

A PATTERN-GUIDED ADAPTIVE EQUALIZER IN 65NM CMOS

by

Shayan Shahramian

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

Copyright © 2011 by Shayan Shahramian

A PATTERN-GUIDED ADAPTIVE EQUALIZER IN 65NM CMOS

Shayan Shahramian

Master of Applied Science, 2011

Graduate Department of Electrical and Computer Engineering

University of Toronto

Abstract

This thesis presents the design, implementation, and fabrication of a pattern-guided equalizer in a 65nm CMOS process. By counting the occurrence of 6 out of 16 4-bit patterns in the received data and utilizing their spectral content, the signal is equalized separately at f_N and $f_N/2$, where f_N is half the bit rate. The design was packaged using a 64 pin Quad Flat No leads (QFN) package. Two different channels were used and the equalizer was able to open the eye for both 13dB and 17dB of attenuation at the Nyquist frequency. The adaptation performance was determined by measuring the vertical and horizontal eye openings for all possible equalizer coefficients. Measured results at 6Gb/s confirm that the adaptation engine opens a closed eye to within 2.6% of optimal vertical opening and 7% of optimal horizontal eye opening while consuming 16.8mW from a 1.2V supply.

Acknowledgements

I would like to thank Professor Sheikholeslami for his continuous help and support throughout this project.

I would like to thank my committee members Professor Chan Carusone, Professor Johns, and Professor T.J Lim for their feedback which has enhanced the quality of this thesis.

I want to thank my family for their unconditional love and support. I want to thank my father for his advice and patience with me throughout this process. I would like to thank my brother, Shahriar, for being my role model and always spending the time to help me. Needless to say, without my family it would have been impossible to accomplish any of this.

I would like to thank Clifford Ting for his invaluable help with the final stages of the project. Cliff, without your help, it would have been impossible to finish this project on time. I would like to greatly thank Kentaro with his help with testing the chip. Thanks for being a solid lab manager and a great friend. I want to thank Hemesh for his help with the PCB design, your footprints were amazing!

I want to thank my closest friends Sarmad Sufi, Safeen Huda, Derek Zhou, Siamak Sarvari, and Hamid-Reza Khazaei. Going through this entire journey in engineering was completely worthwhile for just meeting reliable friends like you. You guys are an extension of my family now and I look forward to the next chapter of all of our lives. Derek, you have been a great friend throughout the entire process. Sarmad, hopefully you will be a CEO soon and can be our industry contact. Safeen, thanks for all the

long discussions about reoccurring topics and the drives home after long days. Siamak, thanks for putting in time to help me out and being a good gym buddy. Hamid, I hope you are enjoying your work and living the dream.

I would like to thank Alain Rousson and Andy Zhang for general comradery and squash games. I would like to thank Dustin Dunwell for being patient with us in the lab and helping me with tutorial prep. I would like to thank Mike Bichan for “lending” me his FR4 channels. I would like to thank Behrooz Abiri for his discussions regarding our tapeouts. I would like to thank Ravi Shivnaraine for his help with testing. Thanks to Meysam for not unleashing the express in the office. A huge thanks goes to Ricardo Aroca for his advice and help with my project. Thanks to Alex Tomkins for always willing to help out no matter how busy he was. I would like to thank Karim and Hamed for their random jokes. I would like to thank Rocky (Yunzhi) Dong for being the professor of BA5000 and helping all of us. I would like to thank Ali Taghvaei for teaching the DEEP course and the undergrad memories. I would like to thank Trevor Caldwell for his help with “understanding” delta-sigma ADCs and losing money in poker.

To the new comers to BA5000, I would like to thank Mario Milicevic for solving puzzles and the tester challenges. Sadegh Jalali, thanks for creating an even friendlier environment and I look forward to being in this office together for the next 10 years.

Finally I would like to thank Tina for supporting me and with your love helping me not lose sight of what’s really important. For being very understanding and considerate and doing whatever you could to make things easier for me.

Always remember, It’s just circuits.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Thesis Objectives | 2 |
| 1.3 | Thesis Outline | 2 |
| 2 | Background | 4 |
| 2.1 | Equalization | 4 |
| 2.2 | Clock and Data Recovery | 7 |
| 2.3 | Non-Data-aided equalization adaptation | 9 |
| 2.3.1 | Adaptive Equalization with slicer swing control | 9 |
| 2.3.2 | Spectrum balancing equalization | 11 |
| 2.3.3 | Improved Spectrum Balancing equalization | 11 |
| 2.4 | Data-aided equalization adaptation | 13 |
| 2.4.1 | Zero-Forcing adaptation | 13 |
| 2.4.2 | Bit-Error-Rate based adaptation | 14 |
| 2.4.3 | Filter pattern equalization | 15 |
| 2.5 | Summary | 15 |
| 3 | Pattern Guided Equalization | 17 |
| 3.1 | Basic Observation | 17 |
| 3.2 | Implementation Requirements | 19 |

| | | |
|----------|--|-----------|
| 3.3 | Digital Adaptation Controllers | 22 |
| 3.3.1 | Algorithm Implementation | 22 |
| 3.3.2 | Algorithm functional simulations | 23 |
| 3.4 | Slicer with shifted threshold | 24 |
| 3.5 | Equalizer Topology | 28 |
| 3.5.1 | Equalizer utilizing highpass filters (Topology 1) | 28 |
| 3.5.2 | Equalizer utilizing bandpass filters (Topology 2) | 29 |
| 3.5.2.1 | Bandpass Filter | 33 |
| 3.5.2.2 | VGA | 34 |
| 3.6 | Conclusions | 35 |
| 4 | Experimental and Simulation Results | 39 |
| 4.1 | Receiver layout and equipment setup | 39 |
| 4.2 | Blind clock operation | 42 |
| 4.3 | Adaptation Performance | 44 |
| 4.4 | Summary | 48 |
| 5 | Conclusion | 52 |
| 5.1 | Thesis Contributions | 52 |
| 5.2 | Future Work | 53 |
| 5.2.1 | Integration with blind DFE and CDR | 53 |
| 5.2.2 | Study of effects of blind vs locked CDR operation | 54 |
| 5.2.3 | Equalizing channels with poorly behaved frequency response | 54 |
| | References | 54 |

List of Tables

| | | |
|-----|---|----|
| 2.1 | Linear vs. Non-linear equalizer | 5 |
| 4.1 | Description of the pin-list | 41 |
| 4.2 | Comparison to state of the art | 48 |
| 5.1 | System performance summary | 53 |

List of Figures

| | | |
|------|--|----|
| 2.1 | Concept of linear equalization | 6 |
| 2.2 | Decision feedback equalizer operation | 6 |
| 2.3 | Timing diagram of CDR clock alignment | 8 |
| 2.4 | Simplified CDR block diagram | 8 |
| 2.5 | Adaptive equalizer with slicer swing control [6] | 10 |
| 2.6 | Adaptive equalizer with bandpass filters [10] | 10 |
| 2.7 | Ideal random binary data spectrum [11] | 11 |
| 2.8 | Block diagram of spectrum balancing equalization [11] | 12 |
| 2.9 | Block diagram of adaptive spectrum balancing equalization [12] | 12 |
| 2.10 | An implementation of a zero-forcing adaptation scheme [13] | 14 |
| 2.11 | Block diagram of Bit-Error-Rate based adaptation [14] | 14 |
| 2.12 | Filter pattern equalization [15] | 15 |
| 3.1 | Patterns of length four categorized based on frequency spectrum | 18 |
| 3.2 | Pattern attenuation through a channel | 19 |
| 3.3 | Simplified system block diagram | 20 |
| 3.4 | Conventional vs. Shifted threshold slicer | 21 |
| 3.5 | Implementation of pattern counters | 22 |
| 3.6 | Implementation of digital adaptation controllers | 24 |
| 3.7 | Timing diagram of the synthesized digital adaptation controllers | 25 |
| 3.8 | Circuit implementation of variable threshold slicer | 26 |

| | | |
|------|---|----|
| 3.9 | Simulated Output eye diagram of variable threshold shifter | 27 |
| 3.10 | Simulated amount of threshold shift in mV for each digital code | 27 |
| 3.11 | Equalizer topology utilizing highpass filters | 28 |
| 3.12 | Equalizer topology utilizing bandpass filters | 30 |
| 3.13 | Implementation of an active inductor | 31 |
| 3.14 | Inductance vs. Frequency | 32 |
| 3.15 | Quality factor vs. Frequency for the chosen inductor | 32 |
| 3.16 | Final implemented equalizer topology utilizing bandpass filters and VGAs | 33 |
| 3.17 | Circuit implementation of a bandpass filter | 34 |
| 3.18 | Frequency response of the bandpass filter centered at f_N | 35 |
| 3.19 | Circuit implementation of Variable Gain Amplifier | 36 |
| 3.20 | Post layout frequency response of the VGA for various gain settings . . . | 37 |
| 3.21 | Frequency response of the 4 VGAs and Bandpass filter at 3.75GHz . . . | 37 |
| 3.22 | Frequency response of the 4 VGAs and Bandpass filter at 1.875GHz . . . | 38 |
| | | |
| 4.1 | Chip die photo | 40 |
| 4.2 | Measurement Setup | 42 |
| 4.3 | PCB picture | 43 |
| 4.4 | Channel frequency response | 44 |
| 4.5 | Eye diagrams for 13dB channel | 46 |
| 4.6 | Eye diagrams for 17dB channel | 47 |
| 4.7 | 3-D contour for eye openings of 13dB channel | 49 |
| 4.8 | 3-D contour for eye openings of 17dB channel | 50 |
| 4.9 | Adaptation learning curve | 51 |

List of Acronyms

ADC Analog to Digital Converter

BER Bit-Error Rate

CDR Clock and Data Recovery

DFE Decision Feedback Equalizer

DFT Discrete Fourier Transform

Gb/s gigabits per second

HDMI high-definition multi-media interface

ISI Inter Symbol Interference

LA Limiting Amplifier

QFN Quad Flat No leads

SATA Serial Advanced Technology Attachment

SRF Self Resonance Frequency

PCIe Peripheral Component Interconnect Express

PRBS Pseudo-Random Binary Sequence

PVT Process, Voltage and Temperature

UI Unit Interval

VCO Voltage Controlled Oscillator

VGA Variable Gain Amplifier

1

Introduction

With the growth of the Internet, high definition videos and high performance gaming systems, the demand for faster data transmission is increasing. Industry standards such as high-definition multi-media interface (HDMI), Peripheral Component Interconnect Express (PCIe), and Serial Advanced Technology Attachment (SATA) continuously increase the required data rate, thereby making it more and more difficult to attain the desired speeds. New circuit innovations are required in order to keep up with the demand of higher data rates.

1.1. Motivation

In high-speed transceivers, the data is sent from the transmitter to a receiver through a communication channel. There are constantly advances in silicon technologies that allow us to increase the data rates at the transmitter and receiver. The improvement of the communication channels, however, do not follow the same aggressive improvements as the silicon technologies. As data rates increase, these channels exhibit frequency dependent loss. As a result, the broadband signal experiences different amounts of attenuation for different frequencies. This frequency dependent loss leads to Inter Symbol Interference (ISI) which causes bits to have an effect on previous or future bits. This is undesirable as we would like each bit transmitted to be independent of what has been transmitted before. This creates challenges for the data recovery circuits in the receiver.

To counteract the effects of the communication channel, equalization is used. Equalization is the process of reversing the effect of the channel on the data. There are two types of equalization, linear or non-linear equalization. Each of these approaches has its own advantages and disadvantages which will be described in Chapter 2. In this thesis the channels of interest are wired channels for backplane applications.

The characteristics of the communication channel is not always known prior to transmission. As a result, it is much more beneficial to have adaptive equalization. In this case, the receiver adapts the equalizer to compensate for the specific channel that the data is transmitted through. There are several different types of adaptive equalization which again have their own advantages and disadvantages and are described in Chapter 2.

1.2. Thesis Objectives

This thesis presents a new type of adaptation for linear equalization. The main objectives of the thesis are as follows:

- Provide a background and a critique on different types of equalization and adaptation schemes
- Propose a new adaptation algorithm that we call “pattern guided equalization”
- Test chip simulation, implementation, and measurement results to prove functionality

1.3. Thesis Outline

The remaining chapters of this thesis are organized as follows:

- Chapter 2 provides a background on linear and non-linear equalizers as well as several different adaptation schemes.

- Chapter 3 describes the proposed adaptive pattern guided equalization. The design choices as well as the final implementation are also discussed. Block level simulations are provided to demonstrate low-level functionality.
- Chapter 4 provides measurement results of the fabricated test chip.
- Chapter 5 concludes the thesis and provides the future directions for this work

2

Background

This chapter discusses a fundamental problem in high-speed receivers which is frequency-dependent signal attenuation. The data that is transmitted through a channel experiences some frequency-dependent loss due to the low-pass frequency nature of wireline channels. This will cause ISI which is the interference of past or future bits with the present transmitted bit. Ideally, we would like the transmitted bits to have no effect on their neighboring bits, hence ISI is undesirable. ISI will lead to eye closure, which makes data recovery difficult and will degrade the performance of the receiver. To compensate for this frequency-dependent loss, equalization is used to counteract the effects of the channel. Section 2.1 will discuss different types of equalization. Section 2.2 will give some background on Clock and Data Recovery (CDR). Sections 2.3 and 2.4 will discuss two categories of adaptive linear equalization.

2.1. Equalization

This section presents the concept of equalization. Equalization can be performed both at the transmitter and the receiver. This section will focus on receiver equalization. The two types of equalizers at the receiver that can be used to compensate for ISI are linear and non-linear equalizers. Linear equalizers are placed at the receiver front-end and provide high-frequency boost to compensate for channel loss, as shown in Figure 2.1. The advantages of linear equalizers are that they can work even with a completely closed

eye and they do not require a low Bit-Error Rate (BER) to function. The disadvantages are that they also amplify high-frequency noise. Non-linear equalizers require a slightly open eye to function properly, however, they do not boost high-frequency noise.

A type of non-linear equalizer called a Decision Feedback Equalizer (DFE) is shown in Figure 2.2. Figure 2.2(a) shows the effect of frequency dependent loss of the channel in the time domain. As mentioned earlier, this causes ISI which in turn means that a pulse of 1 Unit Interval (UI) will spread and affect bits transmitted in other UIs, shown as h_1 and h_2 . h_1 is known as the first post-cursor ISI and h_2 is known as the second post-cursor ISI. Based on the channel, the current pulse might effect any number of subsequent or previous bits. To counteract this effect on subsequent bits, the topology shown in Figure 2.2(b) is used. The system samples the current UI and subtracts it from the incoming signal by a scaling factor. The topology shown in Figure 2.2(b) will allow us to remove the first post-cursor ISI, by subtracting that value from the next bit, removing its effect. The DFE system can be extended to any number of taps, by adding more delay elements after the slicer. The feedback loop leads to timing problems in the equalizer. The main issue is that the signal needs to go through a subtracter and a slicer within 1 UI to be subtracted at the appropriate time. As data rates increase, it becomes more and more difficult to meet this timing constraint.

Another problem with the DFE architecture is the error propagation. If a bit is detected incorrectly by the slicer, the wrong value will be subtracted from the next bit, and this will propagate for each of the DFE taps. Table 2.1 summarizes the advantages and disadvantages of both types of equalization.

| Type of equalizer | Advantages | Disadvantages |
|-------------------|-----------------------|--|
| Linear | Works with a high BER | Amplifies HF noise |
| Non-linear (DFE) | No HF noise boost | Timing difficulty Requires a low BER Error propagation |

Table 2.1: Linear vs. Non-linear equalizer

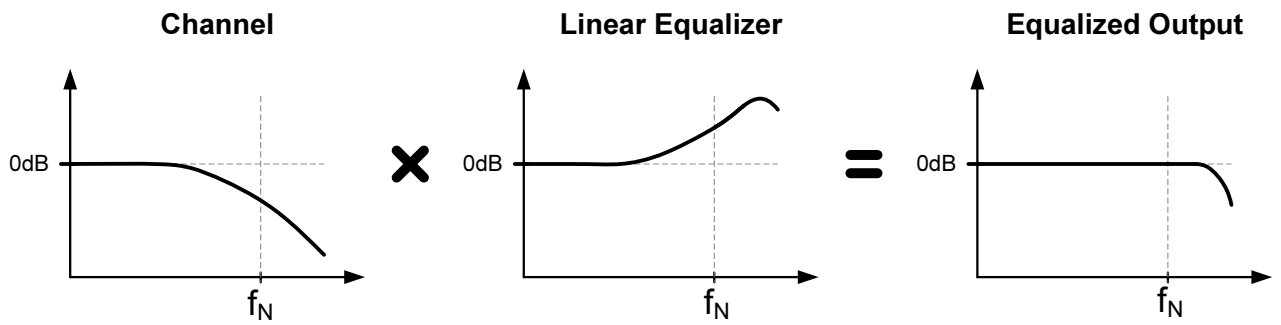


Figure 2.1: Concept of linear equalization

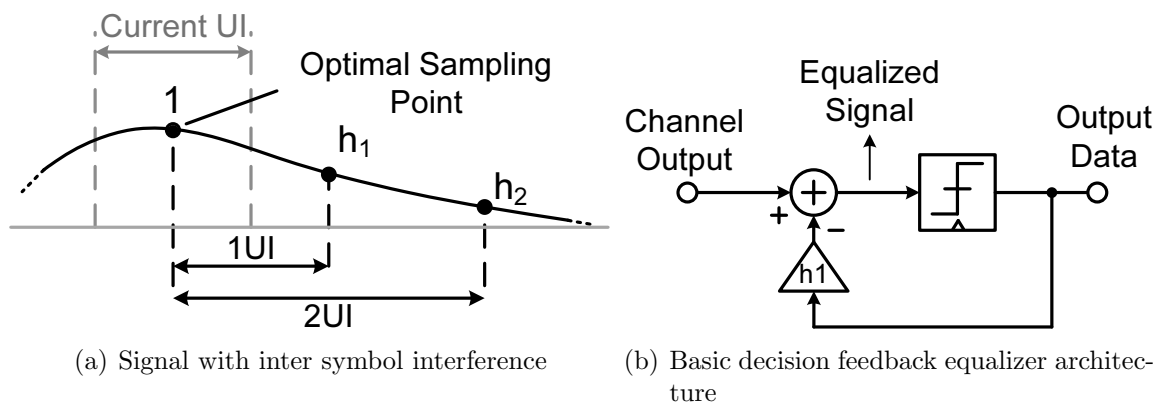


Figure 2.2: Decision feedback equalizer operation

Ultimately, due to their advantages and disadvantages most receivers implement both types of equalizers. Together they alleviate many of the issues; the linear equalizer is used to open the eye to a degree where the BER is lowered sufficiently such that the DFE can operate properly. There are other advances made to DFEs to mitigate the disadvantages associated with them [1–4] but they will not be discussed in this thesis.

So far, we have discussed equalization for pre-determined channel properties, however, the characteristic of the channel are not often known before startup and it is desirable to have the system adaptively determine how much equalization is required. There are two types of linear equalization adaptation. Data-aided and non-data-aided equalization. Data-aided schemes require the recovered data in order to make a decision about the amount of equalization required. Non-data-aided schemes utilize information about the signal's frequency content for equalization. Following Section 2.2 which provides a quick background on CDR, Section 2.3 describes several approaches to non-data-aided adaptive equalization and Section 2.4 describes data-aided adaptation schemes.

2.2. Clock and Data Recovery

The ultimate goal of high speed transceivers is to correctly recover the transmitted bits at the receiver. It is possible to transmit the phase aligned clock and data through two separate channels. However, the different delays of the channels makes it difficult to maintain the phase alignment of the data and clock. It is then possible that there would be a large phase difference between the transmitted clock and the data at the receiver. As a result, the receiver needs to determine the proper phase of the clock to sample the data. Figure 2.3 shows the concept behind the receiver operation. The clock's phase needs to be aligned such that its rising edge is in the middle of the UI where there is the largest eye opening. The recovered clock is then used to sample the incoming data and recover the transmitted bits.

Figure 2.4 shows the simplified block diagram of a CDR. The CDR will need to adjust

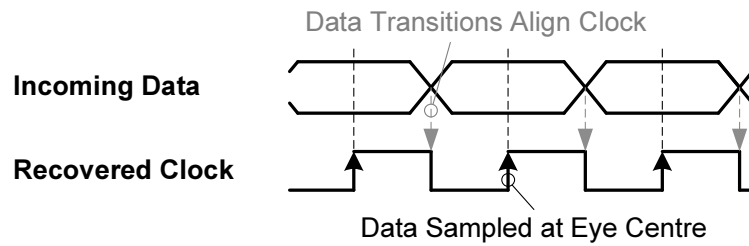


Figure 2.3: Timing diagram of CDR clock alignment

the phase of the clock to sample the data in the center of the UI. The phase detector uses the data transitions to generate a phase error Φ_{Error} which is related to the sampling point of the clock. The Φ_{Error} signal is low-pass filtered and used to control a Voltage Controlled Oscillator (VCO) which can subsequently adjust the phase of the receiver clock [5]. It is important to note that it is possible for the transmitter and receiver to have clocks operating at different frequencies. Even if these frequencies are designed to match, it can be difficult to ensure that after fabrication they operate at the same rate. As a result, the CDR requires a frequency detector to adjust the frequency of the VCO in the receiver to match that of the transmitter as shown in Figure 2.4.

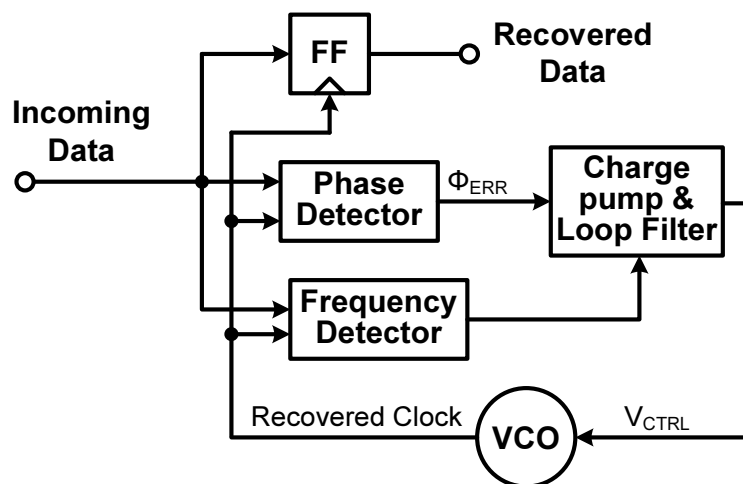


Figure 2.4: Simplified CDR block diagram

2.3. Non-Data-aided equalization adaptation

There are many different types of non-data-aided equalization that have been proposed in the past. The different approaches are explained in Sections 2.3.1, 2.3.2, and 2.3.3.

2.3.1. Adaptive Equalization with slicer swing control

Figure 2.5 shows an adaptive equalizer that uses slicer swing control [6]. To determine the amount of high-frequency boost required, this approach compares the high frequency content of the output of the equalizer with the output of a slicer in the receiver. Assuming the data has been recovered correctly, the output of the slicer should have an identical frequency content to that of the transmitted data. The only problem is the voltage swing at the output of the slicer in the receiver might differ from the transmitter. Older techniques in [7], [8], [9], do not employ any fix to this issue. This could lead to the false conclusion that the system requires equalization where it might only require amplification. To correct for this problem, the low frequency content of the equalized data and the slicer are also compared and the slicer swing is adjusted to ensure these are equal. This will guarantee that the difference in high frequency content is purely due to the channel loss.

Figure 2.6 shows an approach utilizing bandpass filters instead of high-pass filters to guide equalization [10]. This approach is useful when the frequency of the equalizer peaking range is narrow and needs to be accurately controlled. The design of the loops needs to be carefully considered to ensure that the loops do not interfere. The time constants were designed such that the slicer loop control would converge much faster than the high-frequency boost control.

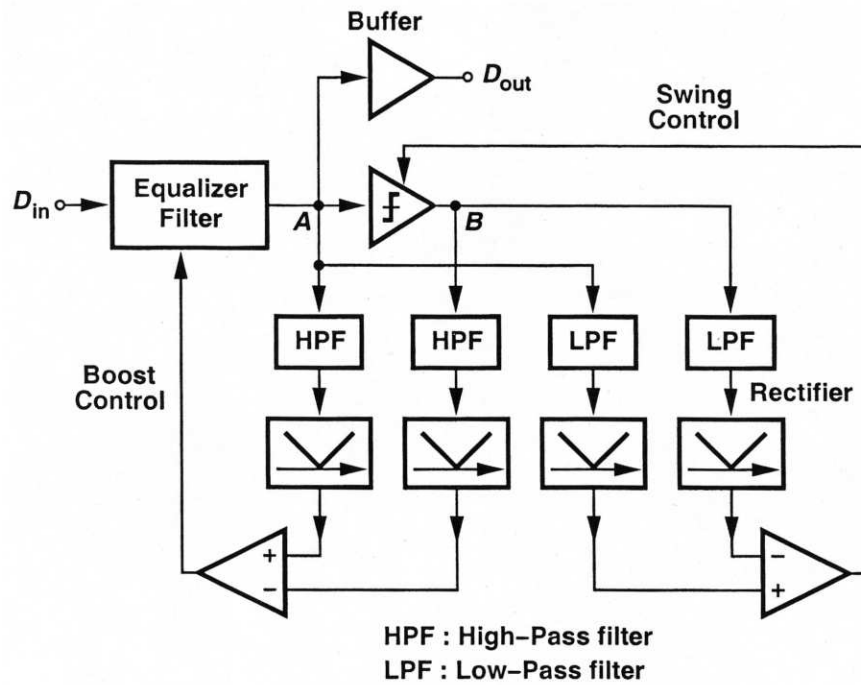


Figure 2.5: Adaptive equalizer with slicer swing control [6]

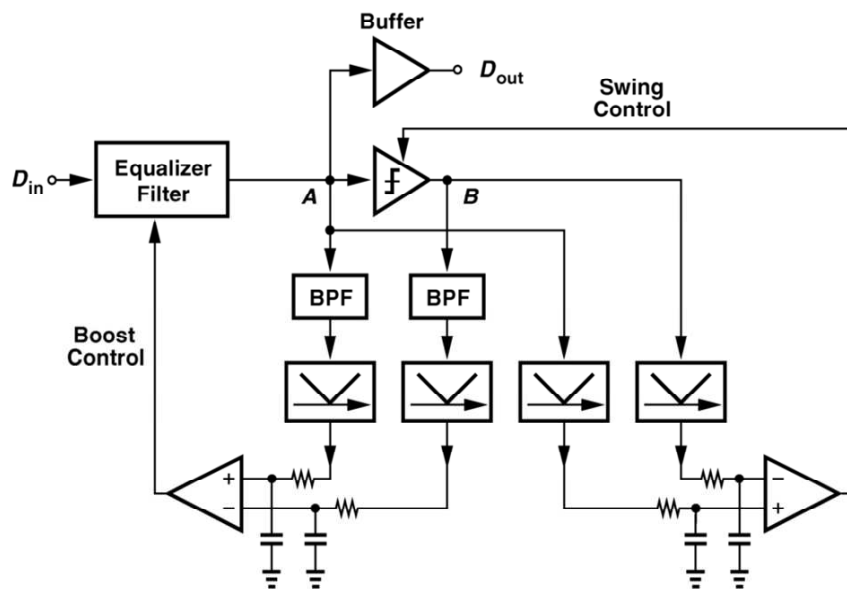


Figure 2.6: Adaptive equalizer with bandpass filters [10]

2.3.2. Spectrum balancing equalization

Figure 2.7 (a) shows the spectrum of ideal random binary data. With a known spectrum, a frequency can be determined which equally splits the spectrum into low and high frequency components. By comparing the power difference in the two portions of the spectrum, a decision can be made regarding equalization. Figure 2.7 (b) shows how the spectrum will differ if there is more low or high frequency content at the receiver.

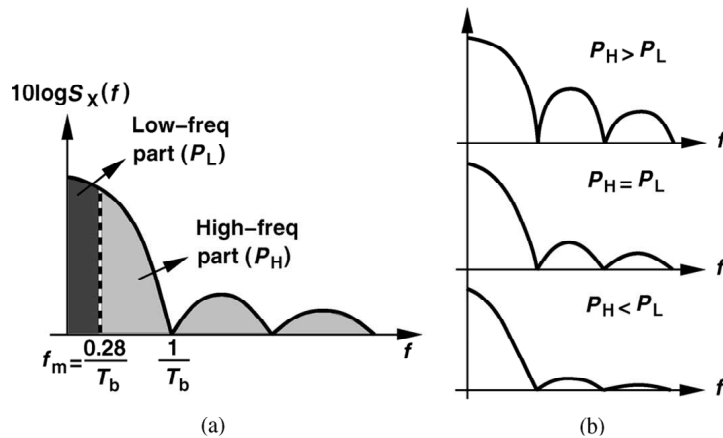


Figure 2.7: Ideal random binary data spectrum [11]

Figure 2.8 shows the block diagram of a system that implements an adaptive equalizer based on the spectrum balancing method [11]. If there is a lack of power in the high frequency portion of the data spectrum, the equalizer gains are increased. Similarly, a high amount of low frequency data would lower the equalizer gain. High-pass and low-pass filters separate the frequency content of the received data into two portions. The rectifier compares the power difference between the low and high frequencies and decides whether to increase or decrease equalization.

2.3.3. Improved Spectrum Balancing equalization

The main issue with the spectrum balancing equalization presented in [11] is the assumption that there is ideal random binary data being transmitted. If the data has a large amount of low frequency content such that it no longer resembles a $\text{sinc}^2(f)$, that scheme

of poorly behaved channels. This concept is explained further in Chapter 3.

The main issue with approaches that require analog blocks (rectifier, high-pass and low-pass filters) is the Process, Voltage and Temperature (PVT) variations. If an entirely digital algorithm could be developed, it would not only alleviate PVT constraints but also have the benefits of scalability across different process nodes.

2.4. Data-aided equalization adaptation

Data-aided equalization utilizes the recovered data to control the high-frequency boost. This allows for digital adaptation implementations that do not require analog filter design and are more robust to PVT variations.

2.4.1. Zero-Forcing adaptation

One of the traditional adaptive equalization algorithms is known as zero-forcing. Figure 2.10 shows a simplified block diagram of an implementation of a zero-forcing adaptation scheme. y_k is the received signal which has been attenuated by the channel. \tilde{d}_k is the equalized signal. The detector performs the task of data recovery, and hence \hat{a}_k is ideally the recovered data. The variable d_k represents the ideal equalized data and is created from \hat{a}_k using g_k . The shift register is used to have access to previous data bits which are the main contributors of ISI. The correlator looks at the difference between the ideal equalized data (d_k) and the equalized data (\tilde{d}_k) and correlates it with the past recovered bits in order to control the equalizer coefficients (c) to equalize the signal. If there are enough previous bits and the equalizer has sufficient number of taps, e_k will be forced to zero; this indicates that optimal equalization has been achieved. The main disadvantage of this approach is the requirement for the large number of taps for the equalizer which leads to added area, power consumption and complexity. Zero-forcing is also not optimal in terms of noise performance, the only criteria is to remove ISI even if this leads to an increase in the amount of noise.

2.4.3. Filter pattern equalization

Another approach to data-aided-equalization is presented in [15]. Figure 2.12 shows the concept behind filter pattern equalization. The data is sampled both at the center and the edge of the eye. By looking for specific patterns and the sliced values of the data at transitions, decisions are made regarding equalization. For example, if a 0001 pattern was transmitted, the system would sample D2 and B3 as shown in Figure 2.12 and based on the detected values would either increase or decrease equalization. The main issue with this approach is that if the required filter pattern does not occur, the equalizer would halt and the equalization time would suffer greatly.

Chapter 3 presents an adaptation architecture which combines the benefits of non-data-aided and data-aided schemes. By separating data into several different frequency bins, poorly behaved channels can be equalized. By using a digital adaptation algorithm, PVT variations will be minimized and the design will allow for easy scalability.

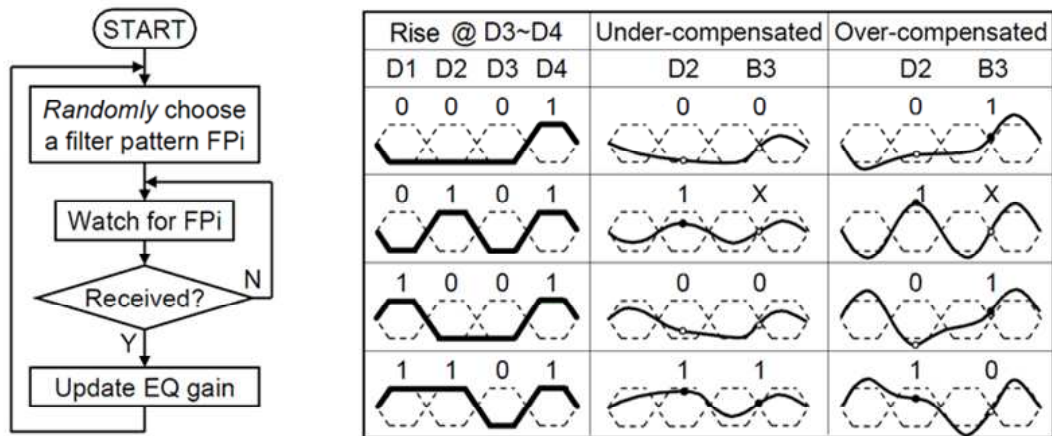


Figure 2.12: Filter pattern equalization [15]

2.5. Summary

Linear and non-linear equalizers were introduced as a solution to frequency-dependent channel loss. Two categories of linear equalizers were introduced as data-aided and non-

data-aided equalization. Previous works relating to both types of linear equalizers were presented.

3

Pattern Guided Equalization

This chapter will outline the proposed equalization scheme. First, a few observations regarding data patterns and their frequency spectra will be presented. Second, the implementation requirements of the design will be introduced. Third, the implementation details of the digital adaptation engine and equalizer will be presented including block level simulation results.

3.1. Basic Observation

By observing some properties of 4-bit data patterns (i.e. 0000, 0001, ... , 1111) an entirely digital equalization algorithm can be developed. The Discrete Fourier Transform (DFT) equation, 3.1, is applied to each of the 4-bit patterns. This will result in a 4-point DFT of the signal, which has frequency content from $[0, 2\pi]$. The frequency of the DFT samples can be obtained from equation 3.2. From this equation it can be observed that the 4-point DFT can give us information about the frequency content at 0, $1/4T$, and $1/2T$. This corresponds to DC, $f_N/2$, and f_N , where f_N is half of the bit rate.

Figure 3.1 shows several different 4-bit data patterns along with their frequency spectra. It can be derived from the DFT that Type 1 patterns (0101, 1010) have frequency content at f_N , and Type 2 patterns (0011, 0110, 1001, 1100) have frequency content at $f_N/2$. The remaining patterns have identical signal power at f_N and $f_N/2$. In our proposed equalization scheme, we would like to use Type 1 and Type 2 patterns to guide

equalization. This would require an equalizer which has independent gain control at the required frequencies.

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{2\pi i}{N} kn} \quad k = 0, \dots, N-1 \quad \& \quad N = 4 \quad (3.1)$$

$$f_k = \frac{k}{NT} \quad \text{for } 0 \leq k \leq \left\lceil \frac{N-1}{2} \right\rceil \quad \text{where } T = \text{sampling period} \quad (3.2)$$

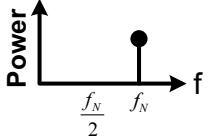
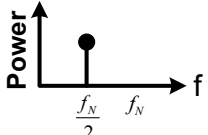
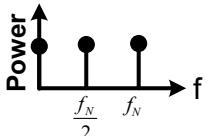
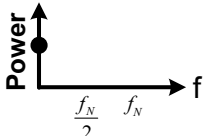
| | Data Patterns | Power Spectrum | Guide Equalization |
|---------------|--|--|--|
| Type 1 | 1010, 0101 |  | Count these to guide equalization at f_N |
| Type 2 | 0011, 0110, 1001, 1100 |  | Count these to guide equalization at $f_N/2$ |
| Type 3 | 0001, 0010, 0100, 1000, 1110, 1101, 1011, 0111 |  | No action |
| Type 4 | 0000, 1111 |  | No action |

Figure 3.1: Patterns of length four categorized based on frequency spectrum

Figure 3.2 shows the basic concept behind the proposed equalization scheme. Due to the low-pass nature of the channel, patterns with higher frequency content (shown at the top, Type 1) are more severely attenuated than patterns with lower frequency content (shown at the bottom, Type 2). As a result Type 1 patterns have an amplitude that is smaller than Type 2 patterns ($B < C$). Consequently, more equalization is required at f_N vs. $f_N/2$. Ideally, the equalizer would ensure that both Type 1 and Type 2 patterns have the same amplitude as the transmitted signal. Two coefficients (as opposed to only one) provide greater flexibility for high frequency compensation as would be necessary

for poorly behaved channels.

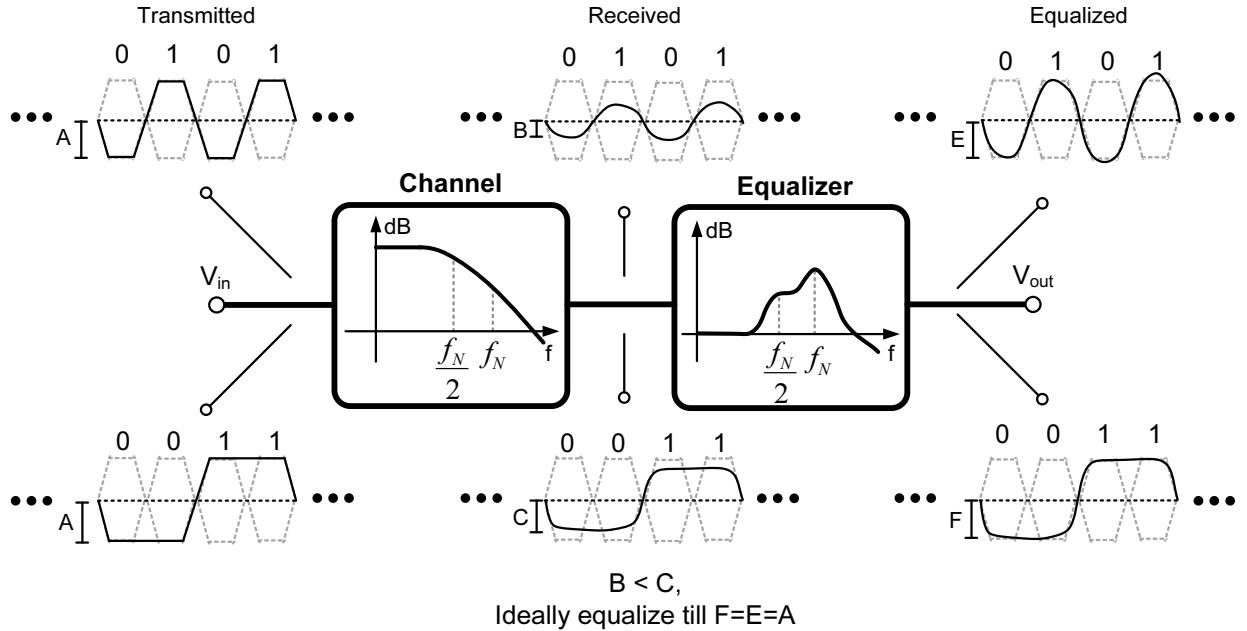


Figure 3.2: Pattern attenuation through a channel

3.2. Implementation Requirements

This section outlines the major implementation blocks required for the design. Following a system block diagram, each of the major blocks and their importance to the design are described.

As shown in Figure 3.3, the incoming data is attenuated by the channel and subsequently boosted by the equalizer with two independent, adjustable gains, C_1 and C_2 . The equalized signal is sampled by two slicers (S1 and S2) whose thresholds differ by ΔV (mV). If the signal amplitude is above ΔV (mV), the outputs of S1 and S2 will be identical, signifying a vertical eye opening larger than ΔV (mV). The adaptive controller adjusts C_1 and C_2 to equalize the vertical eye opening to ΔV for both Type 1 and Type 2 patterns. This is achieved by counting each pattern at the output of the two slicers and forcing their respective differences to zero. This completes the equalization for a given ΔV . The details of the adaptation for ΔV will be discussed in Section 3.3.

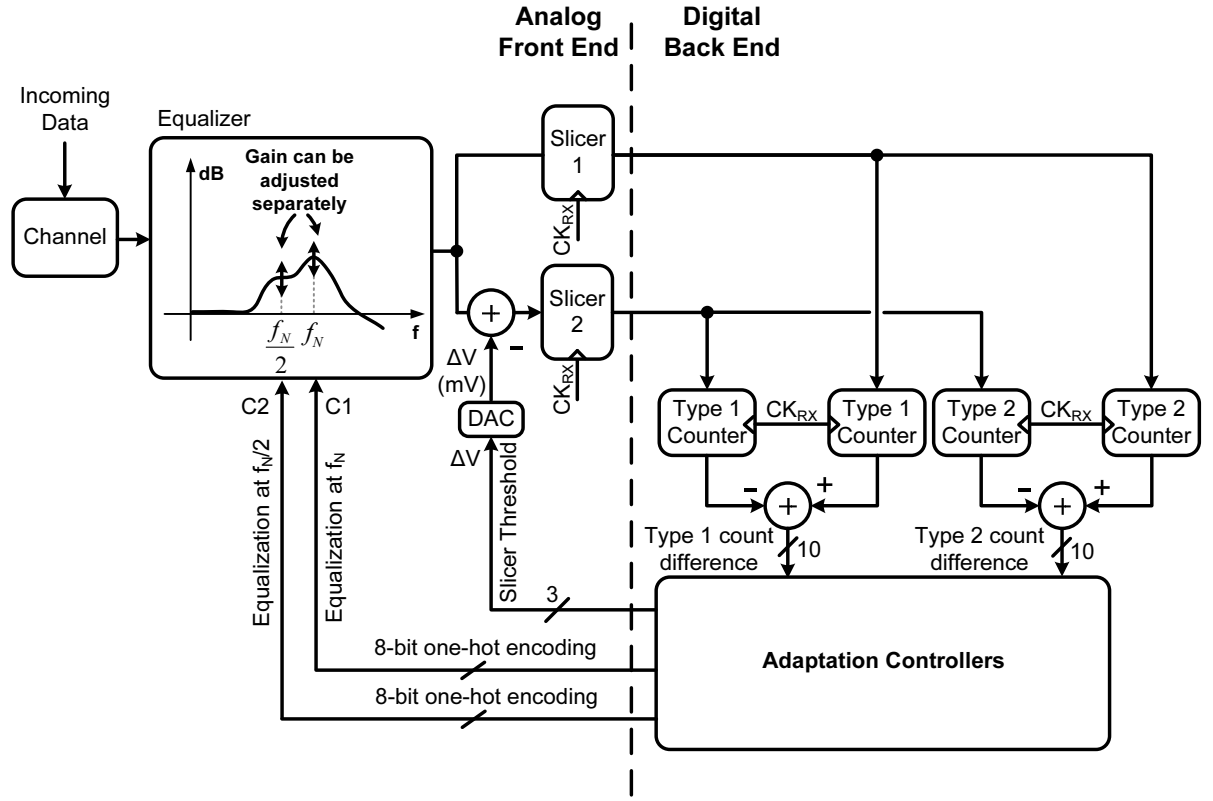


Figure 3.3: Simplified system block diagram

The desired transfer characteristic of the secondary slicer (S2) is shown in Figure 3.4. The slicer has a threshold value, ΔV , which can be shifted to change the point where bits are detected as a 0 or a 1. This block is vital to the operation of the algorithm. If the received eye is open and the signal has not been attenuated, both slicers will produce identical outputs. However, once the eye opening gets smaller, at some point, the secondary slicer (S2) will produce errors with respect to S1. Because of the low-pass nature of the channel, Type 1 and Type 2 patterns, described in Section 3.1, will be attenuated and the reduced amplitude will cause possible errors at the output of S2. The digital algorithm looks at the difference in the number of pattern in the same category between S1 and S2 and determines the amount of equalization required. The details of the digital algorithm will be explained in Section 3.3.

The CDR was directly taken from [16] including the layout. The CDR uses an Alexander Bang-Bang phase detector [17] along with a conventional charge pump. The VCO is

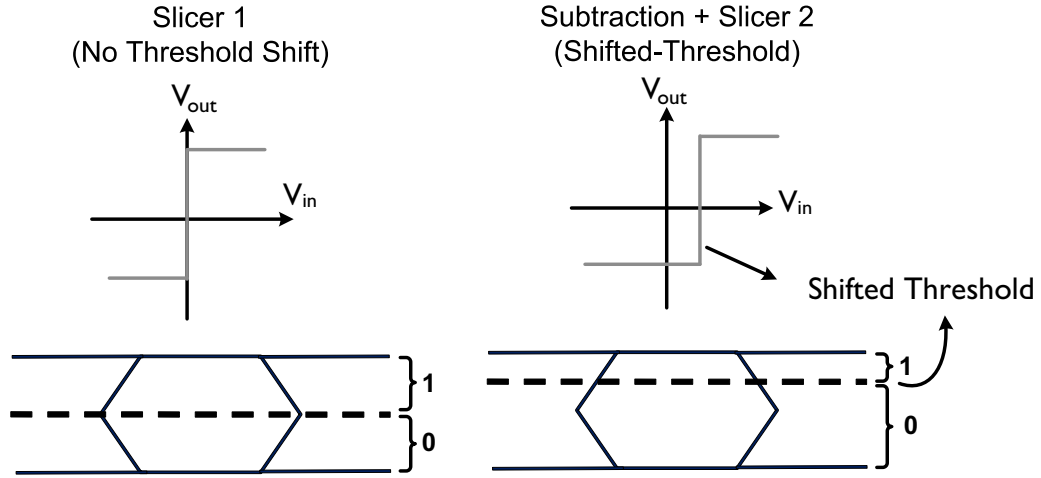


Figure 3.4: Conventional vs. Shifted threshold slicer

a 4 stage ring oscillator which has a simulated center frequency of 7.5GHz with updated models. The design was originally designed for 10Gb/s operation, however, measurements as well as newer models indicated that the maximum operating data rate is 7.5Gb/s. As a result, the current work was designed to be compatible with the 7.5Gb/s CDR. The frequency detection scheme used is a quadricorrelator used in [5]. The CDR's frequency detection requires sharp transitions for data since it is being used as the clock of flip-flops.

As discussed, the design requires an equalizer with independent gain control for two different frequencies. During the design phase, two equalizer topologies were considered that satisfied this requirement. The first topology, discussed in Section 3.5.1, has independent gain control at low frequencies as well as the Nyquist frequency. The second topology, discussed in Section 3.5.2, has independent gain control at f_N and $f_N/2$ using bandpass filters. Ultimately, the bandpass filter approach was chosen and the adaptation controllers were designed to use Type 1 and Type 2 patterns.

The design of the digital controllers, slicer with an offset, and equalizer will be discussed in Sections 3.3, 3.4, and 3.5, respectively.

3.3. Digital Adaptation Controllers

The digital adaptation controllers will set the equalizer gains at f_N , $f_N/2$, and S2's threshold. Section 3.3.1 will discuss the algorithm's operation while Section 3.3.2 will provide functional simulations.

3.3.1. Algorithm Implementation

Figure 3.5 shows the deserializers as well as the pattern counters. The data from S1 is deserialized to a 16-bit word. The 9-bit counters determine the number of Type 1 and 2 patterns in each word and accumulate a running total over 128 words (i.e. 2048 bits). However, word boundaries can also hide patterns. For example, there is a Type 1 pattern ("1010") that crosses the word boundary in "0010 1011". Hence, the design instantiates 4 sets of counters to cover all possible word boundary cases. The additional counters increase the algorithm's robustness and consistency as compared to when only one counter is used. The highest count is chosen to be the output. When the counting cycle finishes, the outputs are held constant for another 2048 bits while the slicer threshold, equalizer gains, and CDR phase alignment are updated. The data from S2 is processed by identical counters, however, the output is chosen to correspond to S1 (e.g. the S2_T1 and S1_T1 outputs are based on the same word boundary).

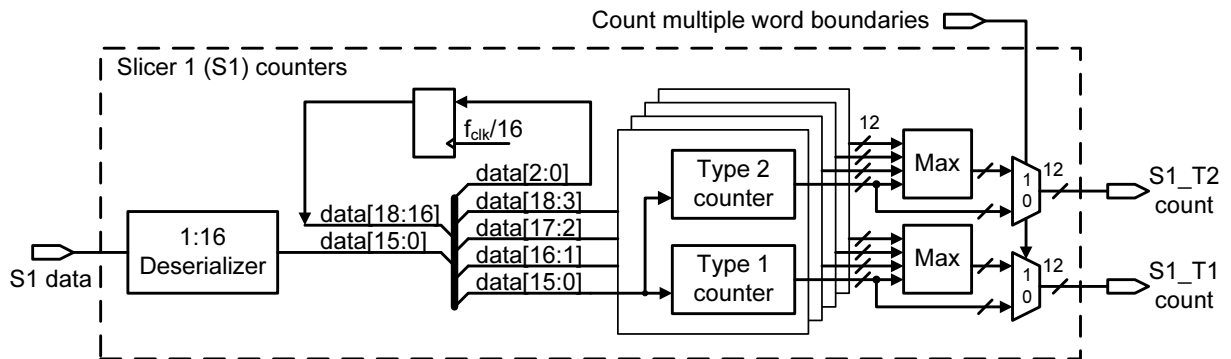


Figure 3.5: Implementation of pattern counters

Figure 3.6 shows the C1, C2, and ΔV controllers. The RTL finalization, synthesis,

and place and route of this design is carried out by Clifford Ting [18]. The C1 controller iteratively adjusts the gain at f_N until it converges to the lowest value such that the Type 1 count difference is less than or equal to an error tolerance that is programmable between 0 and 50. Eventually, C1 will reach steady state if it toggles between two adjacent values (e.g. 4,5,4,5,4,5,4,5), decreases to zero (i.e. 0,0,0,0,0,0,0,0), or increases to maximum (i.e. 7,7,7,7,7,7,7,7). This is identified via a 7-stage shift register as shown. The C2 controller is identical to the one for C1, except that it reads Type 2 counter differences.

The ΔV controller maximizes vertical eye opening by searching for the greatest ΔV that the C1 gain can compensate while avoiding bit errors on S2's output. It starts at the lowest setting ($\Delta V=1$) and iteratively increments C1 until the equalizer can no longer amplify the eye opening to ΔV (mV). The controller is based on C1 instead of C2 because the former compensates for signal attenuation at f_N , which is more severe than the attenuation at $f_N/2$. The ΔV controller implementation switches from C1 to C2 if the counters do not detect Type 1 patterns. For simplicity, this logic is excluded from Figure 3.6. After all three coefficients converge, the controllers lock them to their final values.

3.3.2. Algorithm functional simulations

Figure 3.7 shows a functional simulation of the digital controllers. The analog equalizer and slicers are emulated by a block that generates errors unless specific coefficients have been achieved by the digital algorithm. In the emulated scenario, if C1 (gain at f_N) is smaller than 6 or if C2 (gain at $f_N/2$) is smaller than 3, S2's output will have errors compared to S1. The system also needs to adapt the slicer threshold which also has been emulated and needs to reach an optimal level of 4; anything larger than four and the system would produce errors regardless of the equalizer gain settings. From Figure 3.7 it can be seen that the initial gains for C1 and C2 are set at 7. The system then adapts C2 while holding the value of C1 at 7. Once the C2 levels toggle between two adjacent

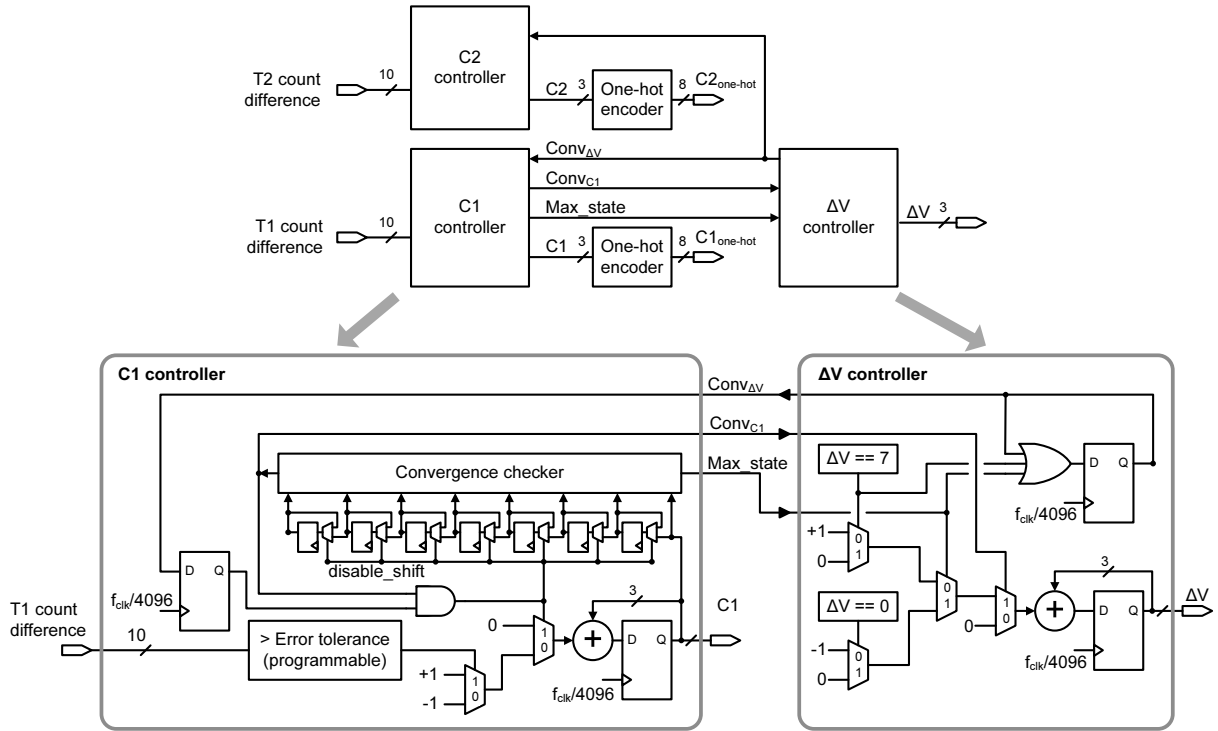


Figure 3.6: Implementation of digital adaptation controllers (Implementation by Clifford Ting [18])

levels, the gain is locked and C1 is adapted to reach the optimal value of 6. The system then attempts to increase the eye opening by increasing the threshold of S2 (ΔV). The threshold is increased and C1 and C2 are re-adapted, until the system reaches a threshold value of 5, where the system cannot equalize and maximizes the gain levels. Once the gain levels reach the maximum, if errors persist at S2's output, the threshold is lowered to the previous working level and locked. The system is then adapted one final time and coefficients are locked.

3.4. Slicer with shifted threshold

To create a slicer with an adjustable threshold, the circuit shown in Figure 3.8 was used. This circuit lowers the DC voltage of one of the differential nodes by drawing additional current through the load resistor. The overall effect is that the differential signal's DC voltage level is shifted. This shifted signal is then sliced using a flip-flop. The circuit in

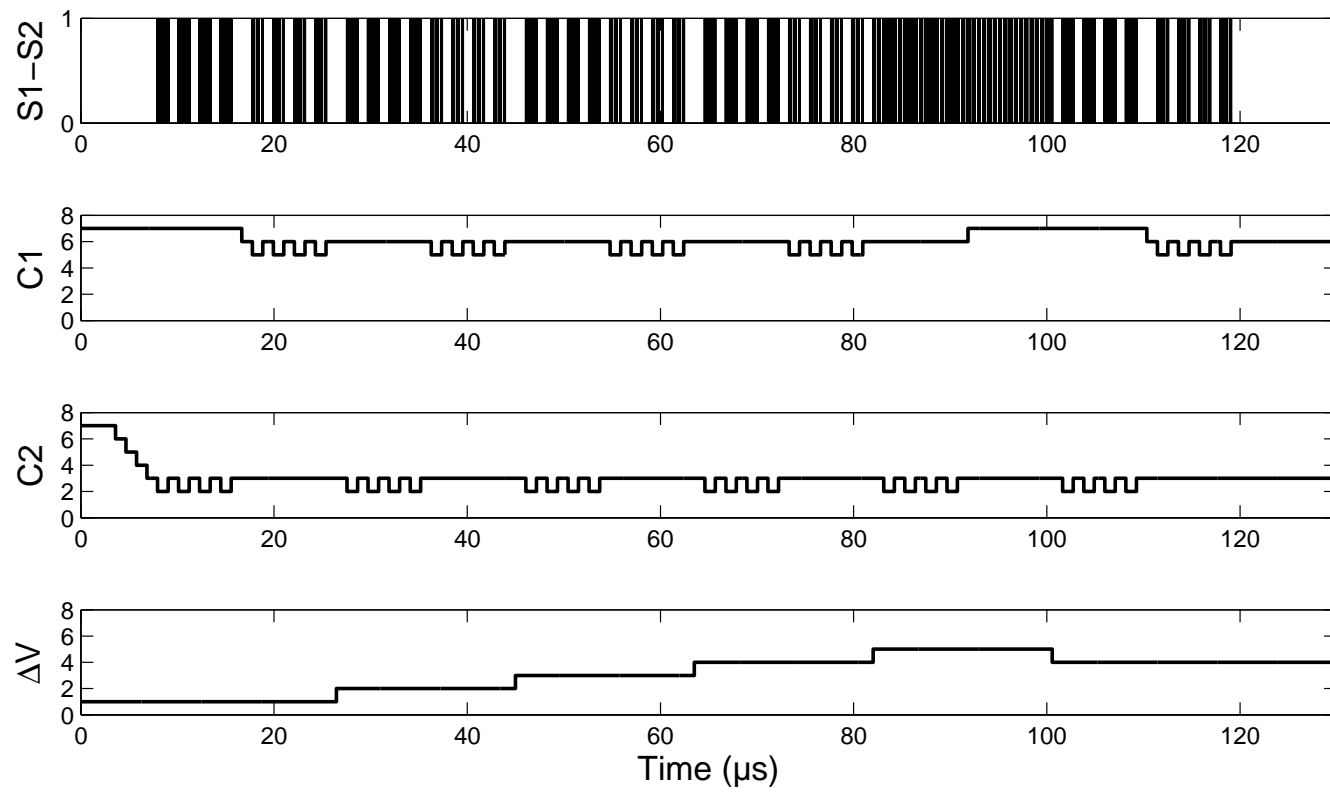


Figure 3.7: Timing diagram of the synthesized digital adaptation controllers

Figure 3.8 together with a flip-flop produce the desired transfer characteristic of a slicer with a shifted threshold. The amount of threshold shift, ΔV (mV), is controlled using the 3-bit output from the adaptation engine as described in Section 3.3. The 3-bit input is converted to current using the current mode DAC.

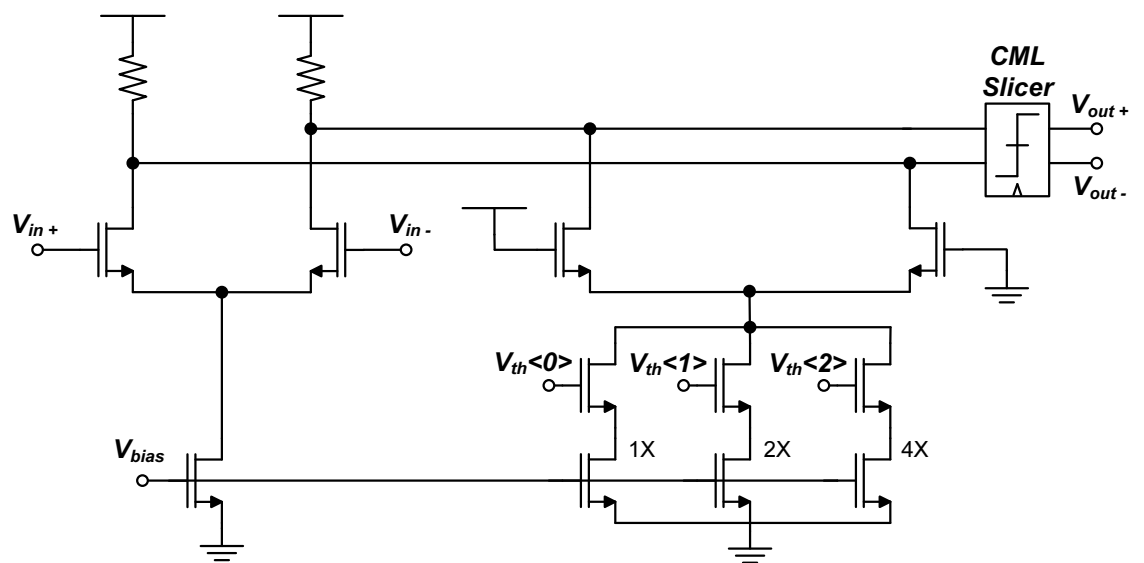


Figure 3.8: Circuit implementation of variable threshold slicer

Figure 3.9 shows the simulated eye diagrams at the output of the threshold shifter stage for various 3-bit inputs. The algorithm ensures that S2's threshold is always initially at the second lowest level to ensure that S1 and S2 always have different thresholds. From the eye diagrams it can be seen that as the threshold code is increased, there is a larger amount of DC shift in the signal. Figure 3.10 shows the eight possible threshold shift levels based on the 3-bit digital control. There may be small amounts of offset present at the input of the differential pair which lead to an overall smaller threshold shift. However, the smallest shift value is 90mV and should be much larger compared to the offset present in the differential pair.

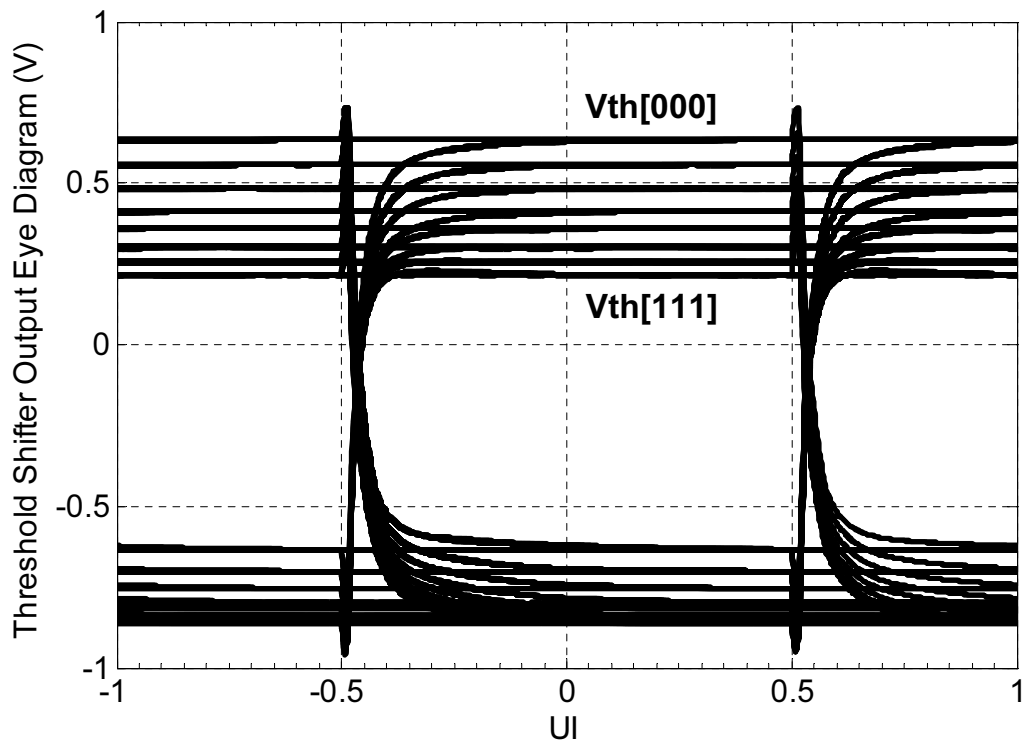


Figure 3.9: Simulated Output eye diagram of variable threshold shifter

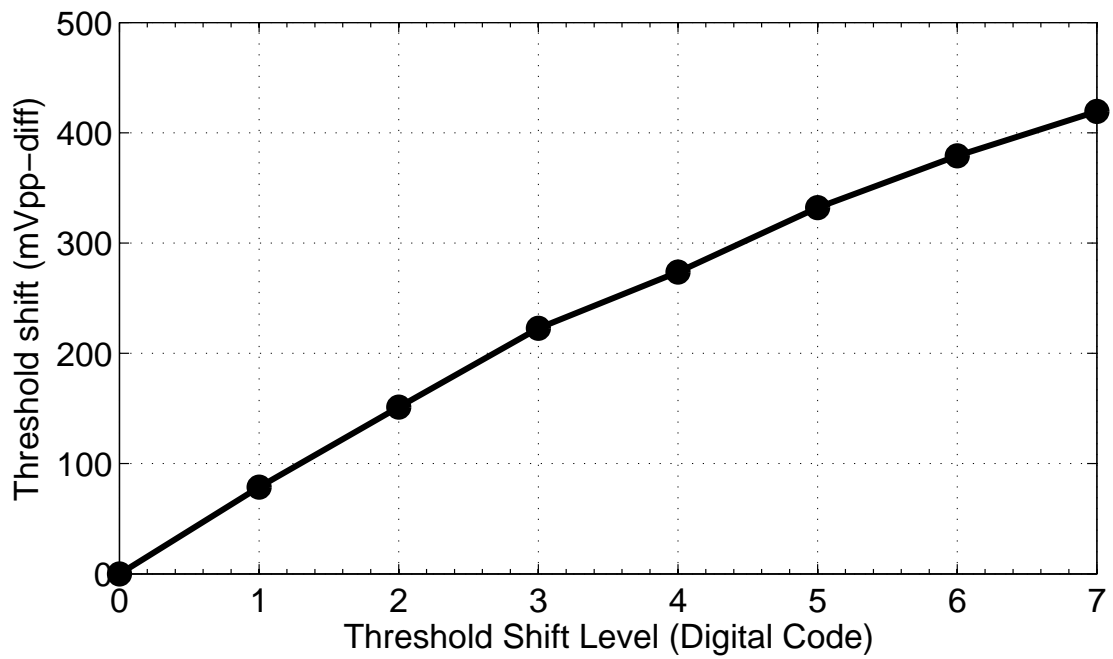


Figure 3.10: Simulated amount of threshold shift in mV for each digital code

3.5. Equalizer Topology

As discussed, the design requires an equalizer with independent gain control for two different frequencies. The candidates for the equalizer topology are discussed in Sections 3.5.1 and 3.5.2 with the latter being chosen.

3.5.1. Equalizer utilizing highpass filters (Topology 1)

Figure 3.11 shows the concept behind an equalizer with two gain controls at DC and f_N [19]. The first stage uses resistive and capacitive degeneration to control the location of the first zero. Varying R_N changes the location of the first zero as well as the DC gain. The pole in the feedback path, translates to another zero, whose location can be controlled with R_P . This architecture has one variable, R_N , which has more control over DC gain, and another, R_P , which prominently affects high frequency gain. This would require that we use pattern types with frequency content at DC and f_N . Equation 3.3 represents the transfer function of this system.

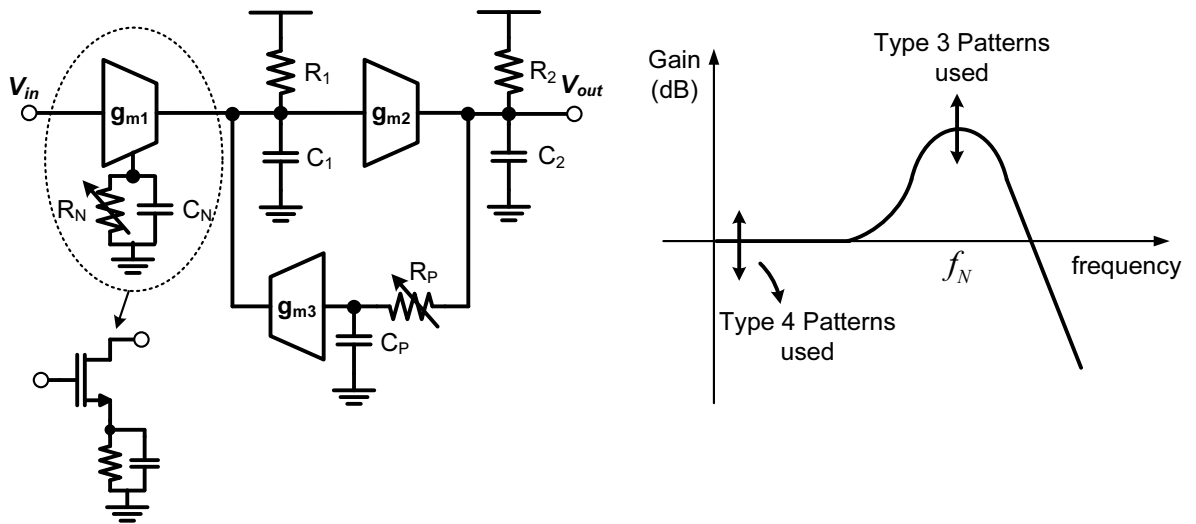


Figure 3.11: Equalizer topology utilizing highpass filters

$$\frac{V_{out}}{V_{in}} = \frac{G_{m1}/G_{m3}}{1 + (1 + sC_1R_1)(1 + sC_2R_2) \left(\frac{1}{R_1R_2G_{m2}G_{m3}} \right)} \quad (3.3)$$

where:

$$\begin{aligned} G_{m1} &= \frac{g_{m1}}{1 + g_{m1} \left(\frac{R_N}{1 + sC_N R_N} \right)} \\ &\approx \frac{1}{R_N} + sC_N \end{aligned} \quad (3.4)$$

$$G_{m2} = g_{m2} \quad (3.5)$$

$$G_{m3} = \frac{g_{m3}}{1 + sC_P R_P} \quad (3.6)$$

$$\begin{aligned} \frac{V_{out}}{V_{in}} &= \frac{(1/R_N + sC_N)(1 + sC_P R_P)1/g_{m3}}{1 + (1 + sC_1 R_1)(1 + sC_2 R_2)1/(R_1 R_2 g_{m2} g_{m3})} \\ &\approx \frac{1}{R_N g_{m3}} (1 + sC_N R_N)(1 + sC_P R_P) \end{aligned} \quad (3.7)$$

Equation 3.7 shows the approximation for the gain at lower frequencies of the highpass topology. It is evident that R_N controls the DC gain and the location of the first zero. The reason this topology was not used was due to the fact that the lower frequency gain also affected the amount of high frequency boost due to the first zero. This makes it very difficult to use the patterns with independent frequency content, as the equalizer gain does not provide independent gain. This highpass topology can be more effectively used with a simpler algorithm provided in [14], which does not require independent gain control.

3.5.2. Equalizer utilizing bandpass filters (Topology 2)

This section describes the bandpass topology which was implemented in the proposed design. As mentioned before, this architecture allows independent gain control at f_N and $f_N/2$. The main concept is illustrated in Figure 3.12. Three separate paths are created to give the desired frequency response. The first path resonates at f_N and is used to amplify signals at the Nyquist rate, no low frequency data will pass through this path.

Similarly, another path provides amplification for signals at $f_N/2$. A third path provides a path for low-frequency data. The sum of the three paths provide the desired transfer function with 0dB DC gain, and independent gain control at f_N and $f_N/2$. The transfer function of the system is shown in equation 3.8.

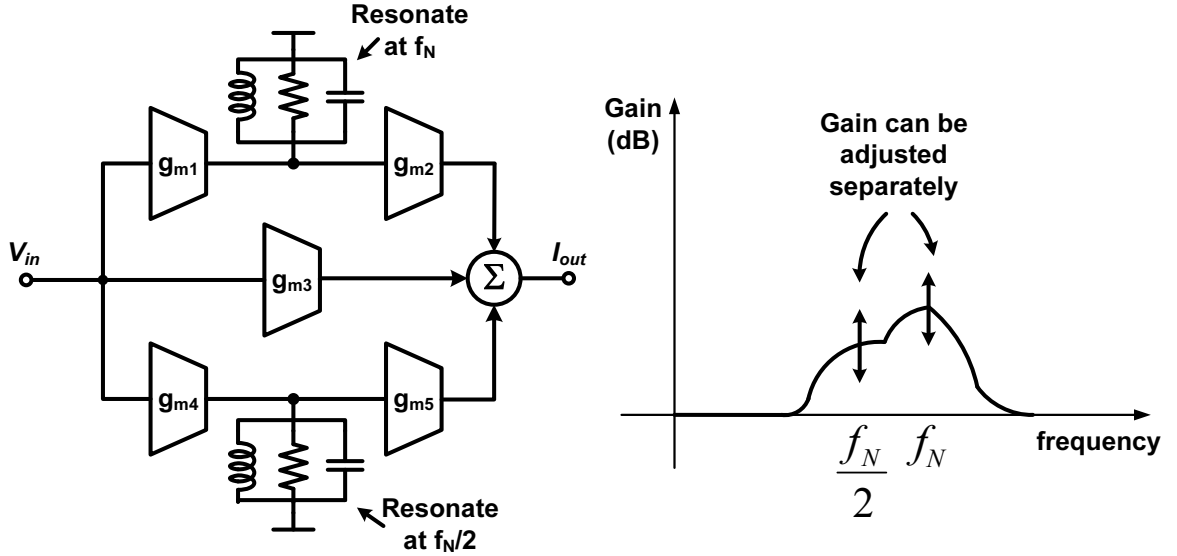


Figure 3.12: Equalizer topology utilizing bandpass filters

$$\frac{I_{out}}{V_{in}} = \underbrace{g_{m1} \times g_{m2} \times Z_{RLC}(f_N)}_{\text{Gain at } f_N} + \underbrace{g_{m4} \times g_{m5} \times Z_{RLC}(f_N/2)}_{\text{Gain at } f_N/2} + \underbrace{g_{m3}}_{\text{Gain at DC}} \quad (3.8)$$

The main disadvantage of this topology is that it requires inductors, which may occupy a large area and are difficult to model accurately. If the inductance were to vary significantly from the designed value, the two bandpass filters may no longer have separate gains at f_N and $f_N/2$ since the center frequency would shift. To compensate for inductor variations, varactors can be used to adjust the center frequency of the bandpass filter, explained further in Section 3.5.2.1.

To try and mitigate the area issue of inductors, active inductors were considered. The basic concept behind an active inductor is shown in Figure 3.13. At low frequencies, the capacitor across the transistor source/gate is open, and therefore looking into the source the impedance is $\frac{1}{gm}$. As the frequency increases, the capacitor shorts V_{gs} and as a result

the impedance seen is R_G . The range of frequencies where the impedance shifts from $\frac{1}{gm}$ to R_G is where the device behaves like an inductor. The impedance for an RLC tank is shown in equation 3.9 and the impedance equation for the active inductor is shown in equation 3.10. The main issue with this approach is the reduced common mode level at the output of the bandpass filter. Since the transistor requires $V_{GS} > V_t$ to stay in saturation, the output common mode would be very low from a 1.2V supply, which would also limit the swing at the output. The active inductors were not pursued further due to their swing/common mode limitations.

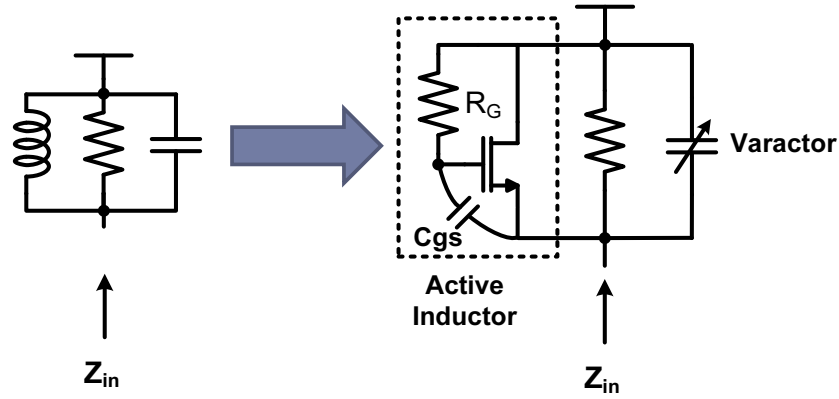


Figure 3.13: Implementation of an active inductor

$$Z_{in} = \frac{\frac{s}{C}}{s^2 + \frac{s}{RC} + \frac{1}{RCL_{eq}}} \quad (3.9)$$

$$L_{eq} = \frac{1 + sRC_{gs}}{g_m + sC_{gs}} \quad (3.10)$$

Passive inductors were used from Fujitsu's process. The desired value of 2nH was chosen for the bandpass filters. The same inductance was used for both bandpass filters at $f_N/2$ (1.75 GHz) and f_N (3 GHz); the capacitance was varied to get the appropriate center frequency. The inductance has 3 turns and occupies an area of 35um x 35um. The inductance vs. frequency is shown in Figure 3.14. For frequency regions of operation, the inductance is the desired 2nH. Figure 3.15 shows the quality factor of the inductor

and the Self Resonance Frequency (SRF) of 18GHz, well over the operating range of the inductor.

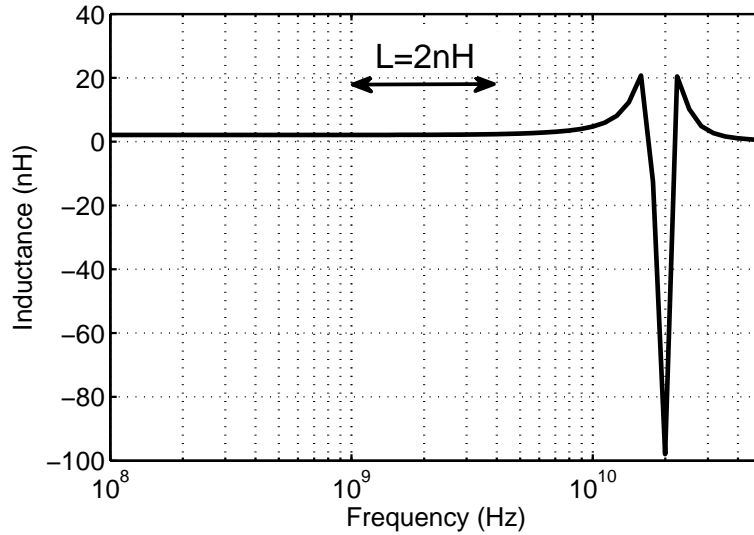


Figure 3.14: Inductance vs. Frequency

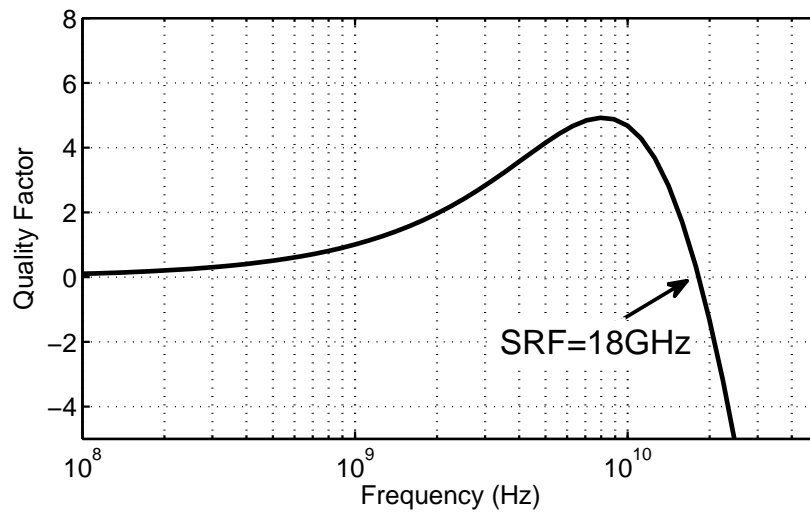


Figure 3.15: Quality factor vs. Frequency for the chosen inductor

The final implemented equalizer is shown in Figure 3.16. The two bandpass filters are followed by 4 VGAs to allow more amplification at f_N and $f_N/2$. The low-pass path consisting of 5 buffers is designed to give 0dB gain as required by this topology. The output of the three paths are connected and summed in current mode. Sections 3.5.2.1

and 3.5.2.2 discuss the design of the bandpass filter and the Variable Gain Amplifier (VGA), respectively.

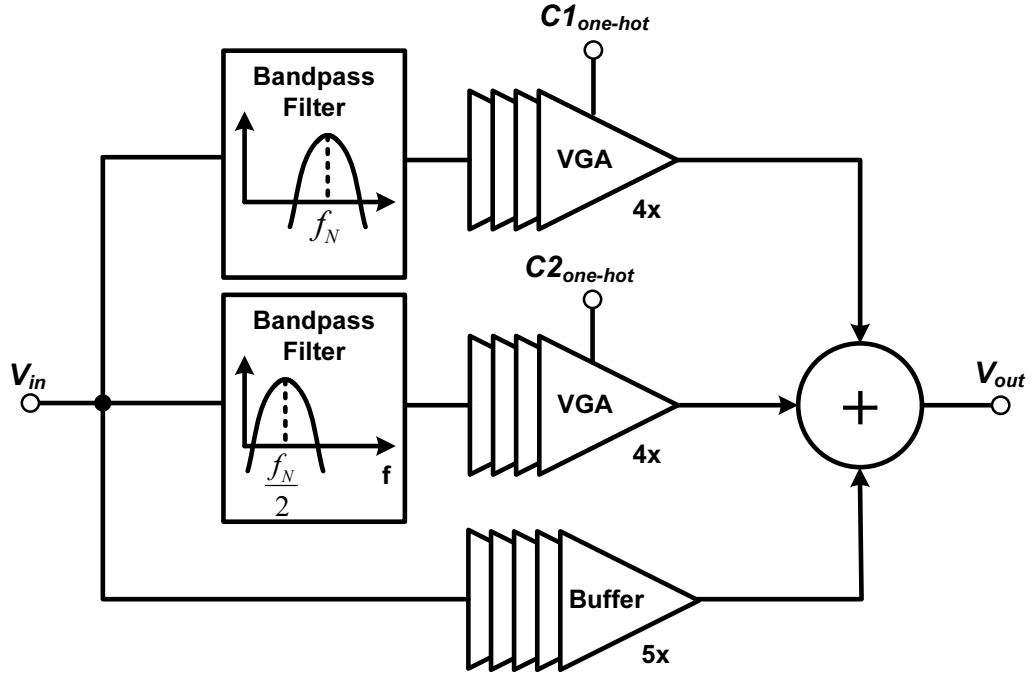


Figure 3.16: Final implemented equalizer topology utilizing bandpass filters and VGAs

3.5.2.1. Bandpass Filter

Figure 3.17 shows the schematic of the bandpass filter that was created using a differential pair with an RLC load. The RLC load is designed to resonate at f_N and $f_N/2$ for the two different bandpass filters. Varactors are used to allow tuning of the center frequency to compensate for any inductor variation. A common mode resistor is used to set the output common mode to 900mV. Since the metal for the inductors have a resistance associated with them, the DC gain of this circuit is not zero. To alleviate this problem, a degeneration capacitor is used to further reduce the DC gain of the circuit.

Figure 3.18 shows the extracted simulation of the bandpass frequency response with the varactor's desired tuning range. A peak at f_N , 3.75GHz, can be obtained with a control voltage of 600mV. The peak can be shifted to 4.45GHz or down to 3.3GHz with a control voltage of 1.2V and 600mV, respectively. The amount of boost changes as the

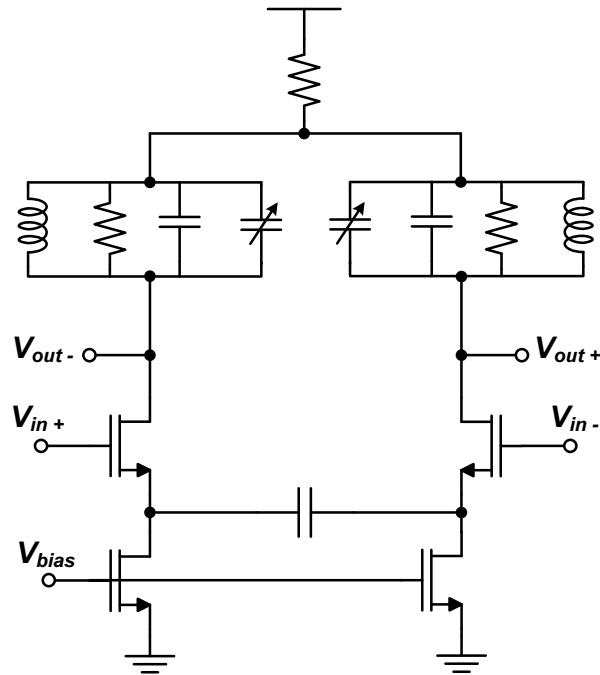


Figure 3.17: Circuit implementation of a bandpass filter

capacitance changes, but that will be compensated for by the following VGA stages.

3.5.2.2. VGA

The schematic of the VGA is shown in Figure 3.19. A differential pair is used with resistive degeneration to change the gain. A total of 8 levels are possible and are controlled by one-hot-encoded switches. Increasing the resistance seen at the source of the input transistor pair decreases the gain of the stage. The four VGAs used in each bandpass path are controlled using the same coefficients. One-hot-encoding was used to reduce the variation between stages compared to using binary weighted resistors.

Figure 3.20 shows the extracted simulation of the frequency response of the VGA. The maximum low frequency gain range is 7dB. For the lower gain setting, there is slight attenuation but this is acceptable since for the lowest gain setting we would like to provide almost no equalization. The signal will pass only through the low-pass path shown in Figure 3.16 and not the VGAs in the bandpass paths.

Figure 3.21 shows the frequency response of the bandpass filter at 3.75GHz (f_N) followed

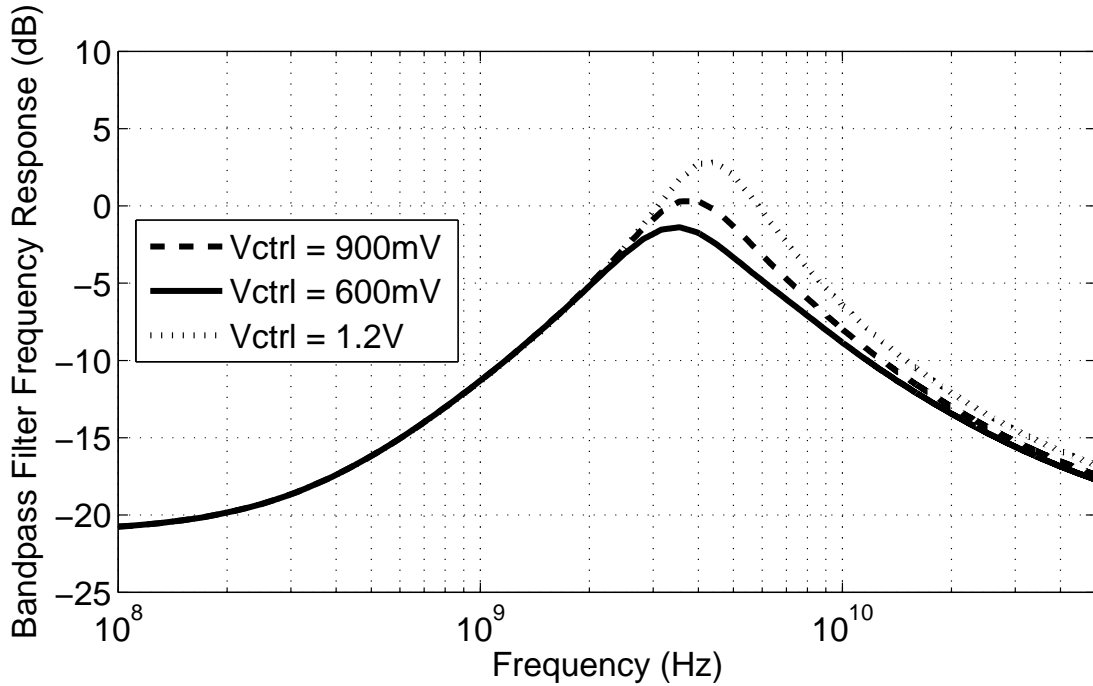


Figure 3.18: Post layout frequency response of the bandpass filter centered at f_N

by 4 VGA stages. Ideally, we would like this path to have no low frequency gain. But it is evident that there is finite DC gain. This is due to the parasitic resistance of the inductor. The series resistance with the inductor provides a path which creates finite gain at low frequencies. This can also be seen from the transfer function of the bandpass filter shown in Figure 3.18. While this is undesirable, it should be noted that the final amount of equalization is a function of the difference in high frequency gain and low frequency gain. Figure 3.22 shows the same result for the bandpass filter centered at 1.875GHz ($f_N/2$) followed by 4 VGA stages.

3.6. Conclusions

This chapter introduced the proposed equalizer adaptation along with its implementation. The output of two slicers are compared to ensure that the equalized signal has a minimum eye opening. Two different equalizer topologies were discussed with the topology utilizing

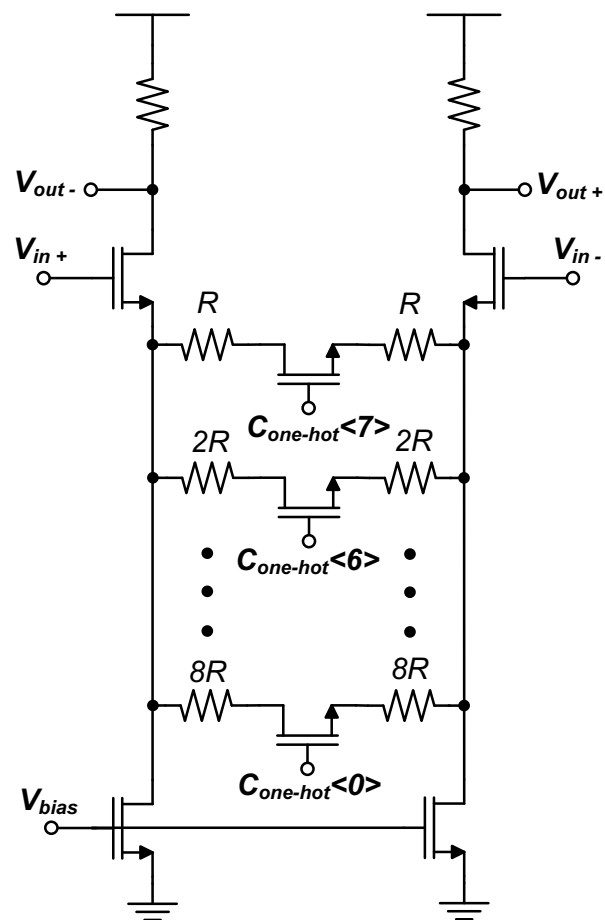


Figure 3.19: Circuit implementation of Variable Gain Amplifier

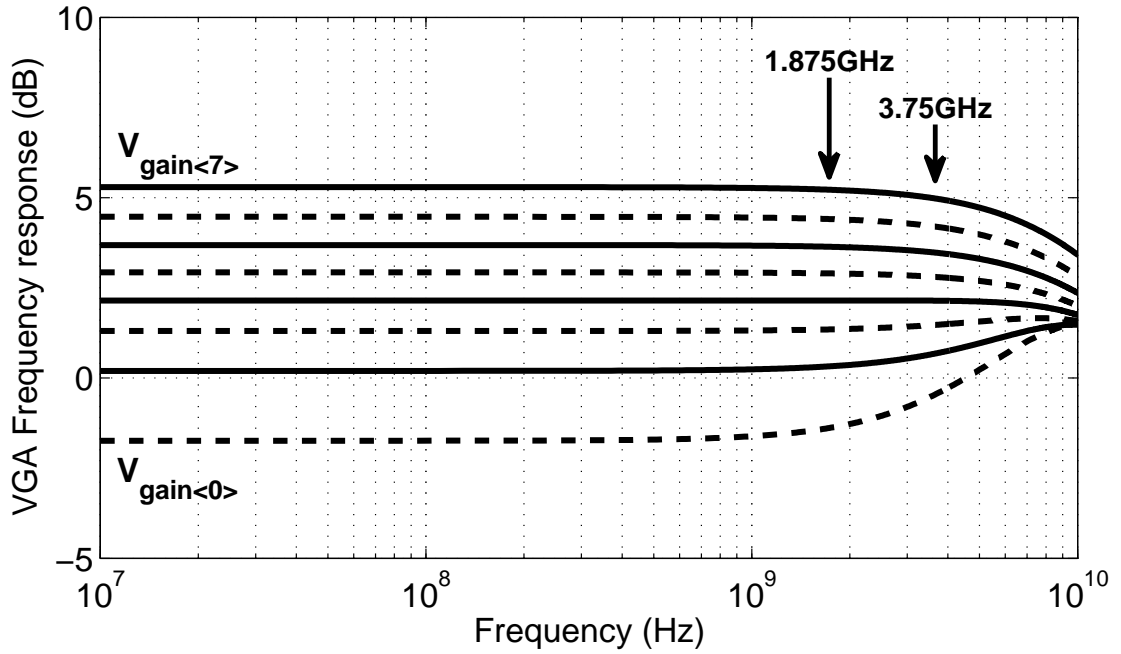


Figure 3.20: Post layout frequency response of the VGA for various gain settings

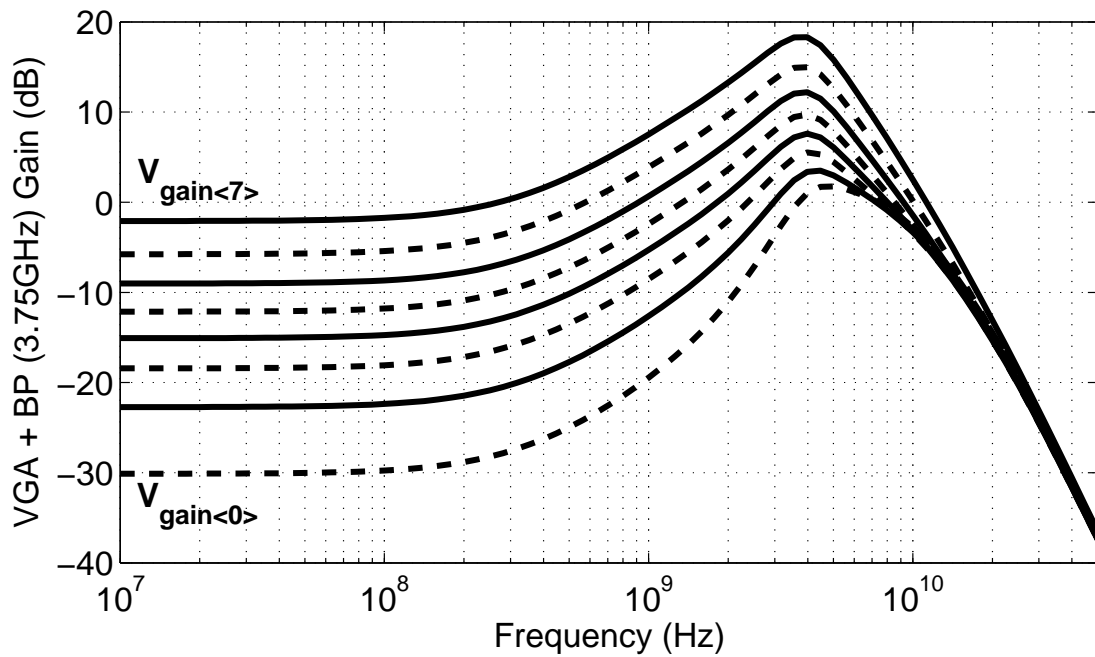


Figure 3.21: Post layout frequency response of the 4 VGAs and Bandpass filter at 3.75GHz

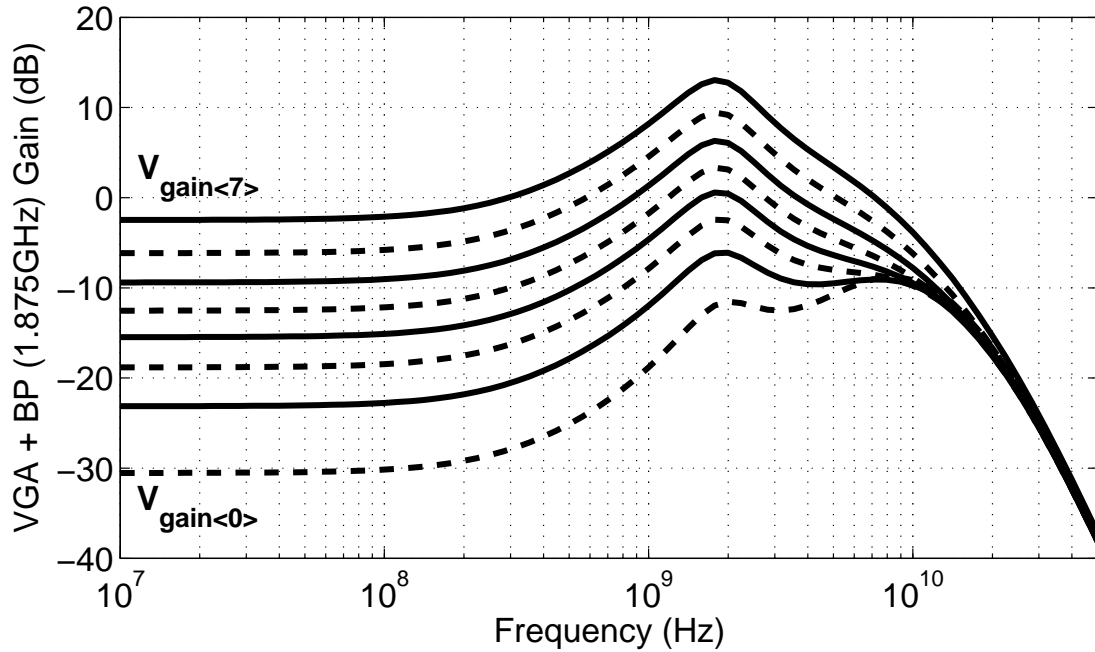


Figure 3.22: Post layout frequency response of the 4 VGAs and Bandpass filter at 1.875GHz

bandpass filters being chosen for final implementation. A digital adaptation engine was designed to control the equalizer coefficients as well as the slicer threshold offset. Post-layout simulation results were provided for each of the analog blocks included in the equalizer. Functional simulations of the digital adaptation were also provided.

4

Experimental and Simulation

Results

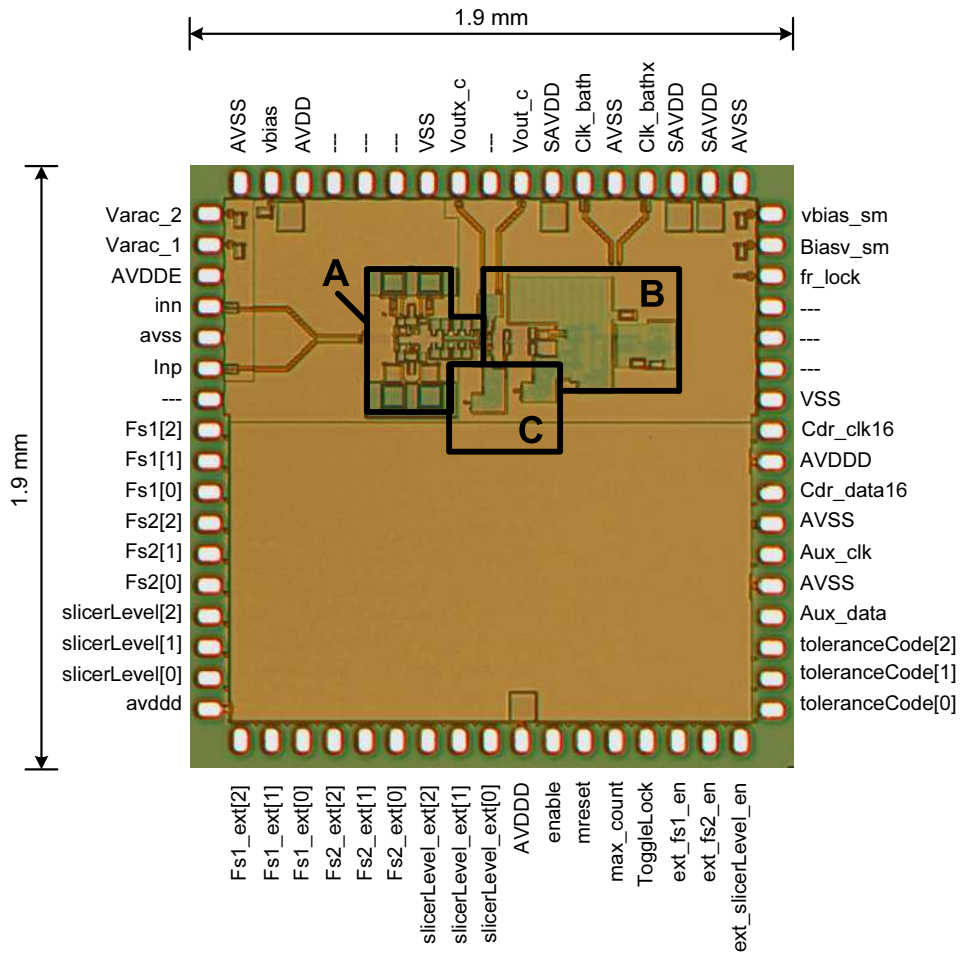
This chapter will present the measurement results of the fabricated test chip. First, the receiver layout and equipment setup will be presented. Second, the operation of the system with a blind clock will be described. Third, the measurement results of the equalizer adaptation will be presented.

4.1. Receiver layout and equipment setup

The design was fabricated in a 65nm CMOS process from Fujitsu. The die photo of the test chip along with pin names are shown in Figure 4.1. The test-chip consists of the equalizer, CDR, and the digital adaptation controllers. The equalizer consumes 60 mW with an area of $0.104mm^2$. The digital portion of the design consumes 16.8 mW with an area of $0.101mm^2$.

Table 4.1 describes the functionality of each of the pins on the chip. The main measurements used the equalizer output ($V_{out.c}/V_{outx.c}$) and using both the adapted coefficients and external coefficients to view equalized eye diagrams.

The measurement setup is shown in Figure 4.2. The test-chip was packaged using a 64-pin QFN package and was soldered on a PCB for measurement. A Centellax OTB3P1A PRBS generator was used as the input to the system. The PRBS data was attenuated using TYCO channels with two different attenuations being measured. The buffered equalizer output was observed with an Agilent Infiniium DCA-J 86100C digital



| | |
|------------------|-----------|
| Process | 65nm CMOS |
| Data Rate | 6Gb/s |
| Supply | 1.2V |

| | | Block description | Area (mm²) | Power (mW) |
|-------------------------|----------|------------------------------|------------------------------|-------------------|
| Analog front end | A | Equalizer | 0.104 | 60 |
| | B | CDR & output buffers | 0.201 | 420 |
| Digital back end | C | Digital counters/controllers | 0.101 | 16.8 |
| | | Total | 0.406 | 469.8 |

Figure 4.1: Chip die photo

| Pin name | Description |
|---------------------------------------|---|
| inp/inn | Differential input data |
| Vout_c/Voutx_c | Buffered equalizer output |
| Varac_1 | Varactor control voltage at f_N |
| Varac_2 | Varactor control voltage at $f_N/2$ |
| vbias | Bias current for equalizer |
| vbias_sm | Bias current for CDR (minus VCO) |
| Biasv_sm | Bias current for VCO |
| fr_lock | Lock flag from CDR |
| Fs1[0:2] | Output gain level for f_N |
| Fs2[0:2] | Output gain level for $f_N/2$ |
| SlicerLevel[0:2] | Slicer threshold output |
| Cdr_clk16 | Demuxed (by 16) clock from CDR |
| Cdr_data16 | Demuxed (by 16) data from CDR by 16 |
| Aux_clk | Auxiliary demuxed clock for debugging |
| Aux_data | Auxiliary demuxed data for debugging |
| Fs1_ext[0:2] | External gain control at f_N |
| Fs2_ext[0:2] | External gain control at $f_N/2$ |
| SlicerLevel_ext[0:2] | External slicer threshold control |
| ext_fs1_en, ex_fs2_en, slicerLevel_en | Enable external coefficients |
| ToggleLock | Enable algorithm lock after reaching steady state |
| max_count | Enable counters to cover data boundaries |
| AVDD | Equalizer power supply |
| SAVDD | CDR power supply |
| AVDDD | Digital power supply |
| AVSS | Ground |

Table 4.1: Description of the pin-list

communication analyzer. The adapted equalizer coefficients were observed with a Tektronix TLA 714 logic analyzer. The control signals were all controlled using toggle/DIP switches placed on the PCB shown in Figure 4.3.

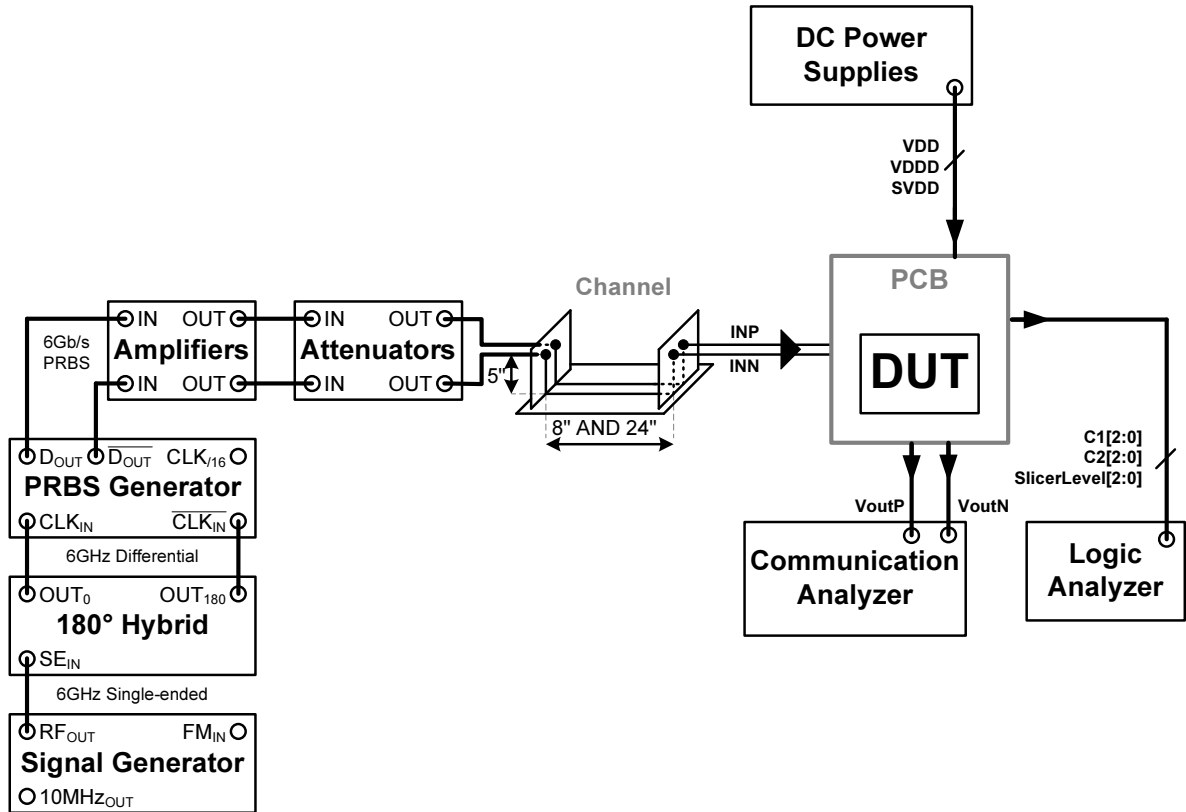


Figure 4.2: Measurement Setup

4.2. Blind clock operation

The CDR in the system was originally designed with a Limiting Amplifier (LA) at the front end. In that system, the Pseudo-Random Binary Sequence (PRBS) data was not transmitted through a channel. In the current scheme, the LA was removed and was replaced by two buffers prior to the slicer inside the CDR to ensure sharp transitions for the CDR frequency detector. The system operated properly during simulations.

Upon measuring the chip, it became clear that the CDR would not lock without the use of the limiting amplifier at the front end. The data transition slopes were simply

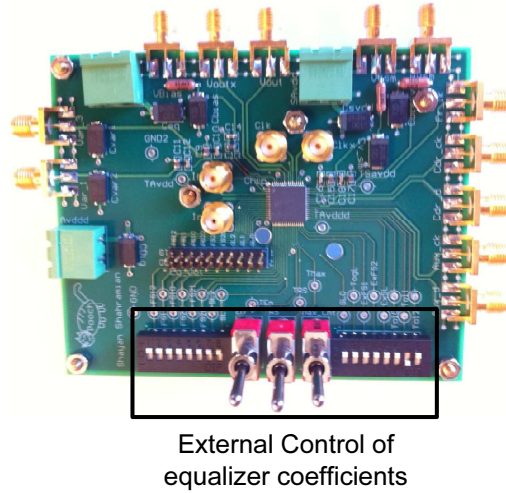


Figure 4.3: PCB picture

not adequate for the frequency detector which uses the data signal as a clock of flip-flops. This is the result of the reduced bandwidth of the fabricated buffers as opposed to simulations. As a result, the VCO inside the CDR would always drift to the lower frequency range of 6.12 GHz, and would not achieve phase or frequency lock. Even when the input data rate was reduced to 6Gb/s the CDR would not lock since the frequency detector would continue to vary the frequency of the VCO outside the lock range.

In an attempt to pursue the measurements without a functioning CDR, we tested the adaptation algorithm with the clock from the free running VCO. $S1$ and $S2$ are both driven by a clock signal, CK_{VCO} . When the sampling phase deviates from the center of the eye, the slicers may under-estimate the eye opening and cause the adaptation controllers to over-estimate the required gains. However, the implemented adaptation algorithm has mechanisms that prevent coefficients from converging to over-equalizing gains.

When there is a frequency offset between CK_{VCO} and the clock embedded in the data, CK_{VCO} may sample the data at the edges (rather than at the center). This may cause the slicers to underestimate eye height, sometimes leading the controller to increase gain erroneously. There were two mechanisms in the $C1$ and $C2$ controllers

which helped prevent this gain increase. First, the programmable error tolerance allows a count difference up to 50 before the controller increases the gain. In the measurement, the design is able to tolerate as much as 25,000ppm of offset with the error tolerance set at 20. Second, due to inherently larger signal slopes at the edge (compared to the center), the convergence checker does not flag the convergence until CK_{VCO} samples closer to the center of the eye.

4.3. Adaptation Performance

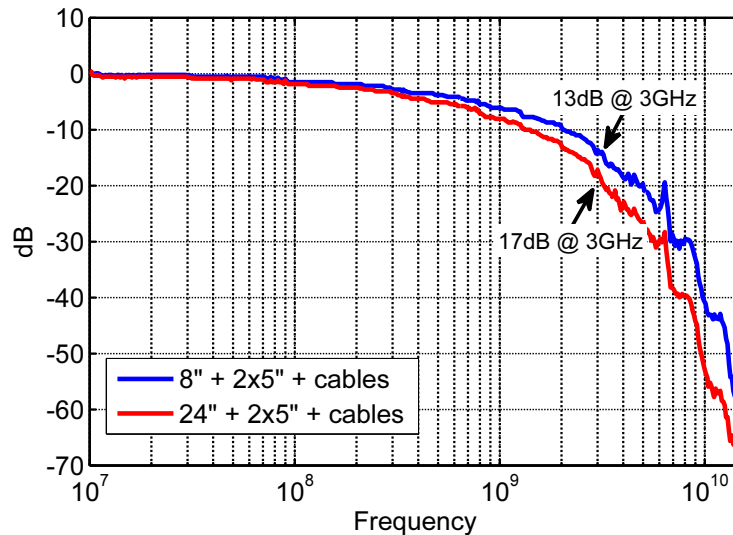


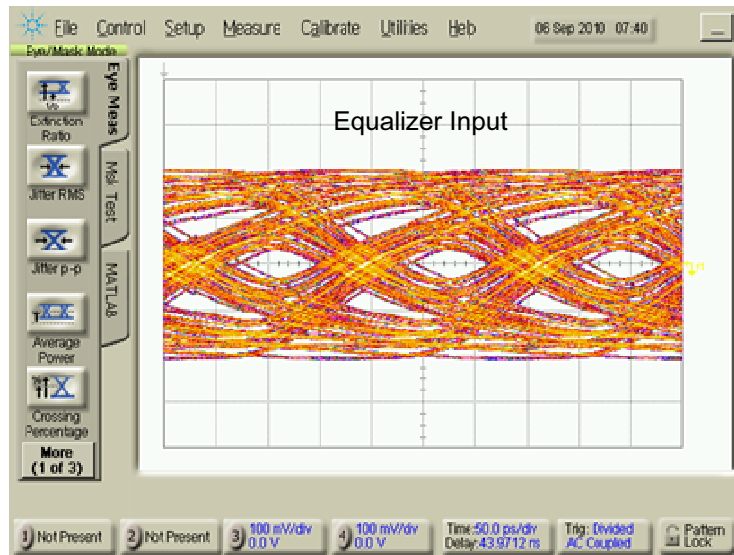
Figure 4.4: Channel frequency response

To determine the performance of the system, two channels were used with attenuations of 13dB and 17dB at 3GHz. The frequency response of both channels is shown in Figure 4.4. For each channel, the adaptation was activated, and the output eye of the equalizer was observed. The launch amplitude is $1.2V_{pp-single-ended}$. Figure 4.5(a) shows the input eye to the equalizer, and Figure 4.5(b) shows the output eye after adaptation of the coefficients for the 13dB attenuation channel. The eye opening after equalization is 512mV with a horizontal opening of 122ps. All the measurements are performed at 6Gb/s with a frequency offset between the receiver and transmitter of 25,000 ppm. Figure 4.6 shows the eye before and after equalizer for the 17dB attenuation channel. The vertical

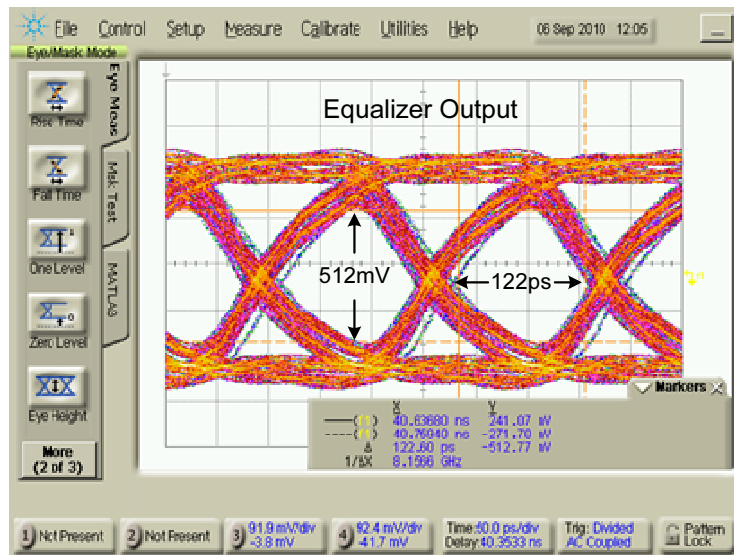
eye opening after equalization is 447mV with 120ps horizontal eye opening.

To determine the quality of the equalization algorithm, we need to find all possible output eyes for the equalizer. With this information, we can determine what are the optimal equalizer coefficients that lead to the largest horizontal and vertical eye opening. Figure 4.7 shows the vertical and horizontal eye openings for all 64 equalizer settings with the adapted coefficient being within 0.2% of the optimal vertical eye opening and within 5.4% of the optimal horizontal eye opening. Figure 4.8 shows the same results for a 17dB attenuation channel. The adapted coefficients are within 2.6% and 7.0% of the optimal vertical and horizontal eye openings, respectively. The reason for the deviation from the optimal vertical eye opening arises from the resolution of slicer with shifted threshold. As the number of bits are increased, finer resolution of the vertical eye opening can be obtained. This comes at the price of increased area and longer equalization time. The horizontal eye opening can also be characterized in terms of peak-to-peak jitter. For the 13dB and 17dB channels the adaptive algorithm adds 12% and 16% to the optimal horizontal peak-to-peak jitter, respectively.

Figure 4.9 shows a sample adaptation curve for the 17dB attenuation channel. The final coefficients do not match the previous results since this measurement is taken from another chip. The output buffers on the original chips were damaged during testing and as a result could not be used to generate the adaptation curves. The figure shows that C1 and C2 adapt to 7 and 0, respectively. The coefficients follow the designed algorithm, where the original S2's threshold offset is small, and slowly increases as the equalizer is able to open the eye past the set threshold. Finally, the maximum opening that can be achieved is chosen and the coefficients are locked to their final values. The abrupt increases in C1 and C2 are attributed to the blind nature of the clock and sampling closer to the edge as explained in Section 4.2.

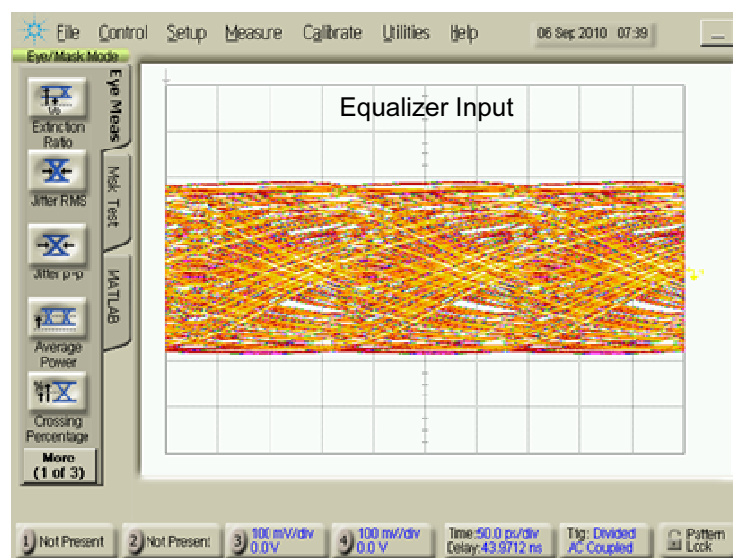


(a) Before Equalization

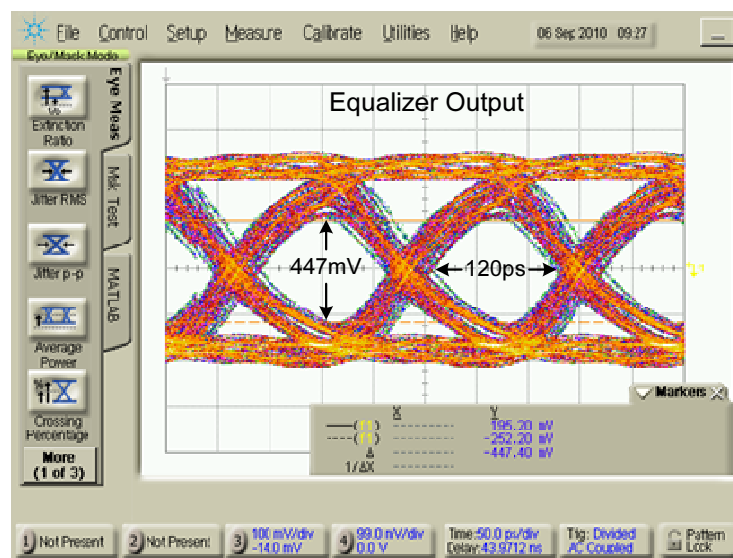


(b) After Equalization

Figure 4.5: Eye diagrams at the input and output of the equalizer for a 13dB attenuation channel. The equalizer adapted to $C1=7$, $C2=1$



(a) Before Equalization



(b) After Equalization

Figure 4.6: Eye diagrams at the input and output of the equalizer for a 17dB attenuation channel. The equalizer adapted to $C_1=6$, $C_2=2$

4.4. Summary

The proposed equalizer and adaptation engine were fabricated in a 65nm CMOS process. The adaptation was able to work with a blind clock with 25,000 ppm offset relative to the transmitter's clock at 6GHz. The system was measured with two channels with 13dB and 17dB of attenuation at 3GHz. The system adapted to within 2.6% and 7.0% of optimal vertical and horizontal eye opening, respectively.

Table 4.2 shows the comparison of this work with the state of the art.

| Specification | [11] | [20] | [12] | [21] | This Work [18] |
|-----------------------------|------|---------|------|----------|----------------|
| Data rate (Gb/s) | 20 | 40 | 6.4 | 10.3 | 6 |
| Attenuation @ f_N | 10 | 10 | 12 | 35 | 17 |
| Vertical eye opening (mV) | 50* | 200* | 222* | N/A | 447 |
| Horizontal eye opening (UI) | 0.6* | 0.6* | 0.7* | N/A | 0.72 |
| EQ. Power (mW) | 60 | 58 | 85 | 107** | 60+16.8 |
| Equalization Type | CTLE | CTLE | CTLE | CTLE+DFE | CTLE |
| Digital Adaptation | NO | NO | NO | YES | YES |
| Power Supply (V) | 1.5 | 1.8/1.5 | 1.8 | 1.2 | 1.2 |
| EQ. Area (mm^2) | 0.2 | 0.54** | 0.35 | 0.214** | 0.205 |
| Technology (nm) | 130 | 90 | 180 | 65 | 65 |

* Estimated from measured results figure

** Entire receiver

CTLE - Continuous Time Linear Equalizer (Discussed in Chapter 2)

Table 4.2: Comparison to state of the art

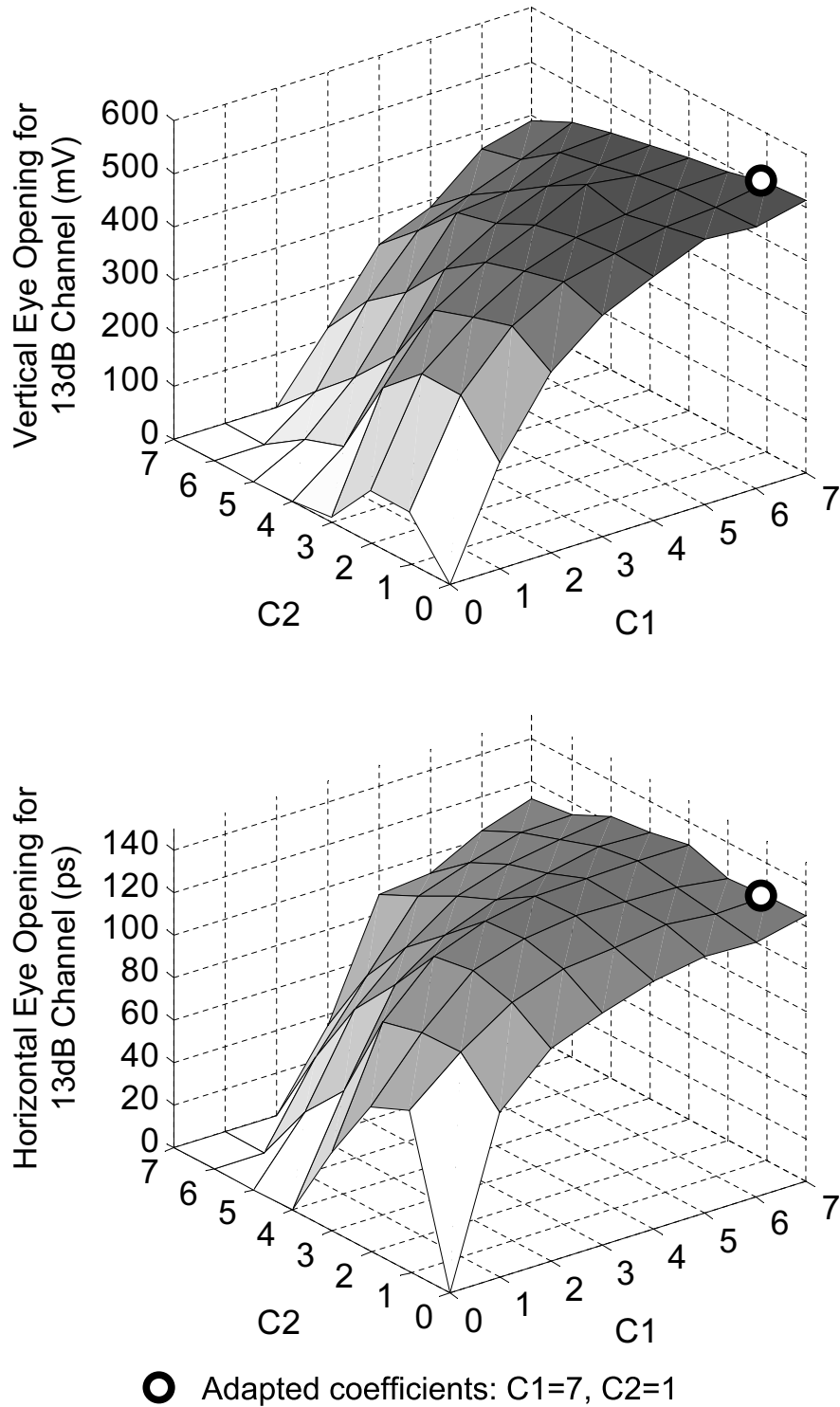


Figure 4.7: Top: Vertical eye opening after equalization for all 64 equalizer settings. Bottom: Horizontal eye opening after equalization for all 64 settings. Both results are for a 13dB attenuation channel

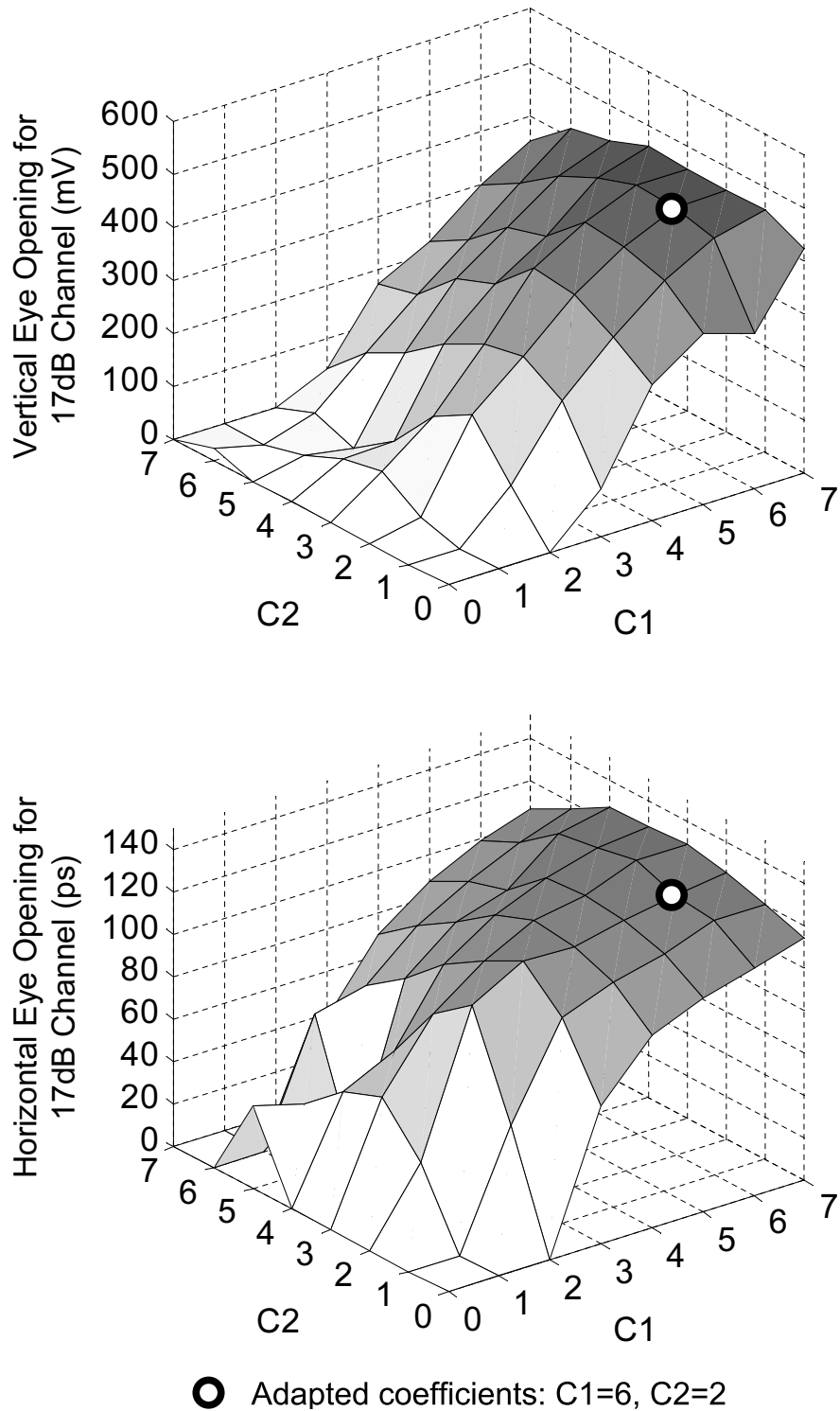


Figure 4.8: Top: Vertical eye opening after equalization for all 64 equalizer settings. Bottom: Horizontal eye opening after equalization for all 64 settings. Both results are for a 17dB attenuation channel

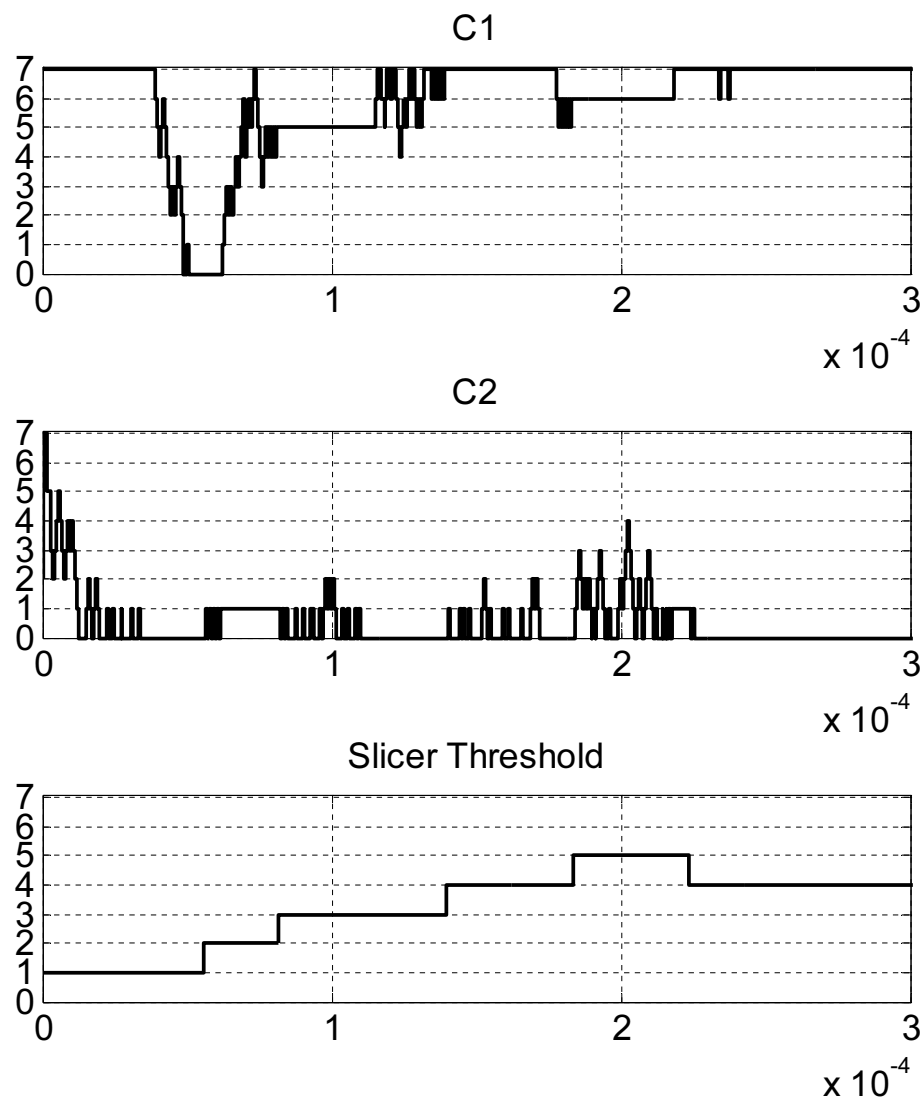


Figure 4.9: Adaptation of equalizer gains (C1 and C2) and slicer threshold voltage for 17dB attenuation channel

5

Conclusion

5.1. Thesis Contributions

This thesis provided a background on the different types of equalizers used to mitigate channel frequency dependent loss. Different adaptation schemes were discussed with advantages and disadvantages of each one highlighted.

A new type of adaptation engine was developed that utilizes patterns to equalize different frequency content of the signal. Two equalizer topologies were considered and the topology utilizing bandpass filters was chosen.

The algorithm and the equalizer were fabricated in a 65nm CMOS process. The CDR in the system was unable to lock to the signal due to the frequency detector topology used. The system operated with a blind clock with a 25,000 ppm frequency offset from the transmitter at 6Gb/s. Two different channels were used and the equalizer was able to open the eye for both 13dB and 17dB of attenuation at the Nyquist frequency. The adaptation performance was determined by measuring the vertical and horizontal eye openings for all possible equalizer coefficients. The eye openings at each of the adapted coefficients were compared to the optimal eye openings possible. The system performance is summarized in table 5.1

A summary of the contributions are:

1. Proposed an adaptation scheme for linear equalizers

| Specification | 13dB attenuation | 17dB attenuation |
|--------------------------------------|------------------|------------------|
| C1 | 7 | 6 |
| C2 | 1 | 2 |
| Vertical eye opening | 512 mV | 447 mV |
| Horizontal eye opening | 122 ps | 120 ps |
| Vertical opening away from optimal | 0.2 % | 2.6 % |
| Horizontal opening away from optimal | 5.4 % | 7.0 % |
| Adaptation Power | 16.8 mW | |
| Equalizer Power | 60 mW | |
| Adaptation Area | 0.101 mm^2 | |
| Equalizer Area | 0.104 mm^2 | |

Table 5.1: System performance summary

2. Implementation of analog part (RTL finalization, synthesis, and place and route was performed by Clifford Ting [18]).
3. A Paper accepted for presentation at ISSCC 2011 [18].

5.2. Future Work

There are three main advances that can be made to this work and they will be described in the following sections.

5.2.1. Integration with blind DFE and CDR

With the blind operation of the adaptive engine, the next natural step becomes to integrate this into a full blind receiver. An entirely Analog to Digital Converter (ADC) based blind CDR has been developed in [22]. This can easily be combined with the analog equalizer and blind adaptation proposed in Chapter 3. A blind DFE has also been implemented for ADC based receivers in [23]. Combining all of these components together would create an ADC based blind receiver including adaptation.

5.2.2. Study of effects of blind vs locked CDR operation

As described in Section 4.2 the system was originally designed to work with a phase aligned CDR but later was determined that it could still adapt with a frequency offset. The reason why the system adapts has been explained, however, the trade offs can be further explored.

The main benefit of a locked CDR would relate to adaptation time. Since the frequency offset causes temporary abrupt gains in the equalizer settings, this increases the total adaptation time before the coefficients stabilize. There should be a correlation with the amount of frequency offset and the increase in equalization time due to the blind clock.

5.2.3. Equalizing channels with poorly behaved frequency response

All the measurements provided in Chapter 4 were for low-pass response channels without any nulls in the frequency response. To verify the functionality of the algorithm further, different channels can be used with nulls placed at f_N and $f_N/2$. The system's adaptation behaviour can be observed for these cases as well as the final adapted values of C1 and C2. If there is severe attenuation at $f_N/2$ the gain coefficient C2 should increase drastically to try and compensate for lack of a specific type of pattern.

References

- [1] John F Bulzacchelli, Alexander V Rylyakov, and Daniel J Friedman. Power-Efficient Decision-Feedback Equalizers for Multi-Gb / s CMOS Serial Links. *RFIC*, pages 507–510, 2007.
- [2] Azita Emami-neyestanak, Aida Varzaghani, John F Bulzacchelli, Alexander Rylyakov, Chih-kong Ken Yang, and Daniel J Friedman. A 6.0-mW 10.0-Gb/s Receiver With Switched-Capacitor Summation DFE. *IEEE International Solid-State Circuits Conference*, 42(4):889–896, 2007.
- [3] Ravinder S Kajley, Paul J Hurst, and James E C Brown. A Mixed-Signal Decision-Feedback Equalizer That Uses a Look-Ahead Architecture. *IEEE International Solid-State Circuits Conference*, 32(3):450–459, 1997.
- [4] Koon-lun Jackie Wong, Alexander Rylyakov, and Chih-kong Ken Yang. A 5-mW 6-Gb/s Quarter-Rate Sampling Receiver With a 2-Tap DFE Using Soft Decisions. *IEEE International Solid-State Circuits Conference*, 42(4):881–888, 2007.
- [5] Bahzad Razavi. *Design of Integrated Circuits for Optical Communications*. McGraw-Hill Science, 2002.
- [6] Srikanth Gondi, Jri Lee, Daishi Takeuchi, and Behzad Razavi. A 10Gb/s CMOS Adaptive Equalizer for Backplane Applications. In *IEEE International Solid-State Circuits Conference*, pages 2004–2006, 2005.

- [7] Alan J Baker. An Adaptive Cable Equalizer for Serial Digital Video Rates to 400Mb/s. In *IEEE International Solid-State Circuits Conference*, pages 10–12, 1996.
- [8] Mohammad Hossein Shakiba. A 2.5Gb/s Adaptive Cable Equalizer. In *IEEE International Solid-State Circuits Conference*, pages 3–5, 1999.
- [9] Guangyu Zhang, Pruthvi Chaudhari, and Michael M Green. A BiCMOS 10Gb/s Adaptive Cable Equalizer. *IEEE International Solid-State Circuits Conference*, (May 1998):343–346, 2004.
- [10] S. Gondi and B. Razavi. Equalization and Clock and Data Recovery Techniques for 10-Gb/s CMOS Serial-Link Receivers. *IEEE Journal of Solid-State Circuits*, 42(9):1999–2011, September 2007.
- [11] Jri Lee. A 20Gb/s Adaptive Equalizer in 0.13m CMOS Technology. In *IEEE International Solid-State Circuits Conference*, 2006.
- [12] Hye-Yoon Joo, Kyung-Soo Ha, and Lee-Sup Kim. A Data-Pattern-Tolerant Adaptive Equalizer Using the Spectrum Balancing Method. In *IEEE Symposium on VLSI Circuits*, volume 57, pages 228–232, March 2010.
- [13] J. W. M. Bergmans. *Digital Baseband Transmission and Recording*. Dordrecht, Netherlands: Kluwer Academic, 1996.
- [14] E-hung Chen, Jihong Ren, Brian Leibowitz, Hae-chang Lee, Qi Lin, Kyung Suk (Dan) Oh, Frank Lambrecht, Vladimir Stojanovic, Jared Zerbe, and Chih-kong Ken Yang. Near-Optimal Equalizer and Timing Adaptation for I / O Links Using a BER-Based Metric. *IEEE Journal of Solid-State Circuits*, 43(9):2144–2156, 2008.
- [15] Yasuo Hidaka, Weixin Gai, Akira Hattori, Takeshi Horie, Jian Jiang, Kouichi Kanda, Yoichi Koyanagi, and Satoshi Matsubara. A 4-Channel 3.1/10.3Gb/s Transceiver

- Macro with a Pattern Tolerant Adaptive Equalizer. In *IEEE International Solid-State Circuits Conference*, pages 31–32, 2007.
- [16] Scott Mcleod, Ali Sheikholeslami, Takuji Yamamoto, Nikola Nedovic, Hirotaka Tamura, and William W Walker. A Digital Offset-Compensation Scheme for an LA and CDR in 65-nm CMOS. In *IEEE Symposium on VLSI Circuits*, pages 16–17, 2009.
- [17] J. D. H. Alexander. Clock Recovery From Random Binary Signals. *Electronics Letters*, 11:541–542, 1975.
- [18] Shayan Shahramian, Clifford Ting, Ali Sheikholeslami, Hirotaka Tamura, and Masaya Kibune. A Pattern-Guided Adaptive Equalizer in 65nm CMOS. In *IEEE International Solid-State Circuits Conference*, volume 43, pages 188–189, 2011.
- [19] 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-um CMOS. *IEEE Journal of Solid-State Circuits*, 40(4):986–993, April 2005.
- [20] Chih-fan Liao and Shen-iuan Liu. A 40 Gb/s CMOS Serial-Link Receiver With Adaptive Equalization and Clock/Data Recovery. *IEEE Journal of Solid-State Circuits*, 43(11):2492–2502, 2008.
- [21] Yasuo Hidaka, Weixin Gai, Takeshi Horie, Jian Hong Jiang, Yoichi Koyanagi, and Hideki Osone. A 4-Channel 10.3Gb/s Backplane Transceiver Macro with 35dB Equalizer and Sign-Based Zero-Forcing Adaptive Control. In *IEEE International Solid-State Circuits Conference*, pages 188–190, 2009.
- [22] Oleksiy Tyshchenko, Ali Sheikholeslami, Hirotaka Tamura, Masaya Kibune, Hisakatsu Yamaguchi, and Junji Ogawa. A 5-Gb/s ADC-Based Feed-Forward CDR in 65nm CMOS. *IEEE Journal of Solid-State Circuits*, 45(6):1091–1098, 2010.

- [23] Siamak Sarvari, Tina Tahmoureszadeh, Ali Sheikholeslami, Hirotaka Tamura, and Masaya Kibune. A 5Gb/s Speculative DFE for 2x Blind ADC-based Receivers in 65-nm CMOS. In *IEEE Symposium on VLSI Circuits*, volume 2, pages 6–7, 2010.