describing crosstalk, we can talk about the *crosstalk channel* that results when a signal from one transmitter is sensed at the receiver on an adjacent channel. We differentiate between near-end crosstalk (NEXT), in which this transmitter-receiver

pair is on the same chip and far-end crosstalk (FEXT), in which the transmitter and receiver are on different chips.

When crosstalk becomes significant, the serial link solution of simply amplifying high frequency signal content is no longer optimal. The crosstalk channel has maximum gain at the same frequencies that need to be amplified for the optimal serial link equalizer. While this equalizer will reduce the amount of eye closure due to ISI, the increase in crosstalk will still result in a closed eye.

In order to find the optimal equalizer for many chip-to-chip channels, both ISI and crosstalk must be taken into account. This chapter discusses a method to pre-determine the optimal transmitter pulse shape for a channel with significant crosstalk. An ideal FIR filter with programmable tap weights and tap spacing is assumed. Sections 3.2 and 3.3 consider the optimal pulse shape when the data rate is 2.7 Gb/s. This data rate was chosen because it is the maximum data rate of the test equipment used in the equalizer proof-of-concept. However, this optimization can be performed for any desired data rate. To know if the given pulse shape search methodology is reasonable it must be compared to measured results. In Section 3.4, optimal pulse shapes are found for higher data rates in order to inform the choice of equalizer specifications in Chapter 5.

3.2 Pulse Shape Search Methodology

We start with the step response of the channel under consideration. This includes the step response of the crosstalk channel and perhaps also the second crosstalk channel. These step responses can come from an electromagnetic simulation of the channel, but the results are more likely to be useful if a measured channel response is used. Such a measured channel response is shown in Figure 3.1. This is the measured step response of the desired channel and two adjacent crosstalk channels for the board-to-board setup shown in Figures 3.2 and 3.3.

Given this channel response, we then compute a figure of merit for each of the candidate filter responses. For each value of total number of taps ($Taps_{total}$) and taps per UI ($Taps_{perUI}$) there is an optimal filter pulse shape. The number of candidate

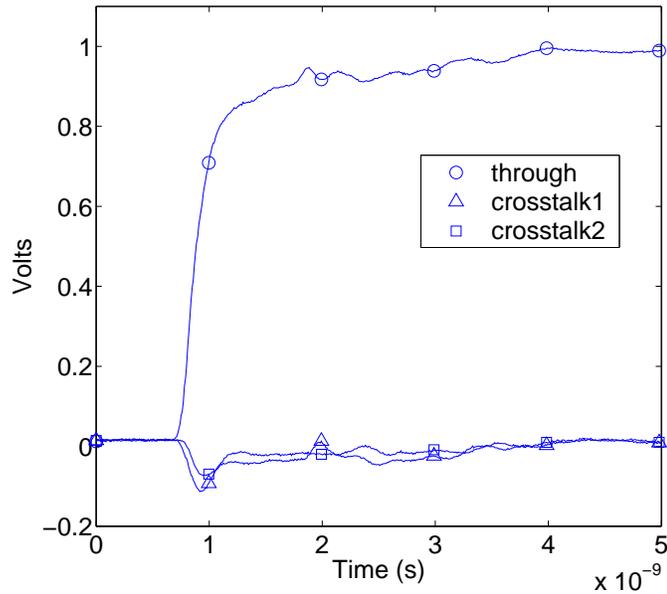


Figure 3.1: Measured step response of the board-to-board channel shown in Figure 3.2.

pulse shapes is determined by several parameters: maximum number of taps ($Taps_{max}$), maximum number of taps per UI ($Taps_{perUI,max}$), and the resolution (in bits), N , of the tap weights. For a given $Taps_{total}$ and $Taps_{perUI}$, we have:

$$\text{Number of Candidates} = (2^N)^{Taps_{total}} \quad (3.1)$$

If, for a given $Taps_{max}$ and $Taps_{perUI,max}$ we consider every candidate for which:

$$Taps_{total} \leq Taps_{max} \quad (3.2)$$

$$Taps_{perUI} \leq Taps_{perUI,max} \quad (3.3)$$

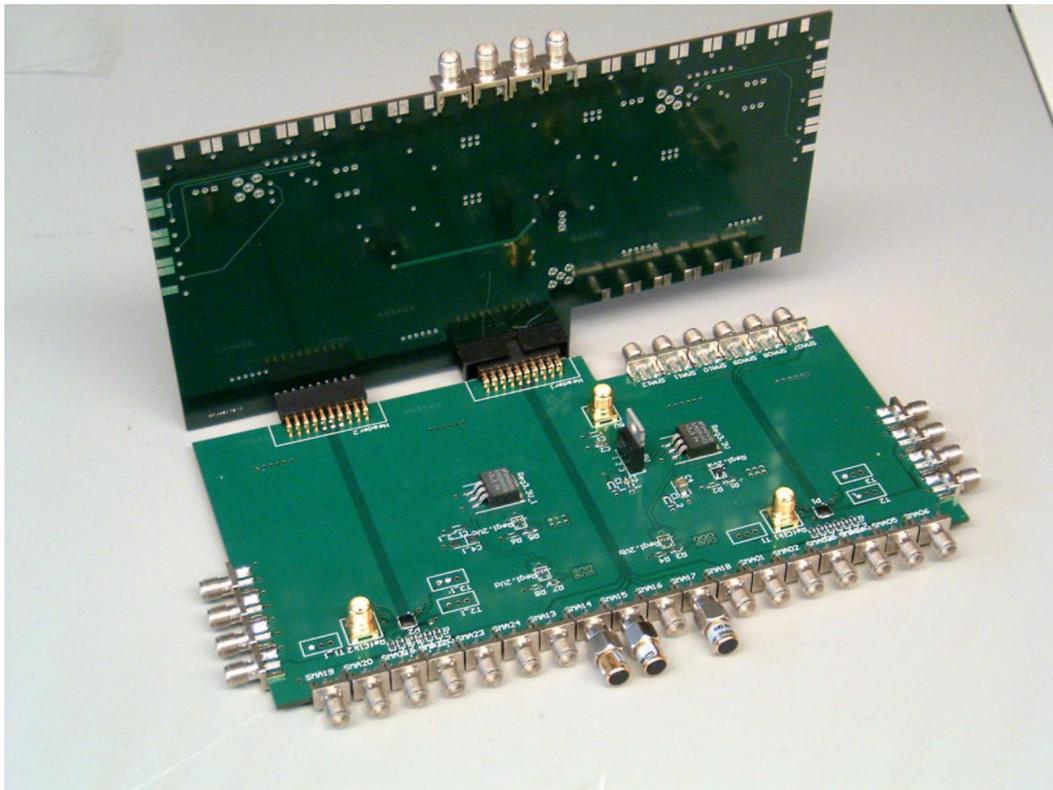


Figure 3.2: Board-to-board communication link.

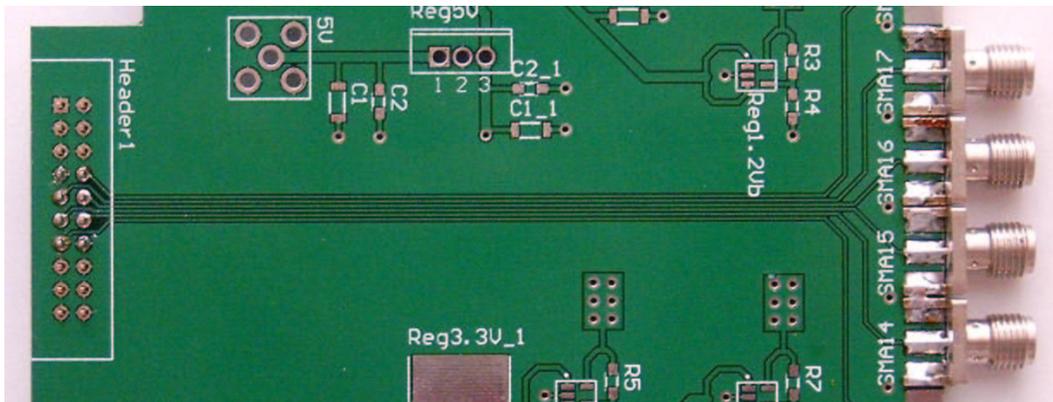


Figure 3.3: Close-up of six adjacent chip-to-chip links.

then the number of candidates increases to:

$$\text{Number of Candidates} = Taps_{perUI,max} \cdot \left((2^N)^1 + (2^N)^2 + \dots + (2^N)^{Taps_{max}} \right) \quad (3.4)$$

$$= Taps_{perUI,max} \cdot \sum_{i=1}^{Taps_{max}} (2^N)^i \quad (3.5)$$

$$= Taps_{perUI,max} \cdot \left(\frac{(2^N) \left(1 - (2^N)^{Taps_{max}} \right)}{(1 - 2^N)} \right) \quad (3.6)$$

using the result that the sum of a geometric series is $\sum_{i=1}^n r^i = \frac{r(1-r^n)}{1-r}$. Now, given the parameters:

$$Taps_{max} = 6 \quad (3.7)$$

$$Taps_{perUI,max} = 6 \quad (3.8)$$

$$N = 3 \quad (3.9)$$

we calculate:

$$\text{Number of Candidates} = 1797552 \quad (3.10)$$

In reality this number is an upper limit on the number of candidates, as we only want to compare pulse shapes on an equal-power basis. Clearly, pulse shapes with more power will tend to have a larger eye opening at the output of the channel. We include the equal-power constraint by immediately discarding any pulse shapes whose tap weights, including sign, do not sum to one. This constraint ensures that we consider only pulse shapes that have the same DC signal swing.¹

In all simulations performed in this chapter, a tap weight resolution of $N = 4$ bits was used. It was found that using a higher resolution produced a marginal performance benefit that did not warrant the greater circuit complexity and increased optimal pulse shape search time.

Now that we have enumerated all possible pulse shapes, we require some method

¹We could have instead chosen to constrain the peak signal swing. This choice may lead to a different optimal pulse shape.

of comparing them to find the optimal pulse shape.

3.2.1 Figure of Merit

To find the optimal transmitter pulse shape, we are going to use the brute-force method of exhaustively scanning through the space of candidates. In order to judge whether one pulse shape is better than another, we need a figure of merit that can be computed quickly while also providing a guide to the size of the resulting eye opening when this pulse shape is used. This figure of merit needs to take into account both ISI and crosstalk so that the optimal pulse shape will represent a tradeoff between reducing ISI and mitigating crosstalk.

I decided to use the ratio of the crosstalk-free eye opening to the maximum possible crosstalk, which I will henceforth refer to as the eye-to-crosstalk ratio (E2C) shown in Equation 3.11.

$$\text{E2C} = \frac{\text{crosstalk-free eye opening}}{\text{maximum possible crosstalk}} \quad (3.11)$$

The crosstalk-free eye opening is simply the eye height at the sampling point seen at the output of the channel. This eye height is determined solely by the ISI of the channel. A simple way of finding this value is to test each candidate pulse shape with a short pseudo-random bit stream (PRBS) sequence. Undesirable pulse shapes will produce bit errors in this short sequence and can be rejected immediately. Other pulse shapes will produce some finite eye opening which can then be used to compare them against other candidates.

The maximum possible crosstalk is calculated by summing the baud-rate samples of the crosstalk response. Taking crosstalk into account in this way acknowledges the fact that adjacent channels can have arbitrary skew relative to the desired channel and that the link will still need to function even with the worst-case skew. However, in many short chip-to-chip links the skew between adjacent channels is minimal and this method may be overly pessimistic. Also implicit in the use of this figure of merit is that each channel is only being considered at a single bit rate. Clearly the same channel at a lower bit rate will have less ISI, and so the E2C will be correspondingly larger.

One drawback of using E2C as a figure of merit is that it does not consider jitter. When using a figure of merit there is a tradeoff between simplicity and descriptive power. The figure of merit used here sacrifices some descriptive power (i.e. it ignores jitter) for simplicity and faster search time.

3.2.2 Searching the Space of Candidate Pulse Shapes

The following procedure was used to find the optimal pulse shape for a given channel at a given bit rate:

1. Obtain the impulse response of the channel, including both *through* and *crosstalk* responses.
2. Establish a figure of merit.
3. Compute the crosstalk-free eye opening using the through response and the bit rate. Eliminate any pulse shapes that do not give an eye opening greater than zero.
4. Compute the maximum crosstalk using the crosstalk response and the bit rate.
5. Compute the figure of merit for each remaining candidate pulse shape.
6. For each number of filter taps and number of taps per UI, choose the pulse shape with the highest figure of merit.

3.3 Results of the Exhaustive Search

Using the measured step response for the PCB channel discussed in Chapter 4, we used the above methodology to find the optimal transmit filter for the given channel.

First we look at the optimal transmit filter in the case where there is no crosstalk from any other link. Then we look at the case where there is a crosstalk channel caused by the proximity of another chip-to-chip link.

3.3.1 PCB Channel with No Crosstalk

To know whether taking crosstalk into account will improve the performance of the chip-to-chip link, it is useful to know which transmit filter is optimal in the absence of

crosstalk. If the optimal filter is the same whether or not we consider crosstalk, then we may as well ignore it. It would just complicate the filter selection procedure for no additional benefit. On the other hand, if the presence of crosstalk produces a different optimal filter then there is some benefit to considering it.

Therefore we first consider the PCB channel with the step response shown in Figure 3.1, but without the two crosstalk responses that are also shown. This simulation does not take any kind of noise into account, so we can no longer use E2C as a figure of merit. For this simulation the figure of merit will be simply the crosstalk-free eye opening. Of course the peak transmitted signal swing is limited so that different pulse shapes can be compared on the same basis.

Figure 3.4 shows the results of the exhaustive search graphically. Each small circle on the graph occurs at an integer value of $Taps_{total}$ and $Taps_{perUI}$, which means that each circle represents a different filter. For each pair of parameter values, the tap weights resulting in the highest crosstalk-free eye opening are found and that value is saved. So Figure 3.4 is a contour plot showing the highest possible crosstalk-free eye opening for each pair of values of $Taps_{total}$ and $Taps_{perUI}$. Between the circles the data is interpolated to better show the shape of the contour plot.

According to this graph, the optimal filter for this channel has five taps and two taps per UI (i.e. tap spacing of 1/2 UI). The tap weights are [1.75 0.5 -1.5 1 -0.75] which results in a pulse shape with significant high frequency content.

The full set of data for the crosstalk-free optimization is shown in Table B.1.

3.3.2 PCB Channel with Crosstalk

Now we again find the optimal filter taps for each value of $Taps_{total}$ and $Taps_{perUI}$, but this time while considering the effect of crosstalk.

Figure 3.5 shows the results of the exhaustive search. According to this graph, the highest E2C occurs with six taps and a tap spacing of 1/3 UI. Figure 3.6 shows a comparison of this optimal six tap pulse with a regular NRZ pulse. The optimal pulse displays a combination of pre-emphasis and slew-rate limiting. For a filter with fewer than six taps, a tap spacing of 1/2 UI is always optimal. Figure 3.7 shows a horizontal slice of the contour plot, in which data is only plotted for filters with a tap spacing of

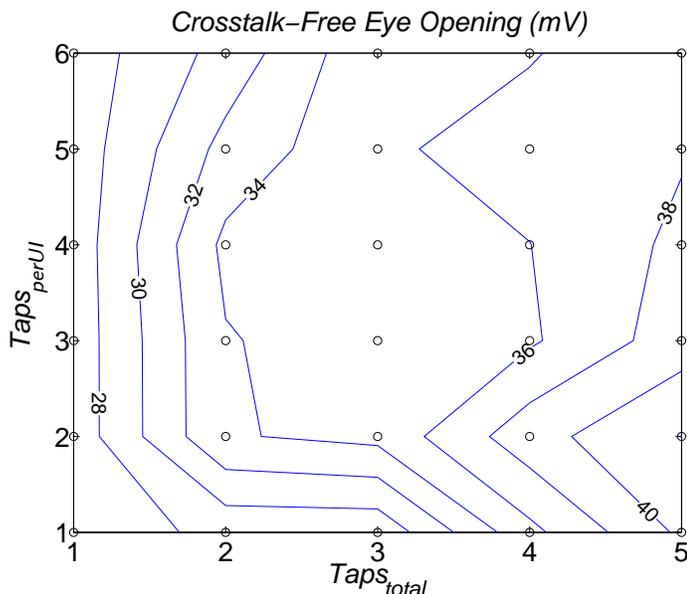


Figure 3.4: Contour plot of simulated crosstalk-free eye opening.

1/2 UI. Clearly, E2C increases monotonically with $Taps_{total}$.

It is interesting to note that the data also illustrates the benefit of a fractionally-spaced filter over a baud-spaced filter. This effect is shown in Figure 3.8. The time granularity of a filter describes the number of filter taps per UI. A filter with many taps per UI is said to be finely grained, while a filter with fewer taps per UI is said to be coarsely grained.

The time extent of a filter is a measure of the number of UI over which the given filter can have an effect. For example, a filter with five taps and two taps per UI has a time extent of 2.5 UI. Figure 3.8 plots E2C against $Taps_{perUI}$ while holding the time extent of the filter constant. We can see that increasing the number of taps per UI increases E2C up to a point. For a filter with an extent of one UI finer granularity is beneficial, but using more than four taps per UI produces diminishing returns. Additionally, increasing the time extent of the filter also results in a higher E2C in general.

The full set of data for the optimization including crosstalk is shown in Table B.2.

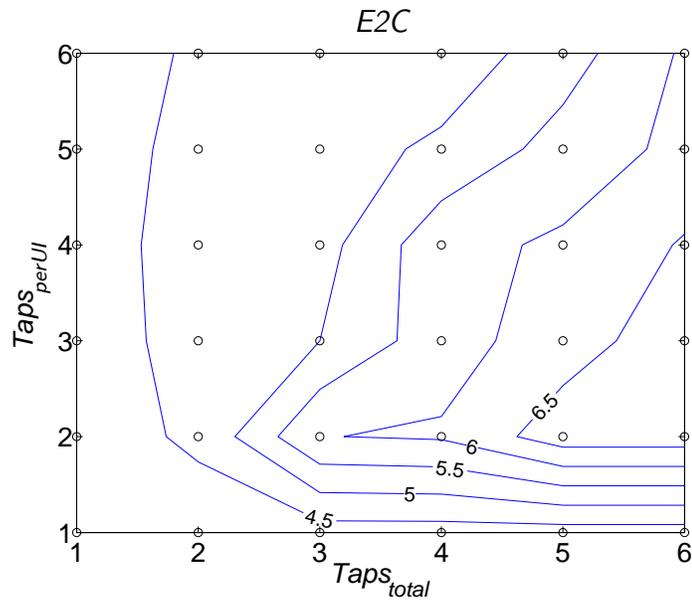


Figure 3.5: Contour plot of simulated E2C against number of taps and taps per UI.

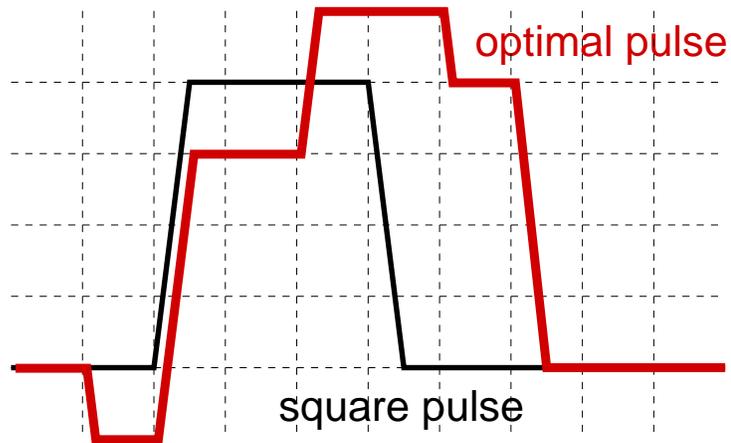


Figure 3.6: Comparison of optimal and regular NRZ pulses.

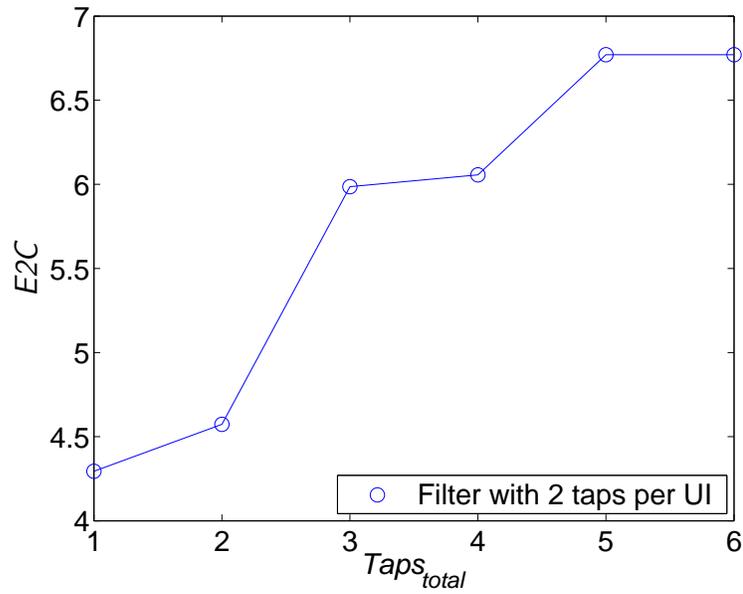


Figure 3.7: Simulated E2C vs. $Taps_{total}$ for a filter with $Taps_{perUI} = 2$.

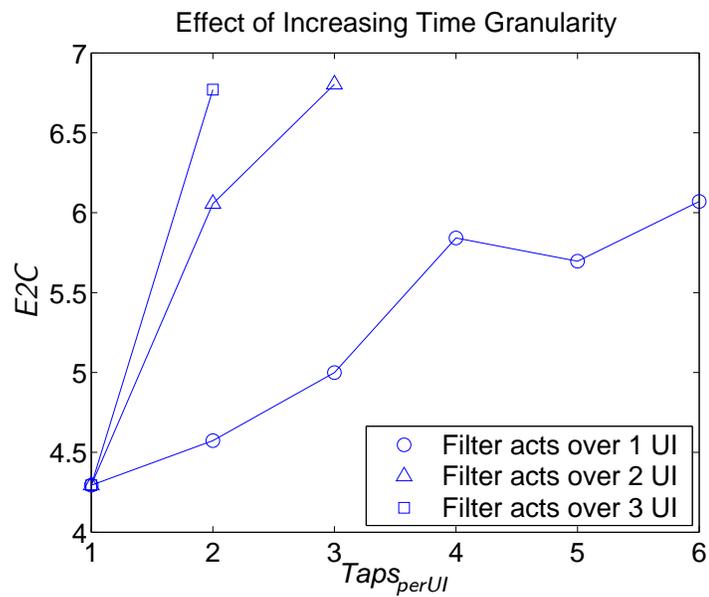


Figure 3.8: Plot of simulated E2C showing effect of time granularity.

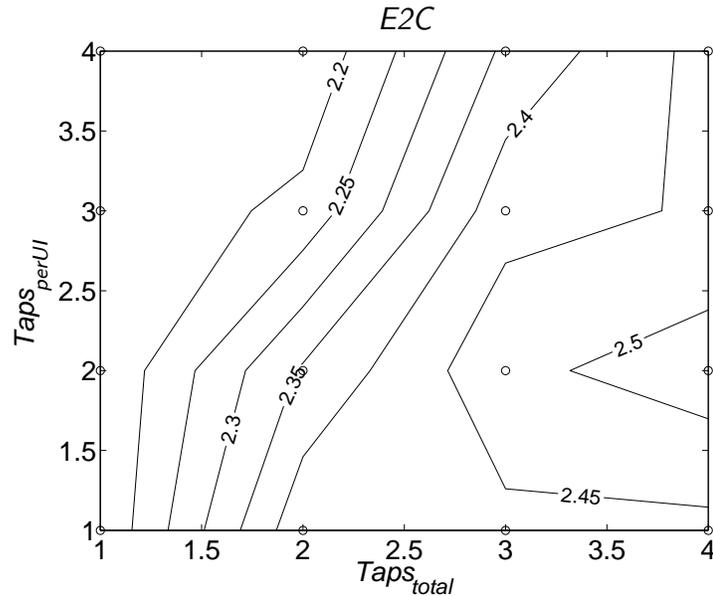


Figure 3.9: Contour plot of E2C for a data rate of 5Gb/s.

3.4 Guideline for Equalizer Specification: Optimal Pulse Shapes above 2.7 Gb/s

As in Sections 3.3.1 and 3.3.2, we want to find the optimal pulse shapes for a given channel. In this case, we would like to find the optimal pulse shape for a data rate higher than 2.7 Gb/s. Figure 3.9 shows the contour plot for a pulse shape optimization at 5 Gb/s. The corresponding time granularity comparison is shown in Figure 3.10.

Similarly, Figures 3.11 and 3.12 show the contour plot of E2C and the time granularity comparison for a data rate of 7.5 Gb/s. The simulation data at these bit rates continue to show the benefit of fractionally-spaced equalizer taps, at least to the level of two taps per UI.

3.5 Summary

This chapter has presented a method of finding the optimal transmitted pulse shape for a given channel. This method starts with the measured step responses of the through

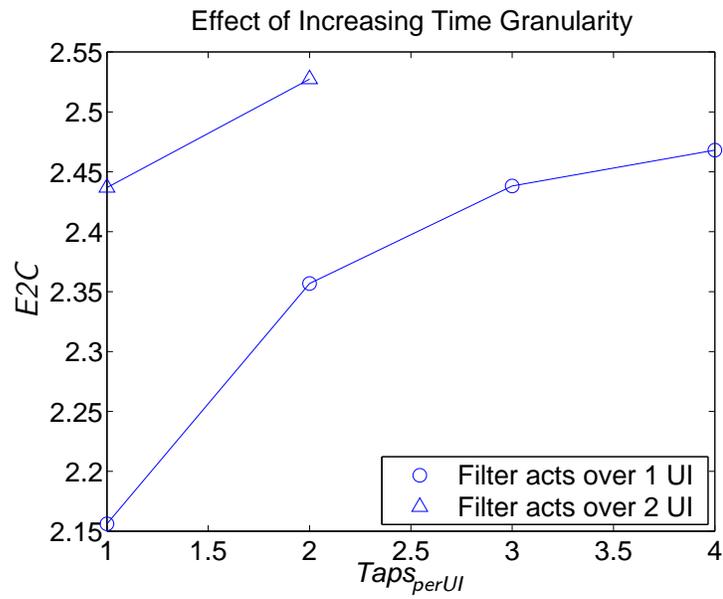


Figure 3.10: Plot of simulated E2C showing effect of time granularity at 5Gb/s.

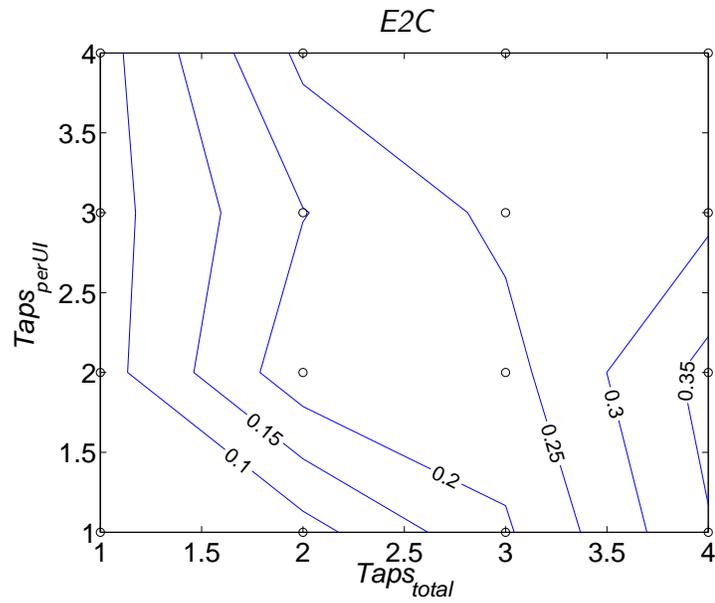


Figure 3.11: Contour plot of E2C for a data rate of 7.5Gb/s.

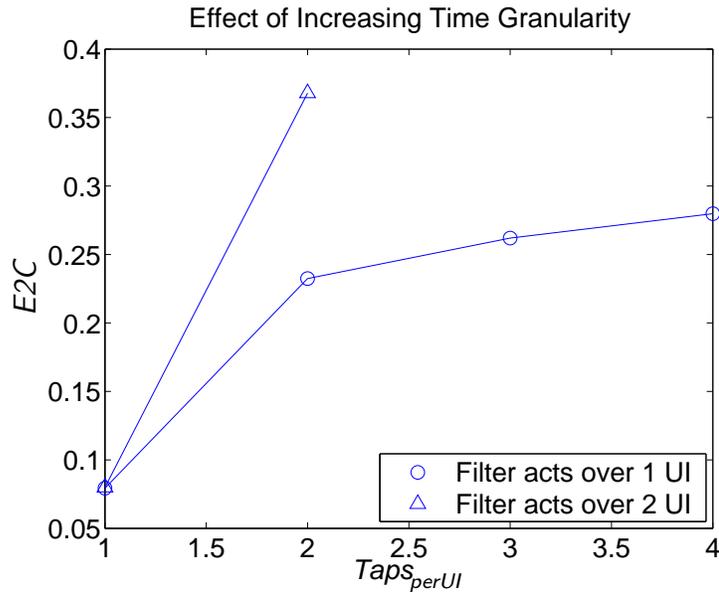


Figure 3.12: Plot of simulated E2C showing effect of time granularity at 7.5Gb/s.

channel and crosstalk channel. A set of candidate pulse shapes is then defined and the candidates are compared based on a figure of merit. The figure of merit used in this search is the ratio of the crosstalk-free eye opening to the maximum possible crosstalk. This ratio takes into account both ISI and crosstalk in an attempt to balance the two and create the largest eye opening at the receiver.

Optimal pulse shapes are found both with and without crosstalk. The optimal pulse shape for the case with crosstalk includes elements of both pre-emphasis and slew rate limiting. The simulation data show the benefit of using a pulse shape with tap spacings of $1/2$ to $1/4$ UI.

Finally, optimizations are performed with the same channel step response but at bit rates of 5 Gb/s and 7.5 Gb/s. These simulations show that a fractionally-spaced transmit filter has some benefit at the higher bit rates that might be targeted in future computer systems.

4 Measurement Results

4.1 Introduction

IN Chapter 3, an optimal pulse shape was found that corresponded to the measured step response of a PCB channel. MATLAB simulation showed that this pulse shape improved the received eye opening. In order to verify this result, the pulse shape must be tested on a real channel with crosstalk.

This chapter presents a proof-of-concept for a crosstalk-aware equalizer. A parallel bit error ratio tester (ParBERT) is used to imitate the function of an equalizing transmit filter, and the output is applied to a board-to-board channel. The ParBERT has nine output modules with a maximum bit rate of 2.7 Gb/s and another two modules with a maximum bit rate of 3.35 Gb/s. For the eye diagram and bit error rate tests three of the 2.7 Gb/s module outputs have been combined together with power combiners. The swing and relative phase of each module can be controlled in software, and so any pulse shape can be reproduced with this setup. The setup is shown in Figure 4.1.

4.2 Time Domain Reflectometry

Time domain reflectometry is a measurement technique that involves injecting a step input into the device under test (DUT)—in this case the chip-to-chip channel—and observing both the output and the reflection at the input. This technique is often used to discover impedance discontinuities anywhere along a channel, which is done by looking at the reflection from the channel. In this case it was used to find the step response of the channel, as well as the crosstalk seen in adjacent channels.

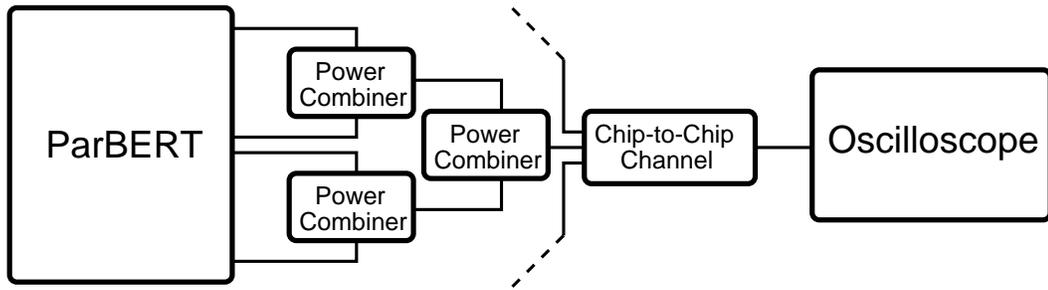


Figure 4.1: Equalizer proof-of-concept test setup.

Figure 3.1 showed the measured step responses of three channels on the test board. The through response (*through*) is shown along with the crosstalk response between adjacent channels (*crosstalk1*) and the crosstalk response between channels that are two places removed from one another (*crosstalk2*).

From the measured step response, an impulse response can be computed by subtracting a shifted copy of the step response from the original step response. The result of this computation is shown in Figure 4.2 for all three channels. The maximum value of the *crosstalk1* impulse response is about 20% of the maximum value of the through response.

Similarly, a frequency response can be computed from the impulse response just obtained. The computed frequency responses of the three channels are shown in Figure 4.3. The differentiating nature of the two crosstalk channels can be clearly seen in this figure; the slope of the crosstalk frequency response is 20 dB/decade at low frequency. The 3 dB frequency of the through response is 550 MHz.

4.3 Output Eye Diagrams

While the frequency response of the channel has already been seen in Figure 4.3, it is helpful to see how the channel distorts an ideal input pulse in the time domain. Figure 4.4 shows the input to the channel at a bit rate of 2.7 Gb/s (the maximum bit rate of the ParBERT). Figure 4.5 shows the corresponding channel output. We can see that at this bit rate there is significant ISI which has caused the eye to close by about

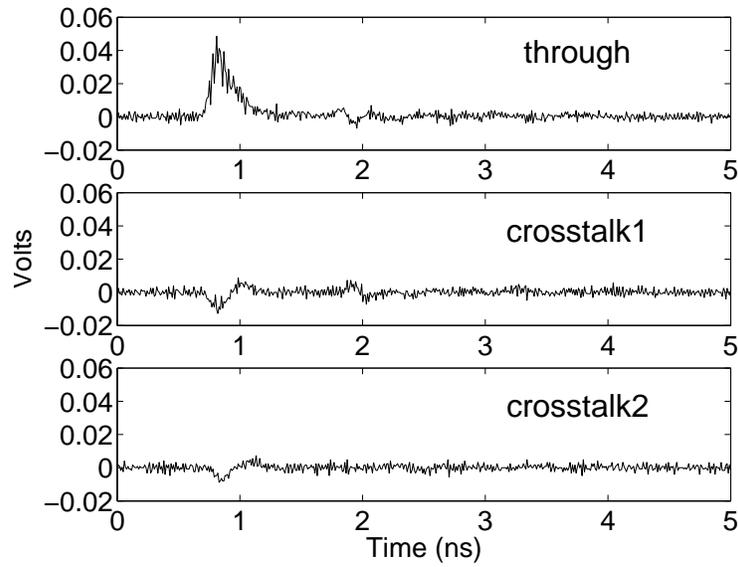


Figure 4.2: Impulse response of the channel, computed from the step response (time step = 10 ps).

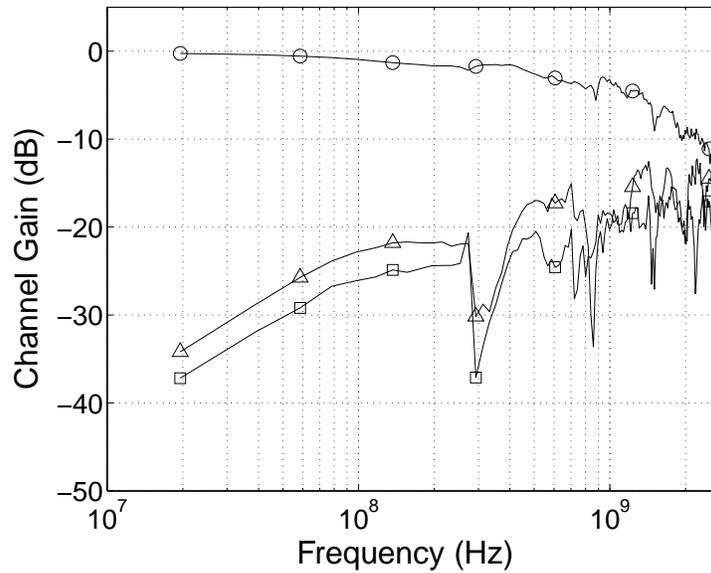


Figure 4.3: Frequency response of the through channel (\circ), first crosstalk channel (\triangle), and second crosstalk channel (\square) computed from the step response (time step = 50 ps).

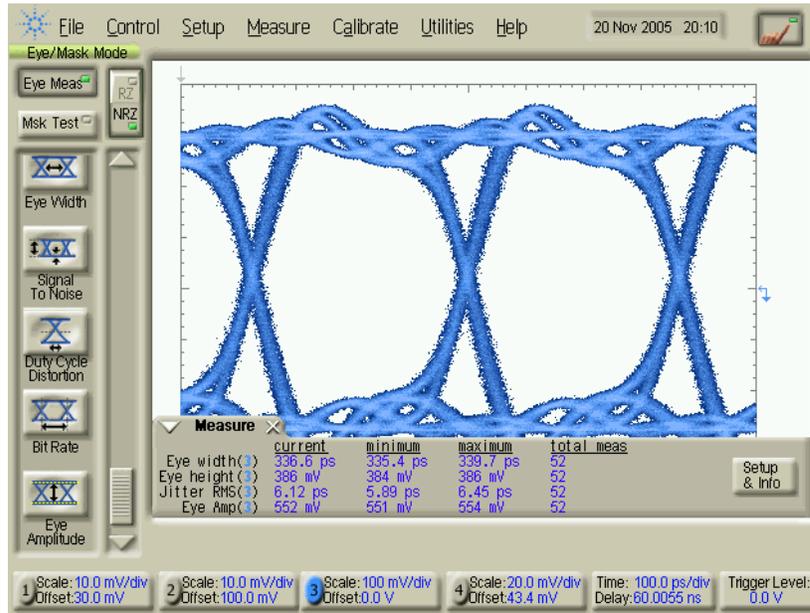


Figure 4.4: Measured eye diagram at 2.7 Gb/s with a PRBS sequence of length $2^{31} - 1$. This figure shows the channel input with no aggressors.

45%.

If ISI were the only channel impairment, then we would be able to design an equalizer as discussed at the beginning of Chapter 3 and be done with it. However a real chip-to-chip channel looks more like the one shown in Figure 4.6, in which some undesirable aggressor signal is feeding through to the receiver in the desired channel.

When we turn on 2.7 Gb/s-signals on both channels adjacent to the channel of interest, we get the eye diagram shown in Figure 4.7. Although the input to the channel in this case was the same as the one shown in Figure 4.4, crosstalk has closed the eye almost completely. According to the oscilloscope, the eye height is 39 mV. To get this eye height, the oscilloscope first finds the average '1' signal level, $V_{1'}$, and the average '0' signal level, $V_{0'}$. Then it computes the 3σ points for each signal level, the points that include 99.7% of the samples between them. The eye height is then the difference between the low 3σ point for the '1' level and the high 3σ point for the '0' level.

To confirm that the optimal pulse shape generated by our exhaustive search from

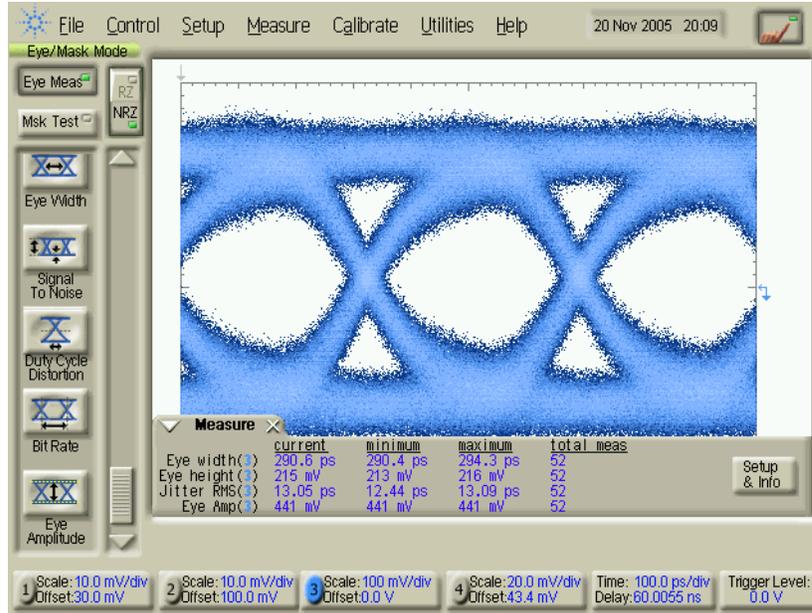


Figure 4.5: Measured eye diagram at 2.7 Gb/s with a PRBS sequence of length $2^{31} - 1$. This figure shows the channel output corresponding to Figure 4.4 with no aggressors.

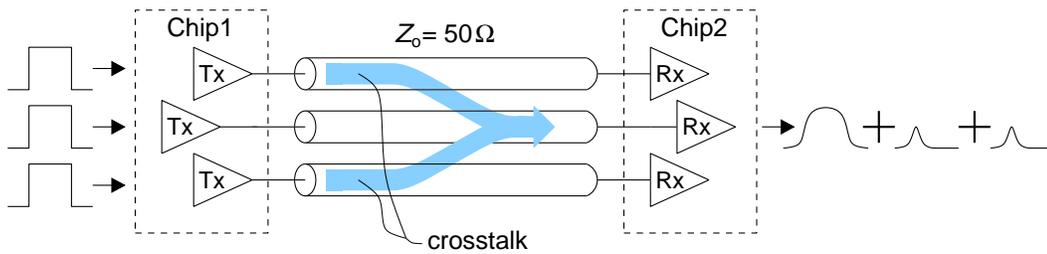


Figure 4.6: Impact of adjacent aggressor signals on desired signal.

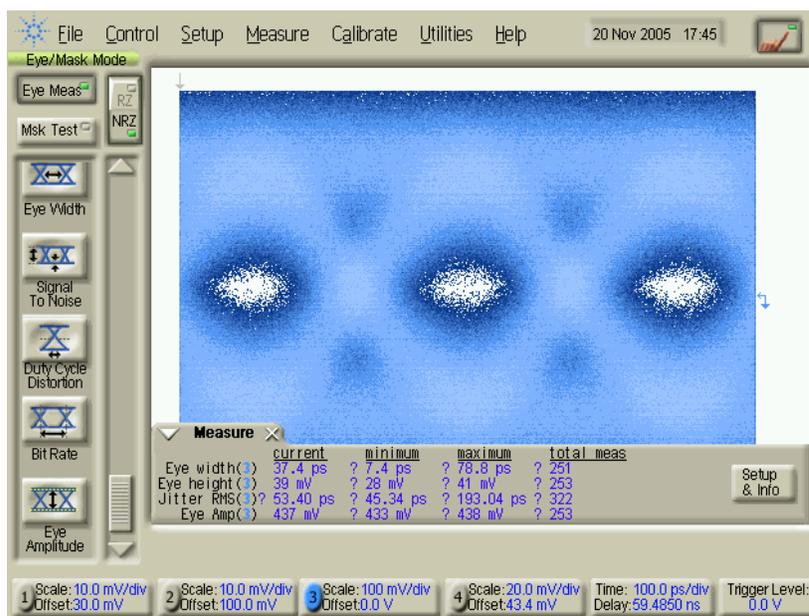


Figure 4.7: Measured eye diagram at 2.7 Gb/s with a PRBS sequence of length $2^{31} - 1$. This figure shows the output of the chip-to-chip channel for square pulse input with two aggressors.

Chapter 3 is beneficial, we need to generate the desired pulse shape and then observe the signal at the output of the channel. The resulting eye diagram is shown in Figure 4.8. Notice that the oscilloscope now measures a 101-mV eye opening, an improvement of 159%.

4.4 Bit Error Rate Testing

While eye diagrams can provide a clue to the performance of a link, they do not tell the whole story. A visual comparison gives a qualitative understanding of how the link has been improved. To gain a quantitative understanding, however, requires a BER test. In this BER test, the ParBERT acts as a receiver at the end of the channel. We can tell how good the transmit filter is by looking at a bathtub plot in which BER is plotted against sampling phase.

Figure 4.9 shows the improvement in BER between a square pulse and the optimal crosstalk-aware pulse shape found in Chapter 3. With a square pulse, the BER is

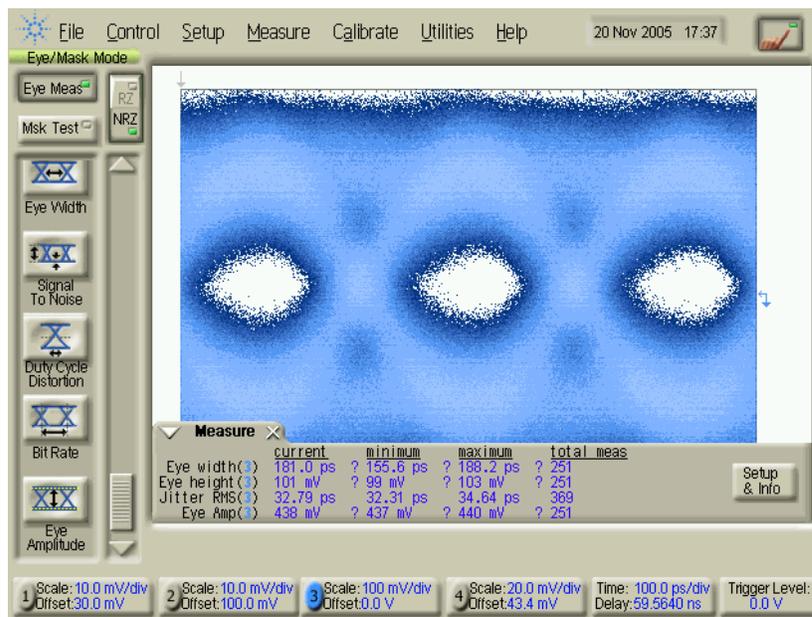


Figure 4.8: Measured eye diagram at 2.7 Gb/s with a PRBS sequence of length $2^{31} - 1$. This figure shows the output of the chip-to-chip channel for crosstalk-aware pulse input with two aggressors.

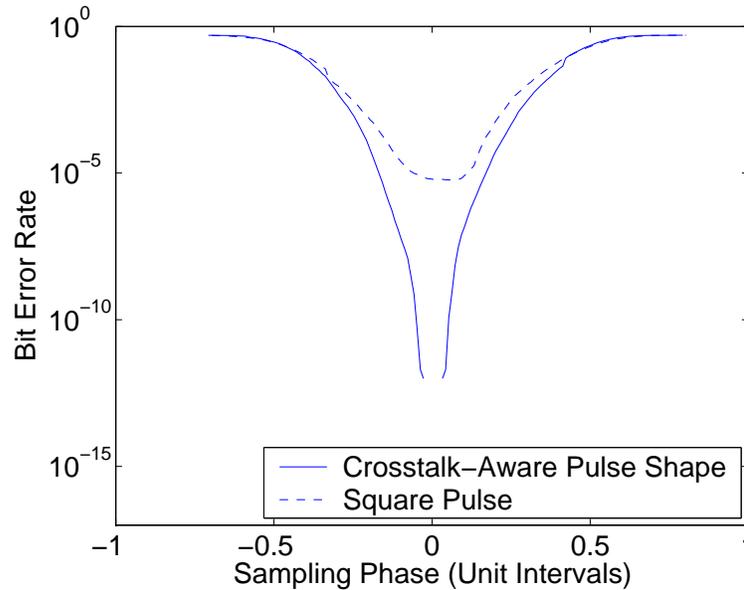


Figure 4.9: Bathtub plot comparing crosstalk-aware and square pulses.

limited to a minimum of 10^{-5} , whereas using the crosstalk-aware pulse reduces the BER below 10^{-12} .

This comparison is not entirely fair, however, because in a typical chip-to-chip link a pre-emphasis pulse would be used, not a square pulse. Due to equipment limitations the signal swing was smaller in this test, which resulted in a higher BER when using the same crosstalk-aware pulse shape as in the first BER test. Figure 4.10 shows that the minimum bit rate achievable with a pre-emphasis pulse is 5×10^{-6} , whereas the crosstalk-aware pulse achieves a bit rate of 5×10^{-8} .

4.5 Summary

This chapter has presented measurements of a board-to-board channel as well as eye diagrams and BER bathtub plots for a proof-of-concept of the proposed equalizer. The step response measurements of the channel were used as the basis for finding the optimal pulse shape in Chapter 3. Received eye diagrams were used to compare the performance of a square pulse with that of a crosstalk-aware pulse. The crosstalk-aware

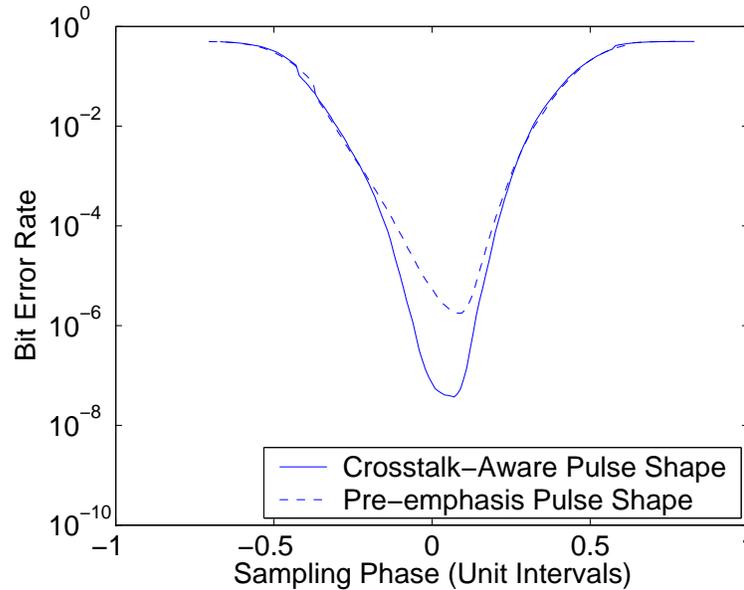


Figure 4.10: Bathtub plot comparing crosstalk-aware and pre-emphasis pulses. BER is higher than in Figure 4.9 because a smaller signal swing was used in this measurement for both pulse shapes.

pulse resulted in the eye opening increasing from 39 mV to 101 mV. Similarly, bathtub plots showed that the crosstalk-aware pulse provided a decrease in BER by more than seven orders of magnitude. Additionally, it was recognized that a pre-emphasis pulse would typically be used in chip-to-chip communication, and a bathtub plot showed that the crosstalk aware pulse provided a decrease in BER of two orders of magnitude over a typical pre-emphasis pulse.

5 Transmitter Design

5.1 Introduction

THIS chapter presents the design of a programmable pulse-shaping transmitter consistent with the requirements discovered through the simulations of Chapter 3 and the measurements of Chapter 4. The fabrication technology used for this chip is IBM 0.13- μm CMOS with eight metal layers and an $f_{T,max}$ of 65 GHz.

Although Chapter 3 focussed on finding the optimal pulse shape for a very specific channel at a bit rate of 2.7 Gb/s, this programmable transmitter is intended to be more generally useful. To that end, the transmitter will consist of eight taps and the number of taps per bit will be adjustable between two and eight. In addition, the transmitter is differential which means that the output can be taken either differentially or single-endedly. This flexibility will enable the transmitter to be useful as a pulse shape test platform for a variety of channels with different ISI and crosstalk characteristics. In addition, the bit rate of the transmitter should be adjustable between 1 Gb/s and 10 Gb/s.

5.2 Transmitter Architecture

A block diagram of the proposed transmitter is shown in Figure 5.1. The transmitter is broken down into eight slices, and each slice contains an output driver and a delay element. These slices are the taps of the filter. The currents of all eight output drivers are summed in a common pair of 50- Ω resistors not shown in the figure. The time delay of the delay elements is controlled by a common delay cell control voltage. The tap

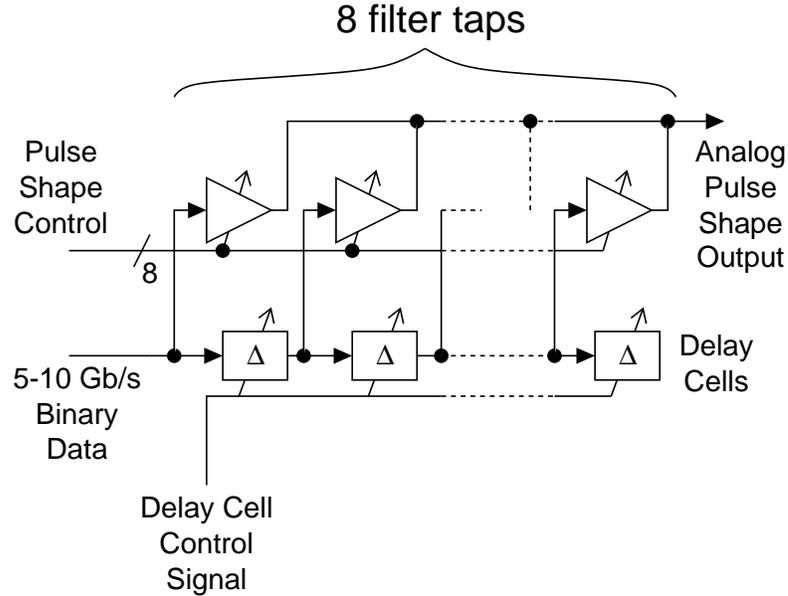


Figure 5.1: Block diagram of the proposed transmitter.

weights are controlled by a parallel bus of binary control voltages that are connected to each output driver cell.

5.2.1 Output Driver

A schematic diagram of one output driver cell is shown in Figure 5.2. Each cell has four bits of output swing tunability, one bit for the sign and three bits for the amplitude. The amplitude is controlled by switching three binary-weighted tail current sources.

The speed limitation on the output driver cell is the fact that all of the cells are connected together at the output. The resulting capacitance that loads the output node slows down the output driver circuit. Compounding this problem is the fact that each output driver differential pair must be large enough to handle the maximum tail current of $I_{tail} + 2I_{tail} + 4I_{tail} = 7I_{tail}$ in Figure 5.2. Since most of the time only one or perhaps two driver cells will be using the maximum current, there is a large output capacitance overhead inherent in this circuit topology. Many output driver cells will be using, for example, I_{tail} current while the size of the differential pair can accommodate

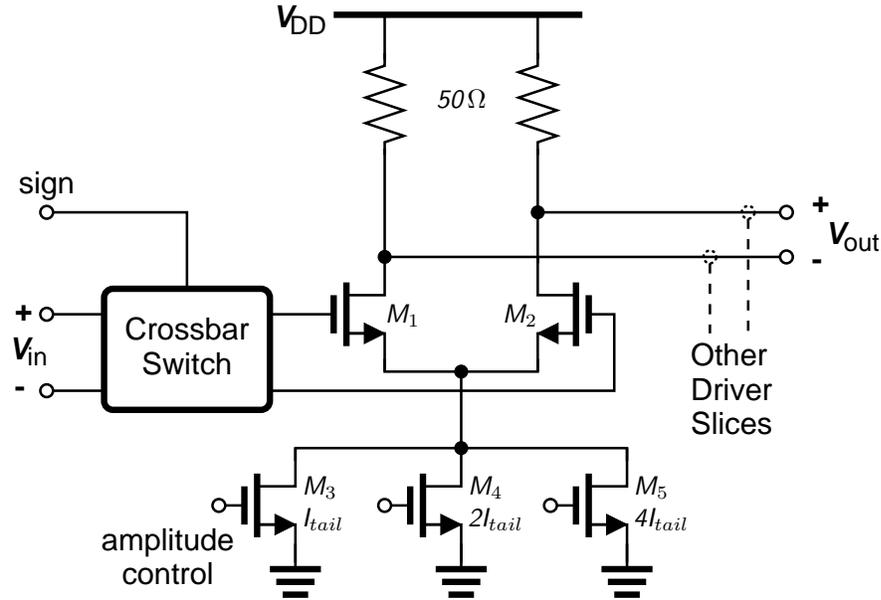


Figure 5.2: Schematic diagram of the output driver cell. Gate lengths are $0.12\ \mu\text{m}$. Gate widths of M_1 and M_2 are $80\ \mu\text{m}$.

$7I_{tail}$ current, resulting in an output capacitance overhead of seven times.

Since the maximum current in the output driver is fixed based on the supply voltage and the required voltage headroom of the differential pair, it is possible to improve on this result. One way to reduce the output capacitance would be to break up the output driver into an array of differential pairs sized to handle I_{tail} current. All of these output driver cells would be in use all the time, and individual cells would be associated with specific filter taps by programming a series of multiplexors. This architecture is illustrated in Figure 5.4.

This improved architecture does reduce the capacitance at the output node, but unfortunately it introduces additional complexity in the multiplexor array. Also, in this case the performance of the output driver is not the limitation on the speed of the entire transmitter. Transmitter performance is limited more severely by the speed of the delay cell.

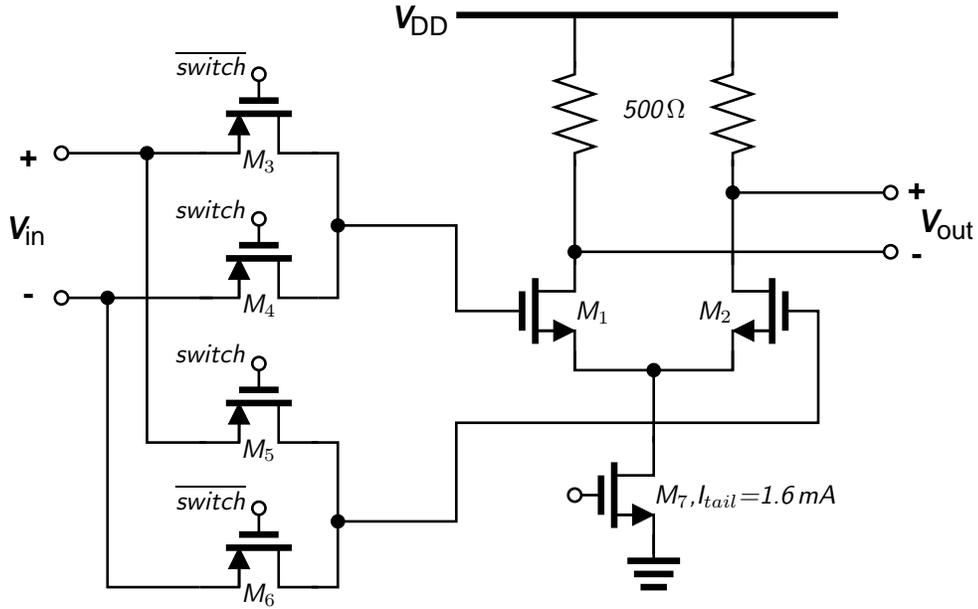


Figure 5.3: Schematic diagram of the crossbar switch. Gate lengths are $0.12\ \mu\text{m}$ and gate widths are $10\ \mu\text{m}$. The pass transistors M_3 – M_6 have $R_{on} = 285\ \Omega$.

5.2.2 Delay Cell

As mentioned above, the delay cell places a more stringent requirement on the performance of the circuit than does the output driver. For example, in order for the circuit to function at 10 Gb/s with four taps per UI, the time delay introduced by each delay cell would need to be:

$$\begin{aligned} \text{delay} &= \frac{1}{4 \frac{\text{taps}}{\text{bit}} \cdot 10\ \text{Gb/s}} \\ &= 25\ \text{ps/tap} \end{aligned} \quad (5.1)$$

For the circuit to function at 1 Gb/s with two taps per bit, the time delay would need to be:

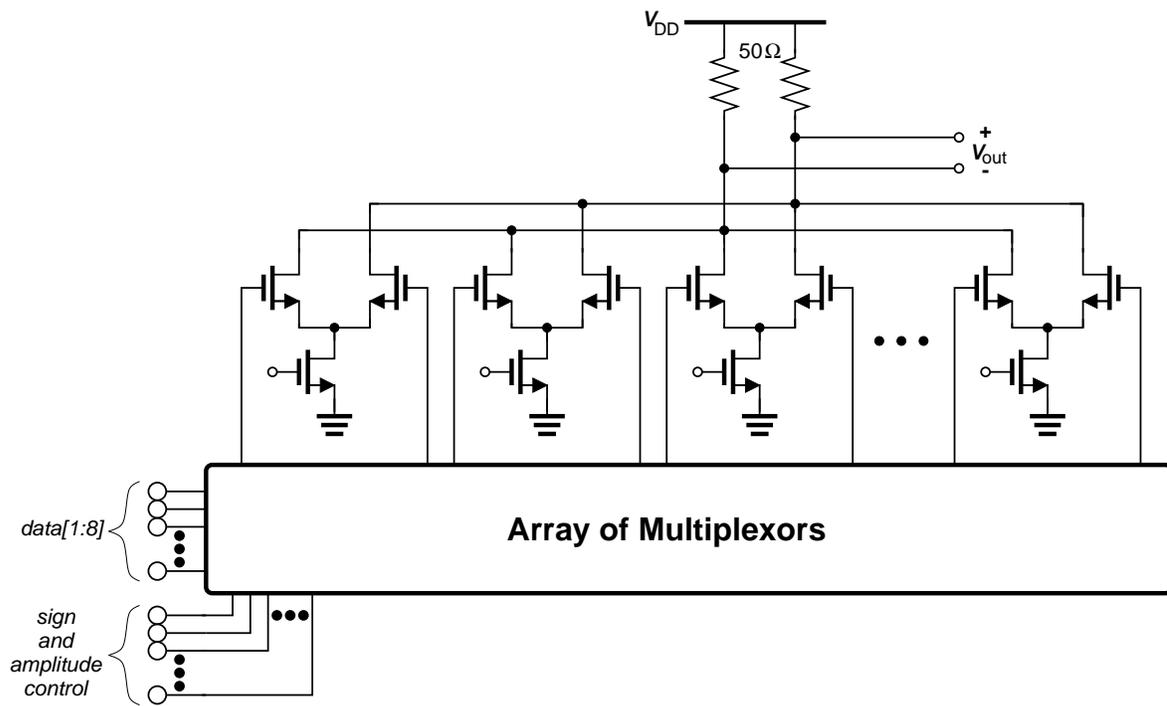


Figure 5.4: Schematic diagram of the improved output driver cell.

$$\begin{aligned}\text{delay} &= \frac{1}{2^{\frac{\text{taps}}{\text{bit}}} \cdot 1 \text{ Gb/s}} \\ &= 500 \text{ ps/tap}\end{aligned}\tag{5.2}$$

Accommodating both of these extremes would require a tuning range greater than an order of magnitude. There are several delay cell topologies that might be useful. Four possible topologies will be compared here, each of them having advantages and disadvantages. The four delay cells are the starved inverter, common source, low voltage, and self-biased symmetric load delay cells.

The *starved inverter* delay cell pictured in Figure 5.5 consumes the smallest amount of power of the three cells when running at its lowest delay. This delay cell is similar to a CMOS inverter and so it will dissipate most of its power while switching and will only draw leakage current when static. However, it is one of the slower delay cells considered here. Another problem with the starved inverter delay cell is its inherently poor power supply rejection ratio. Because it is similar to a CMOS inverter, any power supply noise is fed directly to the output node. And the fact that this delay cell is single-ended instead of differential means that none of that noise is cancelled out.

The power dissipation of the starved inverter delay cell is proportional to data transition frequency and is unaffected by the control voltage. These characteristics are the opposite of the common source and low voltage delay cells, for which power dissipation depends only on control voltage and not on data transition frequency. Having a delay cell with fixed power for a given bit rate is good because it means that the transmitter as a whole would draw the same amount of power independent of how the individual filter taps are programmed.

In addition to the circuit pictured in Figure 5.5, there are other ways of designing this circuit. The analog tuning voltage can be replaced with a digital tuning voltage controlling a bank of binary-sized transistors that limit the current to the CMOS inverter [24].

The *common source* delay cell pictured in Figure 5.6 provides a smaller minimum delay than the starved inverter. Another benefit is that the power dissipation of the

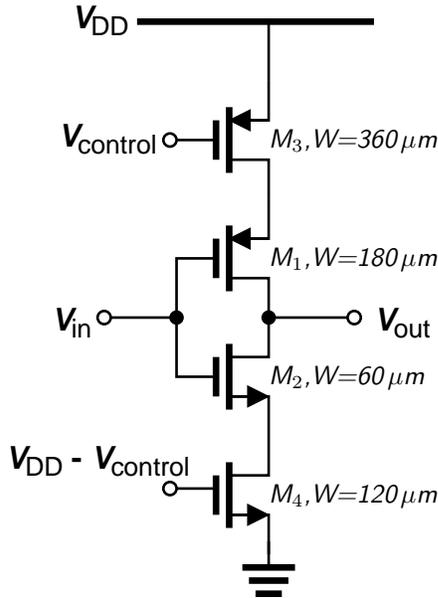


Figure 5.5: Schematic diagram of the starved inverter delay cell. Gate lengths are $0.12 \mu\text{m}$.

delay cell can be set by the tail current, although using less power will also result in a slower delay cell. The common source delay cell has better power supply rejection than the starved inverter cell because of the tail current source and the differential nature of the circuit. The drawback of this delay cell is its small tuning range, similar to that of the starved inverter delay cell.

To improve the tuning range of the common source delay cell, two additions can be made. First, the PMOS load can be augmented with a diode-connected PMOS transistor in parallel. This parallel combination more closely approximates the behaviour of a resistor, and is known as a symmetric load [25]. The voltage $V_{control}$ then controls the resistance of the load. The I-V curves of these two loads are shown in Figure 5.7, and the symmetric-load delay cell is shown in Figure 5.8.

As the resistance of the load increases, the signal swing at the output nodes also increases. Once this swing becomes too high, the tail current transistor starts to operate in the triode region and the delay cell no longer functions. To extend the tuning range of the common source delay cell, the tail current can be made dependent on the control voltage. This change ensures that as the resistance of the load increases,

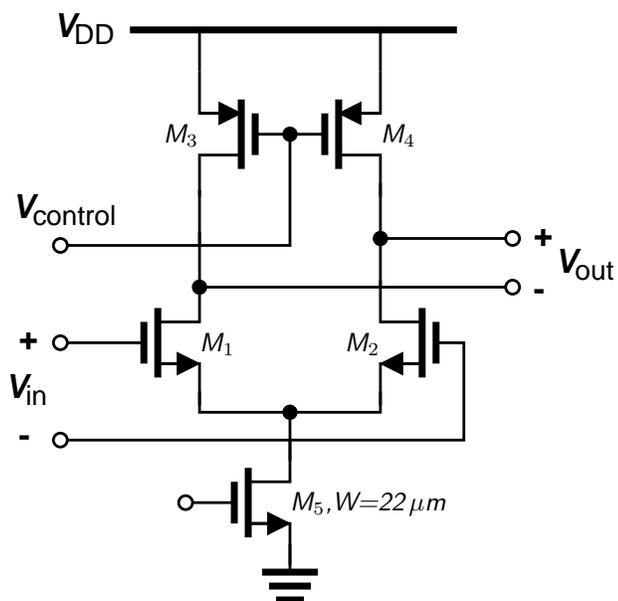


Figure 5.6: Schematic diagram of the common source delay cell. Gate lengths are $0.12 \mu\text{m}$ and gate widths are $10 \mu\text{m}$ unless otherwise stated.

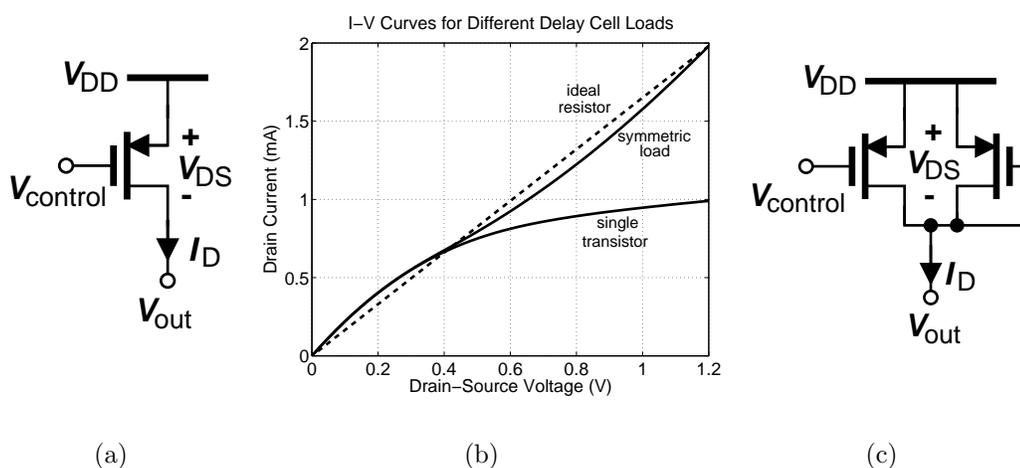


Figure 5.7: (a) Single transistor load, (b) I-V curve comparison, (c) symmetric load.

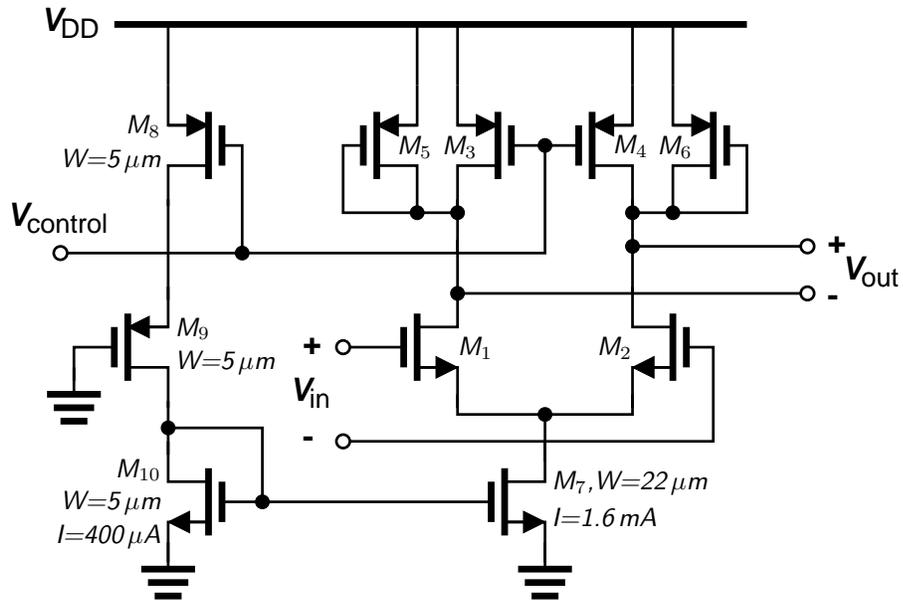


Figure 5.9: Schematic diagram of the self-biased symmetric-load delay cell. Gate lengths are $0.12 \mu\text{m}$ and gate widths are $10 \mu\text{m}$ unless otherwise stated.

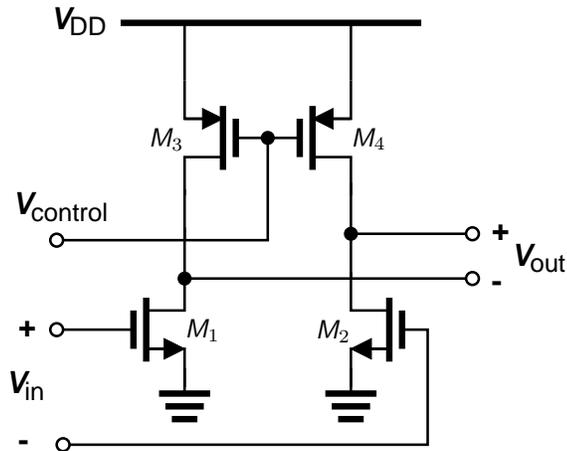


Figure 5.10: Schematic diagram of the low voltage delay cell. Gate lengths are $0.12 \mu\text{m}$ and gate widths are $10 \mu\text{m}$.

Delay Cell	Min. Delay (ps)	Max. Delay (ps)	Max. Power (mW)
Starved Inverter	48	> 200	1.75 [†]
Low Voltage	49	> 200	3.8
Self-Biased Symmetric Load	35	> 200	3.3

[†]Power dissipation at a bit rate of 1 Gb/s.

Table 5.1: Summary of delay cell characteristics.

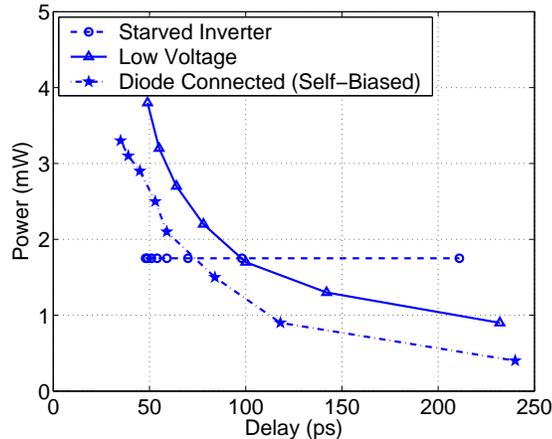


Figure 5.12: Delay cell comparison. Note: simulations conducted at 1 Gb/s. Power dissipation of the starved inverter cell would be higher at the targeted bit rate of 5 Gb/s.

twice as many times in a second, then the power dissipation is doubled. A summary of the characteristics of each cell is shown in Table 5.1.

A comparison of the power and delay range of the low voltage and diode-connected self-biased cells is shown in Figure 5.13. This figure also shows how the time delay and power of each cell scale as the power supply voltage decreases. The voltages marked on the figure denote the highest and lowest supply voltages used for the simulations that produced the corresponding power-delay curves. Although the low voltage delay cell has the benefit—from a voltage headroom point of view—of omitting a tail current source, the self-biased symmetric-load delay cell still performs better at lower supply voltages.

None of the delay cells examined has a tuning range greater than an order of magnitude, and so the transmitter will not be as flexible as initially hoped. However the

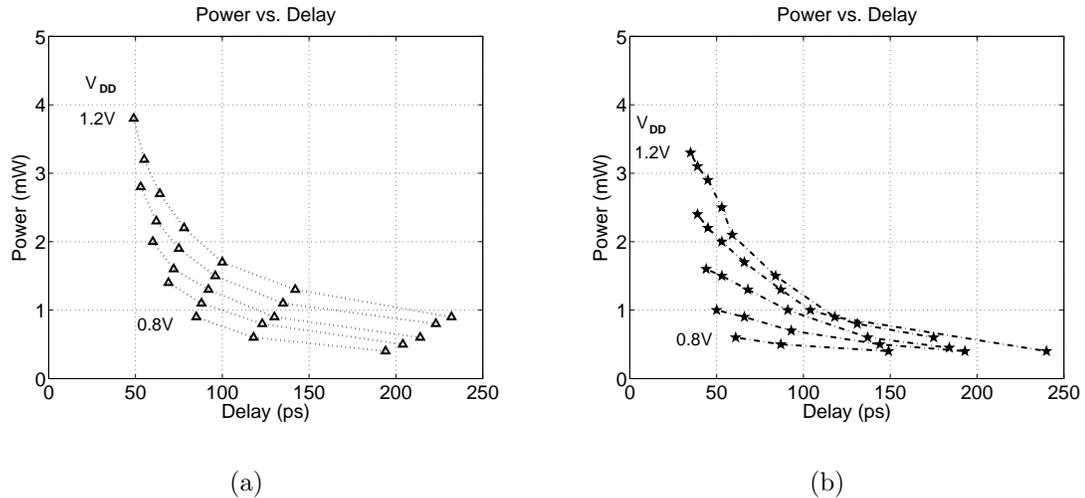


Figure 5.13: Supply voltage scaling: (a) low voltage cell, (b) self-biased, symmetric-load cell.

self-biased symmetric-load delay cell has a tunable range from 35 ps to 200 ps which still allows a range of operating modes. With this range, the transmitter circuit can be operated with $Taps_{perUI} = 4$ from 1.25 Gb/s up to 7 Gb/s or with $Taps_{perUI} = 3$ from 1.67 Gb/s to 9.5 Gb/s.

5.3 Simulation Results

A transmit-side equalizer was designed using IBM's 0.13- μm CMOS design kit using the output driver from Section 5.2.1 and the self-biased symmetric-load delay cell from Section 5.2.2. This equalizer has eight taps, and each of the tap weights can be digitally programmed with 4 bits of resolution. Each tap weight can be either negative or positive. The reference current that controls the tap weights can also be controlled as it is provided from off-chip.

Spectre simulation results showing the versatility of the equalizer can be seen in Figures 5.14 and 5.15. These figures show a sample of the pulse shapes that can be generated by the programmable equalizer circuit. For each pulse shape, the output of

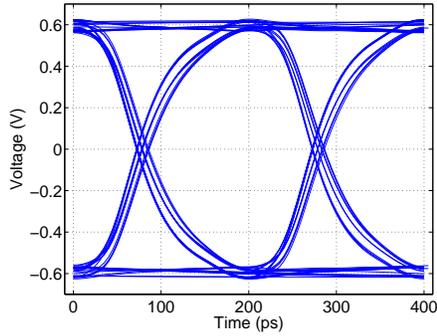
the equalizer (input to the channel) is shown along with the signal output from the channel and the crosstalk output from the channel. These channel outputs are obtained as the convolution of the transient waveform from Spectre and the measured impulse response of the chip-to-chip channel. Several results emerge from an examination of these pulse shapes:

- Using a DC signal swing greater than the eye opening at the receiver decreases that eye opening. This result can be seen by comparing the regular and pre-emphasis pulse shapes. In addition to a larger eye opening, the pre-emphasis pulse shape also produces less crosstalk.
- Using a higher slew rate produces more crosstalk. This result can be seen by comparing the regular, slew-rate limited, and square pulse shapes. This characteristic is obvious in light of the fact that the crosstalk channel acts as a differentiator, as seen in Chapter 2.

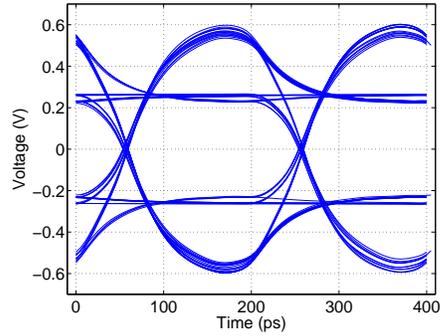
The optimal pulse shape generated by the procedure of Chapter 3 is shown in Figure 5.16. This pulse combines the two results discussed above. The slew-rate is limited and at the same time pre-emphasis is used to limit the DC signal swing. The result is a compromise between increased signal swing and reduced crosstalk to adjacent channels. It is not obvious from Figure 5.16 that the “optimal” pulse does in fact perform better than the pre-emphasis pulse of Figure 5.14. One reason this might be true is that the optimization in Chapter 3 an ideal FIR filter was assumed. In these simulations the finite bandwidth of the output driver at a bit rate of 5 Gb/s is taken into account which changes the results. This change was not seen in the measurements of Chapter 4 because those measurements were taken at the lower bit rate of 2.7 Gb/s. A summary of the characteristics of the transmitter is shown in Table 5.2.

Circuit simulation reveals an additional limit on the delay cell tuning range. Ideally, each of the eight filter taps would provide a signal of identical amplitude, with the only difference being a phase shift. In reality the finite delay cell bandwidth causes the delay cells farther along in the chain to produce more jitter than those at the beginning of the chain.

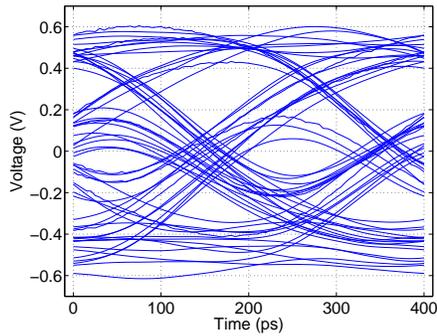
When a delay cell is tuned to produce a longer delay, the bandwidth of the cell is reduced. This drawback is an unavoidable fact of using a delay which is analog in time.



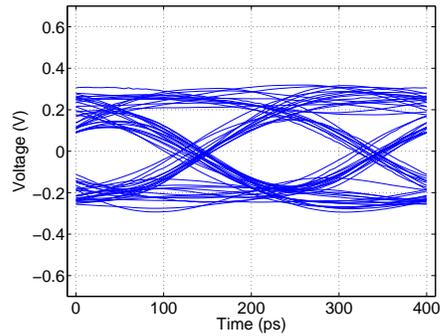
(a) Regular NRZ pulse shape, [1 0 0 0 0]



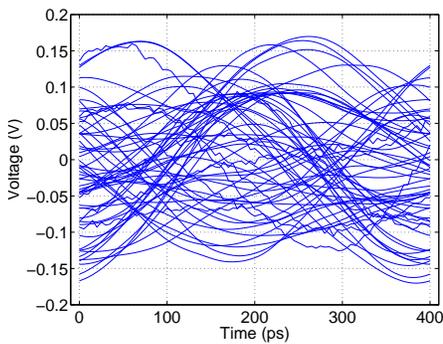
(b) Pre-emphasis pulse shape, [1 0 0 -0.5 0 0]



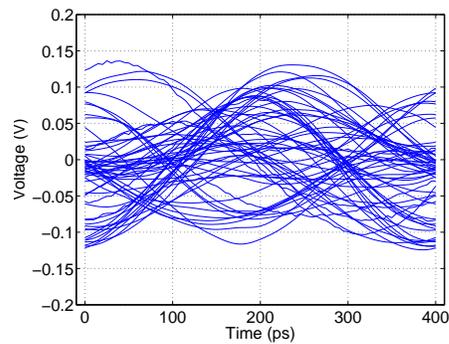
(c) Signal output: regular NRZ pulse



(d) Signal output: pre-emphasis pulse

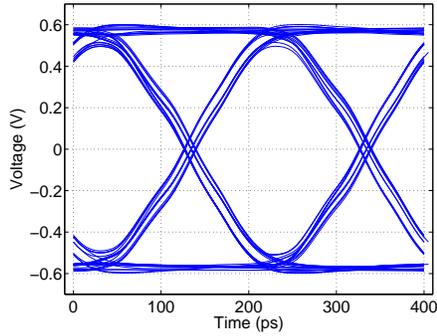


(e) Crosstalk output: regular NRZ pulse

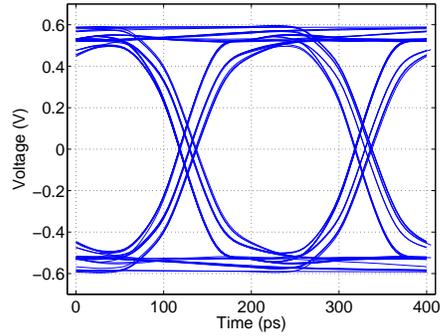


(f) Crosstalk output: pre-emphasis pulse

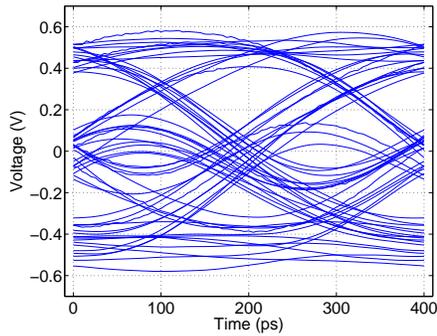
Figure 5.14: Spectre simulation of (a) regular NRZ and (b) pre-emphasis pulse shapes at 5 Gb/s with a PRBS sequence of length $2^7 - 1$. The corresponding signal and crosstalk outputs are shown in (c)-(f) for the chip-to-chip channel. $T_{aps_perUI} = 3$.



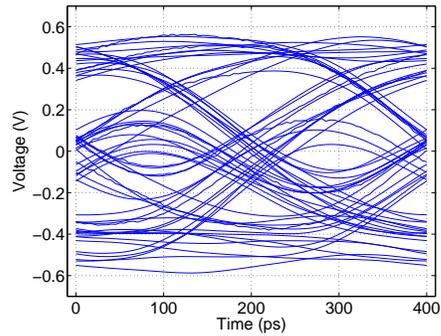
(a) Slew-rate limited pulse shape, $[0.33 \ 0.33 \ 0.33 \ 0 \ 0 \ 0]$



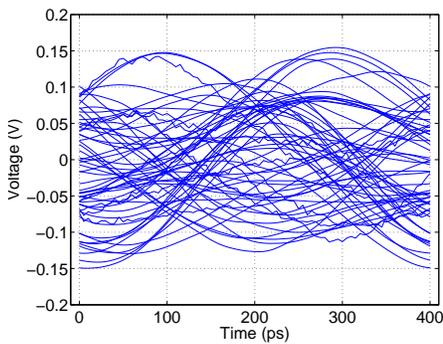
(b) Square pulse shape, $[1 \ -0.1 \ 0 \ 0 \ 0 \ 0]$



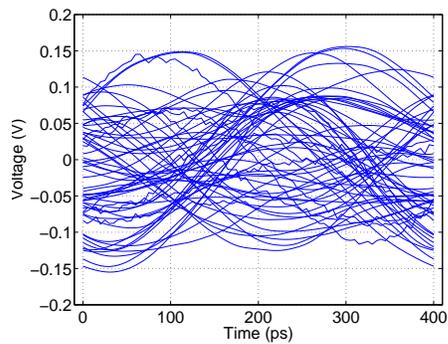
(c) Signal output: slew-rate limited pulse



(d) Signal output: square pulse

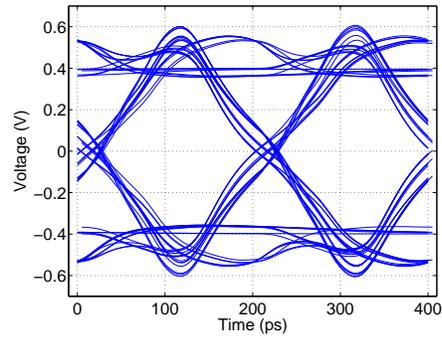


(e) Crosstalk output: slew-rate limited pulse

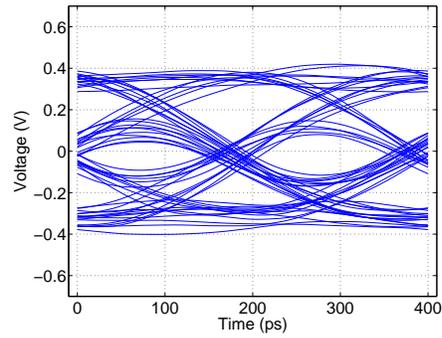


(f) Crosstalk output: square pulse

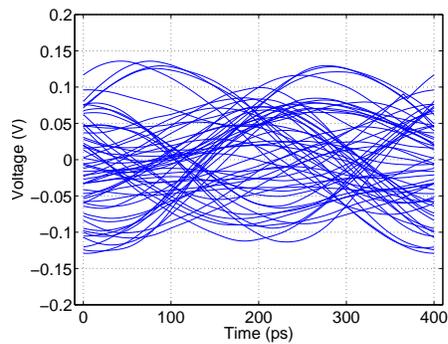
Figure 5.15: Spectre simulation of (a) slew-rate limited and (b) square pulse shapes at 5 Gb/s with a PRBS sequence of length $2^7 - 1$. The corresponding signal and crosstalk outputs are shown in (c)-(f) for the chip-to-chip channel. $Taps_{perUI} = 3$.



(a) Optimal pulse shape, $[-0.25 \ 1 \ 0 \ 0.5 \ 0 \ -0.25]$



(b) Signal output: optimal pulse shape



(c) Crosstalk output: optimal pulse shape

Figure 5.16: Spectre simulation of (a) the optimal pulse shape at 5 Gb/s with a PRBS sequence of length $2^7 - 1$. Also shown are (b) the corresponding signal output and (c) the crosstalk output for the chip-to-chip channel. $Taps_{perUI} = 3$.

The increase in jitter caused by the reduced bandwidth can be seen in Figure 5.17. Low jitter is desirable in all transmitter circuits, and so the link between delay time and bandwidth introduces an additional limit on the tuning range. This limit will depend on the amount of transmit jitter that can be tolerated.

Alternatively, we could consider that the transmit filter is constrained to a certain maximum number of taps. As the number of taps increases, the delay line jitter also increases. This constraint puts a practical limit on the length of the FIR filter that can be used.

In tapped delay-line filters such as the one considered in this chapter, this bandwidth reduction imposes a theoretical limit on the useful tunability of a delay cell. Although the individual delay cells may have a wider tunable range, for delays above a certain value the bandwidth will be too low to be of use.

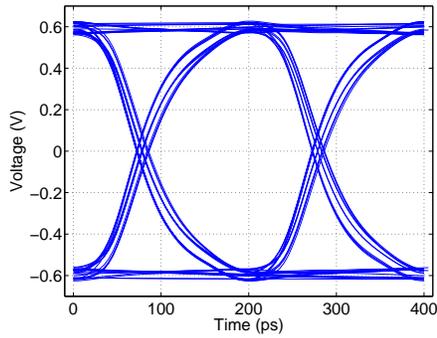
5.4 Test Chip Results

The equalizer circuit was implemented in IBM's 0.13- μm CMOS process, however testing of the circuit was not successful. A short timeline between design kit acquisition and tapeout deadline allowed some errors to go unnoticed. This tight timeline was caused by unforeseen delays in concluding legal agreements between IBM, CMC, MOSIS, and the University of Toronto regarding access to IBM's proprietary design kit.

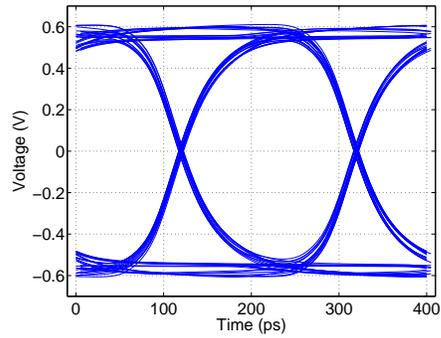
These errors will be rectified and a second test chip produced in the near future. A die photo of the test chip is shown in Figure 5.18.

5.5 Summary

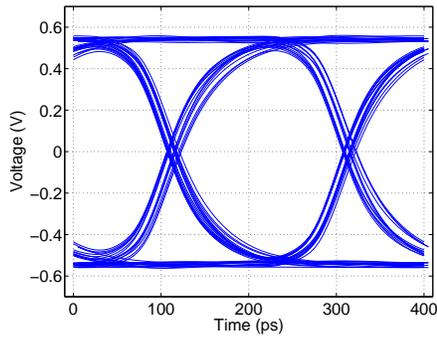
This chapter has described the design of an equalizer for a chip-to-chip communication link. It is designed to be flexible so that it can serve as a platform to test different transmit filters with different fractional spacings. The equalizer consists of a delay line comprised of eight delay cells and eight output driver cells, one for each delay cell. To increase the flexibility of this chip as a test platform, an important characteristic of the delay cell is its tunability range. Several possible cells were considered before settling



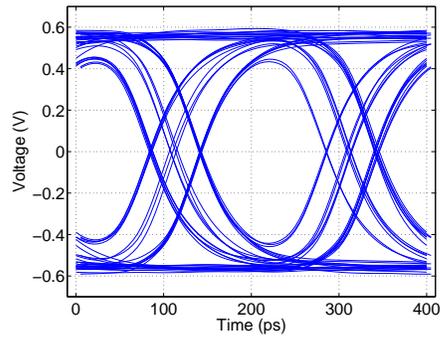
(a) Regular pulse shape using the first delay cell and minimum delay



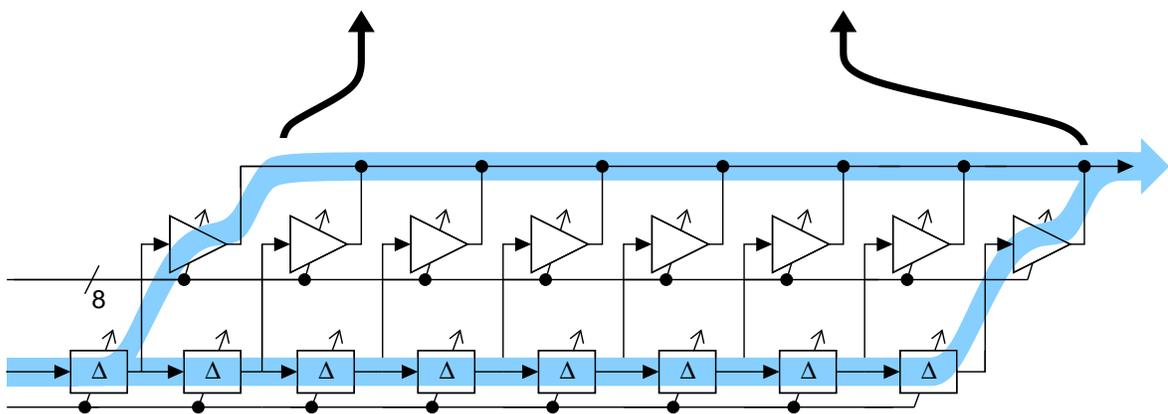
(b) Regular pulse shape using the eighth delay cell and minimum delay



(c) Regular pulse shape using the first delay cell with a longer delay



(d) Regular pulse shape using the eighth delay cell with a longer delay



(e) Transmitter overview: two possible signal paths highlighted

Figure 5.17: Spectre simulation at 5 Gb/s with a PRBS sequence of length $2^7 - 1$. When longer delays are used the bandwidth of the delay cell decreases, increasing jitter in the delay cells farther down the chain.

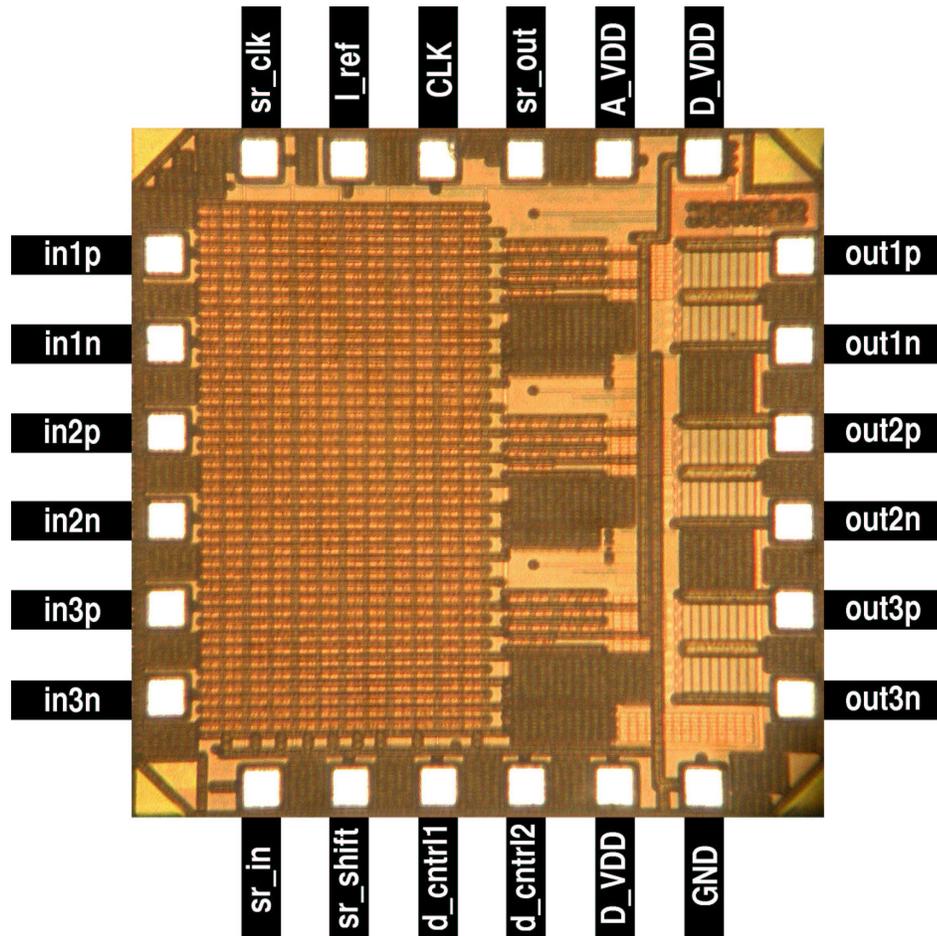


Figure 5.18: Photomicrograph of the test chip in 0.13- μm CMOS. The die dimensions are 1.5 mm \times 1.5 mm.

on the self-biased symmetric-load cell. The equalizer circuit was implemented in IBM's 0.13- μm CMOS process, but was unsuccessful. A second test chip will be produced in the near future.

Bit Rate	5 Gb/s
Number of filter taps	8
Tap Weight Resolution	4 bits
Delay Cell	Self-biased, symmetric load
Delay Tuning Range	35 ps–200 ps
Power Dissipation	70 mW
Supply Voltage	1.2 V
Transmitter Area	$310 \mu\text{m} \times 260 \mu\text{m}$
Technology	0.13- μm CMOS
	8 metal layers
	$f_T = 65 \text{ GHz}$

Table 5.2: Simulated transmitter characteristics.

6 Conclusion

6.1 Conclusion

Designing an efficient chip-to-chip link means more than simply maximizing the bandwidth of a single channel. Differential serial interfaces can transmit the highest bit rates, but limit the total chip-to-chip bandwidth to that of a single channel. To maximize the aggregate bit rate of the link in a pin-constrained chip, it might be beneficial to consider a single-ended rather than a differential signaling scheme. Or a bidirectional scheme might be preferable to a unidirectional one.

However, both single-endedness and bidirectionality increase susceptibility to noise and decrease the achievable bit rate for a single channel. To enhance the usefulness of these schemes, their limitations need to be dealt with. In that vein, this thesis proposes the use of pulse shaping to minimize each channel's impact on its neighbours. Similar techniques have been used in DSL communication where crosstalk between twisted pairs in a bundle is often severe. While pulse shaping has been used in chip-to-chip communication to remove ISI, pulse shapes are typically chosen without considering their effect on crosstalk noise.

Simulations were performed to find the optimal transmitted pulse shape from the point of view of maximum eye opening at the receiver. The measured step responses of the through channel and crosstalk channel were convolved with a sample data pattern and a set of candidate transmitted pulse shapes. The candidates were then compared based on a figure of merit, the ratio of the crosstalk-free eye opening to the maximum possible crosstalk, or E2C. For each $(Taps_{total}, Taps_{perUI})$ pair the optimal tap weights were chosen, and the resulting E2C plotted on a contour plot.

These simulations show the benefit of using a multi-tap, fractionally-spaced trans-

mit filter. They also show that the optimal pulse shape changes depending on whether crosstalk is included or neglected. To verify that the simulation results are applicable to the real world, a proof-of-concept was performed using a ParBERT as a stand-in for an on-chip equalizer. Measurement results showed that crosstalk-aware selection of tap weights led to a two order-of-magnitude improvement in BER over tap weights chosen solely to remove ISI.

Finally, an equalizer chip was designed in IBM 0.13- μm CMOS to serve as a test platform for the ideas explored in this thesis. The key performance targets were delay cell tunability and speed. The equalizer was designed as an eight-tap, fractionally-spaced FIR filter with each filter tap digitally programmable. This structure allows it to be configured in various ways in order to verify that the E2C figure of merit used in Chapter 3 correlated well with better BER performance in operation.

6.2 Suggestions for Future Work

Avenues of further research in this topic fall into three categories: circuit level, system level, and future goals.

General circuit levels goals are reducing power dissipation and area. Designing an equalizer for a wide parallel link implies that each chip has many copies of the same I/O cell. To reduce the overhead of these cells they need to be made power- and area-efficient. In addition, the delay line of the FIR filter is a critical block that determines the output jitter of the entire equalizer. To reduce jitter the delay cells could be powered by a voltage regulator, so the delay cells would be required to run at a lower supply voltage. Also, the delay cell control voltage could be set with a delay-locked loop (DLL). This DLL would lock to an external reference clock in order to provide the right delay with no manual adjustment.

In terms of system level improvements, making the transmit-side equalizer adaptive could be implemented by sending data on a backchannel from the receiver back to the equalizer. While many channels are fixed once implemented, some channels such as multidrop RAM busses can change when daughtercards are added or removed.

State of the art differential, serial links have been shown at bit rates of 20 Gb/s

over short backplane channels [1], however the fastest single-ended, parallel links only operate around 4 Gb/s [12]. Future work should address extending the bit rate of these links towards 10 Gb/s.

A Aggregate Data Rate Derivations

As stated in Chapter 1, several different signalling schemes are possible:

- single-ended vs. differential
- serial vs. parallel
- unidirectional vs. bidirectional
- 2-PAM vs. 4-PAM

Each of these choices modifies Equation 1.1 in a certain way. For example, a single-ended signaling scheme uses one pin per channel and so the equation becomes:

$$\text{aggregate bit rate} = \frac{\left(\frac{\text{bit rate}}{\text{channel}}\right)}{\left(1 \frac{\text{pin}}{\text{channel}}\right)} \times \left(\frac{\text{pins}}{\text{chip}}\right) \quad (\text{A.1})$$

It is clear from (A.1) that a single-ended signaling scheme must only achieve half the bit rate per channel of a differential scheme in order for it to be equally preferable from an aggregate bit rate standpoint. Similarly, the bit rate per channel of a bidirectional scheme need only be half that of a unidirectional scheme in order to be comparable, as shown in (A.2).

$$\text{aggregate bit rate} = \left(\frac{\text{bit rate}}{\text{direction}}\right) \times \left(2 \frac{\text{directions}}{\text{pin}}\right) \times \left(\frac{\text{pins}}{\text{chip}}\right) \quad (\text{A.2})$$

When evaluating different modulation schemes such as 2-PAM and 4-PAM, the equation becomes:

$$\text{aggregate bit rate} = \left(\frac{\text{symbol rate}}{\text{pin}}\right) \times \left(\frac{\text{bits}}{\text{symbol}}\right) \times \left(\frac{\text{pins}}{\text{chip}}\right) \quad (\text{A.3})$$

B Simulation Data

This appendix contains the results of the MATLAB simulations used to produce the optimal transmit filter tap weights.

B Simulation Data

$Taps_{total}$	$Taps_{perBit}$	Eye Opening (mV)	Filter tap weights					
1	1	26.8166	1					
1	2	26.8166	1					
1	3	26.8166	1					
1	4	26.8166	1					
1	5	26.8166	1					
1	6	26.8166	1					
<hr/>								
2	1	28.518	-0.25	1.25				
2	2	33.8236	1.25	-0.25				
2	3	33.8635	1.75	-0.75				
2	4	34.4717	1.75	-0.75				
2	5	32.6515	1.75	-0.75				
2	6	30.7205	1.75	-0.75				
<hr/>								
3	1	28.518	-0.25	1.25	0			
3	2	34.5746	1	0.25	-0.25			
3	3	35.0437	0.25	1.25	-0.5			
3	4	35.5943	0.25	1.75	-1			
3	5	35.6927	0.5	1.5	-1			
3	6	35.661	0.5	1.75	-1.25			
<hr/>								
4	1	35.4566	-0.25	-0.25	1.75	-0.25		
4	2	39.2295	1.75	-1.25	0.75	-0.25		
4	3	35.7174	1.75	-1	0.5	-0.25		
4	4	35.9721	0.5	1.5	-1.25	0.25		
4	5	36.8109	0.5	1.75	-1.5	0.25		
4	6	35.8598	0.5	0	1.75	-1.25		
<hr/>								
5	1	40.3674	-0.25	0	1.75	-0.25	-0.25	
5	2	42.0138	1.75	0.5	-1.5	1	-0.75	
5	3	39.062	1.75	-0.5	-0.75	1	-0.5	
5	4	38.4617	1.75	0	-1.75	1.5	-0.5	
5	5	37.805	-0.25	1.25	-0.75	1.75	-1	
5	6	37.4761	-0.5	1.5	-0.75	1.75	-1	

Table B.1: Optimization data for a chip-to-chip link with no crosstalk

B Simulation Data

$Taps_{total}$	$Taps_{per\ Bit}$	E2C	Filter tap weights						
1	1	4.29409	1						
1	2	4.29409	1						
1	3	4.29409	1						
1	4	4.29409	1						
1	5	4.29409	1						
1	6	4.29409	1						
2	1	4.29409	1	0					
2	2	4.57352	0.75	0.25					
2	3	4.65338	0.5	0.5					
2	4	4.68176	0.75	0.25					
2	5	4.62267	0.5	0.5					
2	6	4.55178	0.5	0.5					
3	1	4.29409	1	0	0				
3	2	5.98612	0.75	0.5	-0.25				
3	3	4.99934	0.75	0.5	-0.25				
3	4	4.80609	-0.25	0.75	0.5				
3	5	4.7613	0.5	0.25	0.25				
3	6	4.64901	0.5	0	0.5				
4	1	4.29409	1	0	0	0			
4	2	6.05604	-0.25	0.75	0.75	-0.25			
4	3	5.78836	0.75	0.25	0.25	-0.25			
4	4	5.84192	1.25	-1	1.5	-0.75			
4	5	5.09919	1.25	-1	1.25	-0.5			
4	6	4.67516	1	-0.75	1.75	-1			
5	1	4.29409	1	0	0	0	0		
5	2	6.7705	-0.25	0.75	0.5	0.25	-0.25		
5	3	6.26198	-0.25	0.75	0.25	0.5	-0.25		
5	4	6.07959	-0.25	1.5	-1	1.5	-0.75		
5	5	5.69639	0.75	0.25	-0.5	1.25	-0.75		
5	6	5.27015	1.5	-1	0	1.25	-0.75		
6	1	4.29409	1	0	0	0	0	0	
6	2	6.7705	-0.25	0.75	0.5	0.25	-0.25	0	
6	3	6.80401	-0.25	1	0	0.5	0	-0.25	
6	4	6.54596	-0.25	1.25	-0.5	0.75	0	-0.25	
6	5	6.1367	0.75	0.25	-0.25	0.5	0.25	-0.5	
6	6	6.07011	1.25	-0.5	-0.25	0.5	0.75	-0.75	

Table B.2: Optimization data for a chip-to-chip link including crosstalk

$Taps_{total}$	$Taps_{perBit}$	E2C	Filter tap weights			
1	1	2.15624	1			
1	2	2.15624	1			
1	3	2.15624	1			
1	4	2.15624	1			
2	1	2.43686	1.25	-0.25		
2	2	2.35677	1.25	-0.25		
2	3	2.21498	1.25	-0.25		
2	4	2.15624	1	0		
3	1	2.43686	1.25	-0.25	0	
3	2	2.48731	1	0.5	-0.5	
3	3	2.43187	1.25	0	-0.25	
3	4	2.36039	1	0.25	-0.25	
4	1	2.43686	1.25	-0.25	0	0
4	2	2.52731	-0.25	1	0.5	-0.25
4	3	2.45538	0.75	0.75	0	-0.5
4	4	2.46809	0.75	0.5	0	-0.25

Table B.3: Optimization data for a chip-to-chip link with crosstalk at 5Gb/s

$Taps_{total}$	$Taps_{perBit}$	E2C	Filter tap weights			
1	1	0.0793435	1			
1	2	0.0793435	1			
1	3	0.0793435	1			
1	4	0.0793435	1			
2	1	0.079939	1.25	-0.25		
2	2	0.232445	0.75	0.25		
2	3	0.198016	0.25	0.75		
2	4	0.262666	0.25	0.75		
3	1	0.193511	1.25	-1.25	1	
3	2	0.232445	0.75	0.25	0	
3	3	0.261969	1	-0.25	0.25	
3	4	0.262666	0.25	0.75	0	
4	1	0.346286	-0.25	1	0	0.25
4	2	0.367852	0.5	0.75	-0.75	0.5
4	3	0.288403	-0.25	0.5	0.5	0.25
4	4	0.279763	-0.75	1.25	0.25	0.25

Table B.4: Optimization data for a chip-to-chip link with crosstalk at 7.5Gb/s

References

- [1] Y. Hur, M. Maeng, C. Chun, F. Bien, H. Kim, S. Chandramouli, E. Gebara, and J. Laskar, "Equalization and near-end crosstalk (NEXT) noise cancellation for 20-Gb/s backplane serial I/O interconnections," *IEEE Trans. Microwave Theory Tech.*, vol. 53, no. 1, pp. 246–255, Jan. 2005.
- [2] P. Chiang, W. J. Dally, M.-J. E. Lee, R. Senthinathan, Y. Oh, and M. A. Horowitz, "A 20-Gb/s 0.13- μm CMOS serial link transmitter using an LC-PLL to directly drive the output multiplexer," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 1004–1011, Apr. 2005.
- [3] S. Gondi, J. Lee, D. Takeuchi, and B. Razavi, "A 10Gb/s CMOS adaptive equalizer for backplane applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2005, pp. 328–329.
- [4] Y. Tomita, M. Kibune, J. Ogawa, W. W. Walker, H. Tamura, and T. Kuroda, "A 10-Gb/s receiver with series equalizer and on-chip ISI monitor in 0.11- μm CMOS," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 986–993, Apr. 2005.
- [5] R. Farjad-Rad, K.-T. Ng, E. M.-J. Lee, R. Senthinathan, W. J. Dally, A. Nguyen, R. Rathi, J. Poulton, J. Edmondson, J. Tran, and H. Yazdanmehr, "0.622-8.0Gbps 150mW serial IO macrocell with fully flexible preemphasis and equalization," in *Symp. on VLSI Circuits Dig. Tech. Papers*, 2003, pp. 63–66.
- [6] J. E. Jaussi, G. Balamurugan, D. R. Johnson, B. Casper, A. Martin, J. Kennedy, N. Shanbhag, and R. Mooney, "8-Gb/s source-synchronous I/O link with adaptive receiver equalization, offset cancellation, and clock de-skew," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 80–88, Jan. 2005.
- [7] V. Stojanovic, A. Ho, B. W. Garlepp, F. Chen, J. Wei, G. Tsang, E. Alon, R. T. Kollipara, C. W. Werner, J. L. Zerbe, and M. A. Horowitz, "Autonomous dual-mode (PAM2/4) serial link transceiver with adaptive equalization and data recovery," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 1012–1026, Apr. 2005.

- [8] K. Chang, S. Pamarti, K. Kaviani, E. Alon, X. Shi, T. Chin, J. Shen, G. Yip, C. Madden, R. Schmitt, C. Yuan, F. Assaderaghi, and M. Horowitz, "Clocking and circuit design for a parallel I/O on a first-generation CELL processor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2005, pp. 526–527.
- [9] H. Higashi, S. Masaki, M. Kibune, S. Matsubara, T. Chiba, Y. Doi, H. Yamaguchi, H. Takauchi, H. Ishida, K. Gotoh, and H. Tamura, "A 5-6.4-Gb/s 12-channel transceiver with pre-emphasis and equalization," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 978–985, Apr. 2005.
- [10] K.-Y. K. Chang, J. Wei, C. Huang, S. Li, K. Donnelly, M. Horowitz, Y. Li, and S. Sidiropoulos, "A 0.4-4-Gb/s CMOS quad transceiver cell using on-chip regulated dual-loop PLLs," *IEEE J. Solid-State Circuits*, vol. 38, no. 5, pp. 747–754, May 2003.
- [11] F. Yang, J. H. O'Neill, D. Inglis, and J. Othmer, "A CMOS low-power multiple 2.5-3.125-Gb/s serial link macrocell for high bandwidth network ICs," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1813–1821, Dec. 2002.
- [12] S.-J. Bae, H.-J. Chi, H.-R. Kim, and H.-J. Park, "A 3Gb/s 8b single-ended transceiver for 4-drop DRAM interface with digital calibration of equalization skew and offset coefficients," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2005, pp. 520–521.
- [13] K.-L. J. Wong, H. Hatamkhani, M. Mansuri, and C.-K. K. Yang, "A 27-mW 3.6-Gb/s I/O transceiver," *IEEE J. Solid-State Circuits*, vol. 39, no. 4, pp. 602–612, Apr. 2004.
- [14] R. J. Drost and B. A. Wooley, "An 8-Gb/s/pin simultaneously bidirectional transceiver in 0.35- μ m CMOS," *IEEE J. Solid-State Circuits*, vol. 39, no. 11, pp. 1894–1908, Nov. 2004.
- [15] J.-H. Kim, S. Kim, W.-S. Kim, J.-H. Choi, H.-S. Hwang, C. Kim, and S. Kim, "A 4-Gb/s/pin low-power memory I/O interface using 4-level simultaneous bidirectional signaling," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 89–101, Jan. 2005.
- [16] B. Casper, A. Martin, J. E. Jaussi, J. Kennedy, and R. Mooney, "An 8-Gb/s simultaneous bidirectional link with on-die waveform capture," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2111–2120, Dec. 2003.

- [17] Y. Massoud, J. Kawa, D. MacMillen, and J. White, "Modeling and analysis of differential signaling for minimizing inductive cross-talk," in *IEEE Design Automation Conf. (DAC) Dig. Tech. Papers*, 2001, pp. 804–809.
- [18] C. Pelard, E. Gebara, A. J. Kim, M. G. Vrazel, F. Bien, Y. Hur, M. Maeng, S. Chandramouli, C. Chun, S. Bajekal, S. E. Ralph, B. Schmukler, V. M. Hietala, and J. Laskar, "Realization of multigigabit channel equalization and crosstalk cancellation integrated circuits," *IEEE J. Solid-State Circuits*, pp. 1659–1670, Oct. 2004.
- [19] S. Reynolds, P. Pepeljugoski, J. Schaub, J. Tierno, and D. Beisser, "A 7-tap transverse analog-FIR filter in $0.13\mu\text{m}$ CMOS for equalization of 10Gb/s fiber-optic data systems," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2005, pp. 330–331.
- [20] X. Lin, S. Saw, and J. Liu, "A CMOS $0.25\text{-}\mu\text{m}$ continuous-time FIR filter with 125 ps per tap delay as a fractionally spaced receiver equalizer for 1-Gb/s data transmission," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 593–602, Mar. 2005.
- [21] X. Lin, H. Lee, and J. Liu, "A continuous-time adaptive FIR equalizer with INV-AIL delay line for 2.5Gb/s data communication," in *IEEE Custom Integrated Circuits Conf. (CICC) Dig. Tech. Papers*, 2005, pp. 413–416.
- [22] J. W. Cook, R. H. Kirkby, M. G. Booth, K. T. Foster, D. E. A. Clarke, and G. Young, "The noise and crosstalk environment for ADSL and VDSL systems," *IEEE Commun. Mag.*, pp. 73–78, May 1999.
- [23] K. B. Song, S. T. Chung, G. Ginis, and J. M. Cioffi, "Dynamic spectrum management for next-generation DSL systems," *IEEE Commun. Mag.*, pp. 101–109, Oct. 2002.
- [24] M. Maymandi-Nejad and M. Sachdev, "A digitally programmable delay element: design and analysis," *IEEE Trans. VLSI Syst.*, vol. 11, no. 5, pp. 871–878, Oct. 2003.
- [25] J. G. Maneatis and M. A. Horowitz, "Precise delay generation using coupled oscillators," *IEEE J. Solid-State Circuits*, vol. 28, no. 12, pp. 1273–1282, Dec. 1993.