

# Energy-Efficient Equalization Circuits for High-Speed Wireline Links

*Yue Lu*



Electrical Engineering and Computer Sciences  
University of California at Berkeley

Technical Report No. UCB/EECS-2016-178

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-178.html>

December 1, 2016

Copyright © 2016, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**Energy-Efficient Equalization Circuits for High-Speed Wireline Links**

by

Yue Lu

B.E. (Shanghai Jiao Tong University) 2008

A dissertation submitted in partial satisfaction  
of the requirements for the degree of  
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Elad Alon, Chair  
Professor Seth R. Sanders  
Professor Paul K. Wright

Fall 2014

The dissertation of Yue Lu is approved.

---

Chair

Date

---

Date

---

Date

University of California, Berkeley  
Fall 2014

Energy-Efficient Equalization Circuits for High-Speed Wireline Links

Copyright © 2014

by

Yue Lu

## Abstract

Energy-Efficient Equalization Circuits for High-Speed Wireline Links

by

Yue Lu

Doctor of Philosophy in Engineering – Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Elad Alon, Chair

The explosive development of various computation and communication platforms has demanded the per-pin I/O bandwidth of wireline links to increase at a commensurate rate, with projections to reach 60+Gb/s in less than 10 years. However, unlike the data-rate requirement, the power consumption of these links cannot increase, making improvements in the energy-efficiency of high-speed architectures and circuits crucial. In particular, equalization circuits such as feed-forward equalizers (FFE) and decision-feedback equalizers (DFE) are faced with the burden of compensating higher channel losses while running at faster speeds – all without allowing any increase in power consumption.

To address these challenges, this thesis first presents a feed-forward equalizing transmitter that utilizes voltage-mode signaling along with a shunting branch technique to improve signaling power. Due to the linear mapping between the equalization strength and output impedance segmentation from this technique, the associated decoding logic can be greatly simplified, and, thus, digital power overhead can be substantially reduced. Regulator-based impedance tracking loops are utilized to reduce the parasitics on the high-speed digital path to further reduce the digital power. A 2-tap prototype based on this architecture was taped out in Fujitsu 65nm LP CMOS process and achieves an overall of efficiency of 1pJ/b when operating at 10Gb/s with 200mV differential output signal amplitude.

To enable energy-efficient DFEs at even higher speeds, this thesis will then describe means to reduce the latency of the circuits within the intrinsic feedback loop of such equalizers. Specifically, techniques such as merged latch and summer, optimized signaling swing, and dynamic latches are combined to enable a multi-tap closed-loop DFE architecture that is capable of running at 60+Gb/s. A 3-tap prototype chip was fabricated in TSMC 65nm GP CMOS process, achieving ~0.7pJ/b at 66Gb/s when cancelling a total inter-symbol interference of ~1.65x of the cursor amplitude. This

design represents by far the fastest DFE demonstrated to date with energy efficiency of better than 1pJ/bit, and highlights that the adoption of such techniques may pave the way forward for continued electrical I/O data-rate scaling.

After the introduction of these equalization circuits, this thesis will present a holistic link evaluation framework that aims to achieve more accurate power and performance estimation of link architectures at the beginning of a link design phase. With a compact circuit modeling methodology, circuit power and noise can be explicitly expressed in terms of both technology and system parameters (e.g. equalization-related parameters) such that a link's overall power and bit-error-rate can be directly estimated. An evaluation example shows that using TSMC 65nm GP CMOS process, it is possible to achieve a 64Gb/s data communication speed over a 1m long coax cable-based platform, which aligns with the previous DFE experimental results and thus proves the potentials of using this framework to guide the continued wireline data-rate scaling.

To My Family



# Contents

Contents .....	ii
List of Figures .....	v
List of Tables .....	ix
Acknowledgements .....	x
Chapter 1 Introduction .....	1
1.1 Background .....	1
1.2 Equalizers .....	3
1.2.1 Feed-forward Equalizer (FFE) .....	3
1.2.2 Decision-feedback Equalizer (DFE) .....	5
1.3 Thesis Organization .....	6
Chapter 2 Design Techniques for Energy-efficient Voltage-Mode Feed-Forward Equalizer Transmitters .....	8
2.1 Current-Mode Driver vs. Voltage-Mode Driver .....	8
2.1.1 VM Driver Signaling Power Advantage .....	8
2.1.2 FFE VM Transmitter Design Challenges .....	10
2.2 FFE Voltage-Mode Transmitter Power Analysis .....	11
2.2.1 Signaling Power Analysis .....	11
2.2.2 Digital Power Analysis .....	14
2.3 Proposed Pre-Emphasis Voltage-Mode Transmitter .....	19
2.4 Transmitter Architecture and Circuit Implementation .....	21
2.4.1 Choice of Impedance Calibration Schemes .....	21
2.4.2 Overall Architecture .....	22
2.4.3 Pre-emphasis Decoder .....	22
2.4.4 Driver Segments .....	23
2.4.5 Impedance Control Loop with Online Comparator Offset Calibration .....	24
2.4.6 Output Amplitude Control Loop .....	26

2.5 Measurement Results .....	27
2.6 Conclusions.....	32
Chapter 3 Design Techniques for Multi-Tap Energy-Efficient Decision Feedback Equalizers.....	33
3.1 Multi-tap Design Challenges .....	33
3.1.1 Closed-loop DFE Timing Constraints.....	33
3.1.2 Loop-unrolling Limitations .....	34
3.2 Proposed Closed-Loop DFE .....	37
3.2.1 Optimization #1: Merged Summer/Latch.....	37
3.2.2 Optimization #2: Reduced Latch Gain.....	38
3.2.3 Optimization # 3: Dynamic Latch Design .....	39
3.2.4 Noise Analysis and Implications .....	43
3.3 Complete 3-Tap DFE Circuit Design.....	46
3.3.1 Dynamic Latch Circuit and Practical Issues.....	47
3.3.2 Overall 3-Tap DFE Architecture .....	49
3.3.3 Clock Distribution.....	51
3.4 Complete test-chip and Measurement.....	52
3.4.1 On-chip Testing Structures and Test Setup .....	53
3.4.2 Measurement Results .....	55
3.5 Conclusion.....	60
Chapter 4 A Holistic Link Evaluation Platform .....	61
4.1 Holistic Evaluation with Circuit Modeling and Device Constraints.....	61
4.1.1 Circuit Component Modeling .....	62
4.1.2 Building Block Interactions .....	65
4.2 Complete Link Framework and Examples .....	68
4.2.1 Signal Flow and Type Conversion.....	68
4.2.2 Complete Framework.....	69
4.2.3 Example: Signaling Platform for 2025 .....	72
4.3 Conclusion.....	75
Chapter 5 Conclusions .....	77

5.1 Thesis Summary .....	77
5.2 Future Directions.....	78
Bibliography.....	79
Appendix A Analysis of PEVM Transmitters.....	87
A.1 CVPEVM .....	87
A.2 CIPEVM .....	88
A.3 IMPEVM .....	88
A.4 Proposed PEVM.....	89
Appendix B Comparison of Dynamic Latch and CML Latch Implementations.....	90
Appendix C DFE Error Propagation Analysis .....	93
C.1 Error Propagation Analysis for 1-tap DFE .....	93
C.2 A Brief Summary for Multi-Tap DFE.....	97

## List of Figures

Figure 1.1: examples of wireline links (source: internet) .....	1
Figure 1.2: per-pin data rate requirement prediction.....	2
Figure 1.3: published efficiency vs. data rate for 10+Gb/s electrical links .....	2
Figure 1.4: a general wireline link architecture .....	3
Figure 1.5: a 3-tap (1-pre and 2-post taps) FFE equalizer implemented on (a) RX side and (b) TX side.....	4
Figure 1.6: a conceptually ideal equalizer (ignored channel's propagation delay for simplicity) .....	5
Figure 1.7: a 2-tap DFE equalizer.....	6
Figure 2.1: (a) simplified model for a terminated CML driver and (b) simplified model for a terminated VM driver when sending a bit "1" .....	9
Figure 2.2: (a) simplified model for CML transmitter and (b) simplified model for VM transmitter .....	10
Figure 2.3: conventional pre-emphasis scheme .....	12
Figure 2.4: (a) constant-current pre-emphasis scheme and (b) its equivalent circuit .....	13
Figure 2.5: (a) impedance modulated pre-emphasis scheme and (b) its equivalent circuit .....	13
Figure 2.6: signaling power comparison of previous drivers.....	14
Figure 2.7: switched branch conductance vs $V_{out}$ for different drivers.....	17
Figure 2.8: normalized digital power vs. # of segments.....	18
Figure 2.9: number of segments vs. $V_{out}$ resolution for different designs.....	18
Figure 2.10: equivalent circuit of proposed PEVM.....	19
Figure 2.11: signaling power comparison .....	20
Figure 2.12: number of segments vs $V_{out}$ resolution .....	20
Figure 2.13: (a) "digital" impedance calibration scheme and (b) "analog" impedance calibration scheme .....	21
Figure 2.14: overall transmitter architecture.....	22
Figure 2.15: decoder and segment schematic.....	23
Figure 2.16: schematic of impedance control loop .....	24
Figure 2.17: implementation of comparator-based regulator loop.....	25
Figure 2.18: illustration of interleaved comparator offset calibration and regulation process: (a) comparator auto-zeroing (calibration) phase and (b) regulation phase.....	26
Figure 2.19: schematic for swing control loop.....	27

Figure 2.20: die photo and TX floor plan .....	27
Figure 2.21: 10" FR4 PCB trace (a) S21 and (b) simulated 10Gb/s pulse response.....	28
Figure 2.22: $2^{23}$ -1 PRBS eye before and after 10" trace with post-tap pre-emphasis turned off (100mV/div vertical, 20ps/div horizontal) .....	28
Figure 2.23: $2^{23}$ -1 PRBS eye before and after 10" trace with post-tap pre-emphasis turned on (100mV/div vertical, 20ps/div horizontal) .....	29
Figure 2.24: TX output swing tuning curve.....	29
Figure 2.25: TX output impedance vs. signal amplitude .....	30
Figure 2.26: signaling power with 250mV differential amplitude swing vs. pre-emphasis setting .....	30
Figure 2.27: TX power vs. output swing and data-rate .....	31
Figure 3.1: (a) full-data-rate 1-tap closed-loop DFE and (b) double-data-rate 1-tap closed-loop DFE architectures.....	34
Figure 3.2: 1-tap loop-unrolled FDR DFE architecture.....	35
Figure 3.3: 1-tap loop-unrolled DDR DFE architecture.....	35
Figure 3.4: 2nd tap critical timing path for (a) 1st tap unrolled and 2nd tap closed-loop DFE and (b) 2-tap closed-loop DFE .....	36
Figure 3.5: illustration of optimization 1 – merged latch and summer .....	37
Figure 3.6: illustration of optimization 2 step 1 – reducing the required output digital level.....	38
Figure 3.7: illustration of optimization 2 step 2 – increasing the input analog level.....	39
Figure 3.8: closed-loop 1st tap with dynamic latch .....	40
Figure 3.9: 1st tap operation waveforms with an input data pattern of 10010 that is distorted by a single tap of post-cursor ISI.....	41
Figure 3.10: close-loop 1st tap with CML latch .....	42
Figure 3.11: power vs. data rate for both CML and dynamic latch-based designs for various total gain requirements.....	42
Figure 3.12: illustration of noise shaping and propagation in the DFE feedback loop .....	43
Figure 3.13: two examples of noise propagation – S1: the latch output signal level lands within the high-gain region of the feedback pair's VTC, and S2: the latch output level lands within the clipping region .....	44
Figure 3.14: noise PDF after 1 and 10 iterations for the two examples in Figure 3.13.....	45
Figure 3.15: noise gain vs. number of propagations with input SNR (i.e. $V_{O,L}/\sigma_{vni}$ ) = 8.45	45
Figure 3.16: noise enhancement vs. latch output signal level under various input SNRs	46
Figure 3.17: amplitude droop due to dynamic latch leakage vs. clock frequency.....	47
Figure 3.18: (a) dynamic latch without tail node reset and (b) with tail reset device to improve the latch's aperture .....	48
Figure 3.19: (a) adjusting the 1st tap correction current $I_{cor}$ by changing the DC gate-bias $V_G$ , and (b) $I_{cor}$ vs. $V_G$ .....	48

Figure 3.20: complete proposed 3-tap DFE design .....	49
Figure 3.21: simulated post-layout eye-diagrams in the TT corner at each node of the 3-tap DFE.....	50
Figure 3.22: simulated post-layout eye-diagrams in the SS and FF corners at the latched buffer (b1) output.....	50
Figure 3.23: simulated post-layout pulse response at the output of the buffer .....	51
Figure 3.24: DFE clock distribution network.....	52
Figure 3.25: complete 65nm GP test-chip including the proposed DFE and on-chip transmitter with channel emulation .....	52
Figure 3.26: transmitter design with DDR PRBS-7 generator and low-pass channel emulation.....	54
Figure 3.27: test-chip die photo.....	55
Figure 3.28: (a) 66Gb/s PRBS-7 single-ended eye-diagram before DFE, (b) 33Gb/s PRBS-7 single-ended eye-diagram after DFE.....	56
Figure 3.29: (a) 66Gb/s single-ended data waveform before DFE, and (b) 33Gb/s single-ended data waveform after DFE.....	56
Figure 3.30: circuit configuration used to characterize the feedback pair's clipping voltage.....	57
Figure 3.31: measured overall VTC.....	57
Figure 3.32: measured bathtub curve with a 90mV differential amplitude input signal...	58
Figure 3.33: DFE power breakdown.....	59
Figure 4.1: a pre-amplifier example.....	62
Figure 4.2: a source-degenerated CTLE example .....	64
Figure 4.3: a RX with CTLE + DFE example.....	66
Figure 4.4: block interactions without offset cancellation.....	67
Figure 4.5: block interactions with offset cancellation.....	68
Figure 4.6: signal flow and type conversion example.....	69
Figure 4.7: overall framework organization .....	70
Figure 4.8: matlab code organization for the CTLE block.....	72
Figure 4.9: Signaling Platform for 2025 (from Intel).....	73
Figure 4.10: a 1-m cable channel for the 64Gb/s link.....	73
Figure 4.11: (a) frequency response of the overall channel and (b) 64Gb/s pulse response with 250mV differential amplitude .....	74
Figure 4.12: channel equalization using 10-tap DFE only - (a) equalized pulse response and (b) residual ISI distribution.....	74
Figure 4.13: link architecture example with unterminated TX driver + 4-tap FFE + 2-stage pre-amp + 2-tap DFE – (a) equalized pulse response and (b) residual ISI distribution.....	75
Figure A.1: Thevenin equivalent circuit for driver analysis .....	87

Figure C.1: 1-tap DFE model.....	93
Figure C.2: 1st-order discrete Markov chain model for the error state transitions.....	94
Figure C.3: steady-state BER vs input SNR wi/wo error propagation.....	96
Figure C.4: BER vs. time with different initial states .....	97
Figure C.5: 1st-order discrete Markov chain model for a 2-tap DFE.....	98
Figure C.6: 1st-order reduction of STD for the 1-tap DFE example.....	99

## List of Tables

Table 2.1: pre-emphasis voltage-mode transmitter comparison .....	32
Table 3.1: comparison to state-of-the-art 20+Gb/s DFE designs .....	59
Table 4.1: reference item list of the technology LUT .....	71



## Acknowledgements

I cannot imagine a better time than the past 6 years I have spent at UC Berkeley as a graduate student. I feel extremely fortunate to have received so much help and support from so many individuals, without whom I can never reach this finish line of my Ph.D journey. I would like to take this opportunity to express my most sincere gratitude to every one of these companions.

I cannot feel luckier to have Prof. Elad Alon as my advisor. Elad's knowledge, patience as well as his inspiring lectures paved the path for me to become a good IC designer. I can hardly think of a better advisor who is always willing to discuss with me whenever I come up with new ideas and provides all sorts of support to implement them. I still remember the hours of technical discussions we had over the phone, some of those lasted deep into the night even when Elad needed to travel early in the next morning. Without his insightful feedback, I could never conquer the obstacles I had faced during my Ph.D studies. My gratitude to Elad is not limited to just research. He was always happy to share with me his experiences working in both industry and academia, and helped me resolve my concerns and difficult situations I encountered, all of which I found very beneficial to shape a better view of my future career.

I would like to thank Prof. Jan Rabaey, Prof. Seth Sanders and Prof. Paul Wright for being on my thesis and qualification exam committee. Their valuable feedback helped me to reshape my ideas when conducting research. I want to thank Prof. Ali Niknejad for his wonderful RF courses and the enjoyable soccer games he invited me to join. I also want to thank Prof. Haideh Khorramabadi and Prof. Borivoje Nikolic for teaching my ADC and digital IC design classes.

Throughout my Ph.D, I have got the opportunities to collaborate with many talents from various research institutes. I would like to thank Ming-Shuan Chen, Amr Hafez and Prof. Ken Yang of UCLA, Kangmin Hu, Rui Bai and Prof. Patrick Chang of Oregon State University, Amr Suleiman of MIT, Prof. Vladimir Stojanovic of UC Berkeley and Prof. Pavan Hanumolu of UIUC for the fruitful discussions and their insightful comments/feedback on my research. I certainly cannot get my work done without the gracious support from our industry partners and therefore I owe my thanks to Bryan Casper, Tanay Karnik, Mondira Pant of Intel, Yasuo Hidaka and William Walker of Fujitsu Laboratories of America as well as Ken Chang and Yohan Frans of Xilinx. I have met many great people in industry who helped me improve my technical strength. In particular, I would like to thank Jared Zerbe, Brian Leibowitz and Wenbo Liu of Apple, Jihong Ren of Altera and Yueyong Wang of Broadcom.

Berkeley Wireless Research Center is a fabulous place with a group of amazing people. I want to thank all the BWRC staff and especially Tom Boot, Brian Richards,

Fred Burghardt, Bira Coelho, James Dunn, Leslie Nishiyama, Olivia Nolan and Sarah Jordan, without whom I cannot finish my projects so smoothly.

I learned a lot of analog/mixed-signal/RF/mm-wave knowledge beyond my research topic from a lot of senior BWRC students: Jiashu Chen, Lingkai Kong, Lu Ye, Chintan Thakkar, Hanh-Phuc Le, John Crossley, Amin Arbabian, Peng Liu, Yida Duan, Wenting Zhou, Jungdong Park, Ashkan Borna, Tsung-Te Liu, Stanley Chen, Namseog Kim, Kyoohyun Noh, Rikky Muller, Simone Gambini, Zhiming Deng, Richard Su and Ping-Chen Huang. I would like to thank all of them for spending their precious time with me.

I would also like to thank my fellow BWRC colleagues - Jun-Chau Chien, Kwangmo Jung, Siva Thyagarajan, Wen Li, Alberto Puggeli, Shingwon Kang, Charles Wu, Matthew Spencer, Jaehwa Kwak, Steven Callendar, Daniel Yeager, William Biederman, Jaeduk Han, Nathan Narevsky, Nai-Chung Kuo, Dongjin Seo, Christopher Sutardja, Nicholas Sutardja, Pengpeng Lu and Constantine Sideris - for working with me on various projects, discussing new ideas and having fun together.

Getting admitted to UC Berkeley for graduate school was the most exciting news for me as an undergraduate. I would like to thank my undergraduate advisor, Prof. Alex Lee for teaching me analog circuit design and writing me an enthusiastic recommendation letter. I want to thank Prof. David Ricketts for providing me the research opportunities when I was an exchange student at Carnegie Mellon University in my junior year, without which I could not discover my interests in integrated circuit design.

I would like to thank my parents for always supporting me to pursue my dream. Even though we are separated far away by the Pacific Ocean, I can still feel the warmth of their love. I owe my thanks to my dearest wife - Qinru Chu. Without her sacrifice and constant support, I can never get through the hardest time of my Ph.D journey.

# Chapter 1

## Introduction

### 1.1 Background

Wireline links are ubiquitous. As shown in Figure 1.1, almost every communication platform requires wireline link to be running in the background. Unsurprisingly, with the explosive development of applications such as cloud storage/computation and sensor networks the future will only witness an increasing demand of high-speed wireline communications.

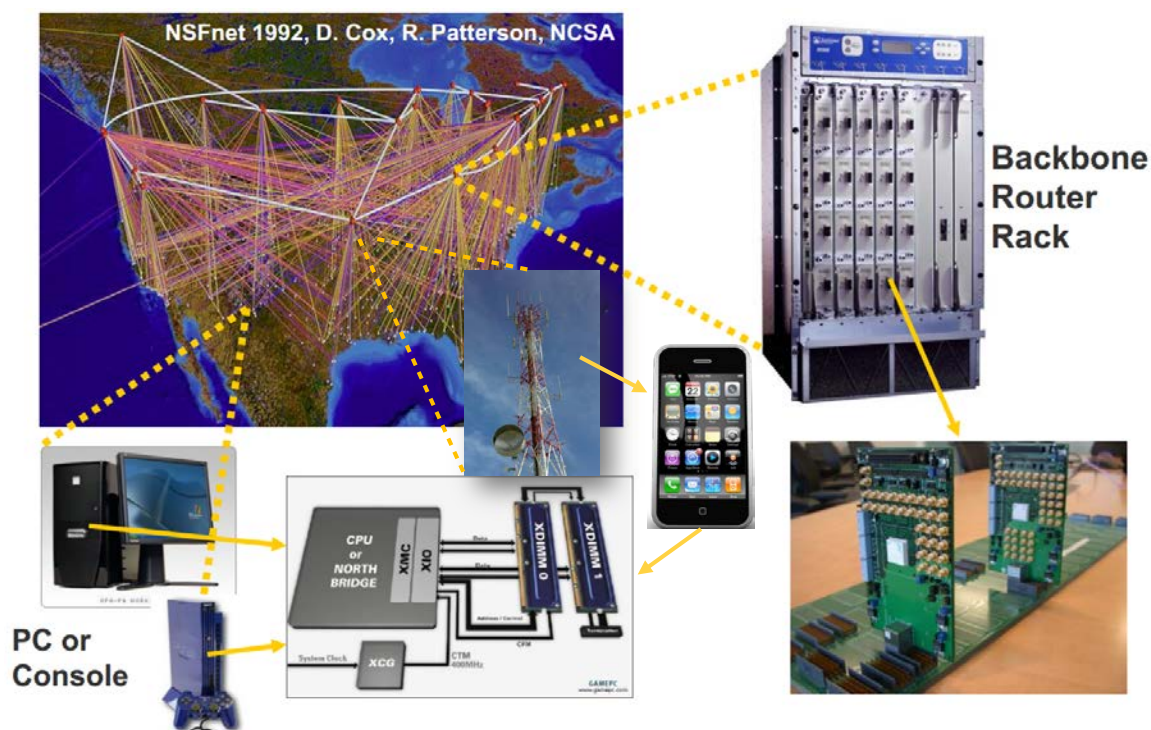


Figure 1.1: examples of wireline links (source: internet)

To get a general trend and estimation of how fast links will need to be, the data compiled from the International Technology Roadmap for Semiconductors (ITRS) can be used as a reference. As shown in Figure 1.2, the requirement on per pin throughput of I/Os will increase at 1.1-1.2X/year, reaching 70+Gb/s in about 10 years ([1]). On the other hand, while the throughput of the I/Os keeps on increasing, their absolute power budget should remain essentially unchanged (<10% as estimated in [2]–[4]), requiring the energy-efficiency of these I/Os to be improved ([5], [6]).

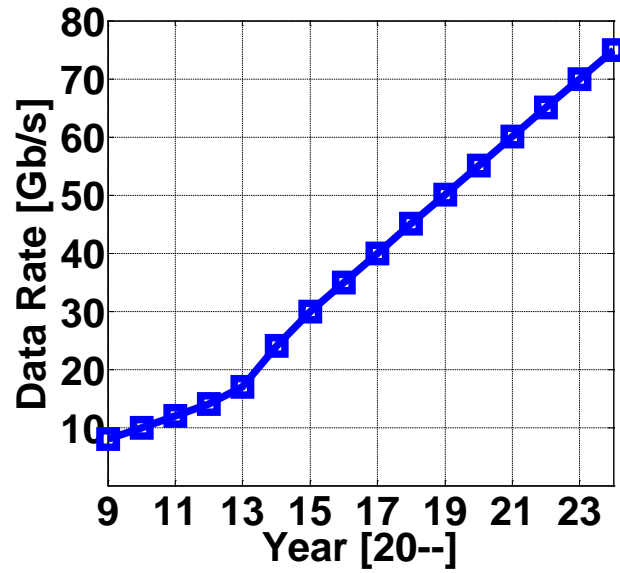


Figure 1.2: per-pin data rate requirement prediction

Driven by this need, many researchers have developed promising approaches to push the limits of building energy-efficient high-speed I/Os. As shown in Figure 1.3, published papers from major conferences such as ISSCC, VLSI and CICC have shown that most designs below 30Gb/s can achieve sub-10pJ/b or even sub-1pJ/b energy-efficiency. On the other hand, while links operating at 30+Gb/s and even 40+Gb/s have also been demonstrated, their energy efficiencies are still clustered at the region beyond 10pJ/b. This data indicates there is still much research on both system and circuit levels to be done to further optimize current high-speed link platforms.

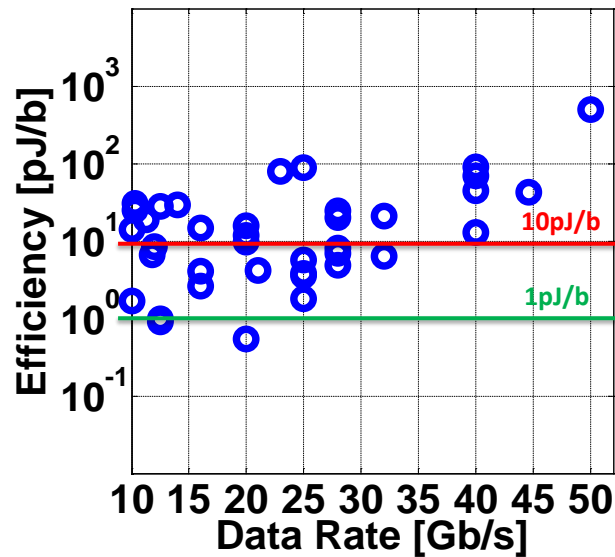


Figure 1.3: published efficiency vs. data rate for 10+Gb/s electrical links

## 1.2 Equalizers

To figure out the power/speed bottlenecks of wireline links, let's begin by examining a typical link architecture as shown in Figure 1.4. Due to the skin effect and dielectric losses, electrical channels often introduce large inter-symbol interference (ISI) that can lead to errors on the receiver side ([7]). To deal with this ISI, equalization circuits – specifically feed-forward equalizers (FFE) and decision-feedback equalizers (DFE) – are widely adopted to restore the transmitted bits on the receiver side to bring down the bit-error-rate to a certain targeted low level (e.g.  $1e-12$ ).

Since the ISI is speed dependent, higher data rate operation will only require faster (i.e. higher circuit bandwidth) and stronger (i.e. larger circuit loading) TX and RX equalizations, making transceiver equalizers a critical and potentially power hungry portion of the transceiver ([5], [8]). For this reason, this thesis will focus on circuit design techniques and modeling methodologies for building energy-efficient equalizers and choosing optimum link architectures such that we can keep on improving link's energy efficiency when we keep on increasing data rates. Before diving into the details, however, some basics of the FFE and DFE need to be introduced.

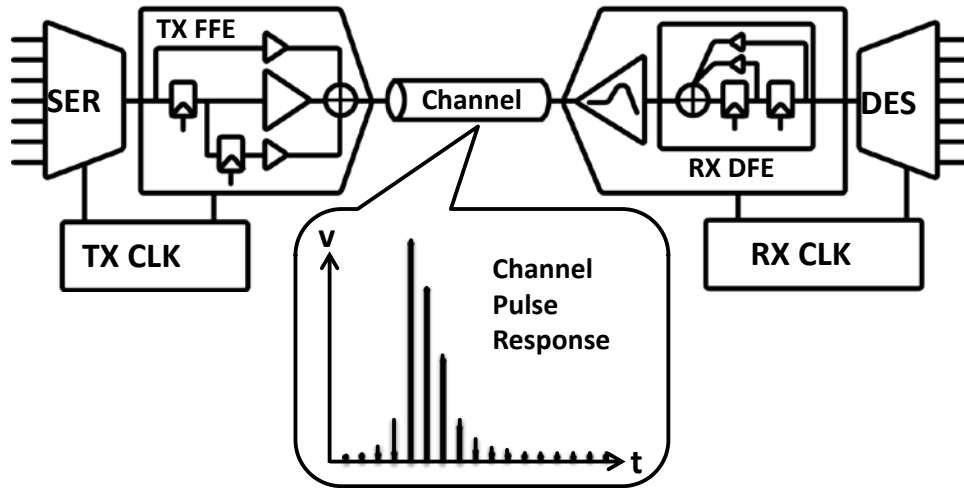


Figure 1.4: a general wireline link architecture

### 1.2.1 Feed-forward Equalizer (FFE)

The basic structure of a FFE can be illustrated in Figure 1.5. While it can be implemented on either RX side (Figure 1.5(a)) or TX side (Figure 1.5(b)),<sup>1</sup> the basic working principle of the FFE remains the same. It generates multi-path signals (i.e. the main and correction

<sup>1</sup> There may be some power and design complexity trade-offs between implementing them on the two sides ([13], [73]).

taps) by applying proper delays from the same signal source before linearly scaling<sup>2</sup> and combining them to cancel the channel ISI.

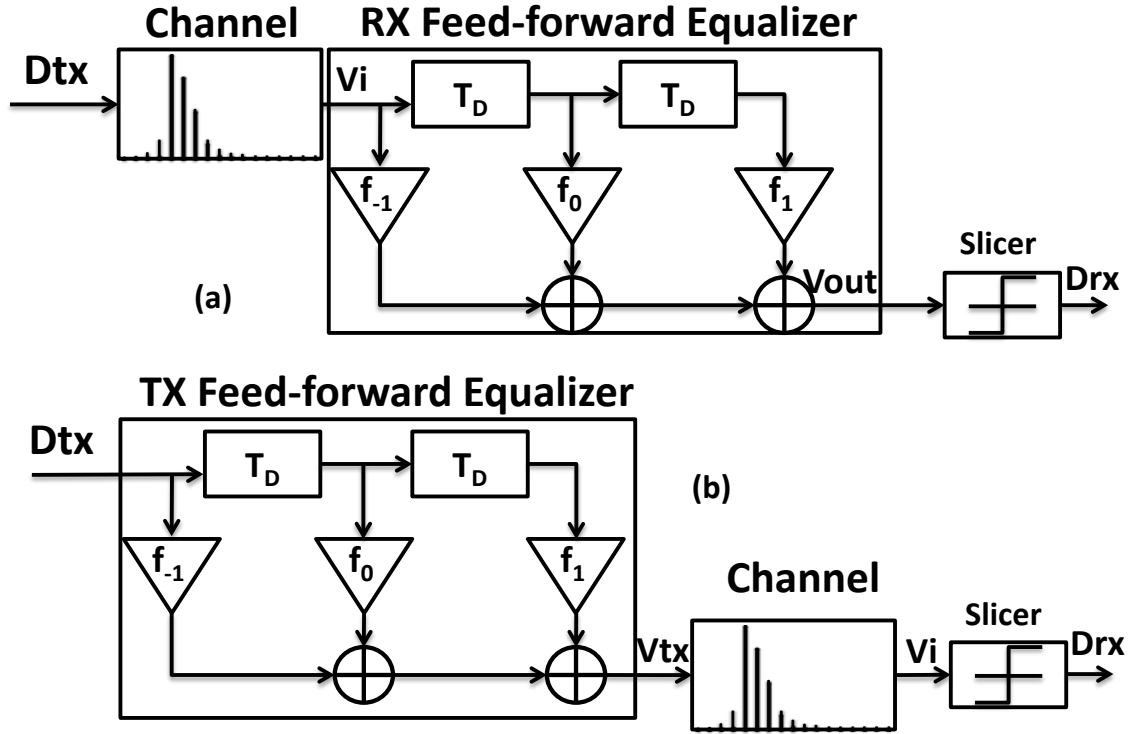


Figure 1.5: a 3-tap (1-pre and 2-post taps) FFE equalizer implemented on (a) RX side and (b) TX side

To further investigate the FFE's linear property, let's write down the difference equation of a simple 2-tap 1-UI spaced FFE with 1 post-tap correction:

$$V_{out}[k] = f_0 \cdot V_{in}[k] - f_1 \cdot V_{in}[k - 1] \quad (1.1)$$

where  $f_0$  and  $f_1$  are the coefficients for the main tap (i.e. current bit) and correction tap (i.e. previous bit) and both are assumed to be positive. The negative sign before  $f_1$  is to implement the high-pass filter function for the FFE such that it can "inverse" or "equalize" the low-pass characteristic of the channel.

When the previous bit and the current bit are different (i.e.  $V_{in}[k] = -V_{in}[k - 1]$  - a clock pattern for instance) and thus a transition happens, the data contains a high-frequency component that will experience a larger loss from the low-pass channel. With the configuration in (1.1), we will thus get the maximum gain of  $(f_0 + f_1)$  from the FFE. On the other hand, when transmitting/receiving consecutive bits (i.e.  $V_{in}[k] = V_{in}[k - 1]$  - a DC pattern for instance), the FFE will result in a minimum gain of  $(f_0 - f_1)$  for the low-frequency data stream. Due to this  $(f_0 + f_1) / (f_0 - f_1)$  gain boosting, the FFE can

<sup>2</sup> Delay choices include 1-UI based data equalization or 1/2-UI based edge+data equalization or other fractional UI equalization ([63], [74], [75]). Scaling coefficients can be adapted based on different algorithms such as sign-sign LMS([50], [76]) and minimum BER ([77]).

extend the overall channel bandwidth by increasing the channel's high-frequency content to match the overall low-frequency gain. However, since the high-frequency gain introduced by the equalizer has to be greater than its low-frequency gain to extend the bandwidth, an FFE would amplify more high-frequency noise (e.g. cross-talk noise) and thus tends to compromise the overall equalized signal-to-noise ratio (SNR) of the link. The more ISI that needs to be equalized (i.e. larger gain boosting), the more high-frequency noise will be amplified.

### 1.2.2 Decision-feedback Equalizer (DFE)

When we take a closer look at the “noise amplification” problem of an FFE, it actually stems from the fact that both the main and correction bits have to go through the channel loss. In other words, due to its linear filter characteristic, we have to combine the “dirty” correction bits to the main signal to equalize the channel ISI. If in some way we can use “clean” correction bits to apply on the main signal path for ISI cancellation, the channel loss will be equalized without introducing any noise amplification. This thought process essentially leads us to a conceptual equalizer architecture as illustrated in Figure 1.6.

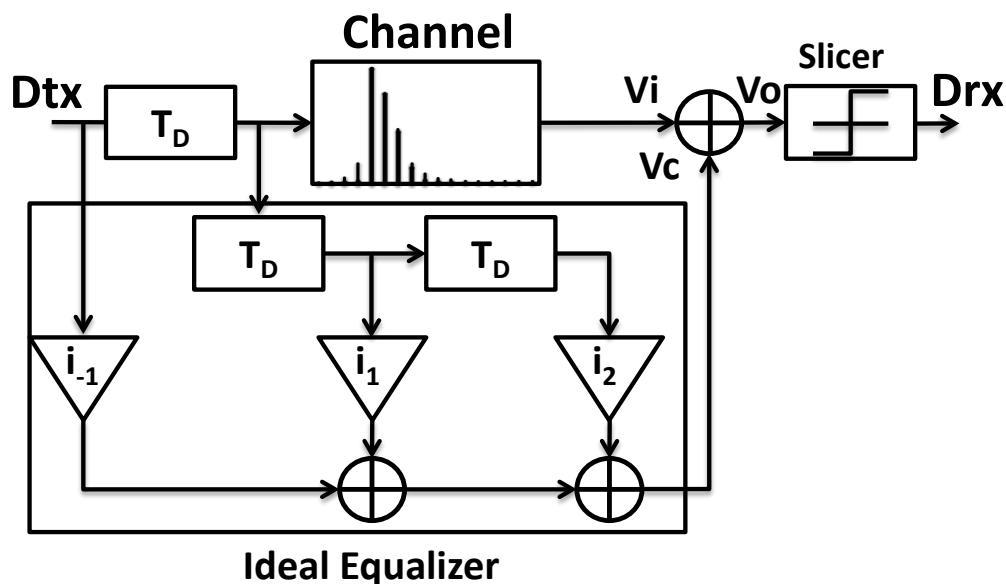


Figure 1.6: a conceptually ideal equalizer (ignored channel's propagation delay for simplicity)

With such a configuration, correction bits are clean digital signals from the transmitter that never get contaminated by the channel loss or any other noise sources, and thus can cancel the channel induced ISI without eating the main signal's dynamic range or introducing any extra noise. Of course, in reality there won't exist such a perfect by-pass channel to send the clean TX data to the RX side or we won't need to build equalizers in the first place. Fortunately, in a practical system, the recovered digital bits

after the RX slicer can serve as good proxies of these clean correction bits, leading to the decision-feedback equalizer architecture as shown in Figure 1.7.<sup>3</sup>

As the slicer essentially erases all the noise and memory effect introduced from the channel, its output can then be treated as the desired channel-independent bit source. However, the penalty we pay for using the received bits as the correction is also clear - due to its feedback nature, a DFE cannot cancel pre-cursor ISI.

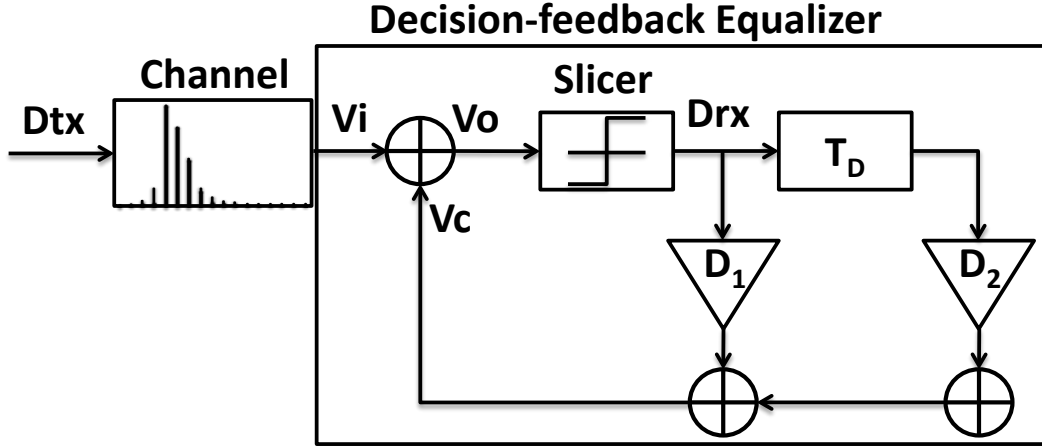


Figure 1.7: a 2-tap DFE equalizer

On the other hand, as we need to utilize the “decisions” to correct post-cursor ISI, one particular concern often raised with DFEs is related to “error-propagation”. This occurs when an incorrect decision is made and causes the feedback correction to be the exact opposite polarity. If an ISI tap is substantial (e.g.  $>0.5$  of the main cursor amplitude), it is very likely the next few bits will continue to be wrong. Fortunately, previous works (e.g [9]) have shown that the differences with and without considering error propagations end up giving negligible steady-state BER differences as often times the target BER would be too low to cause any substantial error accumulation. A brief summary of such an analysis is given in the Appendix C for reference.

## 1.3 Thesis Organization

With the introduction of FFE and DFE in this chapter, the remaining focus of the thesis will then be on the presentation of circuit design techniques to build both energy-efficient feed-forward and decision-feedback equalizers. Starting in Chapter 2, a FFE transmitter architecture that achieves both excellent signaling and digital power consumption will be introduced. Integrated with amplitude control and automatic impedance matching, a 10Gb/s 2-tap transmitter prototype was designed and taped-out in Fujitsu 65nm LP CMOS process and achieved good signaling performance with only 1pJ/bit energy-efficiency. Following that, Chapter 3 introduces a DFE architecture suitable for ultra-high

<sup>3</sup> Unlike a FFE, the tap number definition in a DFE doesn’t include the main-tap. Like an FFE, the delay can also be chosen to be a fraction of UI ([74]).



speed applications. Design techniques and practical considerations will be discussed to implement such a 3-tap DFE capable of running at 66Gb/s while achieving  $\sim 0.7$  pJ/bit efficiency in a standard TSMC 65nm CMOS process. With the knowledge of designing and implementing these equalizers, Chapter 4 presents a holistic link evaluation framework to assess different link architectures. Taking into account both channel and technology device limits, the proposed framework can substantially shorten the entire design cycle by helping system and circuit designers make early decisions on appropriate link architecture choices before running extensive system & circuit simulations. Finally, Chapter 5 summarizes the thesis and points out future directions.

## Chapter 2

# Design Techniques for Energy-efficient Voltage-Mode Feed-Forward Equalizer Transmitters

The pursuit of energy-efficient high-speed links has popularized the use of voltage-mode (VM) transmitters due to their ideally 4x lower signaling power compared to current mode logic (CML) transmitters. These transmitters must typically support impedance matching and pre-emphasis (i.e. FFE function for signal integrity) as well as amplitude control to enable reduce power on clean channels. As will be discussed in more details soon, if not careful enough, the associated digital power overhead when embedding these functions in to a VM transmitter will become dominant and thus compromise the overall energy efficiency. In some cases it may even become more power hungry than a CML design ([10]).

This chapter will analyze the signaling and digital power overhead of pre-emphasis voltage-mode transmitters. After that, an optimized pre-emphasis scheme will be introduced to give both superior signaling and digital power efficiency. Leveraging this technique, a low-power pre-emphasis voltage mode transmitter architecture with output swing control, pre-emphasis coefficient control, and online impedance calibration is proposed and demonstrated. A 65nm LP CMOS implementation of this architecture dissipates only ~10mW from a 1.2V supply when transmitting 10Gb/s 400mV differential peak-to-peak data with a 2-tap FFE function, achieving 1pJ/bit energy efficiency.

## 2.1 Current-Mode Driver vs. Voltage-Mode Driver

### 2.1.1 VM Driver Signaling Power Advantage

To compare the signaling power efficiency of a current-mode driver and a voltage-mode driver, we need to find how much current each driver consumes when sending the same amount of voltage  $V_{sig}$  at the transmitter output. As shown in Figure 2.1(a), for a terminated CML driver where its driver impedance  $R_t$  is equal to the channel characteristic impedance  $Z_0$ , the overall effective load impedance is only  $R_t/2$ . The required steering signaling current to build up a differential amplitude  $V_{sig}$  is then:

$$I_{sig,CML} = \frac{V_{sig}}{R_t/2} = \frac{2V_{sig}}{R_t} \quad (2.1)$$

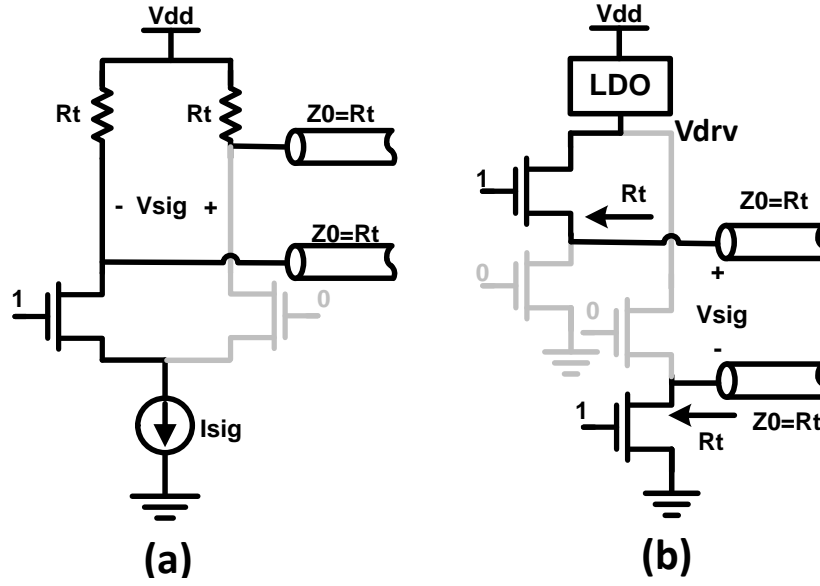


Figure 2.1: (a) simplified model for a terminated CML driver and (b) simplified model for a terminated VM driver when sending a bit “1”

with a supply voltage of  $V_{dd}$ , the signaling power can be computed as:

$$P_{sig,CML} = V_{dd} I_{sig,CML} = \frac{2V_{dd}V_{sig}}{R_t} \quad (2.2)$$

On the other hand, for a VM driver, the mechanism of generating  $V_{sig}$  is very different. Rather than steering current from one-side of the load to the other to create differential voltages, a voltage mode driver essentially forms a voltage divider between the transmitter output and the channel impedance. As shown in Figure 2.1(b), a typical low-swing VM driver requires a low-dropout regulator (LDO) to generate an intermediate voltage  $V_{drv}$  off the supply  $V_{dd}$  in order to build the differential output voltage. In the terminated case, the output impedance of the transmitter is equal to the channel characteristic impedance, and thus the transmitted signal amplitude can be simply expressed as:

$$V_{sig} = \frac{V_{drv}}{R_t + R_t + R_t + R_t} (R_t + R_t) = \frac{V_{drv}}{2} \quad (2.3)$$

and the signaling current can be found as:

$$I_{sig,VM} = \frac{V_{drv}}{4R_t} = \frac{V_{sig}}{2R_t} \quad (2.4)$$

As this current flows from the LDO (i.e. from the supply  $V_{dd}$ ), the signaling power from the supply is then:<sup>4</sup>

<sup>4</sup> A DC-DC converter can also be used to generate  $V_{drv}$ . Depending on its achievable efficiency, it may or may not be more power efficient than a LDO topology.

$$P_{\text{sig,VM}} = V_{\text{dd}} I_{\text{sig,VM}} = \frac{V_{\text{dd}} V_{\text{sig}}}{2R_t} \quad (2.5)$$

Comparing (2.5) and (2.2), we can easily identify that a VM transmitter has the potential advantage of a 4x signaling power efficiency improvement over a CML counterpart.

### 2.1.2 FFE VM Transmitter Design Challenges

While this signaling power benefit looks appealing, we must be careful about the overhead power consumption when integrating the functionalities of equalization, impedance control and swing control to VM transmitter as it requires more complicated controls as compared to a CML transmitter.

To be more specific, as shown in Figure 2.2(a), in a CML driver, since the driver impedance is purely set by the load resistors (ignoring the high-impedance loading from the transistors), amplitude modulation can be simply achieved by replacing the current source with current DACs without compromising termination. However, unlike the CML transmitter whose output swing and impedance are independently set by current DACs and resistors, in a VM transmitter, output swing and impedance are tightly coupled to each other. In other words, as shown in Figure 2.2(b), any changes to the FFE (or pre-emphasis) settings which essentially introduce data-dependent dynamic amplitude modulation will require meticulous controls of the resistor DACs to maintain matched termination. These controls inevitably introduce extra logic on the data-path and thus extra power penalty.

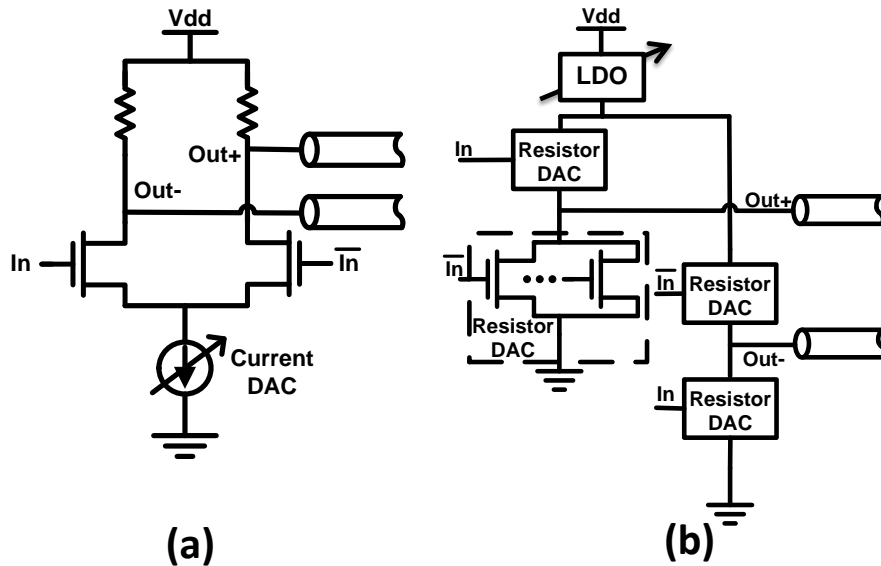


Figure 2.2: (a) simplified model for CML transmitter and (b) simplified model for VM transmitter

At low data-rates where the signaling power consumption is relatively large, the power overhead from extra digital gates on the data-path is mild. However, as the data-rate increases and the digital  $CV^2f$  power rises along with it, the power consumed by the complex pre-driver and per-segment logic necessary to support these feed-forward equalization voltage-mode (FFEVM) TXs can eliminate any benefit from the reduced signaling power. The following sections of the chapter therefore analyze in details both the signaling and digital power consumption of a FFEVM transmitter. With the understanding of this power issue, a TX architecture with a new pre-emphasis scheme will be presented to achieve both good signaling and digital power consumption.

## 2.2 FFE Voltage-Mode Transmitter Power Analysis

Most of the earlier work in FFEVM designs has focused on reducing signaling power overheads due to pre-emphasis. However, if improving the final driver's signaling power results in more parasitic capacitance on the high-speed driver path, the extra digital switching power overhead may easily counteract all of the signaling power savings. Therefore, to improve the overall transmitter power efficiency, both signaling power and digital power must be optimized together. In this section, we will first review how signaling power can be improved by increasing the total supply path impedance. We then analyze the digital power overhead associated with these schemes before proposing our solution. N

### 2.2.1 Signaling Power Analysis

The conventional voltage-mode transmitter design incorporating FFE or pre-emphasis (CVPEVM) was first introduced in [11]. As shown in Figure 2.3 with its equivalent circuit, the output voltage level was lowered (implementing de-emphasis) by creating a path (with a single-ended conductance  $G_{kill}$ ) from  $V_{drv}$  to ground in combination with the main signal path (with a single-ended conductance  $G_{sig}$ ). Since both the  $G_{kill}$  and the  $G_{sig}$  paths are composed of segmented transistors, by moving conductance from one path to the other (i.e. turning on/off transistors in each path) but keeping their sum equal to the channel's characteristic conductance, one can change the output amplitude and hence the pre-emphasis level without sacrificing output termination.

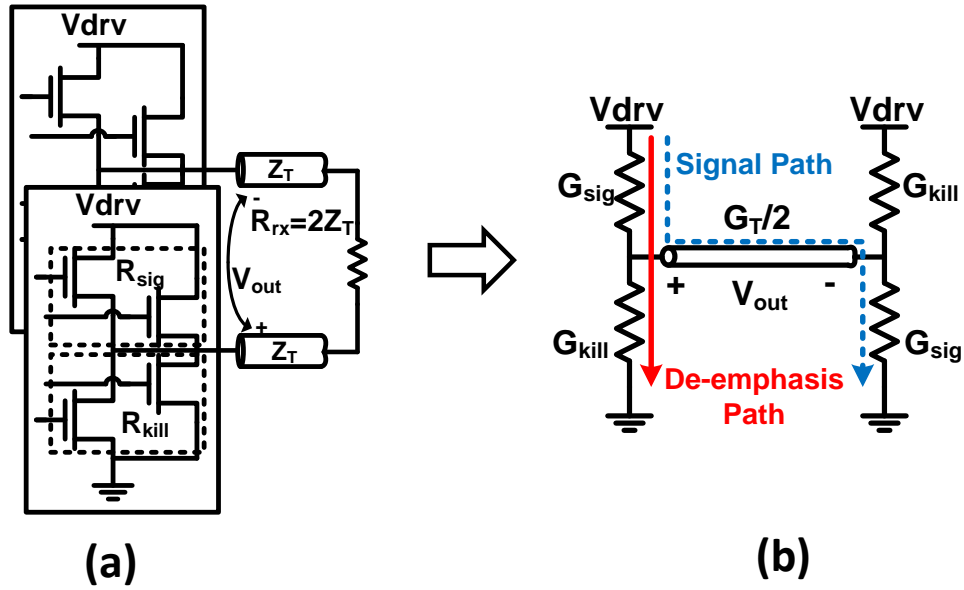


Figure 2.3: conventional pre-emphasis scheme

The major issue with this topology, however, stems from its degraded signaling power efficiency. We can express its signaling current  $I_{\text{sig}}$  in terms of output differential amplitude  $V_{\text{out}}$  as (derived in the Appendix A):

$$I_{\text{sig}} = G_T V_{\text{drv}} \left[ \frac{1}{2} - \left( \frac{V_{\text{out}}}{V_{\text{drv}}} \right)^2 \right] \quad (2.6)$$

where  $G_T$  is the single-ended characteristic conductance of the channel and  $V_{\text{drv}}$  is the supply voltage of the driver. From (2.6), we can see that  $I_{\text{sig}}$  increases when  $V_{\text{out}}$  is reduced. In other words, the efficiency with back-off is poor since more power is consumed when transmitting lower signal power. This occurs because the total supply path impedance from  $V_{\text{drv}}$  to ground is reduced as  $G_{\text{kill}}$  is increased in order to de-emphasize the transmitted signal.

To deal with this particular signaling power issue, an alternative type of driver – which we will refer to as a constant-current pre-emphasis voltage-mode driver (CIPEVM) – was introduced in [10] to maintain constant  $I_{\text{sig}}$  as  $V_{\text{out}}$  varies. As shown in Figure 2.4, an extra path (with a single-ended conductance  $G_{\text{shnt}}$ ) in parallel with the differential channel is used for de-emphasis. The total supply path impedance can thus be held constant regardless of the output voltage, resulting in a constant  $I_{\text{sig}}$ :

$$I_{\text{sig}} = \frac{1}{4} G_T V_{\text{drv}} \quad (2.7)$$

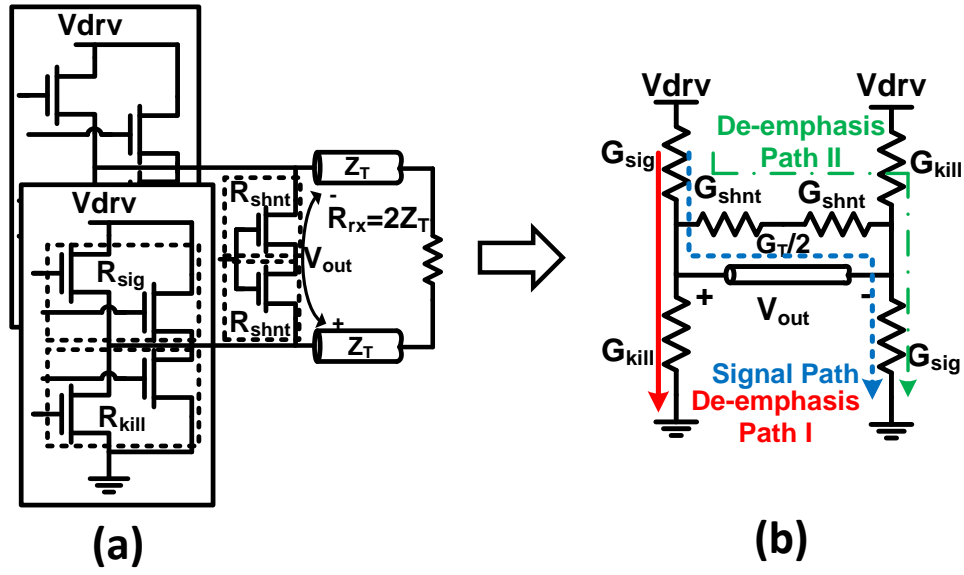


Figure 2.4: (a) constant-current pre-emphasis scheme and (b) its equivalent circuit

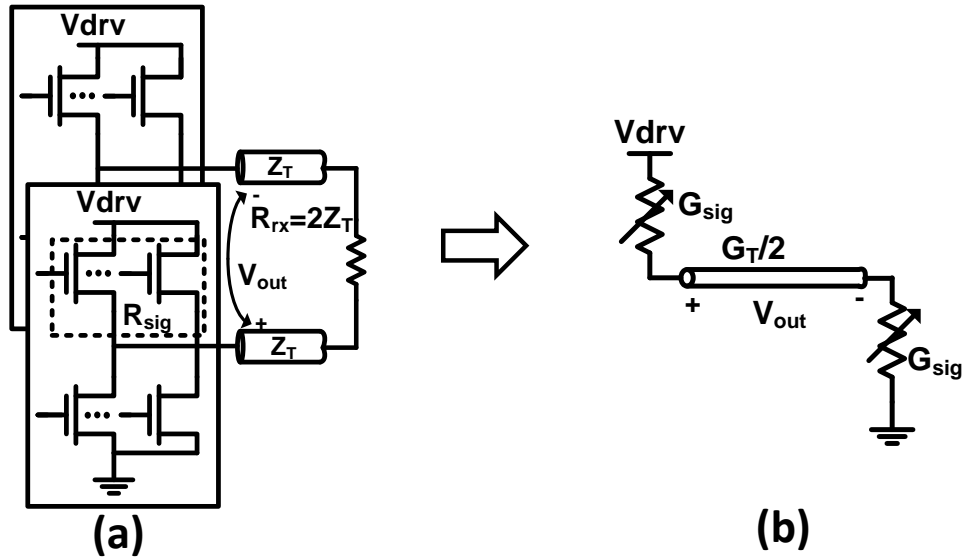


Figure 2.5: (a) impedance modulated pre-emphasis scheme and (b) its equivalent circuit

To further reduce the signaling power, the most effective method<sup>5</sup> is to use the impedance modulated pre-emphasis voltage-mode (IMPEVM) transmitter proposed in [12]. As shown by the equivalent circuit in Figure 2.5,  $V_{out}$  is determined by the voltage divider ratio between the driver's output impedance and the channel's characteristic impedance. By increasing the driver's output impedance we can lower  $V_{out}$  and also

<sup>5</sup> Assuming there is no supply or load impedance modulation.

increase the total supply path impedance. Therefore,  $I_{\text{sig}}$  can be made to scale linearly with  $V_{\text{out}}$ :

$$I_{\text{sig}} = \frac{1}{2} G_T V_{\text{out}} \quad (2.8)$$

As shown in Figure 2.6, although impedance modulation is attractive in substantially reducing signaling power relative to the original scheme, this approach comes at the expense of sacrificing output termination and hence system linearity (and thus perhaps signal integrity). More importantly, as will be discussed in the next subsection, its pre-driver's digital power overhead due to the non-linear driver conductance to  $V_{\text{out}}$  mapping could compromise its overall energy efficiency.

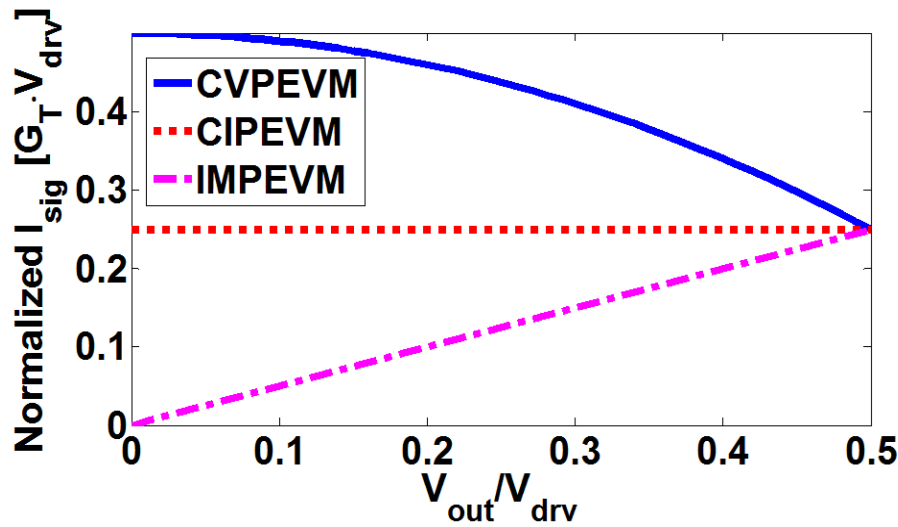


Figure 2.6: signaling power comparison of previous drivers

### 2.2.2 Digital Power Analysis

To evaluate the digital power consumption from the pre-drivers and associated digital logic, we need to know how the driver should be segmented to enable pre-emphasis control. This can be done by finding the mapping between the conductance of each of the transmitter's devices (i.e.  $G_{\text{sig}}$ ,  $G_{\text{kill}}$ , etc) to  $V_{\text{out}}$ .<sup>6</sup> The detailed derivations are shown in the Appendix A; in this section we will provide only the most important equations that highlight the issue at hand.

For CVPEVM, its pull-up conductance ( $G_{\text{sig}}$ ) and pull-down conductance ( $G_{\text{kill}}$ ) as a function of  $V_{\text{out}}$  are:

<sup>6</sup> For all of the analyses below, we will assume that a 1 is being transmitted for the current bit (i.e.,  $V_{\text{out}} \geq 0$ ). The output polarity definition matches that shown in Figure 2.2.



$$G_{\text{sig}}(V_{\text{out}}) = \left( \frac{1}{2} + \frac{V_{\text{out}}}{V_{\text{drv}}} \right) G_{\text{T}} \quad (2.9)$$

$$G_{\text{kill}}(V_{\text{out}}) = \left( \frac{1}{2} - \frac{V_{\text{out}}}{V_{\text{drv}}} \right) G_{\text{T}} \quad (2.10)$$

From (2.9) and (2.10) it is apparent that in order to support the maximum pre-emphasis strength – i.e.,  $V_{\text{out}}$  covering the range from  $V_{\text{drv}}/2$  to  $0 - G_{\text{sig}}$  should take on values between  $G_{\text{T}}$  and  $G_{\text{T}}/2$ , while  $G_{\text{kill}}$  should span from  $0$  to  $G_{\text{T}}/2$ . Therefore, the maximum required switching conductance is  $G_{\text{sw,MAX}} = G_{\text{T}}/2$  in order to achieve the full range of pre-emphasis strength.

To find the required number of segments, we must also know the minimum conductance ( $G_{\text{sw,MIN}}$ ) we need to switch from one branch to the other in order to achieve the required pre-emphasis voltage resolution  $V_{\text{out,LSB}}$ . This conductance is set by:

$$G_{\text{sw,MIN}} \approx V_{\text{out,LSB}} \cdot \min \left( \left| \frac{dG_{\text{sig}}}{dV_{\text{out}}} \right|, \left| \frac{dG_{\text{kill}}}{dV_{\text{out}}} \right| \right) = \frac{V_{\text{out,LSB}}}{V_{\text{drv}}} G_{\text{T}} \quad (2.11)$$

With  $G_{\text{sw,MAX}}$  and  $G_{\text{sw,MIN}}$  known, we can then find the total number of segments  $N_{\text{seg}}$ :

$$N_{\text{seg}} = \frac{G_{\text{br,MAX}}}{G_{\text{br,MIN}}} = \frac{G_{\text{T}}/2}{\frac{V_{\text{out,LSB}}}{V_{\text{drv}}} G_{\text{T}}} = \frac{1}{2} \frac{V_{\text{drv}}}{V_{\text{out,LSB}}} \quad (2.12)$$

For the CIPEVM scheme, the same method can be applied to find the branch conductance mappings, resulting in:

$$G_{\text{sig}}(V_{\text{out}}) = \left[ \left( \frac{V_{\text{out}}}{V_{\text{drv}}} \right)^2 + \left( \frac{V_{\text{out}}}{V_{\text{drv}}} \right) + \frac{1}{4} \right] G_{\text{T}} \quad (2.13)$$

$$G_{\text{kill}}(V_{\text{out}}) = \left[ \left( \frac{V_{\text{out}}}{V_{\text{drv}}} \right)^2 - \left( \frac{V_{\text{out}}}{V_{\text{drv}}} \right) + \frac{1}{4} \right] G_{\text{T}} \quad (2.14)$$

$$G_{\text{shnt}}(V_{\text{out}}) = \left[ -2 \left( \frac{V_{\text{out}}}{V_{\text{drv}}} \right)^2 + \frac{1}{2} \right] G_{\text{T}} \quad (2.15)$$

From the equations above, it should be clear that  $G_{\text{sw,MAX}} = G_{\text{T}}/2$  in order to support the full pre-emphasis range. However, as shown in Figure 2.4, unlike CVPEVM whose branch conductance is linearly proportional to  $V_{\text{out}}$ , the branch conductance of CIPEVM changes more quickly at high  $V_{\text{out}}$  values and saturates at low  $V_{\text{out}}$  values. For this reason, the  $G_{\text{sw,MIN}}$  of CIPEVM is set by low  $V_{\text{out}}$  values:

$$\begin{aligned} G_{\text{sw,MIN}} &\approx V_{\text{out,LSB}} \cdot \min \left( \left| \frac{dG_{\text{sig}}}{dV_{\text{out}}} \right|, \left| \frac{dG_{\text{kill}}}{dV_{\text{out}}} \right|, \left| \frac{dG_{\text{shnt}}}{dV_{\text{out}}} \right| \right) \\ &= V_{\text{out,LSB}} \cdot \left| \frac{dG_{\text{shnt}}}{dV_{\text{out}}} \right|_{V_{\text{out}} \approx V_{\text{out,LSB}}} \approx \frac{4V_{\text{out,LSB}}^2}{V_{\text{drv}}^2} G_{\text{T}} \end{aligned} \quad (2.16)$$

The required number of segments can therefore be estimated as:<sup>7</sup>

$$N_{\text{seg}} = \frac{G_{\text{sw},\text{MAX}}}{G_{\text{sw},\text{MIN}}} = \frac{G_{\text{T}}/2}{\frac{4V_{\text{out,LSB}}^2}{V_{\text{drv}}^2}G_{\text{T}}} = \frac{1}{8} \frac{V_{\text{drv}}^2}{V_{\text{out,LSB}}^2} \quad (2.17)$$

Finally, for the IMPEVM design, the  $G_{\text{sig}}$  to  $V_{\text{out}}$  mapping is:

$$G_{\text{sig}}(V_{\text{out}}) = \frac{V_{\text{out}}}{V_{\text{drv}} - V_{\text{out}}} G_{\text{T}} \quad (2.18)$$

from which we find that  $G_{\text{sw},\text{MAX}} = G_{\text{T}}$ . Similar to CVPEVM,  $G_{\text{sig}}$  is also a non-linear function of  $V_{\text{out}}$  (Figure 2.7), and  $G_{\text{sw},\text{MIN}}$  is also set by the region with low  $V_{\text{out}}$ :

$$\begin{aligned} G_{\text{sw},\text{MIN}} &\approx V_{\text{out,LSB}} \cdot \min \left( \left| \frac{dG_{\text{sig}}}{dV_{\text{out}}} \right| \right) = V_{\text{out,LSB}} \cdot \left| \frac{dG_{\text{sig}}}{dV_{\text{out}}} \right|_{V_{\text{out}} \approx V_{\text{out,LSB}}} \\ &\approx \frac{V_{\text{drv}} V_{\text{out,LSB}}}{(V_{\text{drv}} - V_{\text{out,LSB}})^2} G_{\text{T}} \approx \frac{V_{\text{out,LSB}}}{V_{\text{drv}}} G_{\text{T}} \end{aligned} \quad (2.19)$$

where the last approximation is valid when  $V_{\text{out,LSB}}$  is small, which is usually the case for a reasonable pre-emphasis resolution. Therefore, the total required number of segments for a  $V_{\text{out}}$  range of  $0 - V_{\text{drv}}/2$  can be estimated as:<sup>8</sup>

$$N_{\text{seg}} = \frac{G_{\text{sw},\text{MAX}}}{G_{\text{sw},\text{MIN}}} \approx \frac{G_{\text{T}}}{\frac{V_{\text{out,LSB}}}{V_{\text{drv}}} G_{\text{T}}} = \frac{V_{\text{drv}}}{V_{\text{out,LSB}}} \quad (2.20)$$

From (2.12), (2.17) and (2.20), we see that due to its linear characteristics, CVPEVM requires a substantially smaller  $N_{\text{seg}}$  than the CIPEVM and IMPEVM drivers. As we will describe next, this increase in the number of segments will result in increased switching capacitance, and hence increased digital switching power consumption. Specifically, since each driver segment requires some type of selection logic for pre-emphasis control and buffers for driving the final signaling transistors, the total switching capacitance for the final driver stage can be approximated as ([13]):

$$C_{\text{fin}} = N_{\text{seg}} C_{\text{lgc,tot}} + C_{\text{gt,tot}} \quad (2.21)$$

where  $C_{\text{lgc,tot}}$  is the total capacitance from the logic gates (i.e. MUXes, buffers, etc) and  $C_{\text{gt,tot}}$  is the total gate capacitance of the final driver devices.

<sup>7</sup> Note that although the non-linearity of the driver can be compensated by using non-uniform segment sizes, as long as the smallest segment doesn't hit the minimum size constraint of a given technology, the total loading from the non-uniform distribution can still be approximated by the uniform segment distribution with  $G_{\text{sw},\text{MIN}}$  as the step size.

<sup>8</sup> This range was chosen to have a fair comparison with the terminated PEVM, which can only support swings of up to  $V_{\text{drv}}/2$ . This does not change the minimum step size of the IMPEVM however. In other words, if we want to take advantage of the IMPEVM's potential for larger  $V_{\text{out}}$ , more segments would be required.

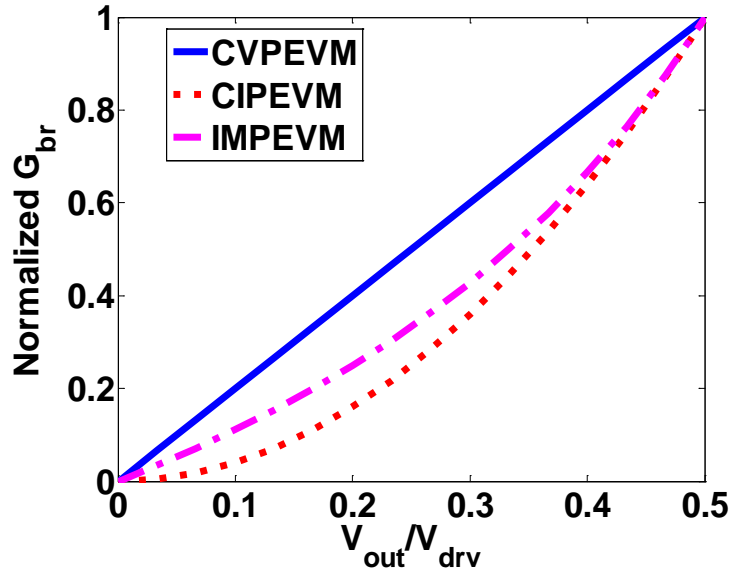


Figure 2.7: switched branch conductance vs  $V_{out}$  for different drivers

The width of the final driver transistors is set by the impedance of the channel, and hence is roughly identical for all of the architectures with the same peak swing. Furthermore, since the impedance of the channel is relatively low, these driver transistors are usually relatively large. Thus, when segmented to enable pre-emphasis control, the size of each segment typically remains larger than minimum (i.e.  $C_{gt,tot}$  remains constant). However, as shown in Figure 2.8, since the capacitive fanout of these digital pre-driver gates is typically greater than 1 (commonly 2-4), the per-segment high-speed gates within each driver cell will often hit the minimum size constraint, resulting in larger final driver switching power for larger  $N_{seg}$ . As shown for example in Figure 2.9 for a 5-bit pre-emphasis resolution,  $N_{seg}$  is  $\sim 62$  for IMPEVM as compared to 31 for CVPEVM. If the pre-driver devices in the CVPEVM have already hit minimum size, the extra 31 segments required for IMPEVM can cause this design to consume 2x more digital switching power than a CVPEM with the same resolution. To provide some numbers, with  $C_{lgc,tot} \sim 10\text{fF}$ , these 31 extra segments will consume  $\sim 1.1\text{mW}$  of additional digital power at 10Gb/s with a 1.2V supply. This 1.1mW is larger than the signaling power of a driver with a 200mV peak differential output amplitude (i.e., 400mV  $V_{drv}$ ), and thus these additional segments would heavily compromise the benefits of low-swing signaling.

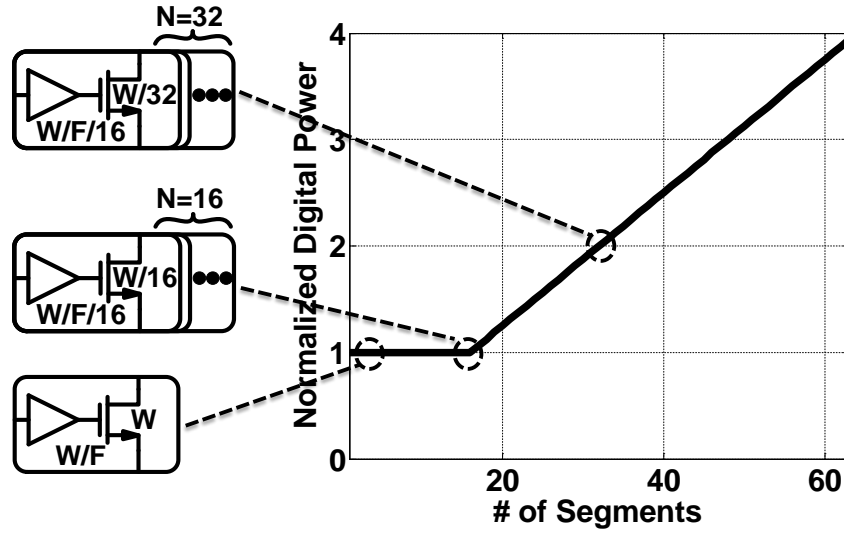


Figure 2.8: normalized digital power vs. # of segments

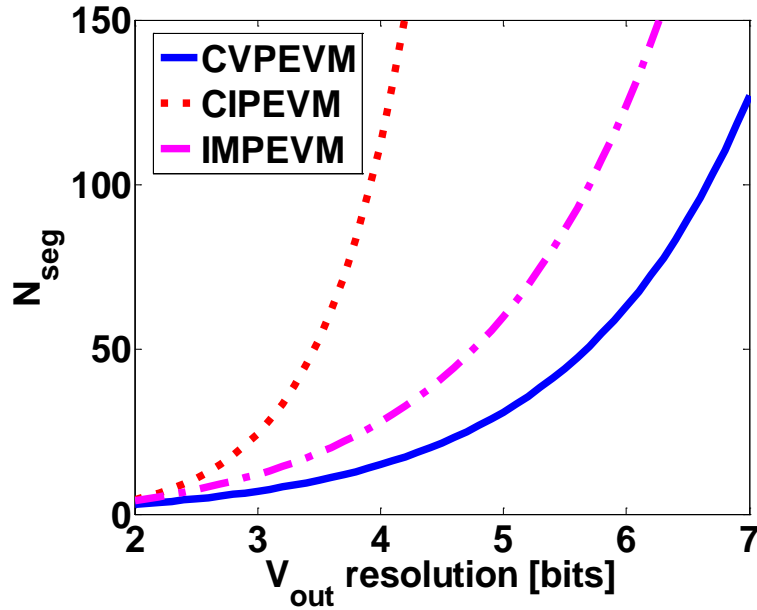


Figure 2.9: number of segments vs.  $V_{out}$  resolution for different designs

Note that if more complex pre-driver and decoder architectures – such as the look-up-tables (LUT) used in [10], [12] – are required to compensate the driver's non-linearity, the digital power overhead will increase even further as compared to a linear driver which requires only simple decoders/unit elements. This combined extra pre- and final-driver switching power can easily exceed the signaling power consumption ([10], [12]), and the benefit of reducing signaling power by utilizing IMPEVM or CIPEVM is compromised.

## 2.3 Proposed Pre-Emphasis Voltage-Mode Transmitter

As discussed in the previous section, the CIPEVM and IMPEVM architectures have improved signaling power efficiency over CVPEVM due to their higher supply path impedance at lower swings. However, both architectures penalize the digital power as compared to a linear CVPEVM driver.

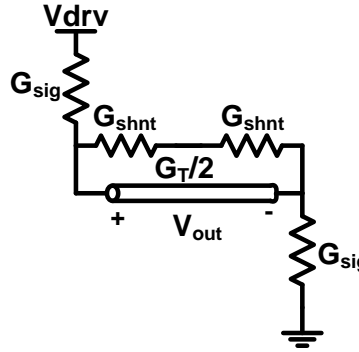


Figure 2.10: equivalent circuit of proposed PEVM

Fortunately, by modifying the means by which pre-emphasis is implemented, we can combine the advantages from IMPEVM and CVPEVM to improve the signaling power and maintain low digital power simultaneously. Specifically, in [11], signaling power is wasted in the crowbar current paths created from  $V_{drv}$  to ground when reduced swing (due to pre-emphasis) is used. Instead, extending the technique used in CIPEVM, the transmitter's output voltage is modulated by using only a shunting path that is in parallel to the channel – as shown in Figure 2.10. When moving conductance from  $G_{sig}$  to  $G_{shnt}$  to reduce  $V_{out}$ , we actually increase the overall supply path impedance from  $V_{drv}$  to ground, and hence reduce signaling current. The output termination is also maintained since the sum of  $G_{sig}$  and  $G_{shnt}$  is kept constant and equal to  $G_T$ .<sup>9</sup>  $V_{out}$  vs.  $G_{sig}$  follows the desired linear relationship shown in (2.22):

$$G_{sig}(V_{out}) = \frac{2V_{out}}{V_{drv}} G_T \quad (2.22)$$

With  $G_{sw,MAX}$  and  $G_{sw,MIN}$  set to  $G_T$  and  $\frac{2V_{out,LSB}}{V_{drv}} G_T$  respectively, this driver requires the same  $N_{seg}$  as that of CVPEVM:

$$N_{seg} = \frac{G_{sw,MAX}}{G_{sw,MIN}} = \frac{G_T}{\frac{2V_{out,LSB}}{V_{drv}} G_T} = \frac{1}{2} \frac{V_{drv}}{V_{out,LSB}} \quad (2.23)$$

<sup>9</sup> Note that if lower pre-emphasis resolution can be tolerated (or equivalently, more segments for the same resolution), this driver can be operated in IMPEVM mode by disabling the shunting branch and using only the on/off the  $G_{sig}$  path transistors.

$I_{\text{sig}}$  can also be shown to scale down with decreased  $V_{\text{out}}$  in the following manner:

$$I_{\text{sig}} = G_T V_{\text{out}} \left( 1 - \frac{V_{\text{out}}}{V_{\text{drv}}} \right) \quad (2.24)$$

As shown in Figure 2.11, the proposed architecture allows signaling current reduction along with the transmitted swing (albeit not quite proportionally as with IMPEVM) while also maintaining output termination. Its required number of segments follows that of a linear driver and is therefore  $\sim 2\times$  smaller than that of IMPEVM (Figure 2.12), thus enabling it to also achieve good digital power efficiency.

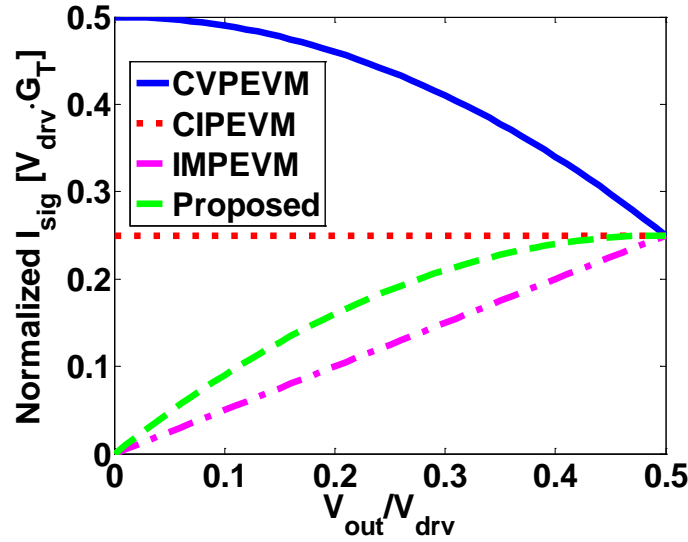


Figure 2.11: signaling power comparison

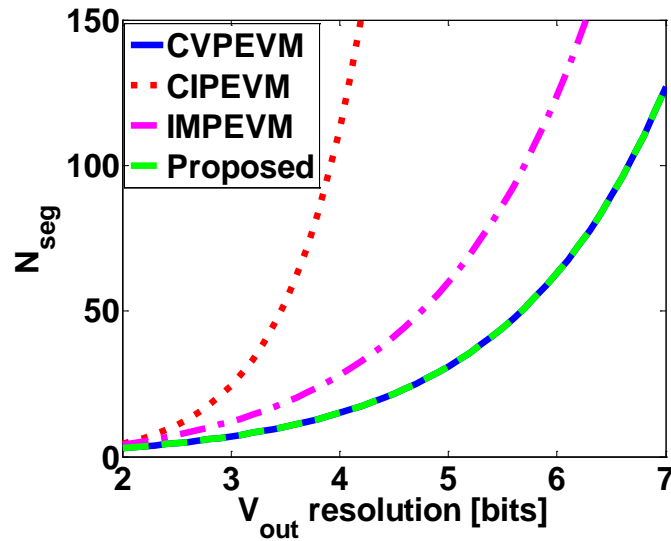


Figure 2.12: number of segments vs  $V_{\text{out}}$  resolution

## 2.4 Transmitter Architecture and Circuit Implementation

Now that we have analyzed the digital power overhead of implementing pre-emphasis, we should also consider the impact of integrating other functionality (i.e., impedance control) on the energy-efficiency of a PEVM. In this section, we will therefore first discuss our choice for the impedance control scheme, and then introduce the overall transmitter architecture. We will then describe the circuit implementations, with an emphasis on the impedance and swing control loops that incorporate an online comparator offset cancellation scheme.

### 2.4.1 Choice of Impedance Calibration Schemes

There are two popular schemes for controlling/setting the output impedance of a transmitter. The “digital” approach (Figure 2.13(a)) incorporates both impedance calibration and pre-emphasis control into the segment design ([10], [14]). Although this may simplify the overall implementation, even with non-uniform segments ([14]), this choice tends to compromise the pre-emphasis resolution for a reasonable total number of segments. For example, the design in [14] used 24 segments to achieve 4.6-bit dynamic range in impedance but only ~2-bit dynamic range in pre-emphasis. Therefore, to decouple impedance calibration and pre-emphasis control, and hence increase the pre-emphasis dynamic range, we choose to use the “analog” approach proposed in [11], [15]. Specifically, as shown in Figure 2.13(b), separate regulator-based control loops adjust the supplies of the pre-drivers in order to control the impedance of the pull-up, pull-down, and shunt devices, leaving segmentation only for pre-emphasis coefficient control.

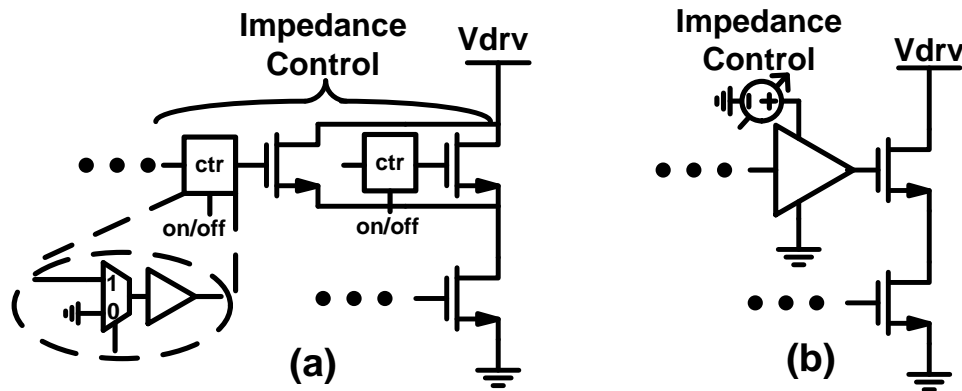


Figure 2.13: (a) “digital” impedance calibration scheme and (b) “analog” impedance calibration scheme

## 2.4.2 Overall Architecture

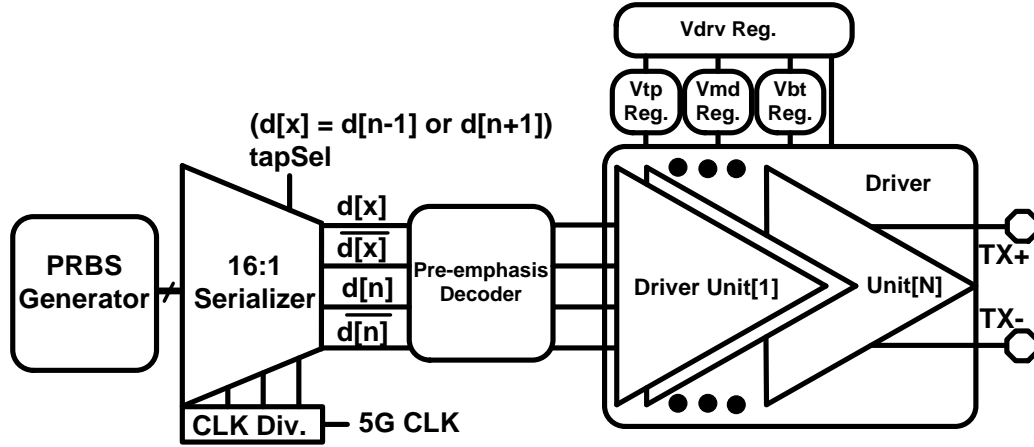


Figure 2.14: overall transmitter architecture

The overall transmitter architecture is shown in Figure 2.14. Since a large number of TX pre-emphasis taps quickly leads to excessive pre-driver power due to complex segment logic, in this design we have chosen to implement only 2 taps of pre-emphasis.<sup>10</sup> The pre-emphasis tap can be assigned to be either the post-cursor or the pre-cursor. Following the 16:1 tree-type serialization, before entering the final driver, the digital data stream is pre-decoded in order to simplify the logic required within each driver segment.

In this design, the final driver comprises of 15 segments, which corresponds to the maximum achievable resolution before the pre-driver logic transistors hit the minimum size constraint. This results in a 4-bit tunable pre-emphasis weight from 0 to 100% with a step of ~6.7%.

## 2.4.3 Pre-emphasis Decoder

Knowing that the target TX FIR filter will be high-pass in nature, the pre-emphasis decoder simply looks for differences/similarities between the current bit and the next or previous bits (using XORs and ANDs) in order to provide appropriate signals for the pull-up, pull-down, and shunt devices. As shown in Figure 2.15, the principle motivation for performing this pre-decoding is that these logic gates can be shared amongst all final driver segments. Each individual segment then only needs to include MUXes to decide whether to use the pre-decoder outputs or simply the raw data bit. In comparison to a design which effectively requires multiple complex logic gates within each segment (e.g. [12]), this design moves some logic gates out of the driver segments to a shared non-minimum-sized pre-driver logic and hence results in significantly reduced digital power overhead.

<sup>10</sup> A multi-tap design can be implemented by incorporating a more complicated decoder to select driver segments. However, multi-tap equalizers typically require higher resolution on a per-tap basis, leading to additional segments and exacerbating the digital switching power issue. For this reason, receiver equalization is often a more efficient means of dealing with higher-loss channels ([13]).



## 2.4.4 Driver Segments

The driver segment includes the interface between the full-swing digital data and the transmitted low-swing analog signal. Since the target maximum swing of this particular design is low ( $<250\text{mV}$  differential amplitude), each segment utilizes an N-over-N voltage mode driver. Similarly, the shunt devices are also implemented with NMOS transistors due to the low output common-mode voltage. The supply voltage of the N-over-N leg is connected to  $V_{\text{drv}}$ , which is generated by the swing control regulator. The gate voltage of the pull-up, pull-down, and shunt devices are connected to 3 buffers, each with a supply of  $V_{\text{tp}}$ ,  $V_{\text{bt}}$  and  $V_{\text{md}}$  generated from the impedance control regulators.<sup>11</sup> The pull-up, pull-down and shunt devices are sized to give roughly equal  $V_{\text{tp}}$ ,  $V_{\text{bt}}$  and  $V_{\text{md}}$  so that the buffer delays on the three paths can be approximately matched. The buffer chains are composed of an even number of inverters to balance rise and fall times across different corners.

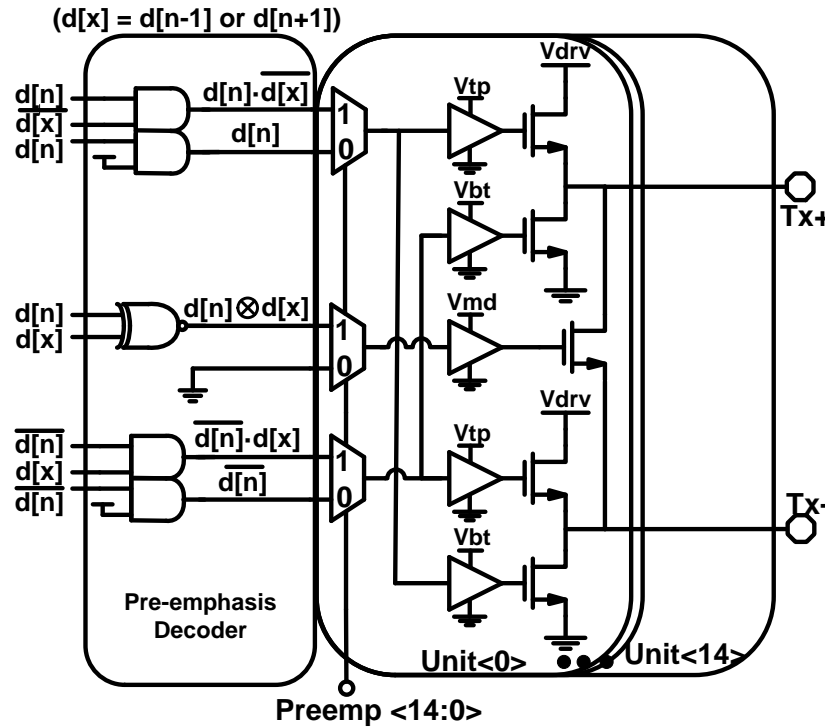


Figure 2.15: decoder and segment schematic

As mentioned earlier, since we have decoupled the impedance control from the pre-emphasis control, as shown in Figure 2.15, MUXes can simply be added to the inputs

<sup>11</sup> For transmitters that must support higher peak swing, the N-over-N voltage mode driver could be replaced with a P-over-N type. An NMOS transistor or a transmission gate could be used for the shunt device of a high-swing TX depending on the output common mode and the maximum pre-emphasis strength. For a P-over-N driver, the ground of the pre-driver for the PMOS output device would be regulated (instead of the supply) in order to control the PMOS transistor's impedance.

of each segment in order to achieve coefficient adjustment. If the segment is enabled for pre-emphasis, the MUX within the segment will connect the pull-up, pull-down, and shunt device gates to the outputs of the pre-decoder that compares between the current and next or previous bits. If the segment is not configured for pre-emphasis, the shunt device will be grounded and the pull-up and pull-down devices will be driven purely by the current bit. Utilizing this simplified segment logic that consists of only 2-input MUXes and level-shifting inverters minimizes the final-driver power overhead to  $\sim 1.7\text{mW}$  from the per-segment high-speed digital gates.

### 2.4.5 Impedance Control Loop with Online Comparator Offset Calibration

As shown in Figure 2.16, impedance control is achieved through the regulated gate voltages  $V_{tp}$ ,  $V_{bt}$ , and  $V_{md}$ . Similar to the technique used in [15], these voltages are generated by replica-bias circuits that force a replica segment to have the same impedance as a reference resistor string.

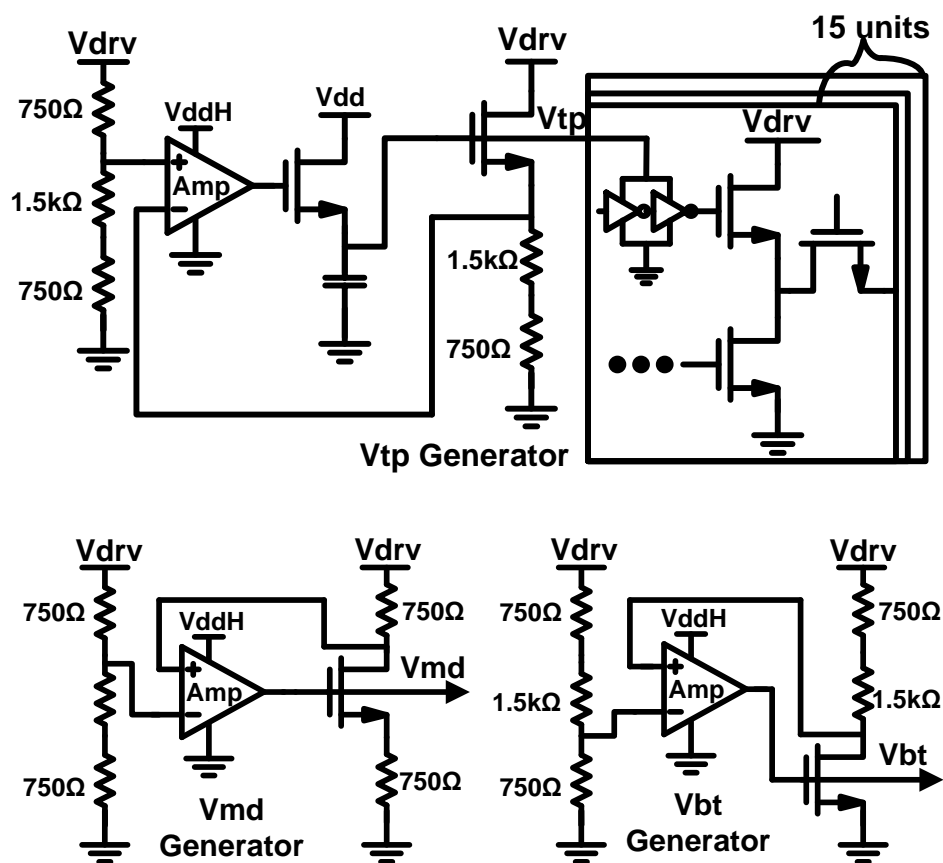


Figure 2.16: schematic of impedance control loop

As highlighted in Figure 2.17, the regulator utilizes a comparator-based architecture with a source-follower output stage in order to reduce the power consumed by its feedback loop ([16]). Since  $V_{tp}$  and  $V_{bt}$  as well as perhaps  $V_{md}$  may need to be very close to  $V_{dd}$ , and since the regulator's power device is an NMOS source-follower, the gate voltage of the power device may need to be higher than  $V_{dd}$ . The regulator feedback control loop therefore uses a higher voltage (but very low current) supply  $V_{ddH}$  (e.g. 1.6V)<sup>12</sup> along with a level shifter to support this higher gate voltage.

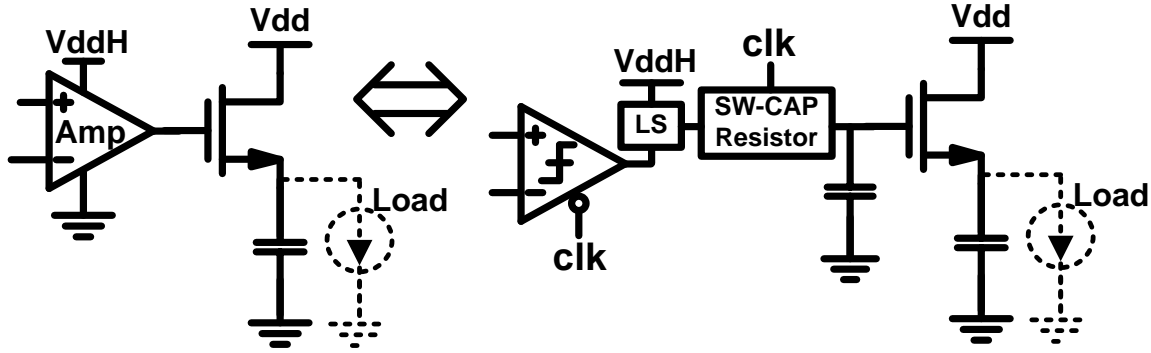


Figure 2.17: implementation of comparator-based regulator loop

Given the broadband low output impedance and  $g_m r_o$  intrinsic PSRR of the NMOS source follower power device, the regulator's feedback control loop need not achieve high bandwidth ([16], [17]). Hence, this design utilizes a switched-capacitor resistor (SCR) at the gate of the power device to implement a low-pass filter and attenuate the output voltage ripple due to the loop's dither to a simulated value of less than 1mV (the capacitance at the gate of the power device is  $\sim 1000X$  larger than that of the SCR). Due to the higher supply  $V_{ddH}$ , the devices in the switched-cap resistor are all thick-oxide to ensure reliability.

The error signal between the replica and the reference resistor string is generated by a comparator. Given the relatively low target swing and hence low common-mode at the comparator's input, the feedback comparator is implemented with a PMOS-input StrongArm latch ([18]). Minimizing the power dissipation of this comparator leads to nearly minimum sized devices. Thus, if left uncompensated, the comparator (and hence the overall regulator) could exhibit  $>100mV$  of offset. Moreover, since changing the output swing will change the output common-mode and hence the input-referred offset, online digital offset cancellation was implemented to continuously eliminate the comparator's offset.

This offset cancellation was achieved by time-interleaving the operation of the comparator so that it auto-zeroes its own offset and also provides feedback for the regulator. In the calibration phase (Figure 2.18(a)), the comparator is disconnected from the regulation loop by shorting its inputs to a capacitor that stores the input common-mode voltage. The output of the comparator changes according to its offset value and is

<sup>12</sup> Since  $V_{ddH}$  is effectively used only to set the DC voltage on a capacitor, it supplies minimal current in steady state. Thus, if a suitable externally generated supply was unavailable,  $V_{ddH}$  could be straightforwardly generated with an on-chip charge-pump ([16]).

fed into an accumulator. The accumulated output controls a 5-bit capacitor-DAC (CDAC) that is connected to the drains of the input devices (similar to [19]) to cancel the comparator's offset. After the calibration phase, the loop enters the regulation phase (Figure 2.18(b)) by re-connecting the comparator inputs to the reference strings/replica cells and by disabling the feedback accumulator. The offset-cancelled comparator then behaves as a normal comparator to amplify the input difference so as to adjust the regulated voltage. In steady-state, the output of the offset accumulator will dither around a certain value. In order to limit this offset dither, the LSB of the CDAC is chosen to be 4mV. The calibration control signal (cal) is a divided by 2 clock pattern from the 78.125MHz clk signal which itself is a divided signal from the global high-speed clock.

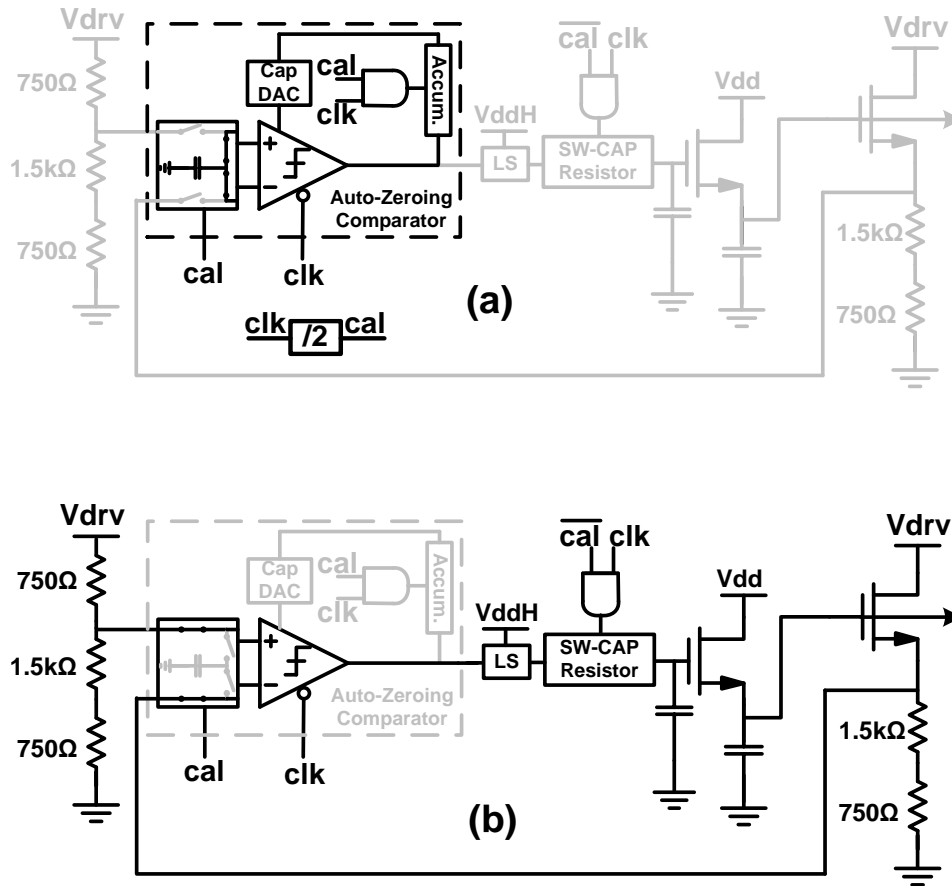


Figure 2.18: illustration of interleaved comparator offset calibration and regulation process: (a) comparator auto-zeroing (calibration) phase and (b) regulation phase

## 2.4.6 Output Amplitude Control Loop

As shown in Figure 2.19, the swing control loop operates in a manner very similar to the impedance control loops. Swing adjustment is achieved through a voltage regulator that sets  $V_{drv}$  based on a reference voltage generated from a voltage DAC. The same

comparator offset cancellation scheme is used here to track common-mode variation induced offset changes as the swing values are adjusted. Since  $V_{drv}$  is designed to be always  $\leq 500\text{mV}$ , no extra  $V_{ddH}$  is required for this NMOS power device.

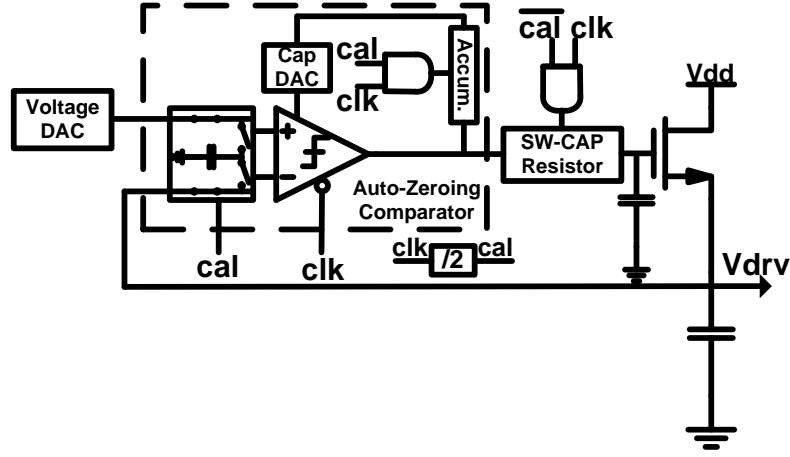


Figure 2.19: schematic for swing control loop

## 2.5 Measurement Results

In order to experimentally verify the proposed TX architecture and circuits, a test-chip including this design was fabricated in a 65nm LP CMOS process. The die photo along with the TX layout is shown in Figure 2.20, highlighting that the TX occupies an area of  $\sim 300\text{ }\mu\text{m} \times 200\text{ }\mu\text{m}$ .

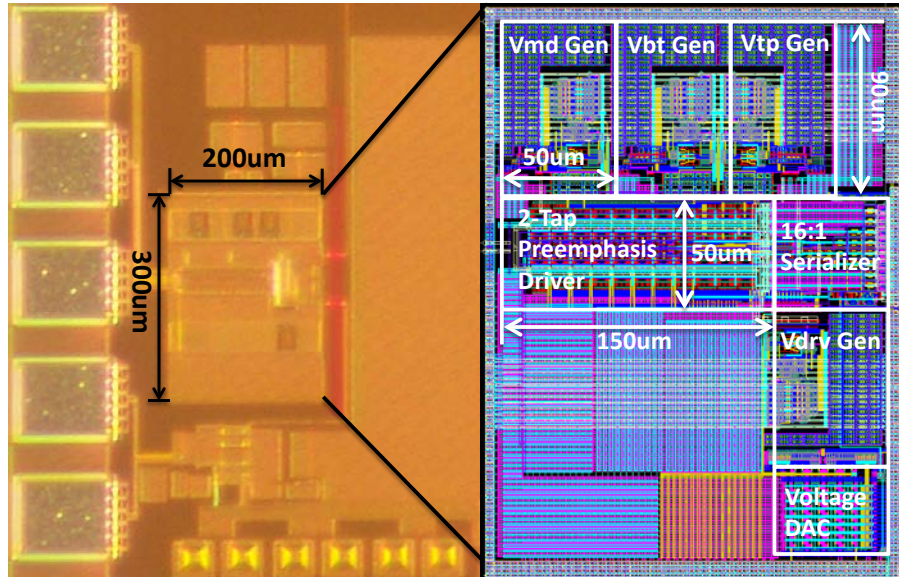


Figure 2.20: die photo and TX floor plan

In order to test the TX pre-emphasis functionality, we used the TX to drive a 10" FR4 PCB trace. Before characterizing the TX itself however, we must first measure the characteristics of this channel. Using the channel's measured S21 to simulate the 10Gb/s pulse response (Figure 2.21) shows ~13dB loss at Nyquist (5GHz) and a dominant post-cursor whose magnitude is ~0.4 of the cursor.

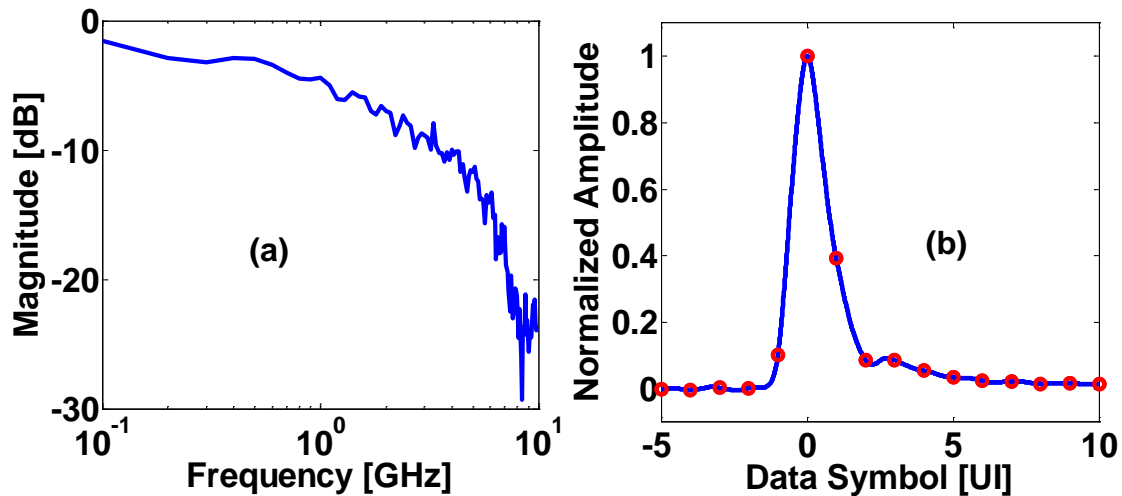


Figure 2.21: 10" FR4 PCB trace (a) S21 and (b) simulated 10Gb/s pulse response

Before turning on pre-emphasis, we measured the eye diagrams with  $2^{23}-1$  PRBS data and  $V_{out} = 250\text{mV}$  before and after this channel, as shown in Figure 2.22. Since the total output pad capacitance including ESD is about 1.3pF and there is ~2" of FR4 PCB trace between the test-chip and the connector, the intrinsic output bandwidth of the TX (even without the additional 10" trace) is limited to ~4GHz. This inherent bandwidth limitation is the reason for the ISI apparent in Figure 2.22. After the 10" trace, the eye is completely closed. However, as shown in Figure 2.23, after configuring the TX to apply a pre-emphasis filter of  $(10x[n]-5x[n-1])/15$ , the eye is opened with 3.19ps/22.22ps RMS/P2P jitter.

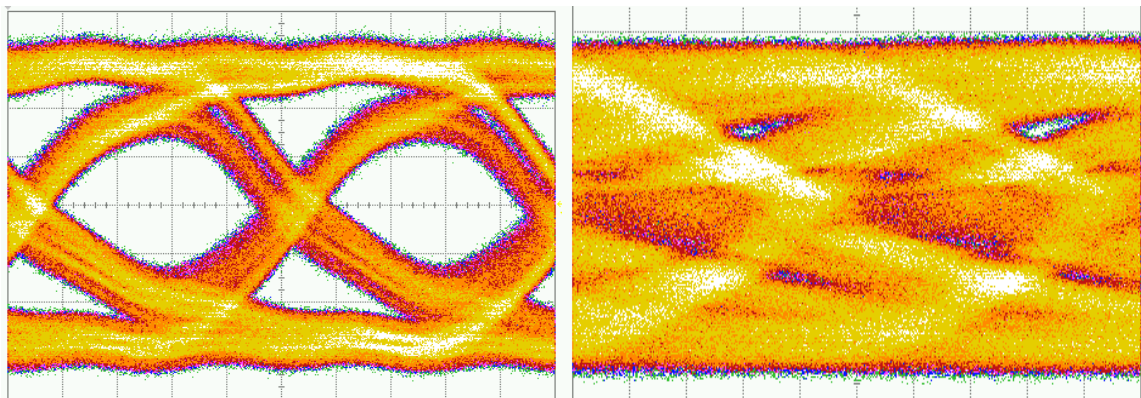


Figure 2.22:  $2^{23}-1$  PRBS eye before and after 10" trace with post-tap pre-emphasis turned off (100mV/div vertical, 20ps/div horizontal)

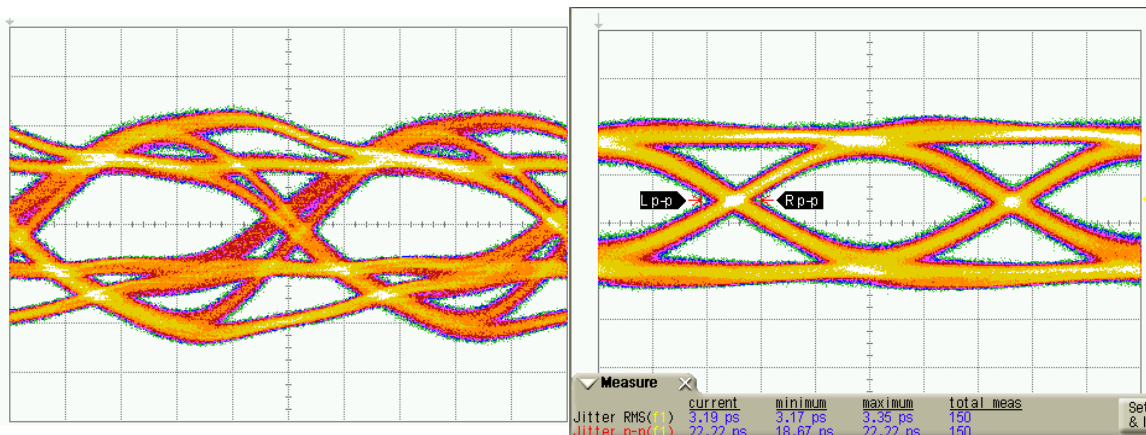


Figure 2.23:  $2^{23}-1$  PRBS eye before and after 10'' trace with post-tap pre-emphasis turned on (100mV/div vertical, 20ps/div horizontal)

To verify the impedance and swing control loops, Figure 2.24 and Figure 2.25 show the measured characteristics of the transmitter. Across output swings and pre-emphasis settings (PE code), the output impedance remains nearly  $52\Omega$ , indicating the effectiveness of the impedance loop. Note that the deviation from the ideal  $50\Omega$  value is due to mismatches between the different reference resistor strings within each regulator.

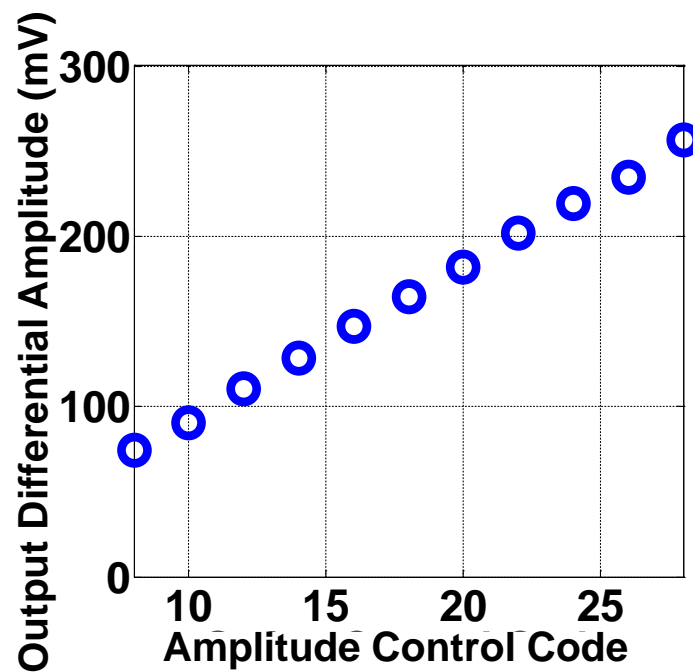


Figure 2.24: TX output swing tuning curve

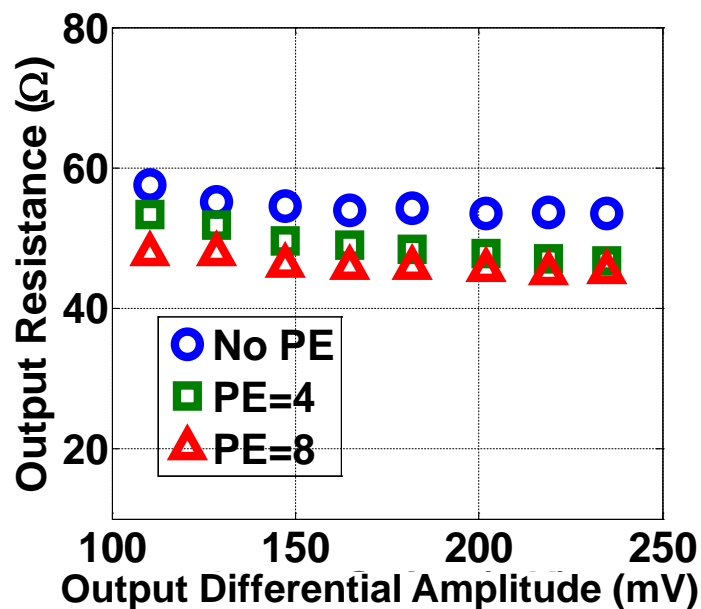


Figure 2.25: TX output impedance vs. signal amplitude

Finally, the power consumption of the TX was characterized. Figure 2.26 shows the signaling power vs. pre-emphasis code at a nominal output swing of 250mV. The measured signaling power drops with lower swing and tracks almost perfectly with the analytical predictions.

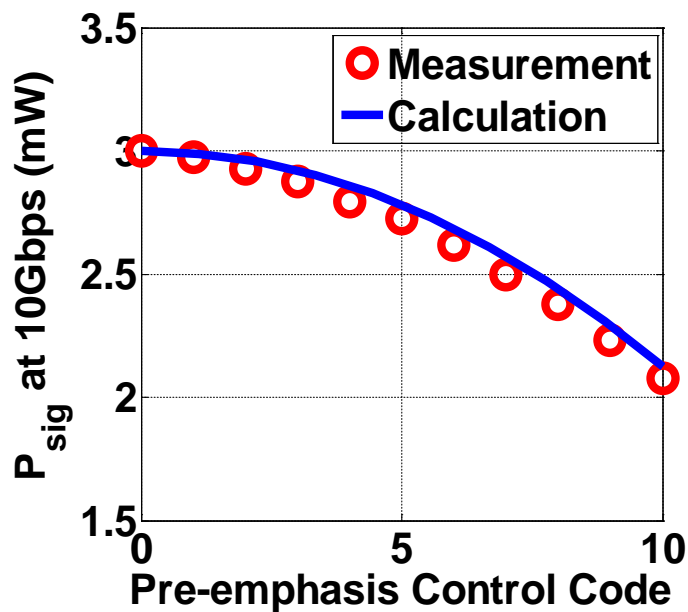


Figure 2.26: signaling power with 250mV differential amplitude swing vs. pre-emphasis setting



The total power vs. output differential amplitude is plotted in Figure 2.27(a), and Figure 2.27(b) shows the total power vs. data-rate. The total power consumption is  $\sim 10\text{mW}$  for a 10Gb/s 200mV differential amplitude PRBS sequence. Based on these trends, we can also extract that the analog power is  $\sim 5\text{mW}$ , with 2.4mW due to the signaling power and the remaining 2.6mW from other biasing and leakage paths.<sup>13</sup> Similarly, the extracted total digital power is  $\sim 5\text{mW}$ , which consists of  $\sim 2.8\text{mW}$  for the 16:1 serializer and clock divider chain,  $\sim 1.7\text{mW}$  for the final driver,<sup>14</sup> and  $\sim 0.5\text{mW}$  for the pre-decoder.

Table 2.1 summarizes the results of this design in comparison with previous PEVM designs with the same number of equalization taps and similar TX swings. In addition to supporting output swing scaling and impedance control, the proposed PEVM works at the highest data rate while achieving  $\sim 2\times$  improved energy-efficiency, largely due to the reduced digital overhead.

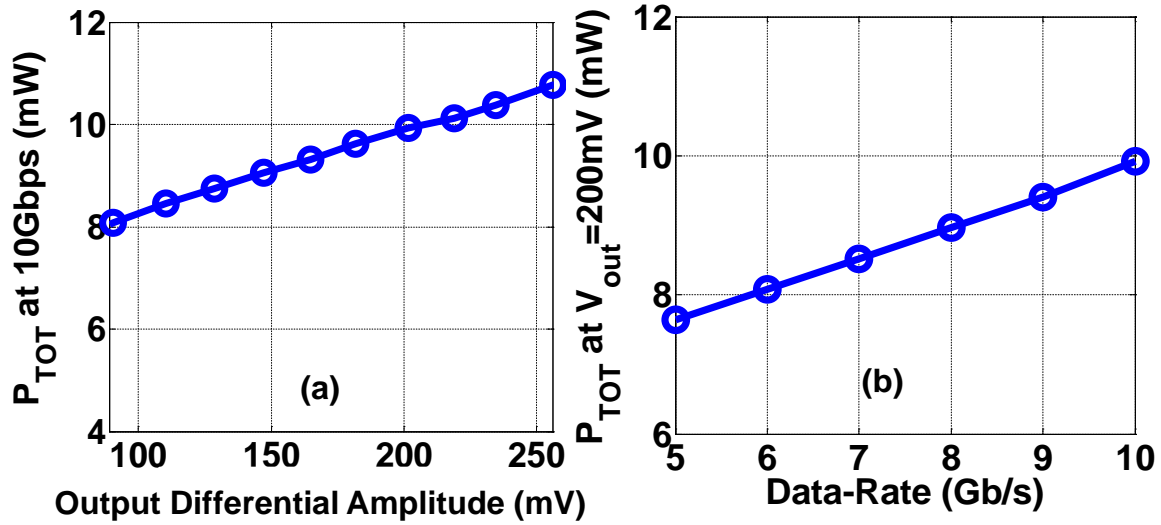


Figure 2.27: TX power vs. output swing and data-rate

<sup>13</sup> This 2.6mW includes 3 resistor reference string currents, 3 transmitter replica currents, and bypass cap leakage current.

<sup>14</sup> This 1.7mW includes gate buffer power drawn from the regulated supplies  $V_{tp}$ ,  $V_{bt}$  and  $V_{md}$  as well as pre-emphasis selection MUX power drawn from  $V_{dd}$ .

Table 2.1: pre-emphasis voltage-mode transmitter comparison

	Hatamkhani, VLSI 03	Dettloff, ISSCC 10	Sredojevic, CICC 10	This Work
<b>Technology</b>	<b>0.18um</b>	<b>45nm SOI</b>	<b>90nm</b>	<b>65nm LP</b>
<b>Supply</b>	<b>1.8V</b>	<b>1-1.65V</b>	<b>1.15V</b>	<b>1.2V</b>
<b>Data Rate</b>	<b>3.6Gb/s</b>	<b>7.4Gb/s</b>	<b>4Gb/s</b>	<b>10Gb/s</b>
<b>Swing</b>	<b>250mV</b>	<b>400mV</b>	<b>≤500mV</b>	<b>≤250mV</b>
<b># of Taps</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>
<b>Equalizer Resolution</b>	<b>35.7mV</b>	<b>33mV</b>	<b>18mV</b>	<b>16.7mV</b>
<b>Power</b>	<b>10mW</b>	<b>32mW</b>	<b>5-17mW</b>	<b>8-11mW</b>
<b>Energy/bit</b>	<b>2.8pJ/bit</b>	<b>4.3pJ/bit</b>	<b>≤4.25pJ/bit</b>	<b>≤1.1pJ/bit</b>
<b>Digital Overhead</b>	<b>N/A</b>	<b>N/A</b>	<b>1.25pJ/bit<sup>a</sup></b>	<b>0.5pJ/bit<sup>b</sup></b>

<sup>a</sup>Decoder only<sup>b</sup>Includes serializer and clock dist.

## 2.6 Conclusions

By analyzing the signaling and digital power consumption of previous PEVM transmitters, this chapter shows that although the CIPEVM and IMPEVM drivers improve signaling power over the CVPEVM driver, they both suffer from increased digital power consumption due to their non-linear conductance to output swing mapping. This non-linear mapping results in an increased number of segments to achieve a given level of pre-emphasis control, and hence results in increased digital power consumption. To improve both signaling and digital power consumption, we propose a pre-emphasis scheme that relies purely on a shunt-to-channel path for signal amplitude de-emphasis. Along with independent impedance control loops to minimize the number of final driver segments required for pre-emphasis control as well as a shared pre-emphasis decoder with simple driver segments, the proposed architecture maintains both low signaling and digital driver power. A 65nm LP CMOS implementation of this architecture dissipates only ~10mW from a 1.2V supply when transmitting 10Gb/s 400mV differential peak-to-peak data with 2-tap pre-emphasis, achieving 1pJ/bit energy efficiency.

## Chapter 3

# Design Techniques for Multi-Tap Energy-Efficient Decision Feedback Equalizers

As previously discussed in Chapter 1, feed-forward equalizers are essentially linear equalizers that suffer from the noise amplification drawback. On the other hand, decision-feedback equalizers (DFE) use clean “digital” decisions to correct the trailing ISI which avoids the high-frequency noise amplification issue. For this reason, to cancel post-cursor ISI, DFE is usually preferred over a FFE to equalize the distorted signal. However, since the operation of a DFE requires a decision from the current bit to cancel the ISI it poses on later bits, it inevitably introduces feedback loops. Closing the timing of these feedback loops – especially the 1<sup>st</sup> tap timing that has only 1 unit interval (UI) – makes the design of DFEs very challenging.

To relieve the timing constraints of the initial tap(s), many current 20-40Gb/s designs utilize a loop-unrolled architecture ([20]–[24]). However, loop-unrolling introduces additional delay into the critical paths of later (non-unrolled) DFE taps due to the selection MUXes, and with its exponential growth in complexity, does not scale well as the number of unrolled taps increases. Perhaps for this reason, no multi-tap DFE solutions with single pJ/bit energy-efficiency have yet been demonstrated at data rates >40Gb/s.

In order to break this barrier and realize an efficient multi-tap DFE operating at such data-rates, in this chapter we will propose techniques to directly close the most timing-critical first tap. Utilizing the proposed techniques, we have demonstrated a 3-tap closed-loop DFE prototype in a 65nm CMOS technology that operates at up to 66Gb/s while consuming only 46mW of power from a 1.2V supply ([25]).

## 3.1 Multi-tap Design Challenges

### 3.1.1 Closed-loop DFE Timing Constraints

Dealing with the timing of the very first tap in the DFE typically drives the architecture and hence timing constraints of the entire design, and thus we will begin the discussion by examining a single tap DFE. As shown in the full-data rate (FDR) 1-tap closed-loop DFE (CLDFE) implementation of Figure 3.1(a), the input analog signal is “sliced” by the flip-flop (FF) in order to convert into a digital signal, which is then fed-back to cancel its the first-tap post-cursor ISI. Therefore, the highlighted path requires the whole operation to be completed within:

$$T_{ckq} + T_{setup} + T_{settle} < 1UI \quad (3.1)$$

where  $T_{ckq}$  and  $T_{setup}$  are the propagation delay and set-up time of the FF while  $T_{settle}$  represents the analog setting time of the summation node.

As bit-rates approach the limits of the device speed, implementing a full-data rate flip-flop (FF) and its associated clocking circuits may not be energy-efficient or practical. Double-data rate (DDR) architectures like the one shown in Figure 3.1(b) are therefore often adopted to relax the circuits' throughput requirements. However, as highlighted in the same figure, moving to a DDR architecture does not modify the 1<sup>st</sup> tap timing constraint.

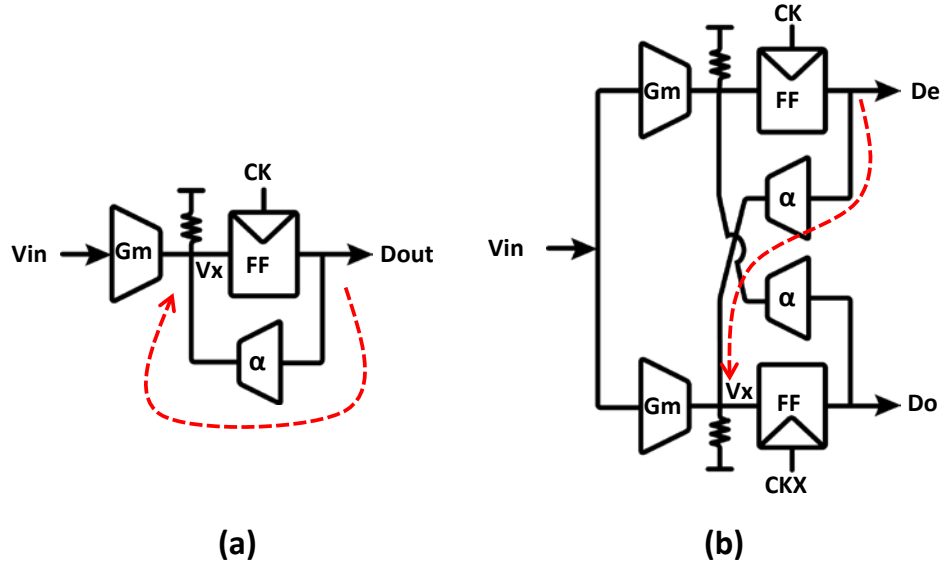


Figure 3.1: (a) full-data-rate 1-tap closed-loop DFE and (b) double-data-rate 1-tap closed-loop DFE architectures

### 3.1.2 Loop-unrolling Limitations

To relax the stringent 1<sup>st</sup>-tap timing constraint, the loop-unrolling (LUDFE) architecture was introduced in [26] and has been widely adopted in many 1-tap DFE designs ever since ([21], [27]–[30]).

The operation of loop-unrolling can be reviewed with the help of Figure 3.2 for a full data-rate 1-tap LUDFE. The operation of loop unrolling behaves quite similar to a carry look-ahead adder ([31]). With the assumptions from the previous bit being either 1 or -1, a static offset with  $-\alpha$  or  $+\alpha$  ( $\alpha$  is the absolute magnitude of the post-cursor ISI) is applied on the incoming analog bits. Thus the outcomes for both cases with ISI corrections can be pre-computed. The final outcome is then selected based on the actual value of the previous bit. It is worth noting that the name “loop-unrolling” is slightly misleading since the feedback loop still exists when choosing which result to select from (i.e., the loop is not truly eliminated). The feedback delay is now moved to the digital

domain and potentially could be faster than the closed loop operation due to the savings on the analog settling time.

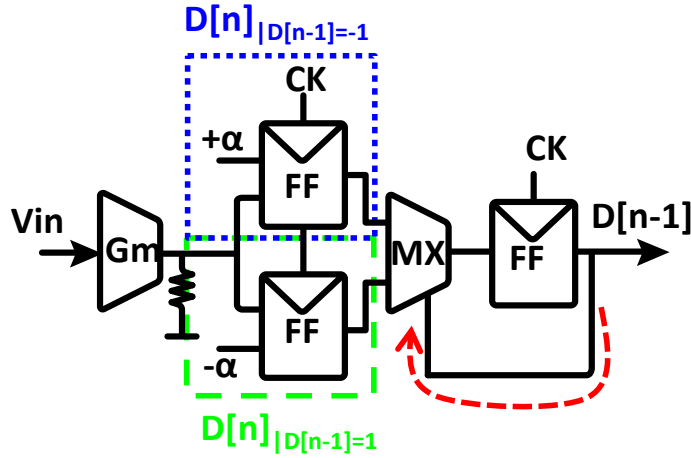


Figure 3.2: 1-tap loop-unrolled FDR DFE architecture

Same as the consideration for the closed-loop case, to cut the clock frequency by half and thus reduce the clocking path's burden, the 1-tap loop-unrolled design can be modified to support the DDR architecture as shown in Figure 3.3 with the same timing constraint; the new loop timing constraint for the 1<sup>st</sup> tap is:

$$T_{ckq} + T_{setup} + T_{s,MX} < 1UI \quad (3.2)$$

where  $T_{s,MX}$  is the selection MUX's digital propagation delay and is presumably smaller than the analog settling delay  $T_{settle}$  in (3.1).

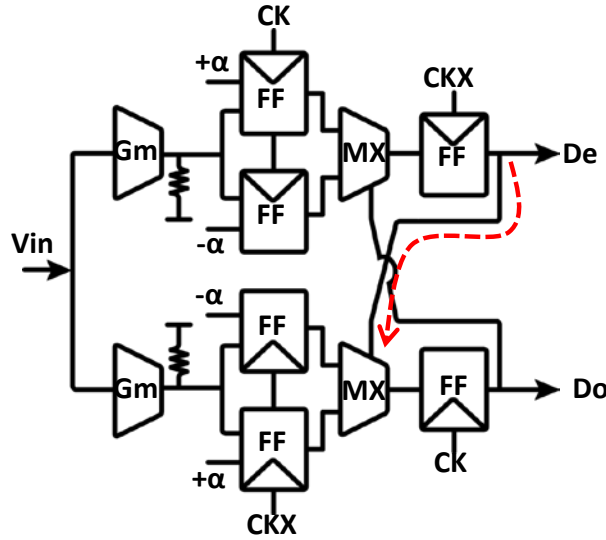


Figure 3.3: 1-tap loop-unrolled DDR DFE architecture

Although this unrolled architecture can improve the 1<sup>st</sup> tap timing, it unfortunately places additional burden on later non-unrolled taps due to the delay from the extra selection MUXes (i.e. the data input delay  $T_{d,MX}$ ).<sup>15</sup> Specifically, as shown in Figure 3.4(a) for a 1<sup>st</sup> tap loop unrolled and 2<sup>nd</sup> tap closed-loop DDR DFE, the timing constraint for the 2<sup>nd</sup> tap is:

$$T_{ckq} + T_{setup} + T_{settle} + T_{d,MX} < 2UI, \quad (3.3)$$

whereas the 2<sup>nd</sup> tap timing constraint for the CLDFE design in Figure 3.4(b) is simply:

$$T_{ckq} + T_{setup} + T_{settle} < 2UI \quad (3.4)$$

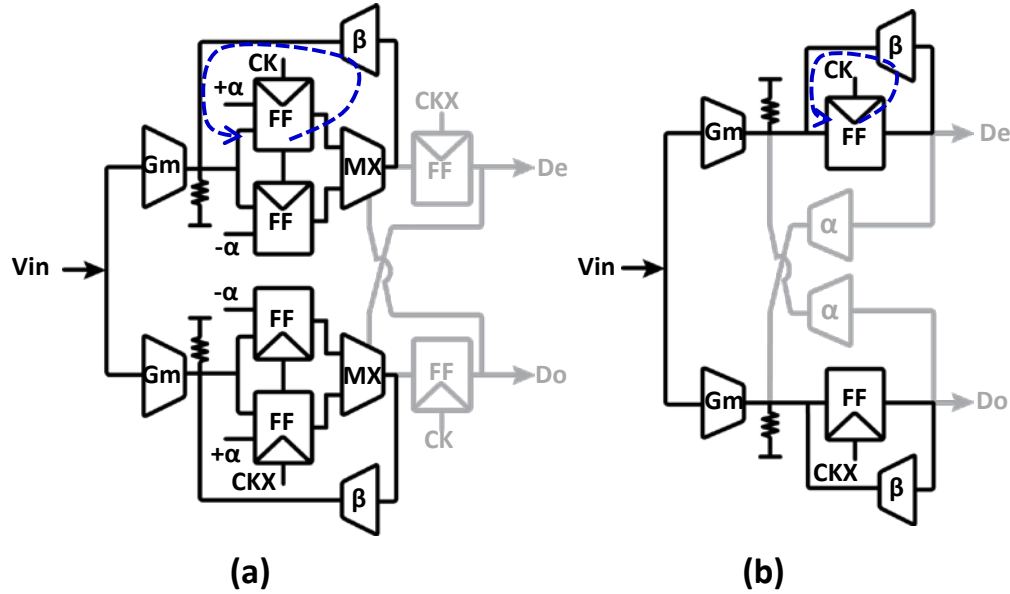


Figure 3.4: 2nd tap critical timing path for (a) 1st tap unrolled and 2nd tap closed-loop DFE and (b) 2-tap closed-loop DFE

In order to alleviate their timing constraints as well, one could unroll further taps (beyond the 1<sup>st</sup> tap) in order to remove their analog settling from the timing loop as well ([22], [23]). However, the exponential growth in required hardware (i.e. slicers, MUXes, etc) as the number of unrolled taps is increased will introduce substantial additional loading due to the parasitic capacitance of the (long) wires and hence inherently degrade the energy-efficiency of the solution.<sup>16</sup>

Note that in contrast to loop-unrolling, if one were able to physically close the timing critical 1<sup>st</sup> tap, meeting the timing for the later taps is relatively easy. This leads us to re-examine techniques to directly close the 1<sup>st</sup> tap.

<sup>15</sup> The exact value of  $T_{d,MX}$  depends on the MUX topology and its loading. However, due to high logical effort as well as self-loading,  $T_{d,MX}$  typically exceeds 1UI at our target speed.

<sup>16</sup> As discussed in [27], receivers with loop-unrolled DFE may require additional edge samplers to maintain a high phase detector update rate, which could further compromise energy-efficiency.



### 3.2.2 Optimization #2: Reduced Latch Gain

While merging the latch and the summer makes a significant step towards improving the DFE timing, pushing the DFE to operate with a UI of  $<1$  fanout-of-four 4 (FO4) inverter delay requires further optimization. In particular, to meet our target of 60+Gb/s,  $T_{dq}$  must be at most  $\sim 16$ ps ( $\sim 0.75$  FO4 in this technology), and thus techniques to reduce this delay are essential.

Recall that the purpose of the latch within the merged structure is to convert the input analog signal into an output digital level. However, as with any other amplifier-like circuit, there is a direct trade-off between the required latch gain,  $A_{lat}$  and its propagation delay,  $T_{dq}$ . Thus, if we can reduce the required latch gain, we can also reduce  $T_{dq}$  and hence the speed of the DFE. As shown in Figure 3.6, the first step in this optimization is to reduce the required digital output level (i.e. the cursor amplitude  $V_d$  after the latch).

To understand the extent to which we can reduce  $V_d$ , we need to keep in mind that in order for the DFE to function as intended, the input devices of the feedback stage  $\alpha$  have to interpret the signal produced by the latch as a digital level. In other words,  $V_d$  can be lowered to the point that it is just greater than the clipping voltage  $V_c$  of the feedback stage and so that any noise that was accumulated on the forward path is attenuated sufficiently that it does not propagate back to the summation node.

The feedback stage is typically implemented by a differential pair, and hence  $V_c$  is directly proportional to the overdrive voltage  $V^*$  of the transistors within this pair.<sup>17</sup> Specifically, when the differential amplitude of the signal fed into the pair exceeds the devices'  $V^*$ , the voltage-transfer curve (VTC) of the stage begins to compress.<sup>18</sup>

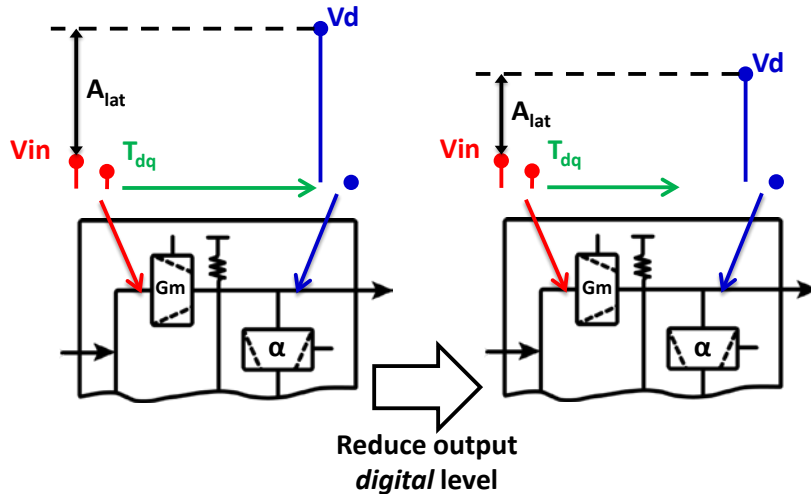


Figure 3.6: illustration of optimization 2 step 1 – reducing the required output digital level

<sup>17</sup> Note that  $V^* = 2 I_{ds}/g_m$  as defined in [47]. It stems from [48] and is equal to the overdrive voltage  $V_{ov} = (V_{gs} - V_{th})$  for square-law devices.

<sup>18</sup> One rule of thumb as derived in [78] for long-channel devices shows  $V_c \approx \sqrt{2}V^*$ . The relationship between  $V^*$  and  $V_d$  in terms of noise performance will however be explored further in 3.2.4.



By choosing the  $V^*$  of the input pair to be small, we can reduce  $V_c$  and thus lower the required output digital level. However, lower  $V^*$  results in lower device transition frequency  $f_T$  and inevitably creates larger capacitive loading to the forward stage. Therefore, the  $V^*$  should typically be lowered only to the point that the net charge (i.e., if  $C_{g,tap}$  is the gate capacitance of the feedback pair, the charge is  $C_{g,tap}V^* \propto V^*/f_T$ ) that must be delivered by the latch is minimized. In a typical 65nm technology, this occurs for a  $V^*$  of  $\sim 200\text{mV}$ , leading to a  $V_c$  of  $\sim 280\text{mV}$ . To further lower the latch gain requirement, as shown in Figure 3.7, we can increase the input analog level (i.e. the cursor amplitude  $V_{in}$ ). In order to achieve this without compromising the receiver's overall input sensitivity, we can simply add gain stages in front of the latched summer. Note that by implementing additional gain outside of the feedback loop, the overall input sensitivity can be maintained without compromising the latch delay or DFE timing constraint.

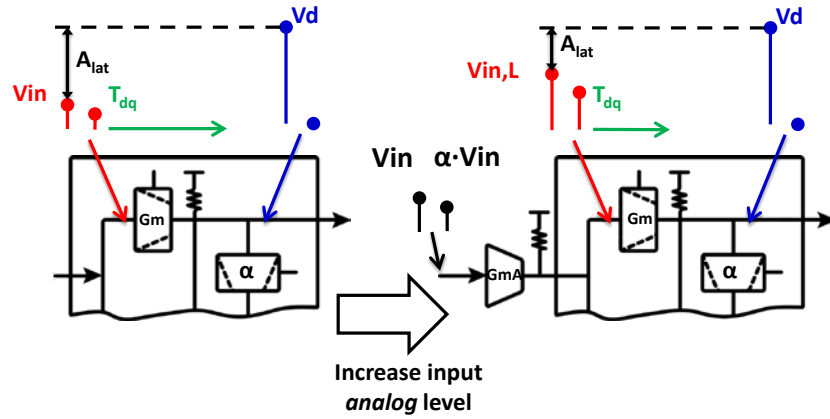


Figure 3.7: illustration of optimization 2 step 2 – increasing the input analog level

In practice, there is however a limit on how much gain can be assigned to this additional front-end stage. Since the first-tap ISI is not removed until the latched summer's output, the input pair of the latch should process the signal coming from the previous stage linearly. This means that unlike the feedback stage, the maximum signal level at the output of  $G_{mA}$  should be less than the clipping voltage  $V_c$  of the latched summer input. To increase the latched summer's input linear range and therefore allow larger front-end gains as well as peak device  $f_T$ , it is typically desirable to choose as large of a  $V^*$  as possible given the available headroom. In this technology with a 1.2V supply, the  $V^*$  of the latched summer input pair is thus set to  $\sim 350\text{mV}$ .

### 3.2.3 Optimization # 3: Dynamic Latch Design

Even after applying the previous two optimizations, as we will describe in further detail next, utilizing a traditional CML latch would still limit the operating speed of the DFE to be below our final target. In order to achieve the true speed limits of a given technology, circuits operating at these speeds should utilize relatively low ratios of input capacitance

to total output capacitance. This means that any structures with large inherent self-loading will either be unable to operate at these speeds or suffer significantly in terms of their power consumption. As analyzed in detail in Appendix B, the regeneration pair of a traditional current-mode-logic (CML) latch design creates substantial self-loading to the circuit. In fact, the bandwidth of such a latch is limited to  $f_T/4$  even with both the gain and electrical fanout (i.e., the ratio of the external load capacitance to the input capacitance of the latch) set to 1. A traditional CML latch is therefore not very well suited to our goals for the latched summer.

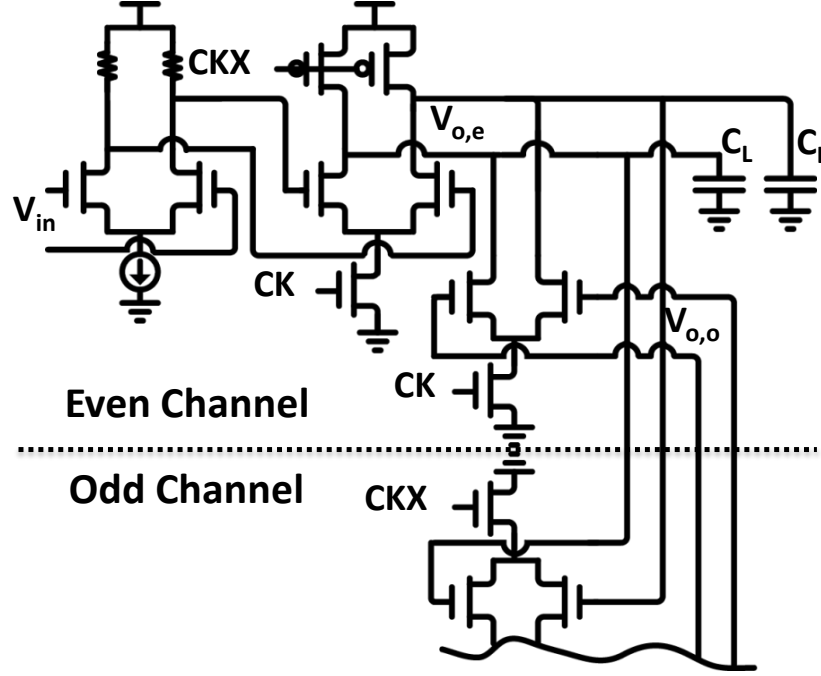


Figure 3.8: closed-loop 1st tap with dynamic latch

Fortunately, we can leverage the earlier optimization of making the latch gain relatively low and instead utilize a dynamic latch design similar to [33] for the latched summer. Figure 3.8 illustrates both the latch and its use in this context, while Figure 3.9 shows representative operating waveforms that highlight how the input data is recovered by this latched summer. When  $CK=1$  ( $CK=0$ ), both the PMOS triode loads and the NMOS tail current source are enabled so that the latched summer in the even (odd) path is in its transparent phase that amplifies the input signal. Meanwhile, the latched summer in the odd (even) channel has both the PMOS load devices and the NMOS tail device turned off so that it is in its opaque phase, and its held output data is fed-back to the correction pair of the even (odd) path to generate the 1<sup>st</sup> tap current pulse  $I_{\alpha,o \rightarrow e}$ . This correction pulse therefore cancels the 1<sup>st</sup>-tap ISI on the input signal at the same time as the input is being amplified by the latch.

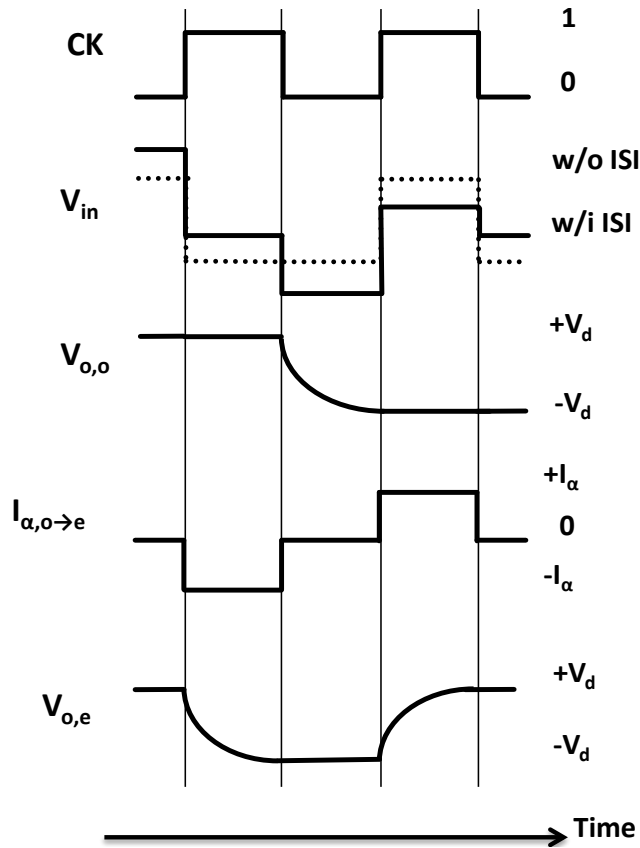


Figure 3.9: 1st tap operation waveforms with an input data pattern of 10010 that is distorted by a single tap of post-cursor ISI

In order to further quantify the advantages of such a dynamic-latch based design over the conventional CML latch-based implementation shown in Figure 3.10. Figure 3.11 compares their optimal power consumption vs. data-rate tradeoffs. Note that the dynamic latch based design includes front-end gain stages, whereas the CML-based design can in principle always achieve sufficient gain during its regeneration phase. The details of the analysis and optimization used to compare the two topologies are provided in Appendix B. Intuitively, since positive feedback effectively “re-uses” the bias current of a single stage to achieve the equivalent of multiple stages of gain, positive feedback only provides substantial power benefits when the overall gain is high. Thus, for low to moderate gain designs, the dynamic latch-based design is always lower power than the CML based design (independent of speed) due to the dynamic design’s reduced self-loading and the fact that each dynamic latch consumes power during only half of the clock period. This case is shown in Figure 3.11 when the overall gain is 4 for both topologies; for this gain the dynamic-latch based design can operate  $\sim 1.6X$  faster than a CML implementation.<sup>19</sup> In contrast, when the required overall gain is 50, the CML based design will be more energy-efficient at lower speeds, as also shown in Figure 3.11.

<sup>19</sup> Note that with a capacitive fanout of 1 and assuming roughly comparable gate and drain capacitance per micron of device width, the dynamic latch itself should ideally be able to operate at  $2X$  higher speed than a

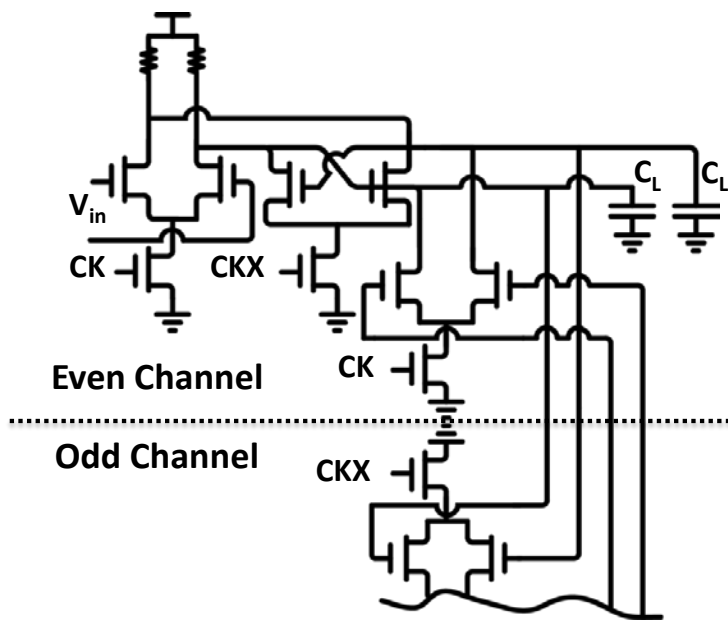


Figure 3.10: close-loop 1st tap with CML latch

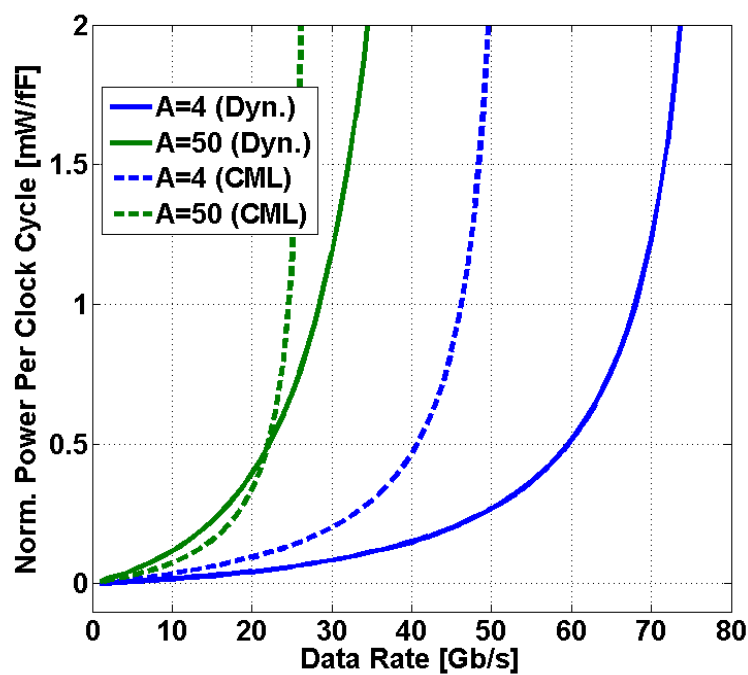


Figure 3.11: power vs. data rate for both CML and dynamic latch-based designs for various total gain requirements

CML latch. In this technology, some of this potential speed benefit is lost due to the increased parasitic capacitance of a PMOS triode-based load (for the dynamic latch) relative to a polysilicon resistor (for the CML latch).

### 3.2.4 Noise Analysis and Implications

As described earlier, reducing the gain of the latch is a key enabler in improving the speed of the DFE. Conceptually, it is appealing to state that one would reduce the gain to the level that the latch's output voltage due to the incoming data signal is just enough to clip the differential pair within the feedback tap (i.e. make  $V_d = V_c$ ). However, if one was to do precisely this, any noise that reduces the magnitude of the latch's output would push the signal back in to the "linear" (or at least more linear) region of the feedback pair's response, and would thus potentially eliminate the noise enhancement advantage from an ideal DFE. Therefore, in order to appropriately set the gain of the latch, we must understand and quantify how the choice of gain impacts the overall noise performance of the link.

Before proceeding to describe this analysis, it is important to note that all DFE implementations must deal with this tradeoff between slicer/latch gain and noise enhancement.<sup>20</sup> The typical approach with positive feedback-based latch designs is simply to increase the gain such that the feedback remains digital even under noise events at the target error probability. While this clearly results in excess latch gain, we believe that this issue has so far not received significant attention because at lower data-rates and with positive feedback this margining perhaps did not result in a dramatic power/performance penalty.

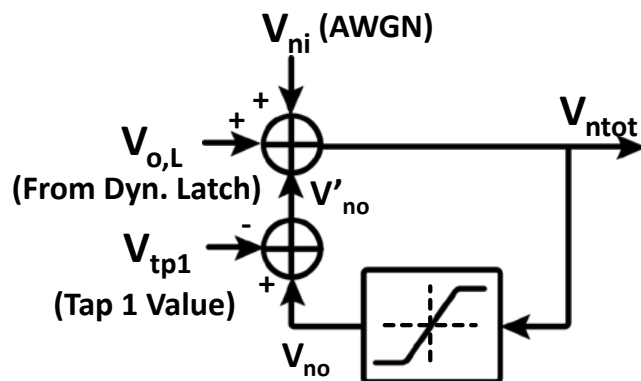


Figure 3.12: illustration of noise shaping and propagation in the DFE feedback loop

In order to simplify the analysis we will assume that all of the errors in the system can be modeled by additive white Gaussian noise (AWGN), but the same basic analysis approach can be extended to include colored noise as well as bounded (deterministic) and/or non-Gaussian error sources.<sup>21</sup> As shown in Figure 3.12, the first step in the analysis is to understand how the added noise (which we will define as  $V_{ni}$ ) would

<sup>20</sup> Note also that during the initial operation of the link, the DFE may be operating on input samples much smaller than in steady-state operation (i.e., after the CDR has locked). Fortunately the BER requirements for CDR locking are not anywhere near as stringent as for steady-state operation, and architectures similar to [79] with separate clock recovery paths can be used to desensitize the CDR locking to any potential noise enhancement within the DFE (which would be on the data path).

<sup>21</sup> The impact of transmitter and receiver clock jitter can also be evaluated in this framework by converting them into voltage noise sources with the methodologies shown in [39][80].

propagate through the feedback tap pair. Focusing on one side of the signal (e.g., +1), this can be achieved by shifting the probability density function (PDF) of the noise by  $V_{o,L}$  (i.e., the nominal latch output level) and then finding the new PDF ( $V_{no}$ ) created by the VTC of the feedback tap pair. We then simply shift the mean of this PDF to remove the effect of the nominal DFE correction signal (i.e., so that  $V'_{no}$  is zero mean). If  $V_{no}$  is still white (i.e., the circuit bandwidth is high enough, which typically requires  $\sim 3\tau$  of settling time ([34]),  $V'_{no}$  will be uncorrelated with the input AWGN. The PDF of the total noise ( $V_{ntot}$ ) will thus simply be the convolution of the PDFs of  $V_{ni}$  and  $V'_{no}$ .

The next step in the analysis is to realize that due to the feedback nature of the DFE,  $V_{ntot}$  will then be passed through the VTC of the feedback pair and convolved with AWGN once again. In fact, this process of  $V_{ntot}$  being passed through the feedback pair VTC and then being added back to  $V_{ni}$  would repeat infinitely. This process will eventually converge to a new, final PDF representing the overall effective noise of the DFE, and the variance of this noise will be larger than that of the original AWGN.

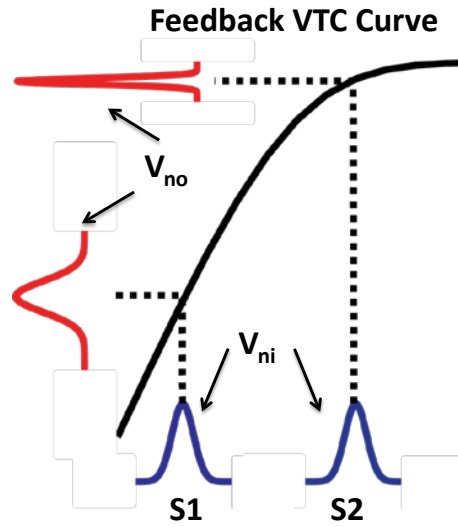


Figure 3.13: two examples of noise propagation – S1: the latch output signal level lands within the high-gain region of the feedback pair’s VTC, and S2: the latch output level lands within the clipping region

As stated originally and illustrated in Figure 3.13, the degree to which the total noise of the DFE is larger than the original input noise depends on the relationships between the nominal signal level, the clipping voltage of the feedback pair, and the variance of the input noise. As an example of these interactions, we have defined two cases in Figure 3.13. In case S1, the nominal output level of the latch is relatively small and lands within the high-gain (linear) of the feedback pair’s VTC, while in case S2 the latch output level lands in the low-gain (clipped) region of the curve.

As shown in Figure 3.14, in case S1 the DFE’s total noise variance increases substantially after only a few iterations through the loop, while in case S2 the increase in noise is relatively mild. To further clarify this effect, Figure 3.15 shows the “noise gain” vs. number of propagations through the DFE loop. In this context, the noise gain is defined as  $\sigma_{eq,DFE}/\sigma_{vi}$ , where  $\sigma_{vi}$  is the standard deviation of the input noise  $V_{ni}$ , and

$\sigma_{eq,DFE}$  is the standard deviation of Gaussian noise that would result in the same BER as the DFE operating with the converged noise distribution  $V_{ntot}$ . As shown in the plot, pushing the feedback pair to nominally operate deeper in its clipped regime reduces both the noise gain and the number of iterations through the loop needed for this noise gain to converge.

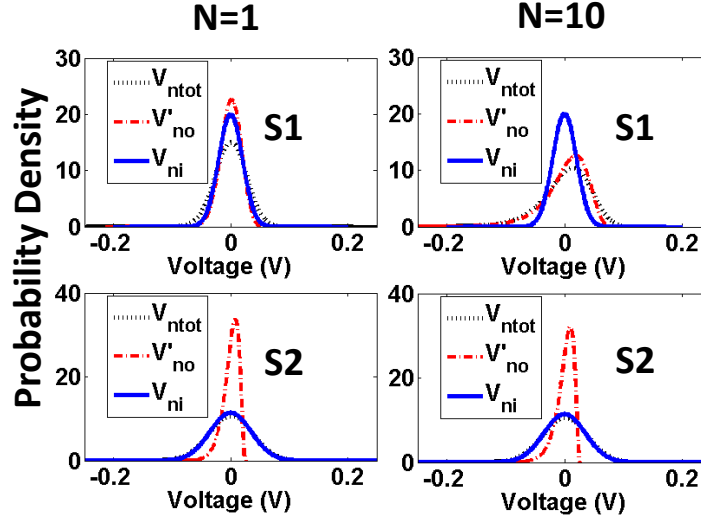


Figure 3.14: noise PDF after 1 and 10 iterations for the two examples in Figure 3.13

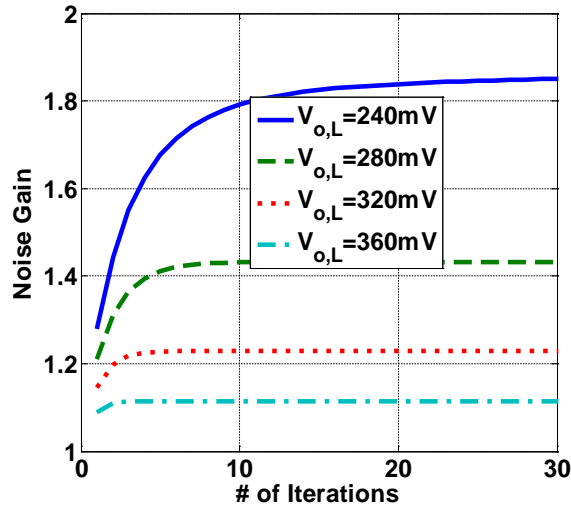


Figure 3.15: noise gain vs. number of propagations with input SNR (i.e.  $V_{O,L} / \sigma_{vni}$ ) = 8

Both the qualitative behavior and the specific numerical values predicted by the above analysis were confirmed to match closely with a time domain behavioral simulation of the DFE. Having verified the analysis through these simulations, we next proceeded to utilize this analysis to decide on the required output swing level (and hence gain) for our dynamic latch design. As shown in Figure 3.16 and using the earlier

definition of noise gain, with our chosen  $V_d$  of  $\sim 350\text{mV}$ , the total noise from the DFE will be increased by  $\sim 10\%$  over the input noise, representing a relatively modest penalty. It is important to further note that the actual thermal noise of the DFE circuits will likely be very small (typically in the range of single mV's) and that other, typically bounded (including residual ISI and offsets, power supply and common-mode fluctuations, etc.) error sources often dominate the BER of the overall link. Thus, in practice the choice of  $V_d$  should be based upon including all of these effects; in the simplest case, one would essentially increase  $V_d$  by the magnitude of the bounded error sources at the target error rate.

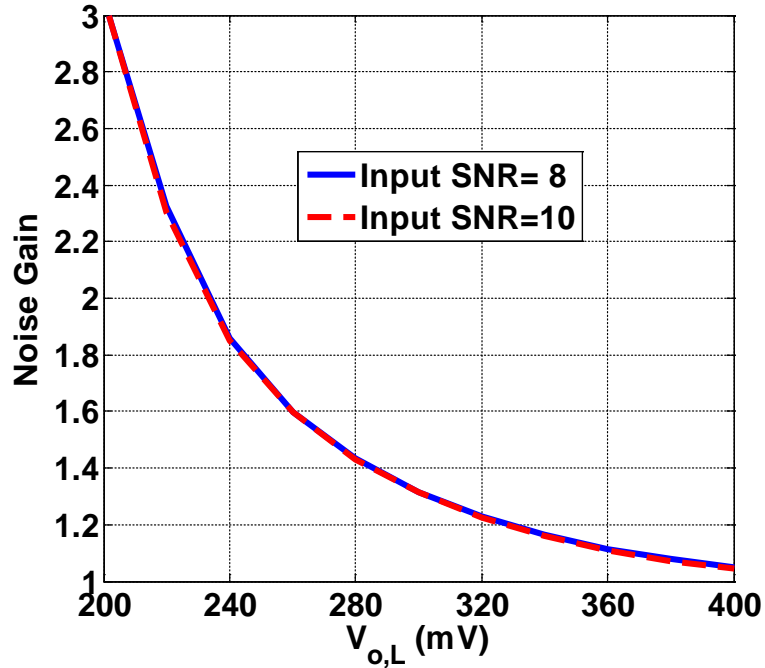


Figure 3.16: noise enhancement vs. latch output signal level under various input SNRs

### 3.3 Complete 3-Tap DFE Circuit Design

Having motivated and analyzed the proposed DFE at an architectural level, in this section we will now address some of the key associated circuit design issues and techniques. In particular, the dynamic latch must be carefully optimized to avoid practical issues related to leakage and sampling aperture, we have so far discussed only how the first-tap loop is closed (and not a complete multi-tap DFE), and clock distribution must be carefully managed to realize the various phases/DC biases required by the design.



### 3.3.1 Dynamic Latch Circuit and Practical Issues

To ensure high-speed operation while retaining energy-efficiency, the dynamic latches rely purely on parasitic capacitors to hold their output voltage during the opaque phase. Since we want to ensure that the output amplitude remains sufficiently large to hard steer the feedback pair, droop in the output voltage due to leakage from the nominally off devices must be considered.

Since the output of the dynamic latch is refreshed every clock cycle, consecutive identical data (CID) will not cause any issues. However, as the data-rate (and hence the frequency of the clock applied to the latches) is decreased, the leakage-induced droop during hold mode will increase. Fortunately, as shown by the simulated amplitude droop vs. clock frequency data (which is half of the baud-rate) in Figure 3.17, even under the worst-case leakage conditions (FF corner and high temperature), operating at  $>30\text{Gb/s}$  will result in negligible amplitude droop. If the same DFE structure must be utilized for data rates lower than this, high-impedance positive feedback “keepers” could be added to eliminate the output voltage droop.

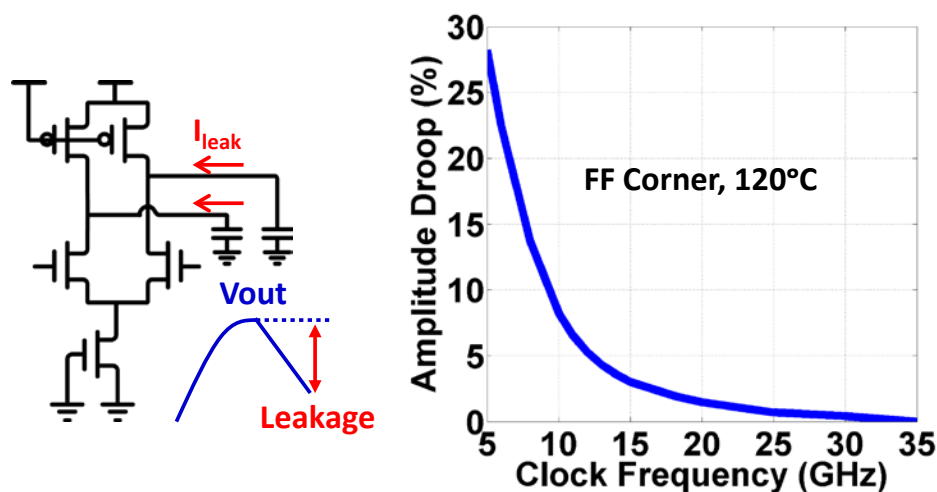


Figure 3.17: amplitude droop due to dynamic latch leakage vs. clock frequency

Another important issue associated with the dynamic latch design is its sampling aperture. As shown in Figure 3.18(a), the input pair is not immediately disabled when we turn off the tail device since the tail node requires some time to be charged up to a voltage high enough to shut off the input pair. During this period of time, any variations on the input signal will be (at least partially) propagated through the latch, potentially causing the effective output amplitude to drop (similar to a hold-time violation in standard digital circuits). To solve this issue, as shown in Figure 3.18(b), we can add a reset switch (similar to [35]) to help quickly charge the tail node when the latch enters its opaque phase. Note that we chose to use an NMOS transistor for this pre-charge not because of its  $f_T$  advantage over a PMOS pull-up, but also because pre-charging the tail node all the way to  $V_{dd}$  would slow down the turn-on transition (i.e., entering the transparent phase).

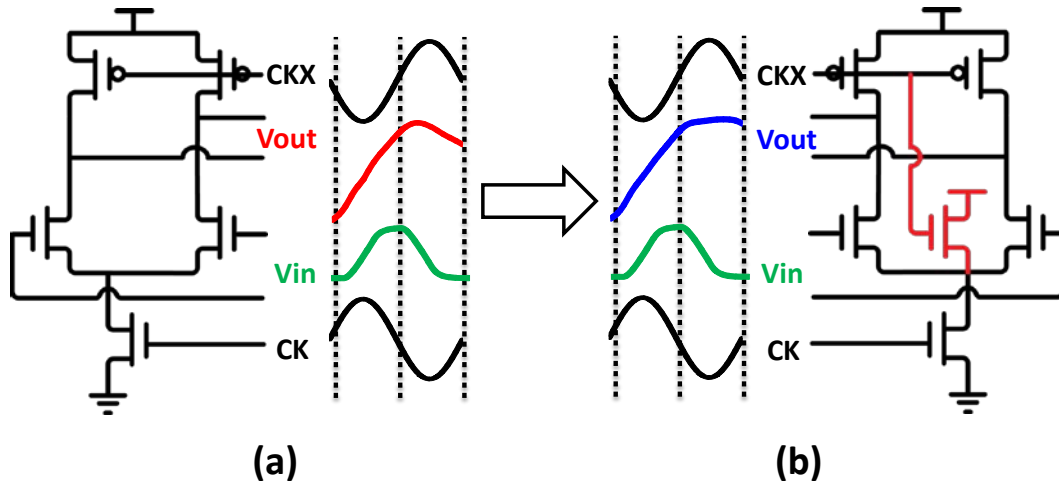


Figure 3.18: (a) dynamic latch without tail node reset and (b) with tail reset device to improve the latch's aperture

Finally, as shown in Figure 3.19(a), to enable adjustment of the 1<sup>st</sup> tap coefficient, we used a voltage digital-to-analog converter (DAC) to control the gate voltage of the feedback stage's tail device. Changing this DC bias voltage directly adjusts the DFE correction current injected into the summation node during the transparent phase. Although the mapping between this gate voltage and the correction current is not linear – as shown in Figure 3.19(b) – as long as this relationship is monotonic, a closed-loop equalizer adaptation will converge to the optimum coefficient.

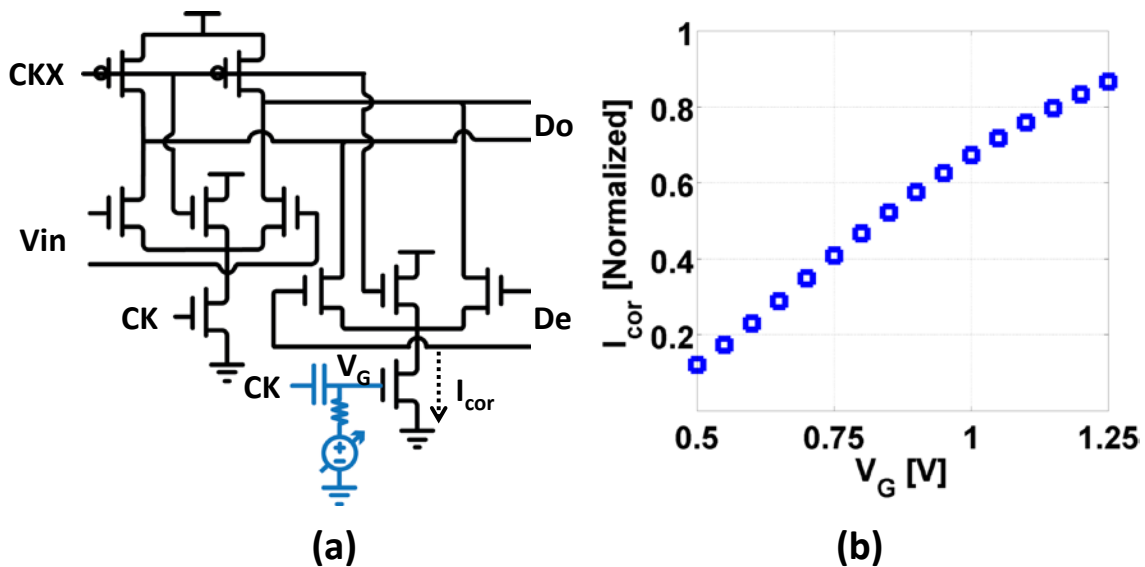


Figure 3.19: (a) adjusting the 1st tap correction current  $I_{cor}$  by changing the DC gate-bias  $V_G$ , and (b)  $I_{cor}$  vs.  $V_G$

### 3.3.2 Overall 3-Tap DFE Architecture

Now that we are able to close the most timing-critical 1<sup>st</sup> post-cursor tap without introducing extra delay to the later taps, it is relatively easy to close the 2<sup>nd</sup> and 3<sup>rd</sup> tap. As shown in Figure 3.20, in order to avoid serialization within the digital feedback path, two separate continuous-time summers (CTS) are used before the 1<sup>st</sup> tap latched summer to cancel the 2<sup>nd</sup> and 3<sup>rd</sup> tap ISI. These linear summation nodes are also convenient locations to independently cancel the overall offset voltage on each path.

In order to enable external measurements of the “analog” (i.e., pre-clipping) eye-diagram after equalization, we added a probe buffer (b2) at the output of the 1<sup>st</sup> latched summer.<sup>22</sup> This additional capacitive loading along with the capacitive loading from the feedback pair, delay latches, and wiring resulted in a relatively high fanout ( $\sim 4$ ) for the latched summer, which would compromise the overall delay and efficiency. We therefore inserted a buffer (b1) at the output of the dynamic latch to help reduce the dynamic latch’s fanout to  $\sim 1.4$  and therefore improve its speed.

Although the overall gain target for our design was  $\sim 4$ , we chose a gain of only  $\sim 1.3$  the linear summer since it is loaded by feedback taps 2 and 3 as well as the offset cancellation. The latched summer and buffer gains are set to  $\sim 1.8$  and  $\sim 1.7$  to achieve the optimal loop delay.

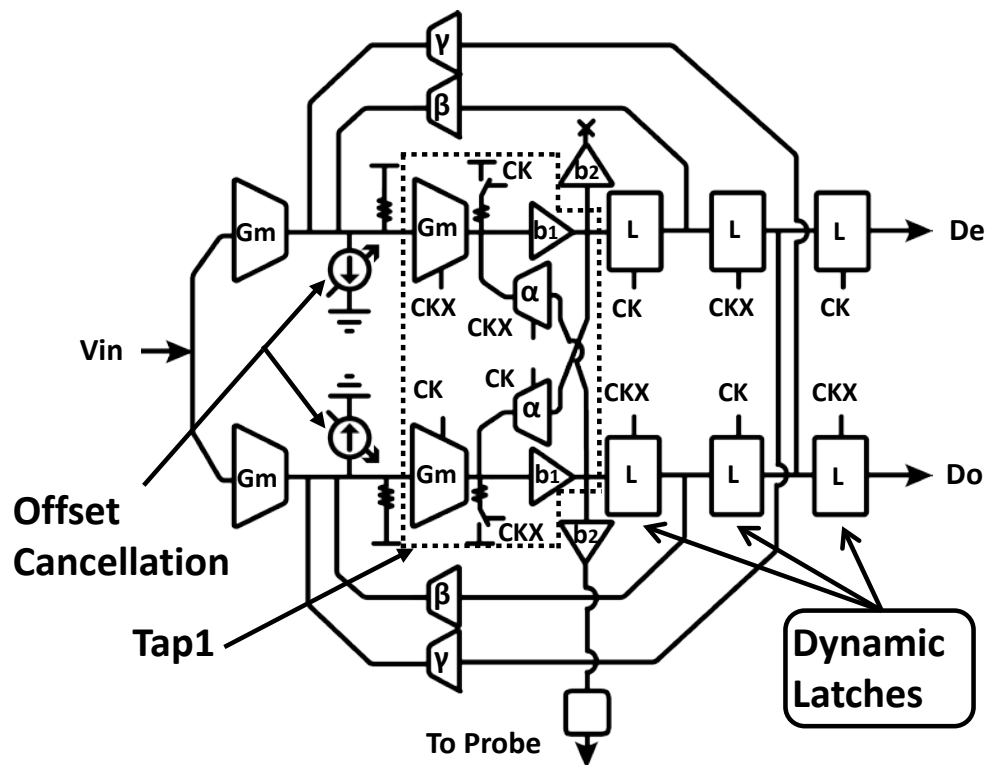


Figure 3.20: complete proposed 3-tap DFE design

<sup>22</sup> This also mimics loading from the clock recovery, adaptation, and/or eye monitoring circuits that would be present in a complete transceiver.

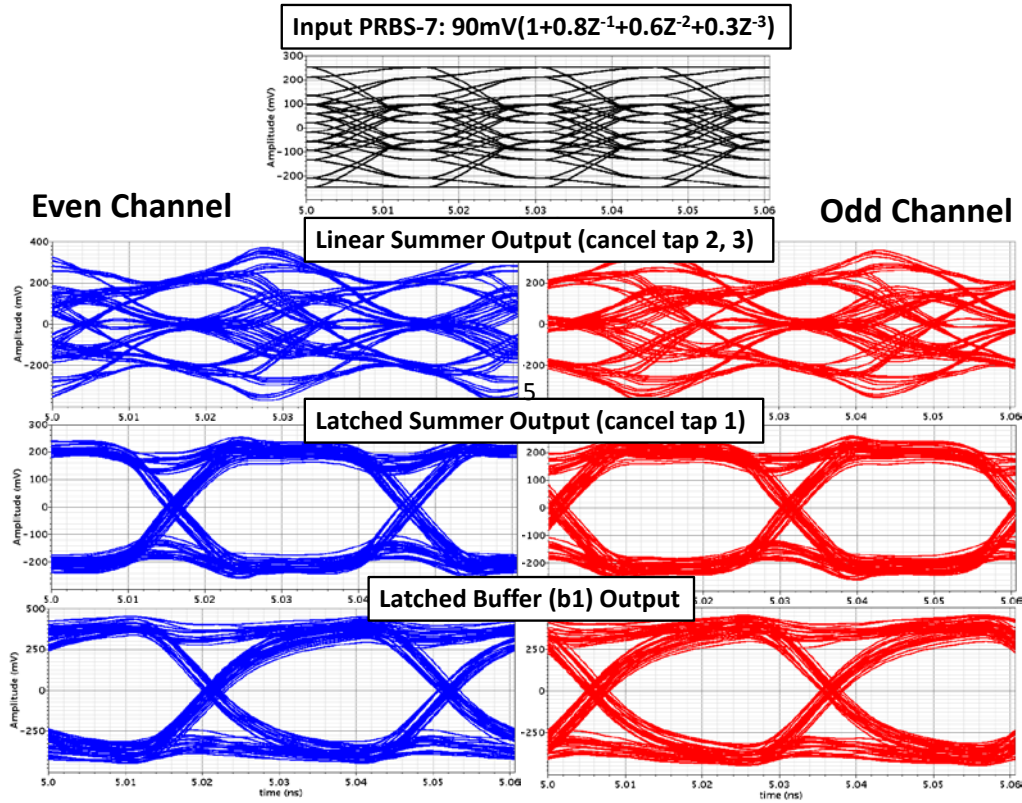


Figure 3.21: simulated post-layout eye-diagrams in the TT corner at each node of the 3-tap DFE

Figure 3.21 shows the simulated eye diagrams in the TT corner at each node of the DFE when it is fed by 90mV differential amplitude input data that has been filtered by a  $(1 + 0.8Z^{-1} + 0.6Z^{-2} + 0.3Z^{-3})$  channel.

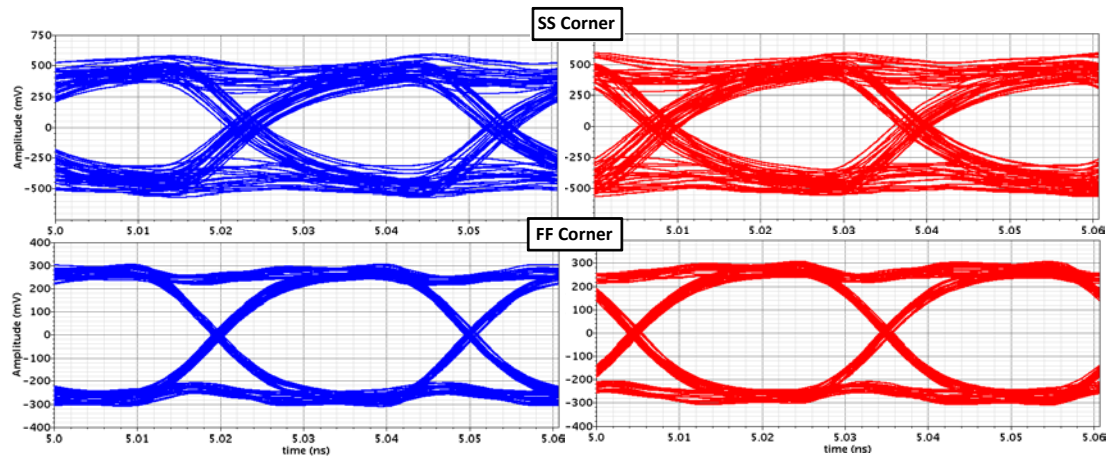


Figure 3.22: simulated post-layout eye-diagrams in the SS and FF corners at the latched buffer (b1) output

Due to the fact that the both the gate bias voltage of the tail NMOS device and PMOS triode load can be adjusted independently as shown in the next section, process corner shifts can be compensated by calibrating these voltages, resulting in good eye margin at the buffer output in both SS and FF corners as shown in Figure 3.22. Both good timing and voltage margins are achieved, verifying the effectiveness of the DFE. In fact, as shown in Figure 3.23, the simulated delay to reach the desired output digital swing level in different corners after calibration is  $\sim 15\text{ps}$ , more than sufficient for the target  $66\text{Gb/s}$  operation.

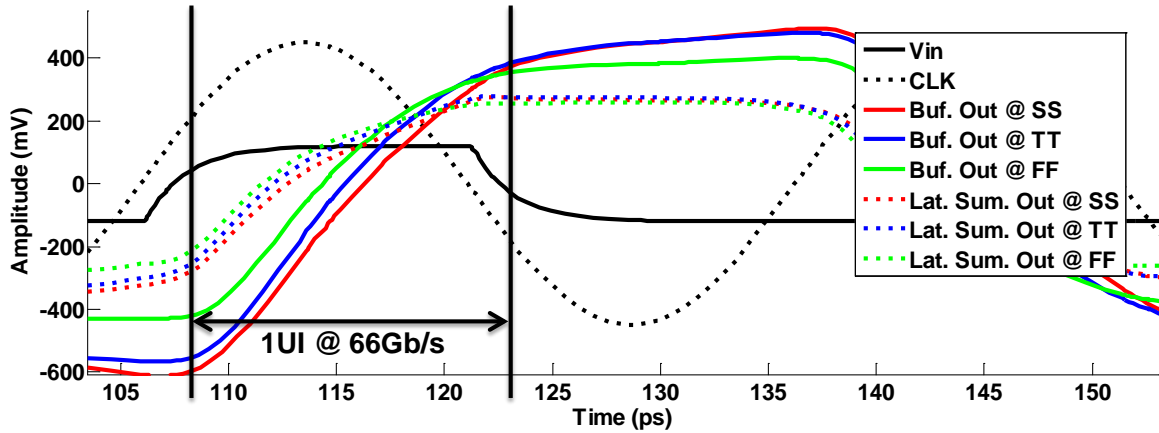


Figure 3.23: simulated post-layout pulse response at the output of the buffer

### 3.3.3 Clock Distribution

Although an on-chip clock generator was not included in our initial prototype DFE design in order to simplify testing, circuits to create the differential clocks with various DC bias points required by the dynamic latch/summer are necessary. First, an on-chip balun was used to convert an external single-ended clock into the differential clocks needed for DDR operation. The differential clocks are then further distributed to each stage as shown in Figure 3.24; note that a similar network could be utilized if the clock was generated on-chip as well. Since the required gate bias voltage for the PMOS load and NMOS tail devices of the dynamic latches are different, we used the balun's secondary center tap to provide the bias voltages for the PMOS devices, but use voltage DACs + AC coupling capacitors to change the gate bias of the NMOS devices. These AC coupling capacitors were implemented with standard MOM capacitors, and as a compromise between signal amplitude loss and area, the capacitors were sized to be  $\sim 10\times$  larger than the total gate capacitance they needed to drive. Even with this small loss from AC coupling, all versions of the clock signals retain 1V of single-ended peak-to-peak swing. With a DC gate bias voltage of 0.5V for the dynamic latch NMOS,  $<0.75\text{V}$  for the 1<sup>st</sup>-tap feedback NMOS, and 0.7V for the dynamic latch PMOS, none of the devices experience a magnitude of more than 1.25V of gate-source voltage stress, thus ensuring device reliability.

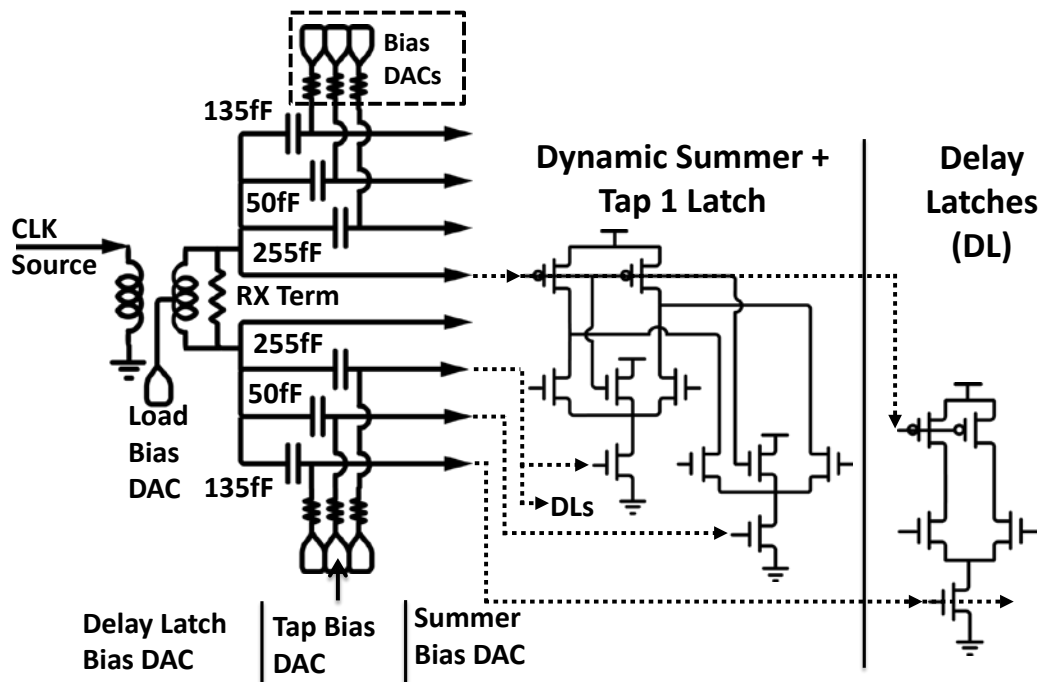


Figure 3.24: DFE clock distribution network

### 3.4 Complete test-chip and Measurement

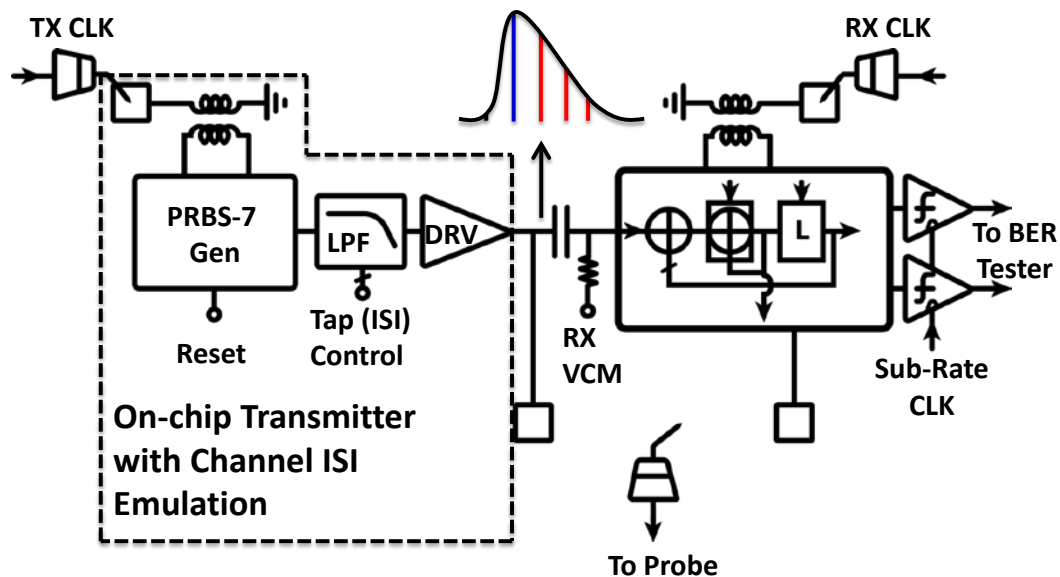


Figure 3.25: complete 65nm GP test-chip including the proposed DFE and on-chip transmitter with channel emulation

In order to experimentally verify the proposed 3-tap DFE design, we taped-out and measured a complete test-chip in TSMC's 65nm G+ technology (Figure 3.25). To enable accurate testing and characterization of the design, both the input to the DFE and the output of the merged latch/1<sup>st</sup>-tap summer within the DFE can be monitored externally.

### 3.4.1 On-chip Testing Structures and Test Setup

The input data to the DFE is generated by an on-chip DDR PRBS-7 generator (similar to [36]) followed by a mixed-signal low pass filter (LPF) with controllable coefficient values and signal amplitude. The use of PRBS-7 is sufficient to emulate an 8B/10B encoded data-stream; it will also create all the 16 possible combinations of the cursor + 3 post-cursor ISI profile. Higher order PRBS sequences can also be supported with the same architecture, but would require additional latches and introduce unnecessary complexity for our purposes.

As shown in Figure 3.26, the LPF filter is implemented with a conventional CML transmitter equalization architecture (e.g. [37]), but with all of the post-cursor coefficients set to be positive. Since the goal of this transmitter is to emulate data that has been filtered by a low-pass channel response, the bandwidth of the transmitter was set relatively low. Specifically, we intentionally allowed the transmitter to introduce a single tap of intrinsic ISI. This allowed us to implement only 3 FIR coefficients (i.e. to add 2 post-cursor ISI) in the TX while still resulting in an overall 4-tap response (1 cursor + 3 post-cursors).

Since we require no gain from the latches and the throughput on each path is half of the target data rate, and since energy-efficiency was not a concern for this test circuit, CML logic was used to implement the delay latches and XORs. This choice reduces the complexity of the TX clock distribution network since the clocks drive only NMOS tail devices and hence require only a single DC bias point. This DC bias for both phases of the clock is conveniently provided by biasing the center tap of the secondary within the balun.

To enable bit-error rate checking of our receiver, StrongArm comparators ([18]) were added to the design to sub-sample the half-rate digital data (i.e. >300mV amplitude). The clocks driving these comparators are generated externally, but are passed through skewed buffers on the die to sharpen their edges and improve the comparator's aperture.

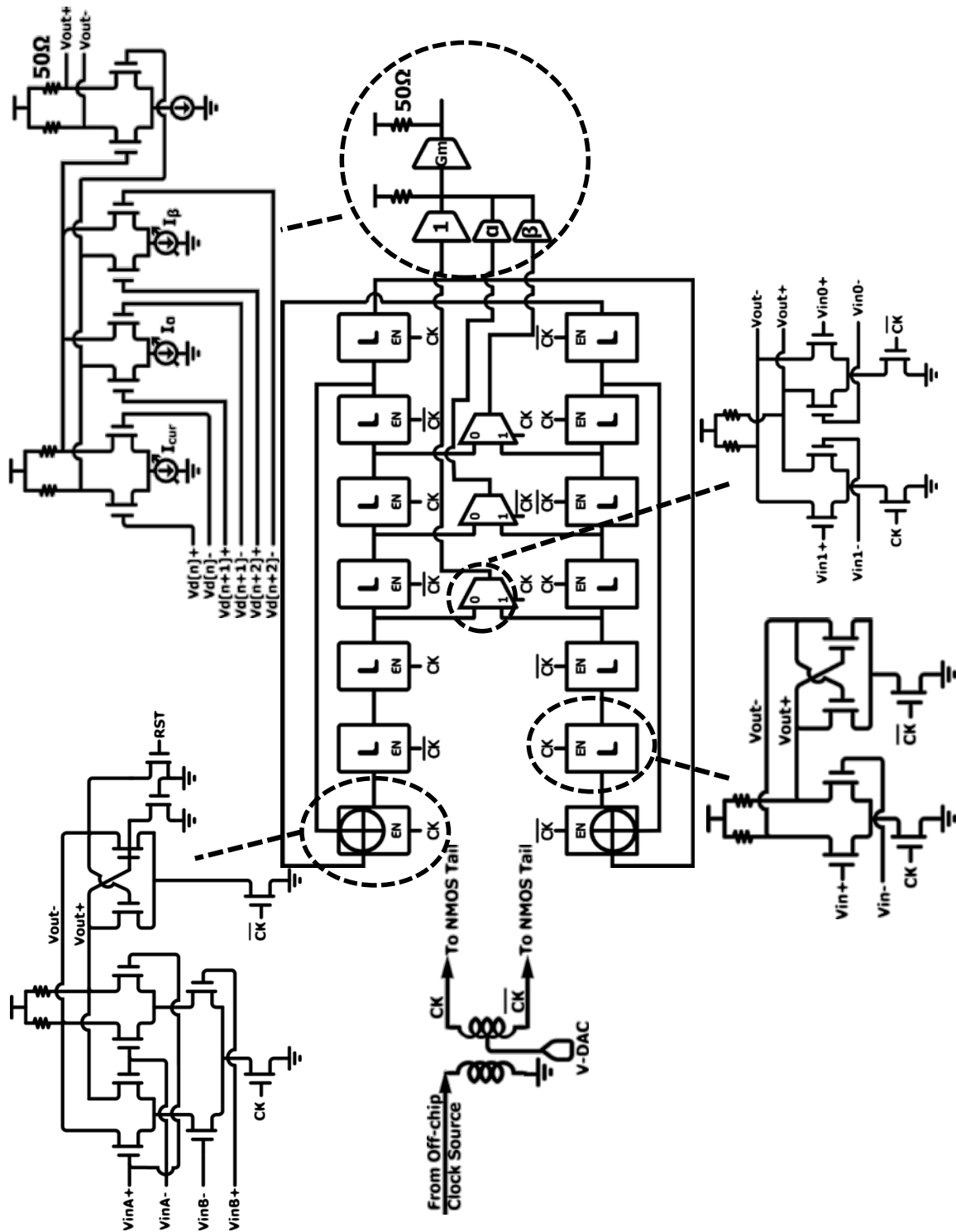


Figure 3.26: transmitter design with DDR PRBS-7 generator and low-pass channel emulation



### 3.4.2 Measurement Results

A die photo of the complete test-chip is shown in Figure 3.27. The chip is wire-bonded on the top side; these wire bonds are used to connect the power supplies, read in digital control bits, receive the sub-sampling clock, and send out the sub-sampled data. Out of the 1.2mm X 1.2mm chip, the DFE occupies only  $\sim 30\mu\text{m} \times 55\mu\text{m}$ ; this compactness is crucial to reducing the parasitics (both capacitive and resistive) in the feedback path.

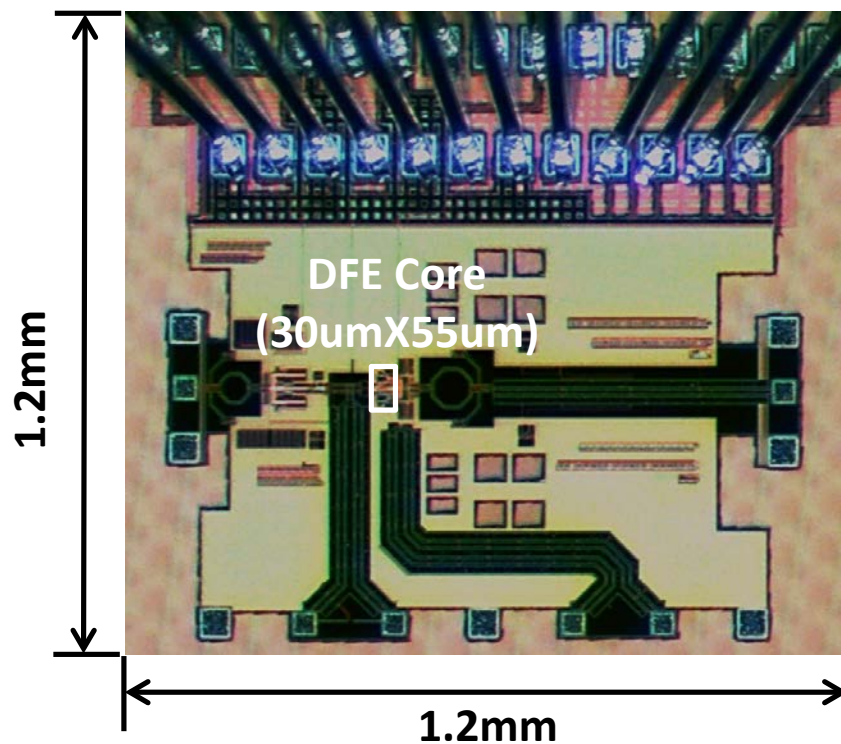


Figure 3.27: test-chip die photo

Probes are used on the other three open sides of the chip to provide the TX (left) and RX (right) clocks as well as monitor the eye diagrams (bottom) before and after DFE (one at a time). Two 40GHz signal generators (Agilent E8257D-540) are frequency locked with an adjustable phase difference to provide TX and RX clocks. An Agilent E8267D was also locked to the two 40GHz signal generators in order to provide the sub-sampling clock. The sub-sampled data is then fed to an Agilent 86130A to conduct bit-error-rate tests. The high-speed signals before and after the DFE were measured through Agilent 86118A 70GHz sampling heads connected to the Agilent 86100C sampling scope.

Figure 3.28 and Figure 3.29 shows the measured signals before and after the 3-tap DFE. The transmitter was configured to send 66Gb/s PRBS-7 data with the LPF set to emulate a  $(1+0.85Z^{-1}+0.6Z^{-2}+0.2Z^{-3})$  channel. The ISI associated with this channel completely closes the eye-diagram at the input of the DFE, as shown in Figure 3.28(a). Note that the PRBS-7 pattern and the precise coefficients of the emulated channel were verified/measured by post-processing the pattern-locked TX waveform shown in Figure 3.29(a). This is achieved by finding the last bit's amplitude for 4 different data patterns

(e.g. 1100, 1010, 0000, and 0001). Since the measured amplitudes can be expressed as a data pattern dependent linear combination of the cursor and 3 post-cursor ISI coefficients (i.e. the 4 unknowns), the 4 data-patterns can be used to construct 4 equations that can then be solved to extract the ISI coefficients.

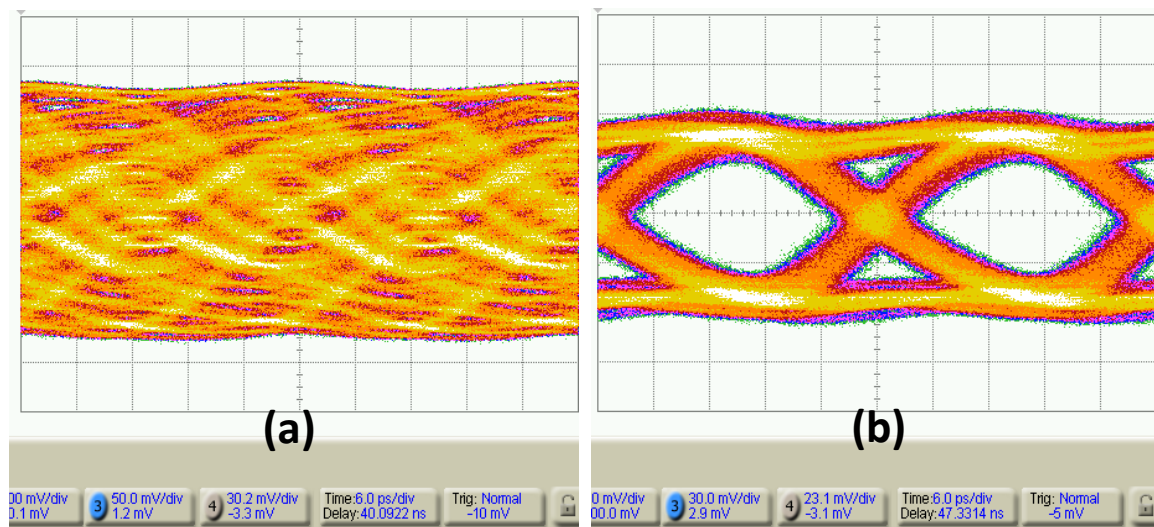


Figure 3.28: (a) 66Gb/s PRBS-7 single-ended eye-diagram before DFE, (b) 33Gb/s PRBS-7 single-ended eye-diagram after DFE

At the output of the DFE's odd-channel merged latch-summer, the half-rate eye is now open and data is equalized, as shown in Figure 3.28(b) and Figure 3.29(b). Note that the eye diagrams/pattern locked waveforms are single-ended and that their amplitudes do not reflect the real on-chip signal amplitude because the  $50\Omega$  buffer (i.e. b2 in Figure 3.20) was designed to have a gain  $< 1/2$  in order to reduce its loading.

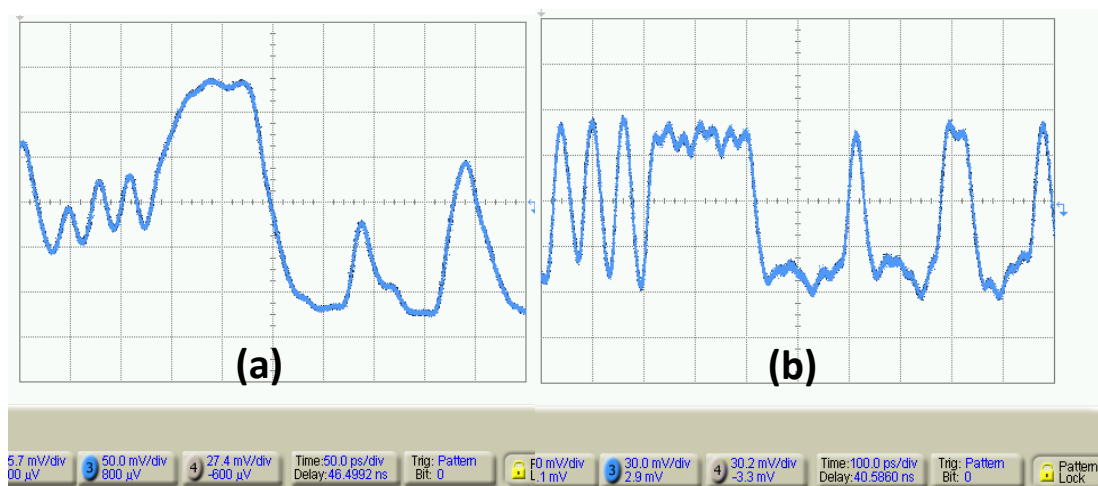


Figure 3.29: (a) 66Gb/s single-ended data waveform before DFE, and (b) 33Gb/s single-ended data waveform after DFE

In order to verify that the feedback correction can indeed be performed in a digital manner, using the circuit configuration shown in Figure 3.30, we next characterized the input signal amplitude required to clip the feedback pair. Since the feedback pair by design has the smallest  $V^*$  and the largest input swing (in this configuration), its non-linearity will set the overall non-linearity of the entire chain. In other words, as we increase the input signal amplitude and measure the output amplitude, we will observe a roughly linear increase until we hit the non-linear regime of the feedback pair.

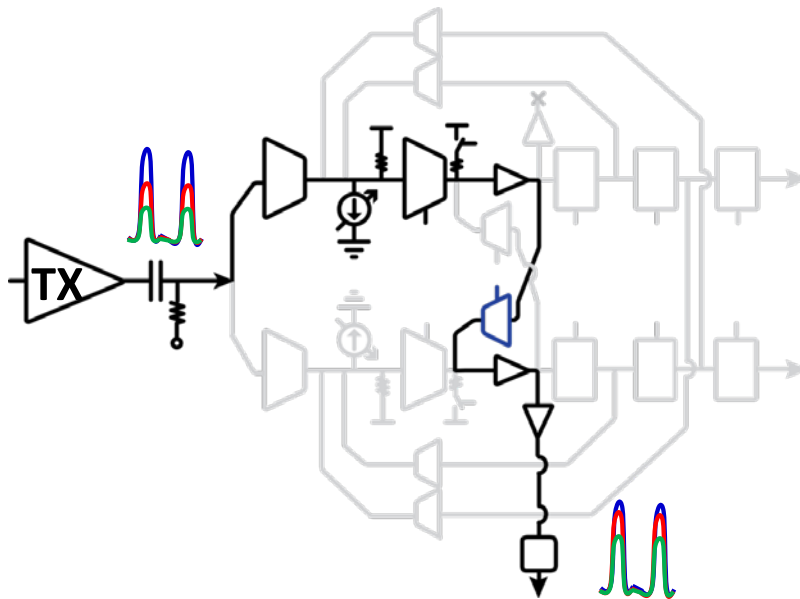


Figure 3.30: circuit configuration used to characterize the feedback pair's clipping voltage

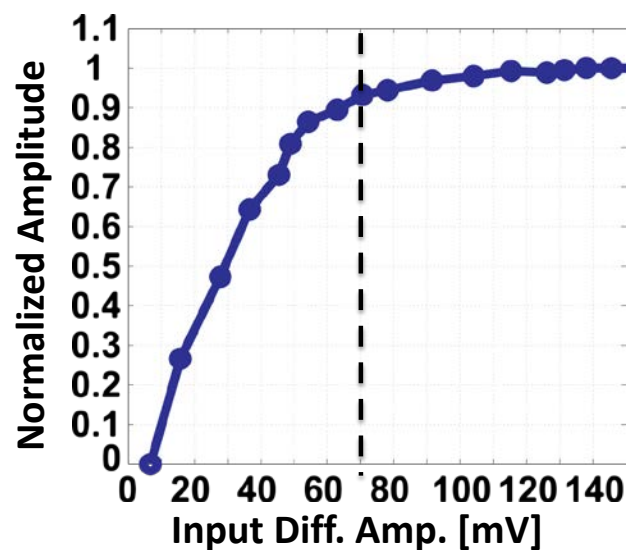


Figure 3.31: measured overall VTC

This measurement therefore allows us to characterize the VTC of the pair, which is shown in Figure 3.31. As implied by the figure, the input cursor amplitude must at the minimum be greater than 70mV to clip the feedback pair. It is worth noting that even if the input signal is less than 70mV, the circuit would still function as an equalizer, albeit as a “soft-decision” equalizer (similar to [38]) – i.e., approaching an analog infinite impulse response (IIR) filter. However, as analyzed in 3.2.4, noise propagation could substantially degrade the BER in this case.

As a final verification of the DFE’s functionality, we once again provided the DFE with a 66Gb/s,  $90\text{mV} \cdot (1 + 0.85Z^{-1} + 0.6Z^{-2} + 0.2Z^{-3})$  PRBS-7 input stream. By changing the RX clock phase from the signal generator and measuring the BER of the sub-sampled 2.0625Gb/s data (i.e. a subsampling rate of 32),<sup>23</sup> we constructed the bathtub curve shown in Figure 3.32. The design achieves  $\sim 0.6\text{UI}$  margin for  $1\text{e}12$  error-free bits; the center of the bathtub was further verified to remain error-free over  $1\text{e}13$  bits.

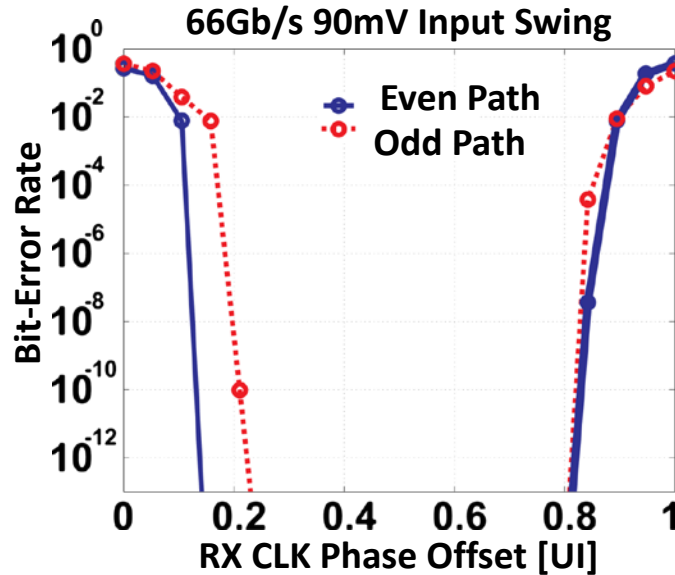


Figure 3.32: measured bathtub curve with a 90mV differential amplitude input signal

The total power consumption of the proposed DFE design is  $\sim 46\text{mW}$  from a 1.2V supply; a breakdown of the power consumed by the various sub-blocks is shown in Figure 3.33. As expected due to its stringent timing constraint, the dominant source of power consumption is the 1<sup>st</sup> tap stage consisting of the dynamic latched summer and

<sup>23</sup> If we label the bits in a PRBS-7 data sequence as 1,2,3,4,5,6,7,8,...127, the down-sampled by 2 sequence will be 1, 3, 5, 7, 9, 11, 13, 15, etc. Given the polynomial  $y=x^7+x^6+1$  we used to generate the original sequence, note that the 8<sup>th</sup> bit in the sub-sampled sequence is 15. In the original sequence, 15 is generated by  $8 \oplus 9$ , which once again applying the polynomial is equal to  $(1 \oplus 2) \oplus (2 \oplus 3) = 1 \oplus 3$ . Bits 1 and 3 appear seven and six entries before bit 15 in the sub-sampled sequence. This means that the sub-sampled sequence obeys the same polynomial as the original (full-rate) PRBS-7 sequence, and hence can also be checked with this same polynomial; this property holds for all sub-sampling rates that are a power of 2. This also implies that all possible bits (i.e., data-patterns) and hence total ISI contained within the original PRBS-7 sequence will appear in the sub-sampled sequence.

buffer. It is worth pointing out that the RX clock power required at the input of our chip was measured to be  $\sim 9.1\text{ mW}$ . However, as shown in Figure 3.24, we added an explicit resistor (RX Term) in parallel with the balun in order to lower its input impedance from  $\sim 200\ \Omega$  to  $50\ \Omega$  in order to simplify matching and testing. If generated on-chip, the DFE would thus require only  $\sim 2.3\text{ mW}$  of clock power.

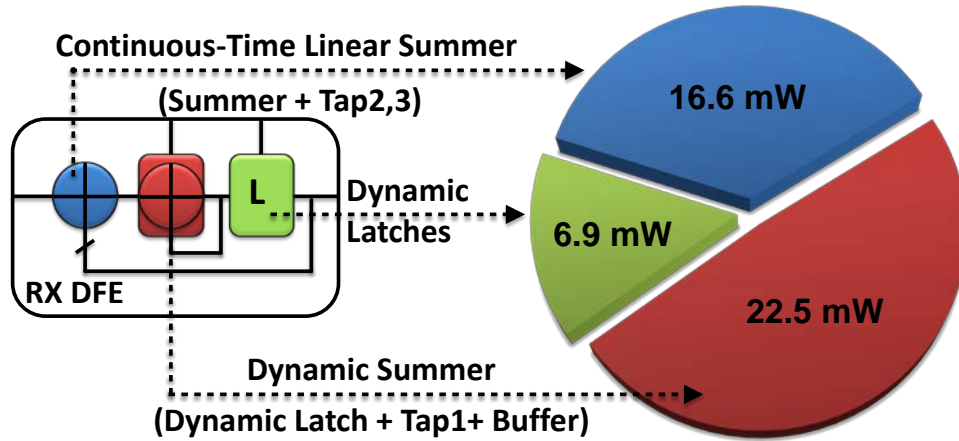


Figure 3.33: DFE power breakdown

In order to further place these results into context, Table 3.1 compares our work with the state-of-the-art 20+Gb/s DFE designs. The design compares especially favorable against the DFE designs in same technology node; the proposed DFE achieves the highest data rate while cancelling a total ISI amplitude of  $\sim 1.7\text{X}$  times the cursor and maintaining  $< 1\text{ pJ/bit}$  efficiency.

Table 3.1: comparison to state-of-the-art 20+Gb/s DFE designs

Reference	[22]	[32]	[24]	<b>Our Work</b>
Process Technology	32nm SOI	65nm GP	65nm GP	<b>65nm GP</b>
Supply Voltage	1.15	1.2	1.2	<b>1.2</b>
Data Rate (Gb/s)	30	21	40	<b>66</b>
# of DFE Taps	15	1	1	<b>3</b>
(Cancelled ISI)/ $V_{\text{cursor}}$	N/A	$0.33^*$	$< 0.63^{**}$	<b>1.65</b>
Power (mW)	92 <sup>***</sup>	19	45	<b>46</b>
Efficiency (pJ/bit)	3.1	0.9	1.13	<b>0.7</b>

\* Calculated using Nyquist EQ boost =  $(1+\alpha)/(1-\alpha)$ , with EQ boost = 6dB as specified in the paper.

\*\* Estimated using the same method as above and assuming complete equalization of the 13dB Nyquist channel loss given in the paper.

\*\*\* Includes CTLE and clock distribution power.

### 3.5 Conclusion

As I/O bandwidth requirements continue to scaling, equalizers – and especially multi-tap DFEs – must continue to increase their operating speeds. As discussed in this chapter, the popular loop-unrolled architecture relaxes the 1<sup>st</sup> tap timing constraint, but degrades the timing of later non-unrolled taps. Since unrolling more taps leads to exponential growth in hardware complexity, solving this issue by further unrolling can result in significantly degraded energy-efficiency.

In order to enable high-speed multi-tap DFEs operating the edge of the technology's capabilities while retaining energy-efficiency, in this chapter we proposed to re-examine closed-loop architectures. In order to solve the critical 1<sup>st</sup> tap closed-loop timing constraint, we proposed the combination of three optimization techniques. First, merging the 1<sup>st</sup>-tap summer and the latch ([32]) relaxes the timing constraint by overlapping (in time) the settling of the summer with that of the latch. Second, reducing the gain required by the latch to the minimum required to clip the feedback pair (and hence achieve nearly noise-free feedback) directly increases the latch's achievable speed. Note that this can be realized without sacrificing input sensitivity by placing additional gain stages in front of the latch, but outside the DFE feedback loop. Finally, leveraging the fact that the latch itself does necessarily need to achieve high gain, a dynamic latch implementation similar to [33] was proposed to overcome the self-loading bottleneck of traditional CML latch implementations.

Since the crux of the latter two optimizations is minimizing the overall gain of the signal chain, and in particular, reducing the gain of the latch itself, we further proposed a methodology to quantify the tradeoff between latch output swing and noise enhancement. The analysis shows that for practically relevant values of latch output swing and feedback pair clipping voltage, the increase in noise can be kept to less than ~10%.

A 3-tap DFE utilizing the proposed architecture and optimization techniques was designed, taped-out, and measured in TSMC's 65nm G+ process. The prototype operates at up to 66Gb/s while achieving 0.7pJ/bit energy-efficiency. These results highlight that with appropriate optimization and circuit architecture, multi-tap DFE designs can continue to be scaled to efficiently operate at even sub-FO4 bit-times, and thus such DFEs show great promise in meeting the needs of future high-speed I/Os.

## Chapter 4

# A Holistic Link Evaluation Platform

While various techniques can be utilized to build efficient building blocks, finding the optimal combination of them (e.g. # of equalizer taps, coefficient values, etc) to achieve the best overall energy-efficiency still remains challenging. Conventionally, the choice of link architecture/building blocks was driven only by system or signal-integrity (SI) considerations at the early phases of the design. As there is no knowledge of the circuit power or noise due to the lack of circuit model and device technology information, such a methodology may result in sub-optimal architecture decisions as the circuit may end up being impractical (i.e. transistors not fast enough) or not meeting BER or power budget.

Fortunately, by leveraging the modeling techniques introduced in the previous chapters, we are able to build a holistic evaluation platform that integrates the necessary information all the way from SI level to circuit and device level to help us find the optimum choice of equalization architectures in a given device technology node. For this purpose, this chapter will introduce the methodologies and practical considerations to build such a framework. With this framework, an example of evaluating a 64Gb/s wireline link in a cable link environment with a 65nm CMOS process node will be shown to prove the effectiveness of this concept. It should be noted that this framework is not intended to replace any circuit design simulators, but rather it serves as an estimator before running extensive simulations to help make better early decisions.

## 4.1 Holistic Evaluation with Circuit Modeling and Device Constraints

As mentioned earlier, the conventional method of specifying link architecture mostly relies on signaling analysis. In such analyses, given a certain channel condition, advanced techniques such as statistical analysis ([39]–[42]) are often adopted to estimate the signaling impairments from channel ISI and timing/voltage noise. However, what are not captured in such analyses are the power and noise contributions of the underlying building blocks. To be more specific, with a particular circuit topology, circuit-level non-idealities such as finite transistor transition frequency, offset, and device generated noise will impact the highest achievable operating speed, power consumption and the overall signal-to-noise ratio. As these circuit-level non-idealities are device dependent, different technology processes will result in different power/BER estimates even with the same circuit architecture. Therefore, with only system-level link budgeting, we can never estimate the true performance for the particular link architecture.

There had been several publications aiming to bridge this gap between the circuit and system modeling. One example ([43]) used analytical circuit models + convex



optimizations to iterate between circuit parameters and system requirements. However, due to the need to enforce convexity, this method was limited to capture a fairly narrow set of circuits (e.g. no DFE modeling).

Another method (e.g. [44]) attempts to optimize the overall system based only on circuit parameters, which can be regarded as a purely circuit-driven approach. To be more specific, in this methodology the amplifier's load resistance is used as a turning parameter for optimizations. While it is a powerful knob to find the circuit level power trade-off, it would contaminate the system-level parameters as it couples to both the gain and bandwidth of the amplifier. In other words, every time a resistor value is changed, a whole complete system level of signaling conditions needs to be re-calculated due to the change of gain/bandwidth. As we include more and more such parameters (e.g. loading capacitances), the complexity of such a framework will grow unbounded, making itself untraceable and inefficient.

Therefore, an ideal framework should give an accurate circuit prediction with all the technology limitations while taking the system-level parameters as the input. In other words, all the circuit related parameters should be derived from these technology and system parameters. Luckily, with the circuit modeling method summarized below, such a goal can be easily achieved.

#### 4.1.1 Circuit Component Modeling

While accurate analog circuit modeling methodologies have been introduced before ([45], [46]), here we will provide a systematic introduction on this method with examples from some of the mostly used building blocks in a link.

##### A. Pre-amp modeling

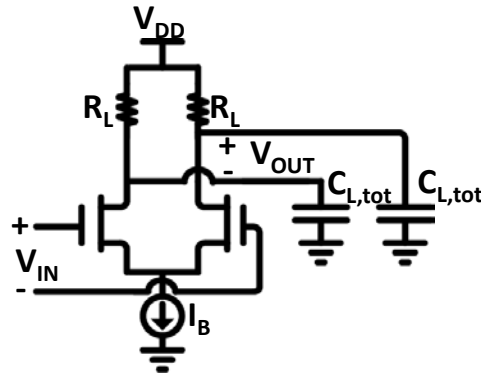


Figure 4.1: a pre-amplifier example

As shown in Figure 4.1, for a pre-amp with a gain of  $A_{pre}$  and a single pole of  $\omega_{p,pre}$ , its transfer function can be expressed simply as:

$$H_{pre}(s) = \frac{A_{pre}}{1+s/\omega_{p,pre}} \quad (4.1)$$



This transfer function only contains the generalized system-level parameters  $A_{\text{pre}}$  and  $\omega_{\text{p,pre}}$  and therefore can be directly used in the SI level of optimizations without introducing any circuit details. To link these system level parameters with circuit and technology details, we can express the gain-bandwidth product of the amplifier assuming a load cap  $C_L$ :

$$A_{\text{pre}}\omega_{\text{p,pre}} = \frac{g_m}{C_{L,\text{tot}}} \quad (4.2)$$

Where  $g_m$  is the target transconductance (i.e. a circuit design parameter) of the pre-amplifier,  $C_{L,\text{tot}} = C_L + C_{\text{dd}}$  includes not only the loading cap  $C_L$  but also the total parasitic drain cap  $C_{\text{dd}}$  from the device itself. We can further express  $C_{\text{dd}}$  in terms of circuit design parameters and technology parameters as:

$$C_{\text{dd}} = \gamma C_{\text{gg}} = \gamma \frac{g_m}{\omega_T} \quad (4.3)$$

Where  $C_{\text{gg}}$  is the total gate cap of the device,  $\gamma$  is the drain-to-gate cap ratio and  $\omega_T$  is the transition frequency (in rad/s) for a given technology and bias condition.

Combining (4.2) and (4.3) and solving for  $g_m$  would then give us the following simple expression:

$$g_m = \frac{A_{\text{pre}}\omega_{\text{p,pre}}C_L}{1 - \gamma \frac{A_{\text{pre}}\omega_{\text{p,pre}}}{\omega_T}} \quad (4.4)$$

As can be directly observed, the circuit design parameter is now expressed in terms of both the system-level and the technology-related parameters. In other words, the circuit parameters can be calculated after the system parameters have been specified to determine the devices' impact on the system performance. As an example, if the required gain or the bandwidth of the amplifier is too large such that the gain-bandwidth product exceeds  $\omega_T$ , the denominator in (4.4) would become negative, indicating an unrealistic circuit implementation in the specific technology with current system-level parameter choices. Similarly the other design parameter  $R_L$  could then be readily derived:

$$R_L = \frac{A_{\text{pre}}}{g_m} \left( 1 + \frac{A_{\text{pre}}}{A_{v0}} \right) \quad (4.5)$$

where  $A_{v0} = g_m r_o$  is again a technology related parameter that could be characterized and stored ahead of time.

We can then further leverage the  $V^*$  (or  $g_m/ID$ ) ([47], [48]) model along with (4.4) to get the power of the pre-amplifier:

$$P_{\text{pre}} = I_B V_{\text{dd}} = g_m V^* V_{\text{dd}} = \frac{A_{\text{pre}}\omega_{\text{p,pre}}C_L V^* V_{\text{dd}}}{1 - \gamma \frac{A_{\text{pre}}\omega_{\text{p,pre}}}{\omega_T}} \quad (4.6)$$

Notice that  $V^*$  is a design-by-choice parameter which is often limited by headroom (i.e. make sure the  $V^*$  is small enough such that with a given  $V_{\text{dd}}$  the devices are still in

saturation) and/or linearity constraint (i.e. make sure the  $V^*$  is large enough such that the max input amplitude won't clip the amplifier).<sup>24</sup>

To figure out the total noise contribution at the output, we can decompose the noise into two parts: 1) noise transfer from the input and 2) noise generation by the transistors. The electrical noise transfer function of this pre-amp is identical to its signal transfer function:

$$|H_{n,tr}(s)|^2 = \left( \frac{A_{pre}}{1+s/\omega_{p,pre}} \right)^2 \quad (4.7)$$

While the input-referred white noise PSD generated from this circuit is also easy to be expressed only in system parameters and device parameters (with  $\alpha$  as the transistor noise coefficient – again it is technology related):

$$|V_{n,i,int}(s)|^2 = \frac{8kT}{C_L} \frac{1-\gamma}{A_{pre}\omega_{p,pre}} \left( \alpha + \frac{1}{A_{pre}} - \frac{1}{A_{v0}} \right) \quad (4.8)$$

With (4.7), (4.8) and with the output noise spectrum transferred from the previous stage we can then easily get the total output noise spectrum for the amplifier:

$$|V_{n,o,tot}(s)|^2 = \left( |V_{n,i,ext}(s)|^2 + |V_{n,i,int}(s)|^2 \right) |H_{n,tr}(s)|^2 \quad (4.9)$$

Note that since we can express both power and noise in the form of system level parameters and loading cap  $C_L$ , we can then easily separate the system and circuit level evaluation. As will be discussed more later in the chapter, unlike [44] where optimizations are directly done through circuit parameters (e.g.  $R_L$ ) and hence making it hard to observe the optimum design point due to the strong coupling between the gain and bandwidth with these circuit parameters, this separation allows us to optimize the link more efficiently.

## B. Continuous-time-liner-equalizer (CTLE) modeling

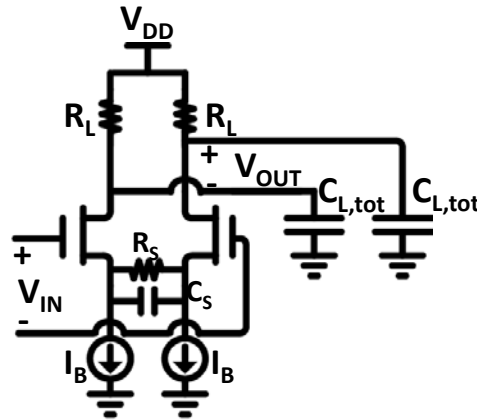


Figure 4.2: a source-degenerated CTLE example

<sup>24</sup> There can be other guidelines on choosing  $V^*$ , e.g. speed or noise requirements

A single-stage CTLE example shown in Figure 4.2 is usually designed to have its two poles located in the same location  $\omega_{p,CTLE}$  to maximize the effective gain-bandwidth. With a peak gain (i.e. the gain at the pole location) of  $A_{pk,CTLE}$  and a zero location of  $\omega_{z,CTLE}$ , the transfer function in terms of system-level parameters can be expressed as ([13]):

$$H_{CTLE}(s) \approx \frac{\omega_{z,CTLE}}{\omega_{p,CTLE}} A_{pk,CTLE} \frac{1+s/\omega_{z,CTLE}}{(1+s/\omega_{p,CTLE})^2} \quad (4.10)$$

It can be derived that the gm/power estimation of a CTLE is no different than that of a pre-amp except we replace  $A_{pre}$  and  $\omega_{p,pre}$  with  $A_{pk,CTLE}$  and  $\omega_{p,CTLE}$ :

$$gm_{CTLE} = \frac{A_{pk,CTLE} \omega_{p,CTLE} C_L}{1 - \gamma \frac{A_{pk,CTLE} \omega_{p,CTLE}}{\omega_T}} \quad (4.11)$$

$$P_{CTLE} = \frac{A_{pk,CTLE} \omega_{p,CTLE} C_L V_{dd}}{1 - \gamma \frac{A_{pk,CTLE} \omega_{p,CTLE}}{\omega_T}} \quad (4.12)$$

Similarly the noise contribution can be found using the standard noise calculation as shown before and summarized in [13]. Again, up to this point, all the power, noise and circuit parameter gm can be expressed with system level and device technology information. Other circuit design parameters can also be easily derived with the following system to circuit parameter relation:

$$A_{pk,CTLE} = \frac{gm_{CTLE} R_D}{\frac{A_{pk,CTLE}}{A_{v0}} + 1} \quad (4.13)$$

$$\omega_{z,CTLE} = \frac{1}{R_S C_S} \quad (4.14)$$

$$\omega_{p,CTLE} = \frac{1 + g_{m,CTLE} R_S}{R_S C_S} \quad (4.15)$$

### C. Decision feedback equalizer (DFE) modeling

More complicated building blocks can be modeled in the same way as described before to directly link the system level parameters with technology information. For DFE in particular, the summer stage which is the dominant power consuming block can be modeled using the pre-amp alike approach with some additional system level information on ISI profiles. Examples on such modeling result can be found in [46], [49], [50] and Appendix B.

## 4.1.2 Building Block Interactions

With the models from each individual blocks, we can connect them together to construct the full data path. As shown before, since we are able to completely isolate the system-

level parameters from the circuit parameters, we can optimize the system-level signal transfers independently before we dive into any circuit details.

To be more specific, when we hook up the blocks, we can first run system-level only evaluations by playing with different system parameters (e.g. gain, pole/zero locations, # of taps). After we confine the design space, circuit modeling could then be turned on to find the required circuit parameters. This step could take longer as it needs to consider the interactions between each block from the offset iterations/ loading changes and noise propagations, etc.

To give an example, shown in Figure 4.3 is a RX front-end consisting of a CTLE and a 1-tap DFE, both of which can be modeled using the previously introduced methodology. As shown in (4.6), (4.8), (4.11) and (4.12), since the power and noise parameters for different blocks are also a function of their loading capacitance, when cascading these blocks together, we have to consider how to instantiate them such that the loading and sizing change of one block won't affect the overall signal/noise flow.

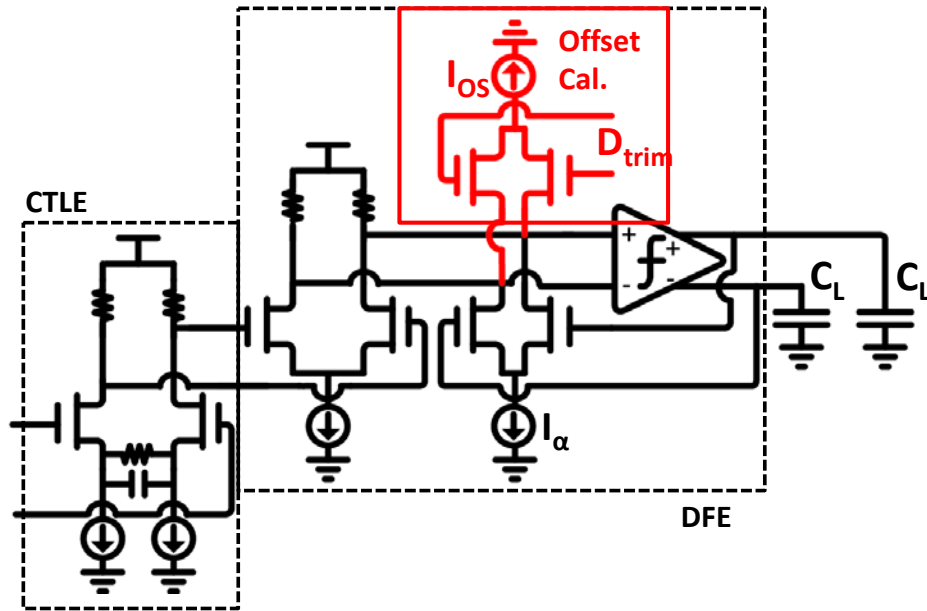


Figure 4.3: a RX with CTLE + DFE example

In the case without considering transistor offsets (i.e. ignore the offset trimming DACs for now), the signal/noise flow is very straightforward. Since the loading cap of a stage comes from the input cap of its next stage, as long as we initiate the blocks reversely from the end to the beginning, the framework would capture the signal and noise propagations properly. To be more specific, as shown in Figure 4.4 for the same RX example without considering the offset trimming, we should first compute the circuit parameters of the “last” stage DFE. The power /noise generation from the DFE summers can be estimated and after getting its input pair's sizing and thus input gate capacitance, we can put it as the loading parameter to the CTLE and kick off the CTLE's power and noise estimation. Then this CTLE noise generation along with its input noise transfer altogether can be transferred back to the DFE to compute the total noise at the input of the slicer that locates inside of the DFE.

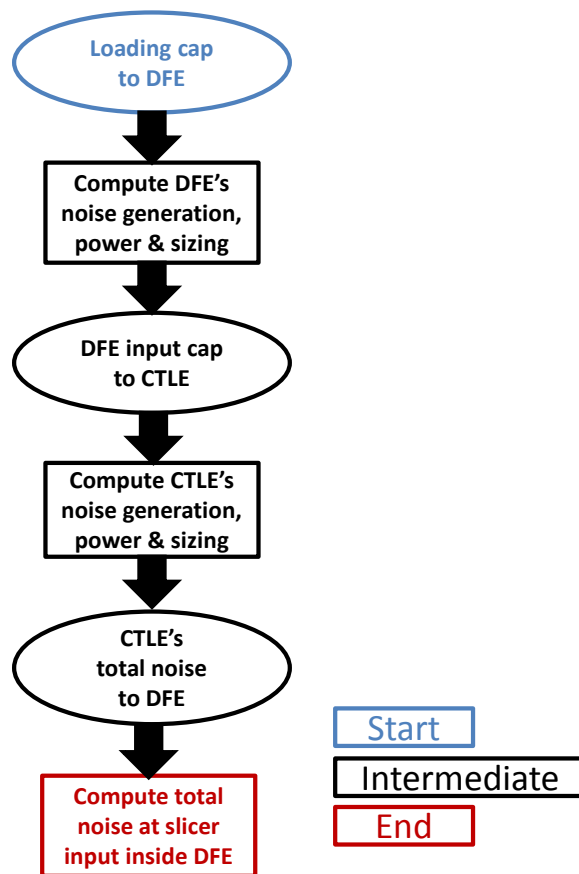


Figure 4.4: block interactions without offset cancellation

On the other hand, when we include the static offset cancellation on the DFE summer node to cancel the overall offset of the amplifier chain, the interactions become more involved. The total offset DAC loading depends on the total accumulated offset to be cancelled and thus relies on the blocks before and after the summer. To fully capture this loading effect from the offset trimming DAC, we need a loop to iteratively update the total offset value when the sizes of the devices are changed.<sup>25</sup>

To be more specific, as shown in Figure 4.5, we can first make an estimate of the total offset value that will put an initial offset DAC loading on the DFE summer. With this loading we can then compute the sizing (and power/noise generation of course) of the DFE and then export this value as the loading to the CTLE to find the CTLE's sizing. With their respective sizing updated we can then find the total offset value at the offset trimming node. If this computed offset value is not within a certain error tolerance (e.g. 1%) of our previously estimated value, we update our estimation to be the newly computed one. This iteration continues until we converge the error to be within the error tolerance range we specify. Then with the converged sizing for both CTLE and DFE, we

<sup>25</sup> It is also possible to get a closed-form expression for the static offset calibration sizing if we lump all the blocks together and write an overall flattened model. However, this complicates the modeling process and reduces the flexibility of trying out different building blocks.

pass the total noise at the output of the CTE to DFE to get the overall noise – i.e. same as we did in Figure 4.4.

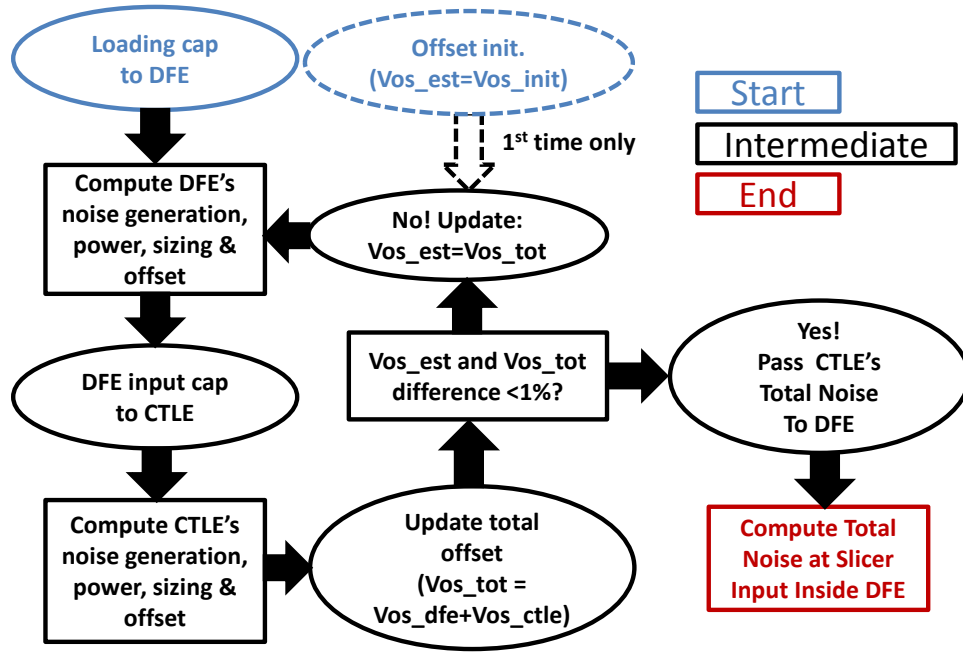


Figure 4.5: block interactions with offset cancellation

## 4.2 Complete Link Framework and Examples

### 4.2.1 Signal Flow and Type Conversion

The previous sections have focused on the circuit modeling methodology and interactions between different blocks. However, since usually both linear and non-linear blocks coexist in a complete link, care must be taken when choosing which domain each block should be evaluated in order to maximize the evaluation efficiency.

To be more specific, blocks such as the channel and amplifiers/CTLE are desired to be linear<sup>26</sup> and thus simply multiplying their signal/noise transfer functions would be the easiest and fastest way of evaluating their cascaded response. Moreover, these blocks only need to carry frequency-domain transfer functions, making the data storage efficient and evaluation fast. On the other hand, blocks such as DFE, FFE and CDR require time-domain responses to find their best coefficients and settings. For this reason, the framework needs to handle time-domain signal processing as well.

Therefore, to maximize the efficiency of the evaluation framework, the signal processing types should be maintained at frequency domain as long as possible. This

<sup>26</sup> While undesired, amplifiers/CTLE could go non-linear. This can be checked and alerted in the circuit modeling within each block. Once this is constrained, we can then simply treat them as linear blocks.

concept is illustrated in Figure 4.6, where an example chain of building blocks for the link is presented. As can be identified, the frequency to time conversion only happens at the FFE block. Before this FFE, all the blocks are supposed to be linear and thus only frequency-domain information needs to be carried and passed by transfer functions. On the contrary, after the FFE, as DFE and CDR are non-linear blocks, time-domain processing is required. By doing the domain conversion only at the boundary blocks, we can limit the required # of time-domain convolutions to a small set and thus improving/saving the overall evaluation speed/memory. It is also worth noting that while blocks such as DFE is usually thought of as nonlinear, some of its components such as summer stages should still remain linear. To capture this, as shown in Figure 4.6, the linear part of the DFE can also be moved into the frequency domain to take its bandwidth limitation into account and thus getting a more accurate estimation.

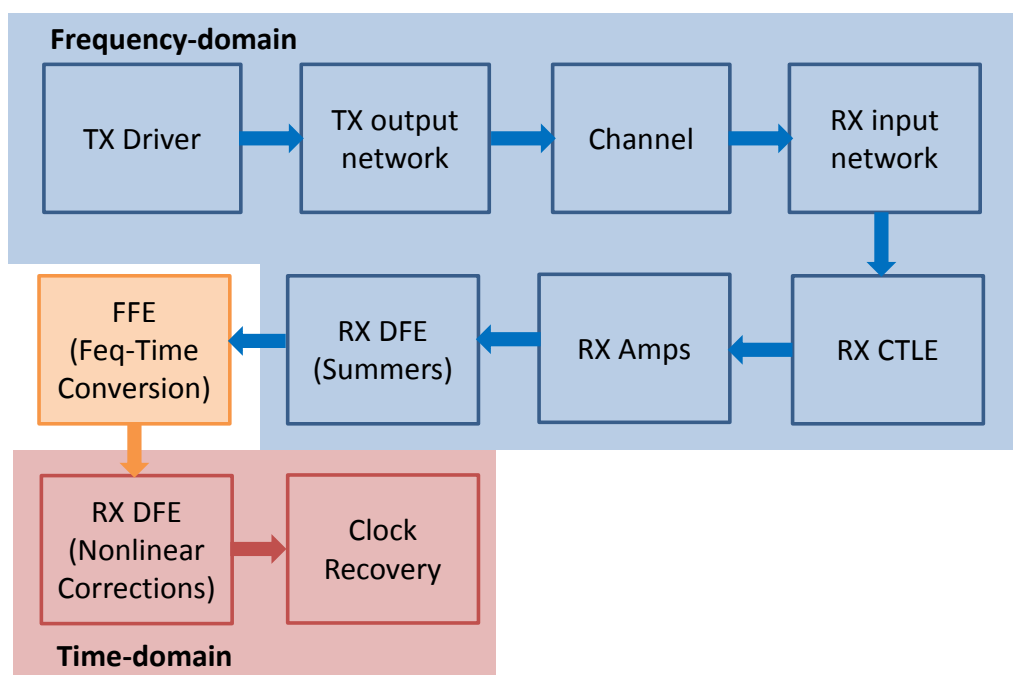


Figure 4.6: signal flow and type conversion example

### 4.2.2 Complete Framework

The overall link framework is shown in Figure 4.7. Testbenches on the top level define what channels and technologies need to be used. They also set the system level parameters for different building blocks and choose what signaling analysis to conduct at each output node of the blocks. Turning on/off circuit modeling and outputting design parameters is also set at this level. In addition, iterations on different system parameters should also be looped at this level and along with the probing results at the output of each building block to achieve both system and circuit level optimizations.

A channel database is essential in the framework as well as it constitutes the basic signal integrity constraints that the link system modeling relies on. The mostly common-used channel sources are s-parameters from either EM simulation or lab measurement. To construct the overall loss effect resulting from the raw channel + packaging losses and any on-chip ESD/bandwidth enhancement elements, the s-parameters for each of these sections should be converted to ABCD matrices ([51]) such that cascading them becomes a simple matrix chain multiplication. The overall voltage transfer function can then be found as the inverse of the cascaded overall A parameter. The resulting overall channel response can then be packed in either frequency domain by directly taking the transfer function or converted to time-domain with inverse Fourier transform to generate the impulse response. Such response will then be used, as explained in the previous sections, in the system modeling of the link to figure out the best signaling strategy.

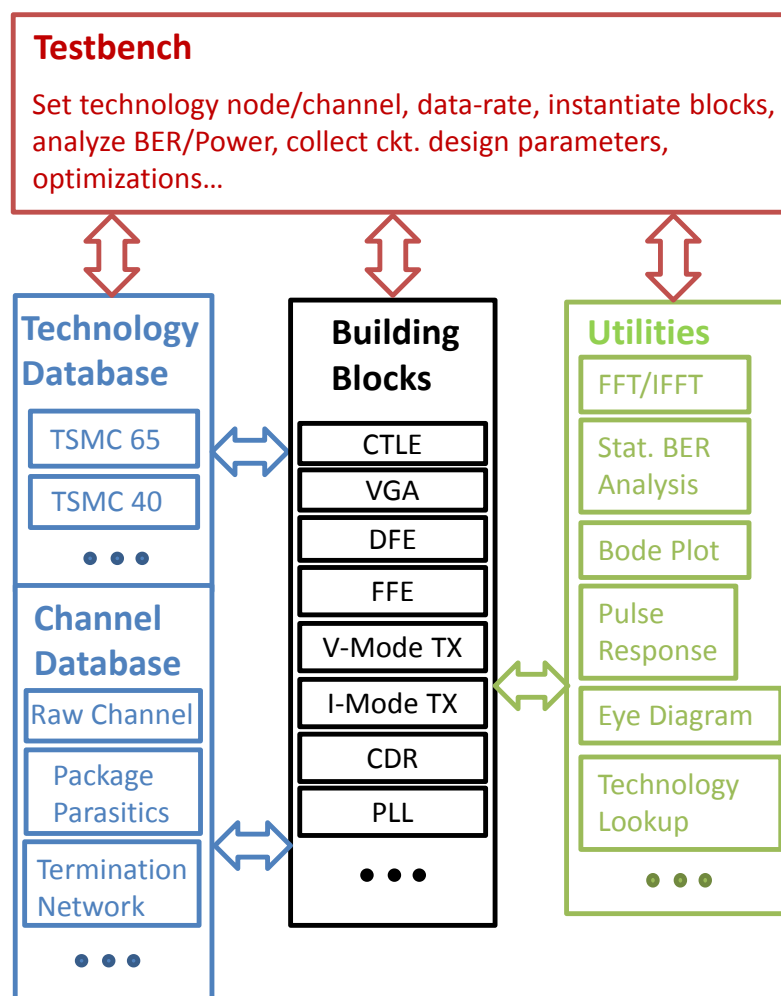


Figure 4.7: overall framework organization

On the other hand, to enable circuit level of optimization for power and noise, technology database is required to store the device parameters that are used in the block circuit modeling. As a reference, this database should contain similar parameters as listed



in Table 4.1 for both analog and digital block evaluations.<sup>27</sup> Device channel length and bias condition should be swept for analog parameters as most of them are bias dependent. Moreover, to capture layout dependent effects such as parasitic caps or well-proximity effect, post-layout reference devices should be used in the characterization testbenches to generate these look-up-tables (LUT). Notice since none of the parameters are unique to a specific technology node, databases for different technology nodes with the exactly same set of parameters can be generated and stored together, making the exploration of technology scaling impact on the same architecture much easier.

Table 4.1: reference item list of the technology LUT

Analog Parameters	Digital Parameters
N/P MOS	L [nm]
L [nm]	$T_{d,F01}$ [s]
$V^*$ [V]	$T_{d,F04}$ [s]
$V_{gs}$ [V]	$R_{up}$ [ohm/um]
$V_{ds}$ [V]	$R_{dn}$ [ohm/um]
$g_{m,unit}$ [S/um]	$C_g$ [fF/um]
$g_{ds,unit}$ [S/um]	$C_d$ [fF/um]
$g_m r_o$ [V/V]	
$C_{gg}$ [fF/um]	
$C_{dd}$ [fF/um]	
$C_{ss}$ [fF/um]	
$f_T$ [Hz]	
$A_{vt}$ [mV/um]	
$\gamma_{noise}$	

Since the building block modules need to interact with the technology and channel database, a well-structured code organization is needed for the system and circuit designers to follow. A source degenerated CTLE example written in Matlab with an object-oriented programming style (OOP) can be found in Figure 4.8, where only 4 Matlab functions are needed to complete the module. The `srcDegenCTLE.m` serves as the initialization function where both system and circuit engineers should access and modify to send their parameters of interests into the design (e.g. pole, zero location/gain of the CTLE from the system designer,  $V^*$ , headroom from the circuit designer). `getUpdate.m` is the interface function to pass these parameters into system/circuit models and get

<sup>27</sup> For completeness, the table lists all the required parameters for modeling. But some of them can be derived from the other ones so only a small set of characterization testbenches is needed.

corresponding output update. Every time the designer changes parameters during his or her trials, this function needs to be called to update the modeling results. For system designers, `getSystemModel.m` is their major playground. Signal/noise flows in either frequency or time-domain before/after the building block will need to be modeled here without touching any technology database. Circuit designers, on the other hand, should focus on the `getCircuitModel.m`. This function encapsulates the circuit modeling methodology as described in section 4.1.1. With some pre-defined utilities, technology parameters could easily be accessed within this function as well.

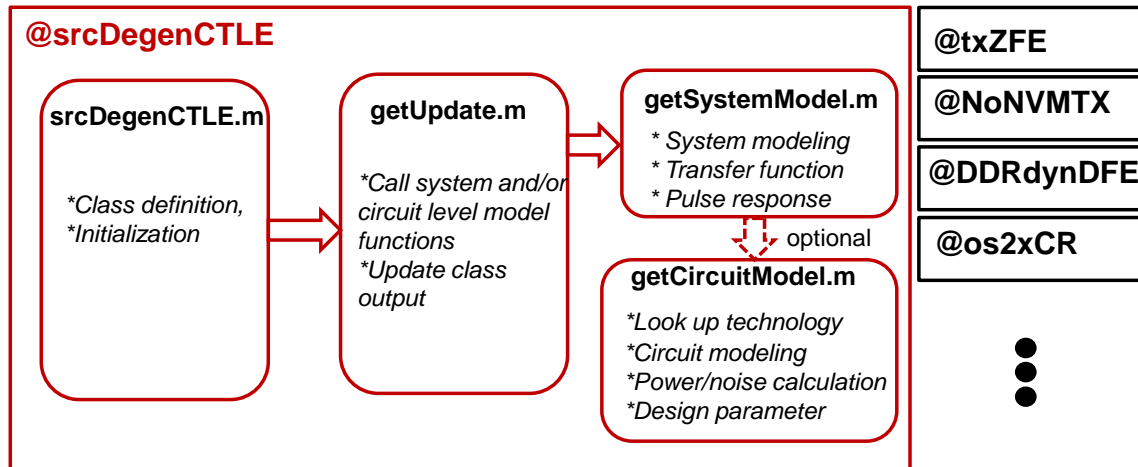


Figure 4.8: matlab code organization for the CTLE block

Finally, in addition to accessing technology database, tools for time /frequency-domain signaling analysis are also included in the utilities module to ease the code management. Example signaling functions include Fourier/inverse-Fourier transforms, pulse-response based ISI analysis, statistical BER calculator, bode plot, eye diagram plot, equalizer coefficients adaptation algorithms, etc. Any user-defined functions can be simply added into this pool as plug-ins for additional analysis without breaking the flow.

### 4.2.3 Example: Signaling Platform for 2025

As mentioned before, the power of this framework lies in its ability to give an early assessment of link architectures under certain channel and technology constraints. With realistic device extraction parameters to provide additional information on circuit generated noise and power, it helps the system and circuit designer make appropriate architecture choices even before building any circuits and running extensive simulations. To validate these ideas, in this section we apply this framework to a platform that is targeted for 60+Gb/s electrical wireline communications.

As shown in Figure 4.9, the signaling platform targeting for 60+Gb/s communications is based on a previously proposed cable environment ([52], [53]). However, in addition to providing physical support, an active transceiver chip is inserted into the connector to separate the loss dominated cable channel environment from the reflection dominated packaging channel environment. With such an active connector, we

could send several low-speed parallel data over the reflective environment first before serializing them up to a 60+Gb/s data stream to deal with the loss dominated environment. Without compromising the overall bandwidth density, each type of link can then be optimized to achieve their maximum energy efficiency.

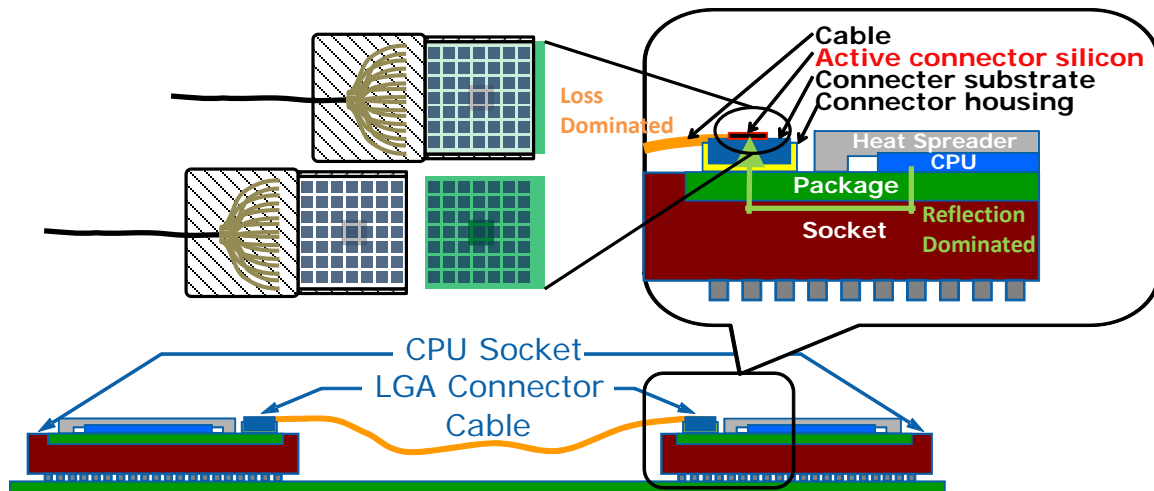


Figure 4.9: Signaling Platform for 2025 (from Intel)

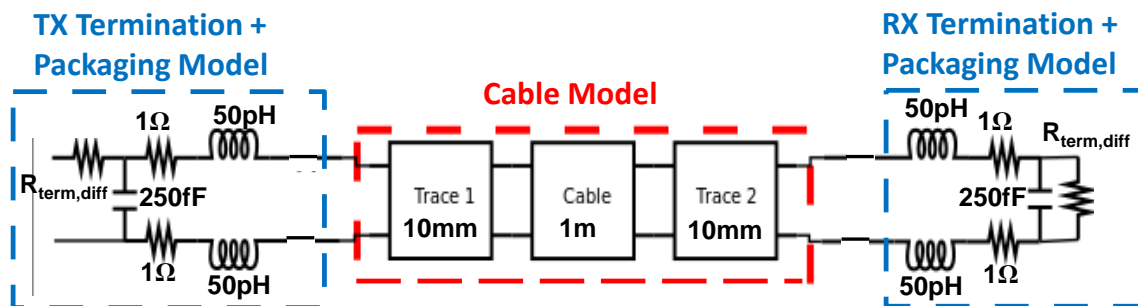


Figure 4.10: a 1-m cable channel for the 64Gb/s link

Focusing on the high-speed portion of the platform, the overall differential channel response including the cable + estimated package parasitics can be constructed with the model in Figure 4.10. With the previously mentioned channel response construction method, both the frequency domain transfer function and time-domain pulse response can be derived and plotted in Figure 4.11. This pulse response shows a substantial amount of ISI that need to be equalized before achieving any useful BER.

To equalize this channel response in a given technology, several architectures could be tested. Figure 4.12 shows an example of using only DFE as the equalizer to equalize the channel pulse response from Figure 4.11(b). As found from both the equalized channel pulse response and the residual ISI distribution, with only DFE, a substantial amount of pre-cursor ISI still remain to be cancelled. Furthermore, with a 65nm post-layout extracted technology database and turning on the circuit modeling for both TX and DFE (based on the model used in [54] for TX and [46] for DFE), we can get both the power and circuit noise estimation. While the TX consumes ~48mW power for

sending the 250mV differential amplitude over the terminated channel, the DFE, due to the large self-loading from the many taps, consumes about ~307.7mW. The generated circuit noise, however, isn't very large due to this large power consumption (i.e. large loading capacitances) and thus contributes  $\sigma_{V_n} \approx 1.27\text{mV}$ . But since the link is heavily ISI dominated, the calculated BER is around 0.106, indicating a failure of the link for robust data communications.

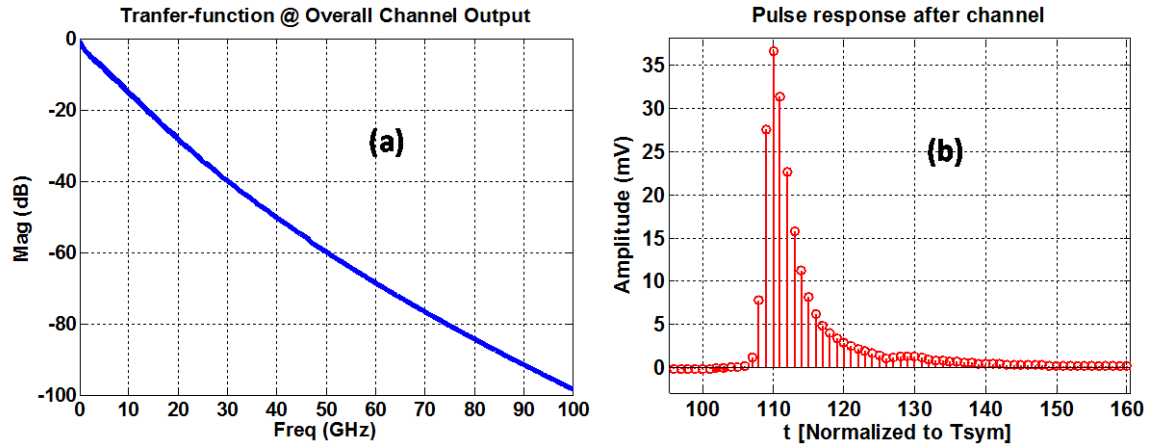


Figure 4.11: (a) frequency response of the overall channel and (b) 64Gb/s pulse response with 250mV differential amplitude

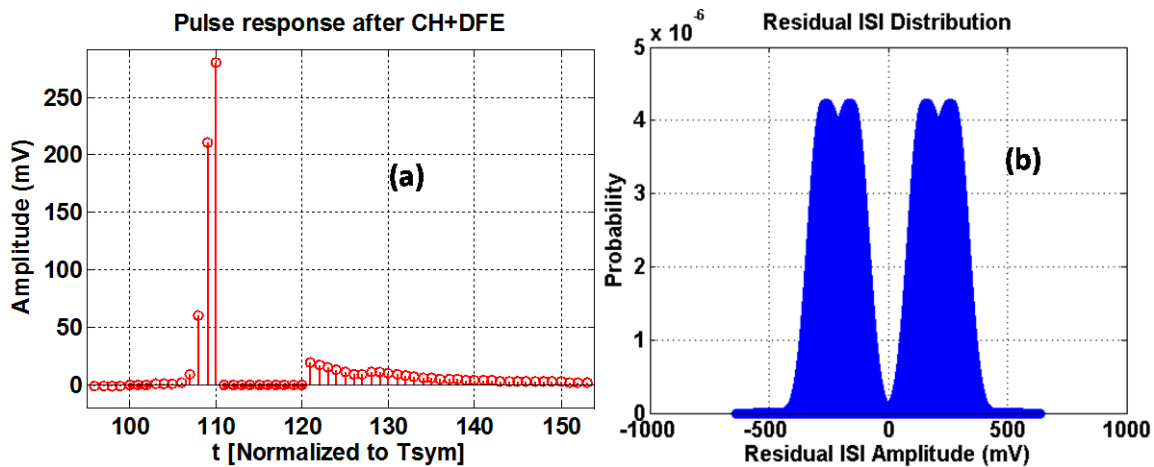


Figure 4.12: channel equalization using 10-tap DFE only - (a) equalized pulse response and (b) residual ISI distribution

As another attempt, realizing the high-loss channel could potentially filter out far-end cross talk from the TX and we have pre-cursor ISI to cancel, we can intentionally lower the TX termination impedance to increase effective transmitter swing while incorporating FFE into the link to cancel pre-cursor ISI. With a link configuration of 30Ω TX single-ended termination resistance + 4-tap FFE (2 pre-tap + 1 post-tap with the zero-

forcing algorithm) + a 2-stage pre-amplifier + a 2-tap DFE, we can get the pulse response and residual ISI distribution as shown in Figure 4.13.

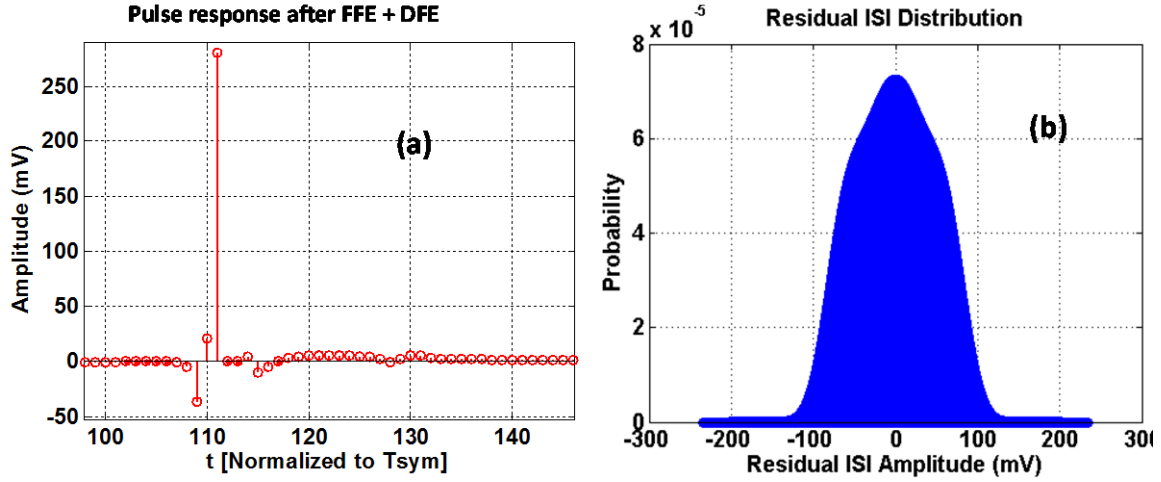


Figure 4.13: link architecture example with unterminated TX driver + 4-tap FFE + 2-stage pre-amp + 2-tap DFE – (a) equalized pulse response and (b) residual ISI distribution

The power consumption for this configuration requires  $\sim 81\text{mW}$  from the TX due to its lowered impedance but only requires  $\sim 7.7\text{mW}$  from the pre-amplifier and  $\sim 18.6\text{mW}$  from the DFE. The calculated overall circuit noise is around  $\sigma_{V_n} = 14.4\text{mV}$  due to the substantially lowered RX power consumption. With such residual ISI and noise profile, the BER is found to be  $\sim 4.9\text{e-}22$ , proving that this is a useful configuration for 64Gb/s data transmission. As can be seen, with a proper choice of the link architecture, both BER and overall power consumption can be improved.

## 4.3 Conclusion

When we keep on pushing the data-rate limit, the conventional SI-driven evaluation method tends to result in inaccurate or infeasible link architecture choices. This issue arises because no circuit or device technology information is taken into account in the selection of link architecture and equalization parameters. As the data-rate requirement approaches the device speed limitation, both the power and noise contribution from the circuits become more sensitive to variations on the system parameters. Thus, to make better architecture decisions, it is highly desirable to be able to estimate the circuit building block's power and noise at the very early stage of the design cycle. For this purpose, this chapter proposed a holistic evaluation link framework that integrates all the information from channel, circuit and device technology to give accurate power and performance estimations for various link architectures.

The core of this framework relies on the compact analytical circuit modeling methodology that was introduced and applied to various building blocks. With this method, the system level parameters are used as the constraint and along with the

extracted device technology database, power and performance of the circuits for each individual blocks can be evaluated. With some extra care on dealing with the interactions and signal/noise flows when putting different building together, a complete framework can thus be constructed. Applying such a framework to a cable-based link platform that was designed for future 60+Gb/s wireline communication, the pros/cons of various link architectures were clearly identified. Furthermore, an energy-efficient combination of building blocks was found to meet the BER requirement even in a relatively “old” process node, proving the framework’s effectiveness in helping the designers make quick architecture decisions.

Finally, although not yet included in this framework, an optimization engine could be integrated into this framework as well to automate the design-space exploration. With the desired characteristic of decoupled system and circuit parameters from this framework, separate optimization loops could be applied to the system and circuit level of evaluation respectively, making the exploration fast and efficient.

## Chapter 5

### Conclusions

#### 5.1 Thesis Summary

While the throughput requirement of wireline I/Os is predicted to increase at a steady state in the future, the total power budget for these links unfortunately stays pretty flat, making it extremely critical to find ways to build high-speed and more energy-efficient wireline links. At higher speeds, equalization circuits need to cancel more ISI while needing a wide bandwidth, making themselves very power hungry. Therefore building low-power equalization circuits is the key to achieving energy-efficient wireline links.

Voltage-mode signaling is widely used in transmitters due its signaling power advantages over analog signaling. However, to deal with different channel losses while still maintaining good signal integrity, multiple functionalities such as finite-impulse-response feedforward equalizer, signal amplitude control and impedance calibration need to be included. While previous publications ([10]–[12]) showed various approaches to incorporate these functions into a voltage-mode driver, achieving simultaneous digital and signaling efficiency still remains challenging. Fortunately, by combining the techniques of shunt-branch based pre-emphasis ([10]), simplified decoder structure for equalization strength control and automatic analog impedance calibration ([11], [15]), this thesis proposed a TX FFE architecture that can drastically reduce the high-speed parasitic loading on the full-swing digital data path and therefore achieves both excellent signaling and digital power efficiency. A 2-tap 10Gb/s prototype fabricated in a low-power Fujitsu 65nm CMOS process shows an overall  $\sim 1\text{pJ/bit}$  energy efficiency that is  $>2.5\times$  better than previous designs, verifying the effectiveness of these techniques.

As the target data rates go beyond 40Gb/s, closing the feedback loop timing in a decision-feedback equalizer becomes even more difficult. While DFE architectures such as loop-unrolling ([28], [30]) could be adopted to relax the 1<sup>st</sup> tap timing, it would make closing later tap's timing more challenging and/or consume more power ([22], [23]). In order to build a low-power DFE, this thesis thus proposed techniques to improve the speed of multi-tap closed-loop architectures instead. With the combined efforts on merged summer/latch design ([32]), reduced latch gain and dynamic latch design ([33]), a 3-tap DFE prototype fabricated in a general purpose TSMC 65nm CMOS process was measured to be able to operate at 66Gb/s, cancelling a total of about 1.65X of the cursor amplitude ISI while consuming only  $\sim 46\text{mW}$  from a 1.2V supply. Comparing to state-of-the-art 20+Gb/s designs, the proposed architecture achieved both the highest speed and lowest power consumption.

From an overall link budget standpoint of view, as the data rate approaches transistor's speed limit, a better tool is required to be able to capture the circuit characteristics at an early stage. With such a tool, both power and noise generated by the

circuit as a function of the device technology parameter and system or signal-integrity constraint can be estimated. As compared to the conventional system-level only driven evaluation tool ([39]–[42]), this will help designers make wiser early decisions to build more efficient links. With this goal in mind, by utilizing the device-technology LUT based circuit modeling techniques ([45], [46], [50]), this thesis proposed a holistic framework to be able to encapsulate all the circuit power and noise details into the system evaluation phase. Applying this framework to a cable-based link platform engineered for 60+Gb/s applications showed clear trade-offs and effectiveness of different equalization architectures.

## 5.2 Future Directions

To continue data rate scaling in the future, various efforts had been shown to use alternative signal carriers other electrical channels to reduce signal losses. For example, plastic waveguides ([55], [56]) have been investigated to help extend the maximum achievable bandwidth of a channel at a relatively low cost. Optical links ([57]–[59]), on the other hand, could achieve even better energy efficiency due to the channel’s superior loss characteristics, but reducing their cost still remains a challenge at this moment. Regardless of which type of channel to be used for wireline communications, the evaluation framework presented in this thesis can always be used to determine the optimal link architecture once the channel model is updated. Circuit techniques introduced in this thesis can also be adopted when building equalization circuits for these links such that their channel capacities could be fully utilized.

While this thesis focuses on FFE and DFE for binary links, other equalization architectures such as the Tomlinson-Harashima precoder ([60], [61]) and model-predictive control equalizer ([62]) have been published and benchmarked to show alternative solutions to equalize channel dispersions. Passive equalization schemes ([63], [64]) have been proposed to take the advantage of reduced bit time at high-data rates such that smaller LC delay lines could be used to save area. Furthermore, time-domain signaling processing techniques such as pulse-width modulation ([65]) have also been proposed to explore alternatives to conventional voltage-domain equalization method. To deal with extreme channels, modulation schemes such as 4-PAM, duo-binary signaling ([27], [61], [66]) are potential candidates to achieve robust communications. Even though these architectures are somewhat different than the topologies discussed in the thesis, the underlying circuit implementation could face similar issues such as gain/latency trade-off and digital power overhead. By adopting and modifying circuit techniques introduced in this thesis, these challenges can be addressed that will lead to new opportunities in achieving energy-efficient high-speed links.



## Bibliography

- [1] "ITRS 2012 Updates (Assembly & Packaging)." [Online]. Available: <http://www.itrs.net/Links/2012ITRS/Home2012.htm>.
- [2] A. Leon, K. Tam, and J. Shin, "A power-efficient high-throughput 32-thread SPARC processor," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 7–16, 2007.
- [3] M. Pant, "Microprocessor Power Impacts," in *GLSVLSI*, 2010, no. May.
- [4] "The Problem of Power Consumption in Servers." [Online]. Available: <http://www.infoq.com/articles/power-consumption-servers>. [Accessed: 24-Sep-2014].
- [5] F. O'Mahony, G. Balamurugan, J. E. Jaussi, J. Kennedy, M. Mansuri, S. Shekhar, and B. Casper, "The future of electrical I/O for microprocessors," *2009 Int. Symp. VLSI Des. Autom. Test*, pp. 31–34, Apr. 2009.
- [6] B. Casper, "Energy efficient multi-Gb/s I/O: Circuit and system design techniques," in *Workshop on Microelectronics and Electron Devices* ( ... , 2011, pp. 1–1.
- [7] V. M. Stojanovic, "Channel-Limited High-Speed Links: Modeling, Analysis and Design," Stanford University, 2004.
- [8] Y. Hidaka, W. Gai, T. Horie, J. H. Jiang, Y. Koyanagi, and H. Osone, "A 4-Channel 1.25–10.3 Gb/s Backplane Transceiver Macro With 35 dB Equalizer and Sign-Based Zero-Forcing Adaptive Control," *IEEE J. Solid-State Circuits*, vol. 44, no. 12, pp. 3547–3559, Dec. 2009.
- [9] P. Monsen, "Adaptive equalization of the slow fading channel," *Commun. IEEE Trans.*, vol. C, no. 8, pp. 1064–1075, 1974.
- [10] W. D. Dettloff, J. C. Eble, L. Luo, P. Kumar, F. Heaton, T. Stone, and B. Daly, "A 32mW 7.4Gb/s protocol-agile source-series-terminated transmitter in 45nm CMOS SOI," in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2010, pp. 370–371.

- [11] H. Hatamkhani and R. Drost, "A 10-mW 3.6-Gbps I/O transmitter," in *2003 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.03CH37408)*, 2003, pp. 97–98.
- [12] R. Sredojevic and V. Stojanovic, "Digital link pre-emphasis with dynamic driver impedance modulation," in *IEEE Custom Integrated Circuits Conference 2010*, 2010, pp. 1–4.
- [13] K. Jung, Y. Lu, and E. Alon, "Power analysis and optimization for high-speed I/O transceivers," in *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2011, pp. 1–4.
- [14] A. Amirkhany, J. Wei, N. K. Mishra, J. Shen, W. T. Beyene, C. Chen, T. J. Chin, D. Dressler, C. Huang, V. P. Gadde, M. Hekmat, K. Kaviani, H. Lan, P. Le, C. Madden, S. Mukherjee, L. Raghavan, K. Saito, D. Secker, A. Sendhil, R. Schmitt, S. Fazeel, G. S. Srinivas, T. Wu, C. Tran, A. Vaidyanath, K. Vyas, L. Yang, M. Jain, K.-Y. K. Chang, and X. Yuan, "A 12.8-Gb/s/link Tri-Modal Single-Ended Memory Interface," *IEEE J. Solid-State Circuits*, vol. 47, no. 4, pp. 911–925, Apr. 2012.
- [15] J. Poulton, R. Palmer, and A. Fuller, "A 14-mW 6.25-Gb/s transceiver in 90-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 42, no. 12, pp. 2745–2757, 2007.
- [16] G. W. den Besten and B. Nauta, "Embedded 5 V-to-3.3 V voltage regulator for supplying digital IC's in 3.3 V CMOS technology," *IEEE J. Solid-State Circuits*, vol. 33, no. 7, pp. 956–962, Jul. 1998.
- [17] E. Alon and M. Horowitz, "Integrated Regulation for Energy-Efficient Digital Circuits," *IEEE J. Solid-State Circuits*, vol. 43, no. 8, pp. 1795–1807, Aug. 2008.
- [18] D. W. Dobberpuhl, "Circuits and technology for Digital's StrongARM and ALPHA microprocessors [CMOS technology]," in *Proceedings Seventeenth Conference on Advanced Research in VLSI*, 1997, pp. 2–11.
- [19] M.-J. E. Lee, W. J. Dally, and P. Chiang, "Low-power area-efficient high-speed I/O circuit techniques," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1591–1599, Nov. 2000.
- [20] S. Ibrahim and B. Razavi, "Low-Power CMOS Equalizer Design for 20-Gb/s Systems," *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1321–1336, Jun. 2011.
- [21] J. E. Proesel and T. O. Dickson, "A 20-Gb/s, 0.66-pJ/bit serial receiver with 2-stage continuous-time linear equalizer and 1-tap decision feedback equalizer in 45nm SOI CMOS," in *2011 Symposium on VLSI Circuits (VLSIC)*, 2011, pp. 206–207.

- [22] T. Toifl, M. Ruegg, R. Inti, C. Menolfi, M. Brandli, M. Kossel, P. Buchmann, P. A. Francese, and T. Morf, "A 3.1mW/Gbps 30Gbps quarter-rate triple-speculation 15-tap SC-DFE RX data path in 32nm CMOS," in *2012 Symposium on VLSI Circuits (VLSIC)*, 2012, pp. 102–103.
- [23] K. Jung, A. Amirkhany, and K. Kaviani, "A 0.94mW/Gb/s 22Gb/s 2-tap partial-response DFE receiver in 40nm LP CMOS," in *2013 IEEE International Solid-State Circuits Conference*, 2013, pp. 42–43.
- [24] C.-L. Hsieh and S.-I. I. Liu, "A 40Gb/s decision feedback equalizer using back-gate feedback technique," in *2009 Symposium on VLSI Circuits (VLSIC)*, 2009, pp. 218–219.
- [25] Y. Lu and E. Alon, "A 66Gb/s 46mW 3-tap decision-feedback equalizer in 65nm CMOS," in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 30–31.
- [26] K. K. Parhi, "High-speed architectures for algorithms with quantizer loops," in *IEEE International Symposium on Circuits and Systems*, pp. 2357–2360.
- [27] V. Stojanovic, A. Ho, B. 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.
- [28] A. Garg, A. C. A. C. Carusone, S. P. S. P. Voinigescu, and S. Member, "A 1-Tap 40-Gb/s Look-Ahead Decision Feedback Equalizer in 0.18-um SiGe BiCMOS Technology," *IEEE J. Solid-State Circuits*, vol. 41, no. 10, pp. 2224–2232, Oct. 2006.
- [29] D. Z. Turker, A. V. Rylyakov, D. J. Friedman, S. M. Gowda, and E. SÁnchez-sinencio, "A 19Gb/s 38mW 1-tap speculative DFE receiver in 90nm CMOS," in *2009 Symposium on VLSI Circuits (VLSIC)*, 2009, pp. 216–217.
- [30] A. Awny, L. Moeller, J. Junio, J. C. Scheytt, and A. Thiede, "Design and Measurement Techniques for an 80 Gb/s 1-Tap Decision Feedback Equalizer," *IEEE J. Solid-State Circuits*, vol. 49, no. 2, pp. 452–470, Feb. 2014.
- [31] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits*, 2/e. Pearson Education, 2003, p. 761.
- [32] H. Wang and J. Lee, "A 21-Gb/s 87-mW Transceiver With FFE/DFE/Analog Equalizer in 65-nm CMOS Technology," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 909–920, Apr. 2010.

- [33] A. Ghilioni, U. Decanis, E. Monaco, A. Mazzanti, and F. Svelto, "A 6.5mW inductorless CMOS frequency divider-by-4 operating up to 70GHz," in *2011 IEEE International Solid-State Circuits Conference*, 2011, pp. 282–284.
- [34] B. Murmann, "Thermal noise in track-and-hold circuits: analysis and simulation techniques," *IEEE Solid-State Circuits Mag.*, vol. 4, no. 2, pp. 46–54, Jun. 2012.
- [35] W. Ellersick, M. Horowitz, and W. Dally, "GAD: A 12-GS/s CMOS 4-bit A/D converter for an equalized multi-level link," in *1999 Symposium on VLSI Circuits. Digest of Papers (IEEE Cat. No.99CH36326)*, pp. 49–52.
- [36] F. Weiss, H. Wohlmuth, D. Kehrer, and A. Scholtz, "A 24-Gb/s  $2^{17}$ -1 Pseudo Random Bit Sequence Generator IC in 0.13  $\mu$ m Bulk CMOS," in *2006 Proceedings of the 32nd European Solid-State Circuits Conference*, 2006, pp. 468–471.
- [37] J. F. Bulzacchelli, M. Meghelli, S. V. Rylov, W. Rhee, A. V. Rylyakov, H. A. Ainspan, B. D. Parker, M. P. Beakes, A. Chung, T. J. Beukema, P. K. Pepeljugoski, L. Shan, Y. H. Kwark, S. Gowda, and D. J. Friedman, "A 10-Gb/s 5-Tap DFE/4-Tap FFE Transceiver in 90-nm CMOS Technology," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2885–2900, Dec. 2006.
- [38] K.-L. J. Wong, A. Rylyakov, and C.-K. K. Yang, "A 5-mW 6-Gb/s Quarter-Rate Sampling Receiver With a 2-Tap DFE Using Soft Decisions," *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 881–888, Apr. 2007.
- [39] V. Stojanovic and M. Horowitz, "Modeling and analysis of high-speed links," *Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE 2003*, pp. 589–594, 2003.
- [40] A. Sanders, M. Resso, and J. D'Ambrosia, "Channel compliance testing utilizing novel statistical eye methodology," *DesignCon*, 2004.
- [41] F. Lambrecht, J. Zerbe, and V. Stojanovic, "Accurate System Voltage and Timing Margin Simulation in High-Speed I/O System Designs," *IEEE Trans. Adv. Packag.*, vol. 31, no. 4, pp. 722–730, Nov. 2008.
- [42] G. Balamurugan, B. Casper, J. E. Jaussi, M. Mansuri, F. O'Mahony, and J. Kennedy, "Modeling and Analysis of High-Speed I/O Links," *IEEE Trans. Adv. Packag.*, vol. 32, no. 2, pp. 237–247, May 2009.
- [43] R. Sredojevi and V. Stojanovi, "Optimization-based framework for simultaneous circuit-and-system design-space exploration: A high-speed link example," in *2008 IEEE/ACM International Conference on Computer-Aided Design*, 2008, pp. 314–321.

- [44] S.-H. Weng, Y. Zhang, J. F. Buckwalter, and C.-K. Cheng, "Energy Efficiency Optimization Through Codesign of the Transmitter and Receiver in High-Speed On-Chip Interconnects," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 22, no. 4, pp. 938–942, Apr. 2014.
- [45] C. Thakkar, L. Kong, K. Jung, A. Frappe, and E. Alon, "A 10 Gb/s 45 mW Adaptive 60 GHz Baseband in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 47, no. 4, pp. 952–968, Apr. 2012.
- [46] Y. Lu and E. Alon, "Design Techniques for a 66 Gb/s 46 mW 3-Tap Decision Feedback Equalizer in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 48, no. 12, pp. 3243–3257, Dec. 2013.
- [47] B. E. Boser, "Analog Circuit Design with Submicron Transistors." [Online]. Available: <http://www.ewh.ieee.org/r6/scv/ssc/May1905.htm>. [Accessed: 10-Jun-2013].
- [48] F. Silveira, D. Flandre, and P. G. A. Jespers, "A gm/ID based methodology for the design of CMOS analog circuits and its application to the synthesis of a silicon-on-insulator micropower OTA," *IEEE J. Solid-State Circuits*, vol. 31, no. 9, pp. 1314–1319, 1996.
- [49] C. Thakkar, L. Kong, K. Jung, A. Frappe, and E. Alon, "A 10 Gb/s 45 mW Adaptive 60 GHz Baseband in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 47, no. 4, pp. 952–968, Apr. 2012.
- [50] C. Thakkar, N. Narevsky, C. D. Hull, and E. Alon, "Design Techniques for a Mixed-Signal I/Q 32-Coefficient Rx-Feedforward Equalizer, 100-Coefficient Decision Feedback Equalizer in an 8 Gb/s 60 GHz 65 nm LP CMOS Receiver," *IEEE J. Solid-State Circuits*, vol. 49, no. 11, pp. 2588–2607, Nov. 2014.
- [51] A. M. Niknejad, *Electromagnetics for High-Speed Analog and Digital Communication Circuits*. Cambridge University Press, 2007.
- [52] H. Braunsch, J. E. J. E. Jaussi, J. A. J. A. Mix, M. B. M. B. Trobough, B. D. B. D. Horine, V. Prokofiev, R. Baskaran, P. C. H. P. C. H. Meier, K. E. K. E. Mallory, M. W. M. W. Leddige, S. Member, D. Lu, and D. Han, "High-Speed Flex-Circuit Chip-to-Chip Interconnects," *IEEE Trans. Adv. Packag.*, vol. 31, no. 1, pp. 82–90, Feb. 2008.
- [53] J. Jaussi, G. Balamurugan, S. Hyvonen, T.-C. Hsueh, T. Musah, G. Keskin, S. Shekhar, J. Kennedy, S. Sen, R. Inti, M. Mansuri, M. Leddige, B. Horine, C. Roberts, R. Mooney, and B. Casper, "A 205mW 32Gb/s 3-Tap FFE/6-tap DFE bidirectional serial link in 22nm CMOS," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 440–441.

- [54] A. A. Hafez, M.-S. Chen, and C.-K. K. Yang, "A 32-to-48Gb/s serializing transmitter using multiphase sampling in 65nm CMOS," in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 38–39.
- [55] S. Fukuda, Y. Hino, S. Ohashi, T. Takeda, H. Yamagishi, S. Shinke, K. Komori, M. Uno, Y. Akiyama, K. Kawasaki, and A. Hajimiri, "A 12.5+12.5 Gb/s Full-Duplex Plastic Waveguide Interconnect," *IEEE J. Solid-State Circuits*, vol. 46, no. 12, pp. 3113–3125, Dec. 2011.
- [56] Y. Tanaka, Y. Hino, Y. Okada, T. Takeda, S. Ohashi, H. Yamagishi, K. Kawasaki, and A. Hajimiri, "A versatile multi-modality serial link," in *2012 IEEE International Solid-State Circuits Conference*, 2012, pp. 332–334.
- [57] F. Liu, D. Patil, and J. Lexau, "10 Gbps, 530 fJ/b optical transceiver circuits in 40 nm CMOS," *VLSI Circuits (VLSIC), 2011 IEEE Symp.*, pp. 290–291, 2011.
- [58] J. Buckwalter, X. Zheng, and G. Li, "A monolithic 25-Gb/s transceiver with photonic ring modulators and Ge detectors in a 130-nm CMOS SOI process," *IEEE J. Solid-State Circuits*, vol. 47, no. 6, pp. 1309–1322, 2012.
- [59] C. Li, R. Bai, A. Shafik, E. Z. Tabasy, G. Tang, C. Ma, C.-H. Chen, Z. Peng, M. Fiorentino, P. Chiang, and S. Palermo, "A ring-resonator-based silicon photonics transceiver with bias-based wavelength stabilization and adaptive-power-sensitivity receiver," in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 124–125.
- [60] Y. Gu and K. Parhi, "High-speed architecture design of Tomlinson–Harashima precoders," *Circuits Syst. I Regul. Pap. IEEE*, vol. 54, no. 9, pp. 1929–1937, 2007.
- [61] M. Kossel, T. Toifl, P. A. Francese, M. Brandli, C. Menolfi, P. Buchmann, L. Kull, T. M. Andersen, and T. Morf, "An 8Gb/s 1.5mW/Gb/s 8-tap 6b NRZ/PAM-4 Tomlinson-Harashima precoding transmitter for future memory-link applications in 22nm CMOS," in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 408–409.
- [62] A. Suleiman, R. Sredojevic, and V. Stojanovic, "Model Predictive Control Equalization for High-Speed I/O Links," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 61, no. 2, pp. 371–381, Feb. 2014.
- [63] A. Momtaz and M. M. Green, "An 80 mW 40 Gb/s 7-Tap T/2-Spaced Feed-Forward Equalizer in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 45, no. 3, pp. 629–639, Mar. 2010.

- [64] M.-S. Chen and C.-K. K. Yang, "A 50–64 Gb/s serializing transmitter with a 4-tap, LC-ladder-filter-based FFE in 65-nm CMOS," in *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*, 2014, pp. 1–4.
- [65] W. Wang and J. Buckwalter, "A 10-Gb/s, 107-mW Double-Edge Pulsewidth Modulation Transceiver," *IEEE Trans. Circuits Syst. I*, vol. 61, no. 4, pp. 1068–1080, 2014.
- [66] J. Lee, M.-S. Chen, and H.-D. Wang, "Design and Comparison of Three 20-Gb/s Backplane Transceivers for Duobinary, PAM4, and NRZ Data," *IEEE J. Solid-State Circuits*, vol. 43, no. 9, pp. 2120–2133, Sep. 2008.
- [67] L. Kong, Y. Lu, and E. Alon, "A multi-GHz area-efficient comparator with dynamic offset cancellation," in *2011 IEEE Custom Integrated Circuits Conference (CICC)*, 2011, pp. 1–4.
- [68] P. J. J. O. Reilly, B. Tech, D. Ph, C. Eng, and S. M. I. E. E. E, "Error propagation in decision feedback receivers," *IEE Proc. F (Communications, Radar Signal Process.)*, vol. 132, no. 7, 1985.
- [69] A. de O. Duarte and J. O'Reilly, "Simplified technique for bounding error statistics for DFB receivers," *IEE Proc. F (Communications, Radar Signal Process.)*, vol. 132, no. 7, pp. 567–575, 1985.
- [70] S. Altekhar and N. Beaulieu, "Upper bounds to the error probability of decision feedback equalization," *Inf. Theory, IEEE Trans.*, vol. 39, no. 1, pp. 145–156, 1993.
- [71] R. Kennedy, B. Anderson, and R. Bitmead, "Tight bounds on the error probabilities of decision feedback equalizers," *IEEE Trans. Commun.*, vol. 35, no. 10, pp. 1022–1028, Oct. 1987.
- [72] J. Tsimbinos and L. B. White, "Error propagation and recovery in decision-feedback equalizers for nonlinear channels," *IEEE Trans. Commun.*, vol. 49, no. 2, pp. 239–242, 2001.
- [73] C. Thakkar, "Design of Multi-Gb/s Multi-Coefficient Mixed-Signal Equalizers," UC Berkeley, 2012.
- [74] K. Wong, E. Chen, and C. Yang, "Edge and data adaptive equalization of serial-link transceivers," *IEEE J. Solid-State Circuits*, vol. 43, no. 9, pp. 2157–2169, 2008.
- [75] S. Song and V. Stojanovic, "A 6.25 Gb/s voltage-time conversion based fractionally spaced linear receive equalizer for mesochronous high-speed links," *Solid-State Circuits, IEEE J.*, vol. 46, no. 5, pp. 1183–1197, 2011.

- [76] J. E. J. E. Jaussi, G. Balamurugan, D. R. D. R. Johnson, B. Casper, A. Martin, J. Kennedy, N. Shanbhag, R. Mooney, and S. S. Member, "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.
- [77] E. Chen, J. Ren, and B. Leibowitz, "Near-optimal equalizer and timing adaptation for I/O links using a BER-based metric," *IEEE J. Solid-State Circuits*, vol. 43, no. 9, pp. 2144–2156, 2008.
- [78] B. Razavi, *Design of Analog CMOS Integrated Circuits*, vol. 5. McGraw-Hill Education, 2002, p. 684.
- [79] G. R. Gangasani, C.-M. Hsu, J. F. Bulzacchelli, S. Rylov, T. Beukema, D. Freitas, W. Kelly, M. Shannon, J. Qi, H. H. Xu, J. Natonio, T. Rasmus, J.-R. Guo, M. Wielgos, J. Garlett, M. A. Sorna, and M. Meghelli, "A 16-Gb/s backplane transceiver with 12-tap current integrating DFE and dynamic adaptation of voltage offset and timing drifts in 45-nm SOI CMOS technology," in *2011 IEEE Custom Integrated Circuits Conference (CICC)*, 2011, pp. 1–4.
- [80] A. A. Abidi, "Phase Noise and Jitter in CMOS Ring Oscillators," *IEEE J. Solid-State Circuits*, vol. 41, no. 8, pp. 1803–1816, Aug. 2006.
- [81] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*. Cambridge University Press, 2004, p. 797.



## Appendix A

### Analysis of PEVM Transmitters

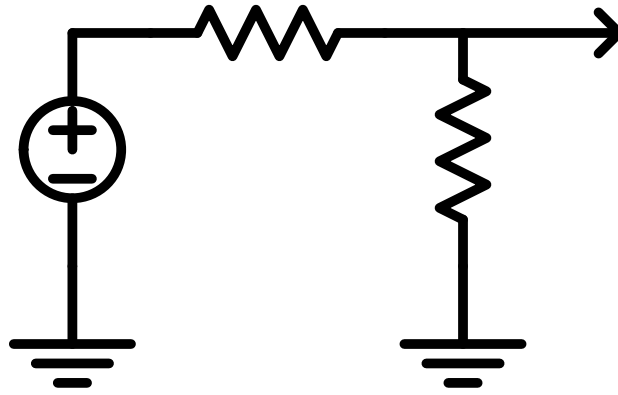


Figure A.1: Thevenin equivalent circuit for driver analysis

All of the voltage-mode drivers can be analyzed using the same Thevenin equivalent circuit shown in Figure A.1. The differential output swing can be expressed as:

$$V_{out} = V_{src} \frac{G_{src}}{G_{src} + G_{ld}} \quad (A.1)$$

Where  $V_{src}$  and  $G_{src}$  are the equivalent source voltage and source conductance from the Thevenin equivalent circuit, and  $G_{ld}$  is the load conductance. With the aid of (A.1), we can now analyze all of the drivers discussed in the chapter.

### A.1 CVPEVM

Assuming the channel as a load and using Figure 2.3(b), the Thevenin equivalent  $V_{src}$  is:

$$V_{src} = \frac{G_{sig} - G_{kill}}{G_{sig} + G_{kill}} V_{drv} = \frac{G_{sig} - G_{kill}}{G_T} V_{drv} \quad (A.2)$$

The last equity is valid because the driver is terminated to the channel so that  $G_{sig} + G_{kill} = G_T$ . Similarly, the Thevenin equivalent  $G_{src}$  is:

$$G_{src} = \frac{G_{sig} + G_{kill}}{2} = \frac{G_T}{2} \quad (A.3)$$

with  $G_{ld} = G_T/2$ ,  $V_{out}$  is therefore:

$$V_{out} = V_{src} \frac{G_{src}}{G_{src} + G_{ld}} = \frac{G_{sig} - G_{kill}}{G_T} V_{drv} \times \frac{\frac{G_T}{2}}{\frac{G_T}{2} + \frac{G_T}{2}} = \frac{G_{sig} - G_{kill}}{2G_T} V_{drv} = \frac{2G_{sig} - G_T}{2G_T} V_{drv} \quad (A.4)$$

Re-ordering Equation (A.4) above and applying the output termination condition results in Equation (2.9). Applying  $G_{sig} + G_{kill} = G_T$  to Equation (2.9) results in Equation (2.10). Since the output common-mode of the driver is fixed at  $V_{drv}/2$ ,  $I_{sig}$  can be found from Figure 2.3(b):

$$\begin{aligned} I_{sig} &= G_{sig}(V_{drv} - V_{out+}) + G_{kill}(V_{drv} - V_{out-}) \\ &= G_{sig}[V_{drv} - (V_{drv}/2 + V_{out}/2)] + G_{kill}[V_{drv} - (V_{drv}/2 - V_{out}/2)] \end{aligned} \quad (A.5)$$

Simplifying (A.5) by plugging in (A.4) then results in Equation (2.6)

## A.2 CIPEVM

Using Figure 2.4(b), we can once again identify the Thevenin equivalent  $V_{src}$  and  $G_{src}$ :

$$V_{src} = \frac{G_{sig} - G_{kill}}{G_{sig} + G_{kill}} V_{drv} \quad (A.6)$$

$$G_{src} = \frac{G_{sig} + G_{kill}}{2} \quad (A.7)$$

Since the load conductance is  $(G_{shnt} + G_T)/2$  from Figure 2.4(b) and  $G_{sig} + G_{kill} + G_{shnt} = G_T$  to maintain output termination, we can derive that:

$$V_{out} = V_{src} \frac{G_{src}}{G_{src} + G_{ld}} = \frac{G_{sig} - G_{kill}}{G_{sig} + G_{kill}} V_{drv} \times \frac{\frac{G_{sig} + G_{kill}}{2}}{\frac{G_{sig} + G_{kill}}{2} + \frac{G_{shnt} + G_T}{2}} = \frac{G_{sig} - G_{kill}}{2G_T} V_{drv} \quad (A.8)$$

Similar to the analysis of CVPEVM, the total signal current for CIPEVM can be expressed as:

$$I_{sig} = G_{sig}[V_{drv} - (V_{drv}/2 + V_{out}/2)] + G_{kill}[V_{drv} - (V_{drv}/2 - V_{out}/2)] \quad (A.9)$$

Plugging in (A.8), we can express  $I_{sig}$  in terms of  $G_{sig}$  as follows:

$$I_{sig} = G_{sig}V_{drv} - G_T V_{sw} - G_T \frac{V_{sw}^2}{V_{drv}} \quad (A.10)$$

Since  $I_{sig} = \frac{G_T V_{drv}}{4}$  is intended to be constant over all  $V_{out}$  values for this architecture,

applying  $\frac{dI_{sig}}{dV_{out}} = 0$  to (A.10) results in:

$$\frac{dG_{sig}}{dV_{out}} = \left(2 \frac{V_{out}}{V_{drv}} + 1\right) \frac{G_T}{V_{drv}} \quad (A.11)$$

Integrating Equation (A.11) and utilizing the known solution  $G_{sig}(V_{out} = V_{drv}/2) = G_T$ , results in the  $G_{sig}(V_{out})$  relationship shown in (2.13).  $G_{kill}(V_{out})$  in Equation (2.14) and  $G_{shnt}(V_{out})$  in Equation (2.15) can then be easily derived using Equation (A.8) and the  $G_{sig} + G_{kill} + G_{shnt} = G_T$  constraint.

## A.3 IMPEVM

From Figure 2.5(b), we can directly derive  $V_{out}$  as a function of  $G_{sig}$ :

$$V_{out} = V_{src} \frac{G_{src}}{G_{src} + G_{ld}} = V_{drv} \times \frac{\frac{G_{sig}}{2}}{\frac{G_{sig}}{2} + \frac{G_T}{2}} = \frac{G_{sig}}{G_{sig} + G_T} V_{drv} \quad (A.12)$$

A simple re-ordering of (A.12) results in (13). Similarly, the signaling current for this driver is:

$$I_{sig} = V_{drv} \left( \frac{\frac{G_T G_{sig}}{2} \frac{2}{G_T + \frac{G_{sig}}{2}}}{\frac{2}{2}} \right) = \frac{1}{2} G_T \left( \frac{G_{sig}}{G_T + G_{sig}} \right) V_{drv} \quad (A.13)$$

Substituting (A.12) into (A.13) results in Equation (2.8).

## A.4 Proposed PEVM

With the aid of Figure 2.10 and the termination constraint  $G_{sig} + G_{shnt} = G_T$ ,  $V_{out}$  is simply:

$$V_{out} = V_{src} \frac{G_{src}}{G_{src} + G_{ld}} = V_{drv} \times \frac{\frac{G_{sig}}{2}}{\frac{G_{sig}}{2} + \frac{G_{shnt} + G_T}{2}} = \frac{G_{sig}}{2G_T} V_{drv} \quad (A.14)$$

which results in (2.22).

The total signaling current can be found in a similar way as the IMPEVM:

$$I_{sig} = V_{drv} \left( \frac{\frac{G_{sig}}{2} \frac{G_T + G_{shnt}}{2}}{\frac{G_{sig}}{2} + \frac{G_T + G_{shnt}}{2}} \right) = V_{drv} \left( \frac{\frac{G_{sig}}{2} \frac{2G_T - G_{sig}}{2}}{\frac{G_{sig}}{2} + \frac{G_T + G_{shnt}}{2}} \right) = \frac{V_{drv} G_{sig}}{2} \left( 1 - \frac{G_{sig}}{2G_T} \right) \quad (A.15)$$

which becomes Equation (2.24) after plugging in (A.14).

## Appendix B

# Comparison of Dynamic Latch and CML Latch Implementations

The dynamic and CML latch implementations we analyze in this section are shown in Figure 3.8 and Figure 3.10, respectively. The definitions listed below will be utilized throughout the rest of the analyses in this Appendix.

### Shared Terms for Both Topologies:

- $C_L$  - External output load capacitance
- $A_{tot}$  - Total required gain
- $V_{DD}$  - Supply voltage
- $V_{cur}$  - Input cursor amplitude
- $V_d$  - Required latch “digital” output amplitude
- $N_\tau$  - Required setting time within 1UI
- $T_{t1}$  - Coefficient of the 1<sup>st</sup> post-cursor ISI tap normalized by the cursor amplitude

$\gamma$  - The ratio of drain to gate capacitance for the transistors; for simplicity, we have assumed this to be a constant regardless of device type and bias

### Dynamic Latch Topology:

- $A_{dyn}, A_g$  - Gain of the dynamic latch and the gain stages ( $A_{tot} = A_{dyn} \cdot A_g$ )
- $A_{0,p,lin}$  - Intrinsic gain of the triode PMOS load, which is usually less than one
- $g_{m,dyn}$  - Transconductance of the latch’s input pair<sup>28</sup>
- $V_{dyn}^*, V_{tp}^*, V_g^*$  - the  $V^*$ ’s of the dynamic latch input pair, tap input pair, and gain stage input pair
- $\omega_{T,dyn}, \omega_{T,p,lin}, \omega_{T,tp}, \omega_{T,g}$  - the  $\omega_T$ ’s of the dynamic latch input pair, triode PMOS load, tap input pair, and gain stage input pair

### CML Latch Topology:

- $A_{pre}$  - Pre-amplification gain of the CML latch (i.e., gain in the pre-amplification phase)
- $g_{m,CML}$  - Transconductance of the CML latch input pair
- $V_{CML}^*$  -  $V^*$  of the CML latch input pair
- $\omega_{T,CML}$  - the  $\omega_T$  of the CML latch input pair

With these definitions and using the same approach as [45], we first find the intrinsic delay of the dynamic latch due to its self-loading:<sup>29</sup>

---

<sup>28</sup>  $g_{m,dyn}$  should be interpreted as the average  $g_m$  of the input pair given the finite clock edge rate. The analysis in [2] can be used to calculate the required peak  $g_m$ .

$$\tau_{o,dyn,slf} = \gamma \frac{A_{dyn}}{\omega_{T,dyn}} + \gamma \frac{A_{0,p,lin}}{\omega_{T,p,lin}} + (\gamma + 1) T_{t1} \frac{V_d}{V_{tp}^*} \frac{1}{\omega_{T,tp}} \quad (B.1)$$

We can then express the time-constant at the output of the dynamic latch as:

$$\tau_{o,dyn} = \tau_{o,dyn,slf} + A_{dyn} \frac{C_L}{g_{m,dyn}}, \quad (B.2)$$

which, in order to meet the timing constraint, needs to satisfy:

$$T_{dq} = N_\tau \tau_{o,dyn} = 1UI = 1/f_{bit} \quad (B.3)$$

With (B.2) and (B.3), the bias current for the dynamic latch during its transparent phase and the associated tap correction current can be found:

$$I_{dyn} = g_{m,dyn} V_{dyn}^* = \frac{A_{dyn} N_\tau f_{bit} C_L V_{dyn}^*}{1 - N_\tau f_{bit} \tau_{o,dyn,slf}} \quad (B.4)$$

$$I_{t1,dyn} = T_{t1} V_{cur} g_{m,dyn} = T_{t1} \frac{V_d}{A_{dyn}} g_{m,dyn} = \frac{T_{t1}}{A_{dyn}} \frac{V_d}{V_{dyn}^*} I_{dyn} \quad (B.5)$$

Knowing the bias current of the dynamic latch, we can find the capacitance driven by the gain stages in front of the latch (i.e., the input capacitance of the latch):

$$C_{gg,dyn} = \frac{g_{m,dyn}}{\omega_{T,dyn}} = \frac{I_{dyn}}{V_{dyn}^*} \frac{1}{\omega_{T,dyn}} \quad (B.6)$$

The resulting bias current of this gain stage is thus:<sup>30</sup>

$$I_g = \frac{A_g N_\tau f_{bit} C_{gg,dyn} V_g^*}{1 - N_\tau f_{bit} \tau_{o,dyn,slf}} = \frac{A_{tot}}{A_{dyn}} \frac{V_g^*}{V_{dyn}^*} \frac{N_\tau f_{bit}}{1 - N_\tau f_{bit} \tau_{o,g,slf}} \frac{1}{\omega_{T,dyn}} I_{dyn} \quad (B.7)$$

where  $\tau_{o,g,slf}$  is the intrinsic delay of the gain stage, which for simplicity we assume is a resistively loaded topology and further ignore any parasitic capacitances from the load resistors.  $\tau_{o,g,slf}$  can then be expressed as:

$$\tau_{o,g,slf} = \gamma \frac{A_g}{\omega_{T,g}} = \gamma \frac{A_{tot}}{A_{dyn}} \frac{1}{\omega_{T,g}} \quad (B.8)$$

With the expressions for the bias current of each stage, it is straightforward to find the average power per cycle for the amplifier + dynamic latch architecture:

$$P_{tot,dyn} = P_{avg,dyn} + P_{avg,t1} + P_{avg,g} = V_{DD} \left[ \frac{1}{2} (I_{dyn} + I_{t1,dyn}) + I_g \right] \quad (B.9)$$

Note that the  $\frac{1}{2}$  factor is applied to the bias current of the dynamic latch input pair and feedback tap since both are active for only half of the clock cycle. After substituting (B.4), (B.5), and (B.7) into (B.9), the total power becomes:

$$P_{tot,dyn} = \left[ \frac{1}{2} \left( 1 + \frac{T_{t1}}{A_{dyn}} \frac{V_d}{V_{dyn}^*} \right) + \frac{A_{tot}}{A_{dyn}} \frac{V_g^*}{V_{dyn}^*} \frac{N_\tau f_{bit}}{1 - N_\tau f_{bit} \tau_{o,g,slf}} \frac{1}{\omega_{T,dyn}} \right] \frac{A_{dyn} N_\tau f_{bit} C_L V_{dyn}^* V_{DD}}{1 - N_\tau f_{bit} \tau_{o,dyn,slf}} \quad (B.10)$$

Note that since  $\tau_{o,dyn,slf}$  is usually larger than  $\tau_{o,g,slf}$ , the highest data rate we can operate is determined by  $f_{bit} < 1/(N_\tau \tau_{o,dyn,slf})$  (i.e.  $P_{tot,dyn} > 0$ ).

We can now move on to compute the power consumed by the CML latch. The time constant of the latch during its transparent phase can be expressed as:

$$\tau_{o,pre} = \tau_{o,CML,slf} + A_{pre} \frac{C_L}{g_{m,CML}}, \quad (B.11)$$

<sup>29</sup> The finite settling time effects can be included by scaling both  $A_{dyn}$  and  $T_{t1}$  by  $\sim 1/(1 - e^{-N_\tau})$ . The finite  $r_0$  effect simply scales  $A_{0,p,lin}$  by a factor of  $\left(1 - \frac{A_{dyn}}{A_{0,N}}\right)$ , where  $A_{0,N}$  is the intrinsic gain of an NMOS in saturation.

<sup>30</sup> In this appendix we will consider only a single gain stage. A multi-stage amplifier (to provide higher gain) can also be integrated in this analysis by adopting the method shown in [81]. It is used to generate the gain=50 case Figure 3.10.

where  $\tau_{o,CML,slf}$  is the intrinsic delay, which is equal to:

$$\tau_{o,CML,slf} = (1 + 2\gamma) \frac{A_{pre}}{\omega_{T,CML}} + (1 + \gamma) T_{t1} \frac{A_{pre} V_d}{A_{tot} V_{tp}^*} \frac{1}{\omega_{T,tp}}, \quad (B.12)$$

Note that due to the self-loading from the regenerative pair (i.e., the additional gate and drain parasitic capacitance of the regenerative pair, there is an additional  $(1 + \gamma) \frac{A_{pre}}{\omega_{T,CML}}$  added to  $\gamma \frac{A_{pre}}{\omega_{T,CML}}$  (i.e. the drain parasitics of the input pair), which substantially increases  $\tau_{o,CML,slf}$ . For simplicity, we also ignored here any parasitic capacitots from the load resistors, which when taken into account would further increase  $\tau_{o,CML,slf}$ .

The total  $T_{dq}$  for a CML latch is composed of two terms – one from settling during the transparent (pre-amplification) phase, and the other from regeneration during the opaque phase. We can therefore derive the following constraint:

$$T_{dq} = N_\tau \tau_{o,pre} + \ln \left( \frac{A_{tot}}{A_{pre}} \right) |\tau_{o,reg}| = \frac{1}{f_{bit}} \quad (B.13)$$

Since the regeneration time constant  $|\tau_{o,reg}| = \frac{\tau_{o,pre}}{A_{pre}-1}$  (as shown in [67]), we can now find an expression for the required CML latch bias current and tap correction current:

$$I_{CML} = \frac{A_{pre} \left[ N_\tau + \ln \left( \frac{A_{tot}}{A_{pre}} \right) \frac{1}{A_{pre}-1} \right] f_{bit} C_L V_{CML}^*}{1 - \left[ N_\tau + \ln \left( \frac{A_{tot}}{A_{pre}} \right) \frac{1}{A_{pre}-1} \right] f_{bit} \tau_{o,CML,slf}} \quad (B.14)$$

$$I_{t1,CML} = T_{t1} V_{cur} g_{m,CML} = T_{t1} \frac{V_d}{A_{tot}} g_{m,CML} = \frac{T_{t1}}{A_{tot}} \frac{V_d}{V_{CML}^*} I_{CML} \quad (B.15)$$

The total average power for the CML latch is therefore:

$$P_{tot,CML} = V_{DD} \left( \frac{1}{2} I_{t1,CML} + I_{CML} \right) \quad (B.16a)$$

$$= \left[ \frac{1}{2} \frac{T_{t1}}{A_{tot}} \frac{V_d}{V_{CML}^*} + 1 \right] \frac{A_{pre} \left[ N_\tau + \ln \left( \frac{A_{tot}}{A_{pre}} \right) \frac{1}{A_{pre}-1} \right] f_{bit} C_L V_{CML}^* V_{DD}}{1 - \left[ N_\tau + \ln \left( \frac{A_{tot}}{A_{pre}} \right) \frac{1}{A_{pre}-1} \right] f_{bit} \tau_{o,CML,slf}} \quad (B.16b)$$

Comparing (B.16b) with (B.10) emphasizes an observation made earlier in the chapter. Specifically, in high-speed but mild gain applications (i.e. our targets in this particular DFE design), the delay for a CML latch is mostly dominated by its transparent phase setting time (since the total gain is low,  $\ln \left( \frac{A_{tot}}{A_{pre}} \right) \frac{1}{A_{pre}-1} \ll N_\tau$ ). Thus, since  $\tau_{o,dyn,slf}$  is  $\sim 2X$  smaller than  $\tau_{o,CML,slf}$ , we are able to achieve higher-speed (or better energy-efficiency for the same achievable speed) using the dynamic latch-based design.

## Appendix C

### DFE Error Propagation Analysis

This appendix summarizes some previous published papers on the analysis of DFE error propagation ([9], [68]–[70]). It is found that for low BER or high SNR applications, including error propagation will only have a minor increase on the steady-state BER as compared to the case without error propagations.

Since the complexity of the analysis grows with the number of DFE taps (i.e. # of channel post-cursor ISI), to help convey the core methodology that is behind the analysis, we will start with a 1-tap DFE example. After that, some model reduction techniques to deal with multiple taps will be briefly summarized in order to introduce interested readers to the provided references for further study.

#### C.1 Error Propagation Analysis for 1-tap DFE

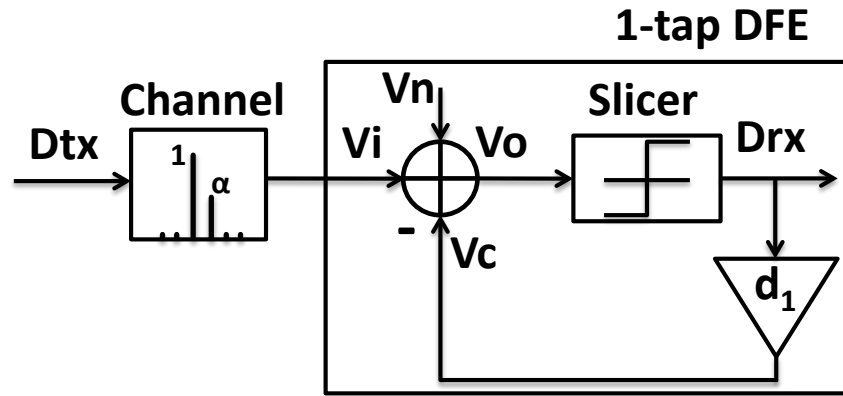


Figure C.1: 1-tap DFE model

A simplified 1-tap DFE model is shown in Figure C.1, where  $D_{tx}$  and  $D_{rx}$  are the transmitted (desired) and received (“estimated”) digital data respectively.  $V_i$  represents the analog signal (i.e. transmitted signal amplitude  $\pm$  ISI amplitude) seen at the DFE input. It is summed with the noise voltage  $V_n$  and feedback correction voltage  $V_c$  ( $=\pm d_1$  depending on the polarity of  $D_{rx}$ ) to generate the input voltage to the slicer  $V_o$ . To express their relationship in a more formal way with the assumption that the DFE has already been properly adapted to cancel the 1-tap channel induced ISI (i.e.  $d_1=\alpha$ ) :

$$\begin{cases} V_o[k] = V_i[k] - V_c[k] + V_n[k] \\ V_i[k] = D_{tx}[k] + \alpha D_{tx}[k-1] \\ D_{rx}[k] = \text{sgn}(V_o[k]) \\ V_c[k] = \alpha D_{rx}[k-1] \end{cases} \quad (\text{C.1})$$

In other words,

$$V_o[k] = D_{tx}[k] + \alpha\{D_{tx}[k-1] - D_{rx}[k-1]\} + V_n[k] \quad (C.2)$$

We can further define the error between the transmitted and received bit as  $e[k] = D_{tx}[k] - D_{rx}[k]$ . This error has 3 possible values: 1) 0 when receiver makes the right decision; 2) +2 when transmitted bit is -1 but receiver makes a +1 decision and 3) -2 when transmitted bit is +1 but receiver makes a -1 decision. With this definition, (C.2) can be modified:

$$V_o[k] = D_{tx}[k] + \alpha e[k-1] + V_n[k] \quad (C.3)$$

Now  $D_{rx}[k]$  can be derived which gives an updated  $e[k]$ . In other words,  $e[k]$  will be completely determined by the previous error  $e[k-1]$  and the current transmitter bit  $D_{tx}[k]$  along with current noise sample  $V_n[k]$ . Assuming the transmitted digital bits and the noise are statistically independent, we can then construct a 1<sup>st</sup> order discrete Markov chain model for the error states as shown in Figure C.2 ([9]).

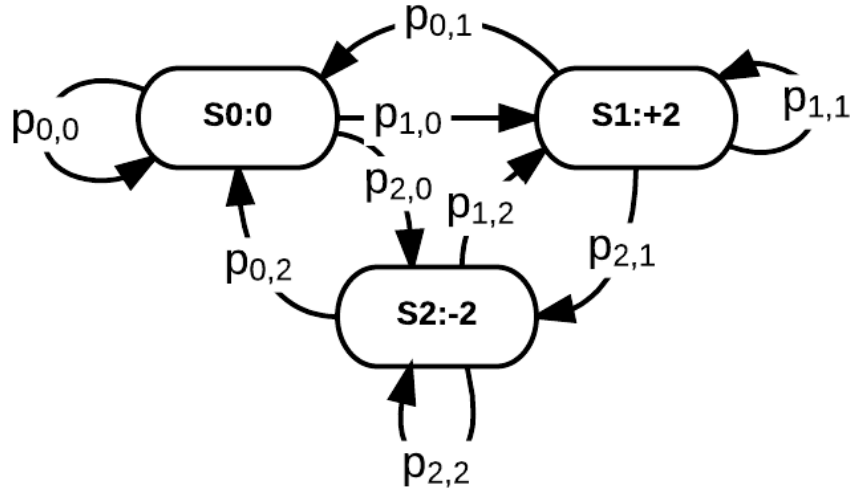


Figure C.2: 1st-order discrete Markov chain model for the error state transitions

Since there are 3 possible error outcomes, 3 distinct states are assigned to them ( $S0 \rightarrow e=0$ ,  $S1 \rightarrow e=+2$  and  $S2 \rightarrow e=-2$ ) and we will have 3 transition probabilities for each states which will result in a transition matrix with a total dimension of  $3 \times 3=9$ :

$$\bar{P} = \begin{bmatrix} p_{0,0} & p_{1,0} & p_{2,0} \\ p_{0,1} & p_{1,1} & p_{2,1} \\ p_{0,2} & p_{1,2} & p_{2,2} \end{bmatrix} \quad (C.4)$$

To find out each transition probability, we need to figure out how each transition happens. As an example, for a transition from  $S0$  to  $S1$ , it really means the receiver slicer make a -1 decision when transmitter sends a +1 and the correction is perfect (i.e. no decision error for the previous bit). We can thus express the probability of this transition as:

$$\begin{aligned} p_{1,0} &= Pr[e[k] = +2 | e[k-1] = 0, D_{tx}[k] = +1] \\ &= Pr[D_{rx}[k] = -1 | e[k-1] = 0, D_{tx}[k] = +1] \end{aligned} \quad (C.5)$$



Assuming the transmitter sends equal number of +1 and -1, we can then simply (C.5) to:

$$\begin{aligned}
 p_{1,0} &= \frac{1}{2} Pr[D_{rx}[k] = -1 | e[k-1] = 0] \\
 &= \frac{1}{2} Pr[(D_{tx}[k] + V_n[k]) < 0] \\
 &= \frac{1}{2} \times \frac{1}{2} \left[ 1 + \operatorname{erf} \left( \frac{-V_{cur}}{\sqrt{2}\sigma_n} \right) \right] = \frac{1}{4} \left[ 1 + \operatorname{erf} \left( \frac{-SNR}{\sqrt{2}} \right) \right]
 \end{aligned} \tag{C.6}$$

where the last equivalence is reached by assuming the transmitted cursor voltage is  $V_{cur}$  and the noise has a Gaussian distribution with a rms noise voltage of  $\sigma_n$ . The signal-to-noise ratio SNR is defined in voltage ratio as  $V_{cur}/\sigma_n$ . Similarly, we can easily compute the probability of transition from S0 to S2:

$$\begin{aligned}
 p_{2,0} &= Pr[D_{rx}[k] = +1 | e[k-1] = 0, D_{tx}[k] = -1] \\
 &= \frac{1}{2} Pr[D_{rx}[k] = +1 | e[k-1] = 0] \\
 &= \frac{1}{2} Pr[(-D_{tx}[k] + V_n[k]) > 0] \\
 &= \frac{1}{4} \left[ 1 - \operatorname{erf} \left( \frac{SNR}{\sqrt{2}} \right) \right] = \frac{1}{4} \left[ 1 + \operatorname{erf} \left( \frac{-SNR}{\sqrt{2}} \right) \right] = p_{1,0}
 \end{aligned} \tag{C.7}$$

Notice the last equivalence is valid because  $\operatorname{erf}$  function is an odd function. The equivalence of  $p_{2,0}$  and  $p_{1,0}$  also makes intuitive sense as the conditions of reaching S1 and S2 from S0 are exactly symmetric in a binary system with only 1-tap of DFE. with  $p_{1,0}$  and  $p_{2,0}$  in hand,  $p_{0,0}$  can be easily derived:

$$p_{0,0} = 1 - p_{1,0} - p_{2,0} \tag{C.8}$$

To consider the transition probability from other states other than S0, the approaches is exactly the same except now the signal amplitude will be modified due to incomplete ISI cancellation (i.e. SNR will be different). For S1:

$$\begin{aligned}
 p_{1,1} &= Pr[e[k] = +2 | e[k-1] = +2, D_{tx}[k] = +1] \\
 &= \frac{1}{2} Pr[D_{rx}[k] = -1 | e[k-1] = +2] \\
 &= \frac{1}{2} Pr[(D_{tx}[k] + 2\alpha + V_n[k]) < 0] \\
 &= \frac{1}{4} \left[ 1 + \operatorname{erf} \left( \frac{-SNR(1+2\alpha)}{\sqrt{2}} \right) \right]
 \end{aligned} \tag{C.9}$$

$$\begin{aligned}
 p_{2,1} &= \frac{1}{2} Pr[D_{rx}[k] = +1 | e[k-1] = +2] \\
 &= \frac{1}{2} Pr[(-D_{tx}[k] - 2\alpha + V_n[k]) > 0] = p_{1,1}
 \end{aligned} \tag{C.10}$$

$$p_{0,1} = 1 - p_{1,1} - p_{2,1} \tag{C.11}$$

Again, the  $p_{2,1} = p_{1,1}$  stems from the fact that it is a binary link with 1-tap DFE correction so the transitions to a +2 or -2 decision error from the same state are symmetric. Finally,  $p_{1,2}$ ,  $p_{2,2}$  and  $p_{0,2}$  are easily found by simply changing the polarities in the ISI correction term:

$$p_{2,2} = p_{1,2} = \frac{1}{4} \left[ 1 + \operatorname{erf} \left( \frac{-\operatorname{SNR}(1-2\alpha)}{\sqrt{2}} \right) \right] \quad (\text{C.12})$$

$$p_{0,2} = 1 - p_{1,2} - p_{2,2} \quad (\text{C.13})$$

Intuitively, comparing to state 1, the transition starting at state 2 should result in smaller probability going back to the healthy state 0 (i.e.  $p_{0,2} < p_{0,1}$ ) due the fact that the incorrect correction from the tap closes the eye in state 2 rather than boosting it in state 1 and thus has much poorer SNR. Such a behavior can also be observed by comparing (C.12) and (C.9).

Putting (C.5)-(C.13) into (C.4) we can therefore get the transition matrix. With little efforts, the steady-state probability for each state can be found:

$$\bar{\Pi} = \lim_{n \rightarrow \infty} (\bar{P})^n \quad (\text{C.14})$$

Since the  $\pi_{0,0}$  entry in  $\bar{\Pi}$  is ultimately the steady-state zero error probability, BER can then easily be derived:

$$\text{BER} = 1 - \pi_{0,0} \quad (\text{C.15})$$

With some math, we can get the steady-state BER vs. input SNR in Figure C.3. It can be immediately found that the penalty on error propagation in a DFE is really negligible, especially at low BER region. As an example, for  $1\text{e-}12$  BER target, the required SNR for the ISI free case (i.e. no error propagation) is  $\sim 7.03$  vs.  $\sim 7.07$  and  $7.13$  for  $\text{ISI}=0.5$  and  $1$  respectively. In other words, for a target BER of  $1\text{e-}12$ , the required SNR improvement is only  $\sim 1.4\%$  for a 1-tap channel that causes a complete eye closure when taking the DFE error propagation into account.

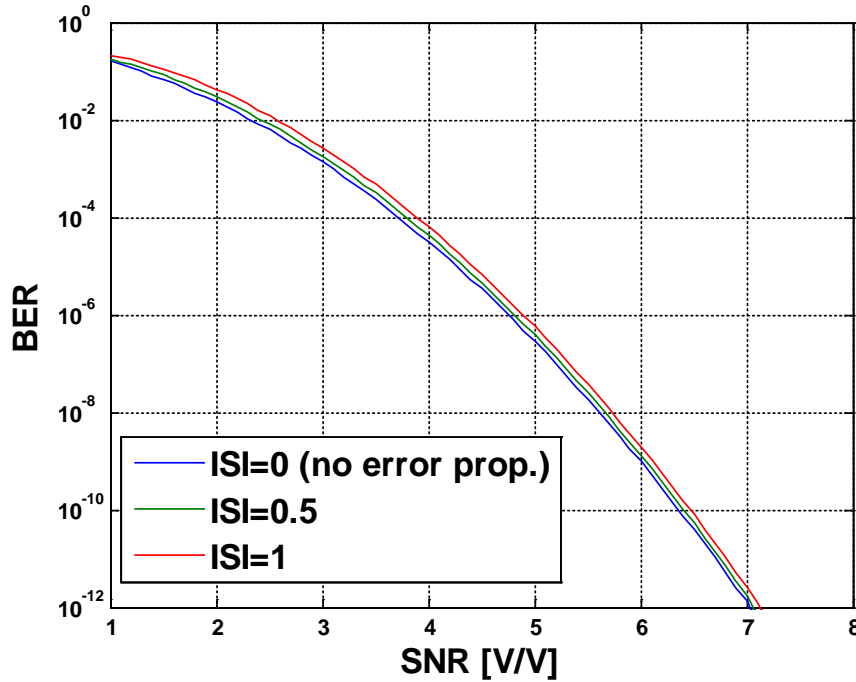


Figure C.3: steady-state BER vs input SNR wi/wo error propagation

Another interesting characteristic we can consider is the time a DFE requires to recover to an “error-free” state once a decision error is made. To study such a behavior, Figure C.4 plots the BER vs. # of decisions (i.e. # of transitions) for the 1-tap DFE example. Here the “error-free” criterion is decided by entering the steady-state target BER. We ran the simulation with an input SNR of 7 assuming the DFE suddenly entering one of the 3 different states mentioned before. Not surprisingly, the worst state is always state 2 where the incorrect ISI cancellation tends to close the eye and thus results in very poor initial BER. It can also be noted that the larger the channel ISI the longer it requires to recover from the bad state.

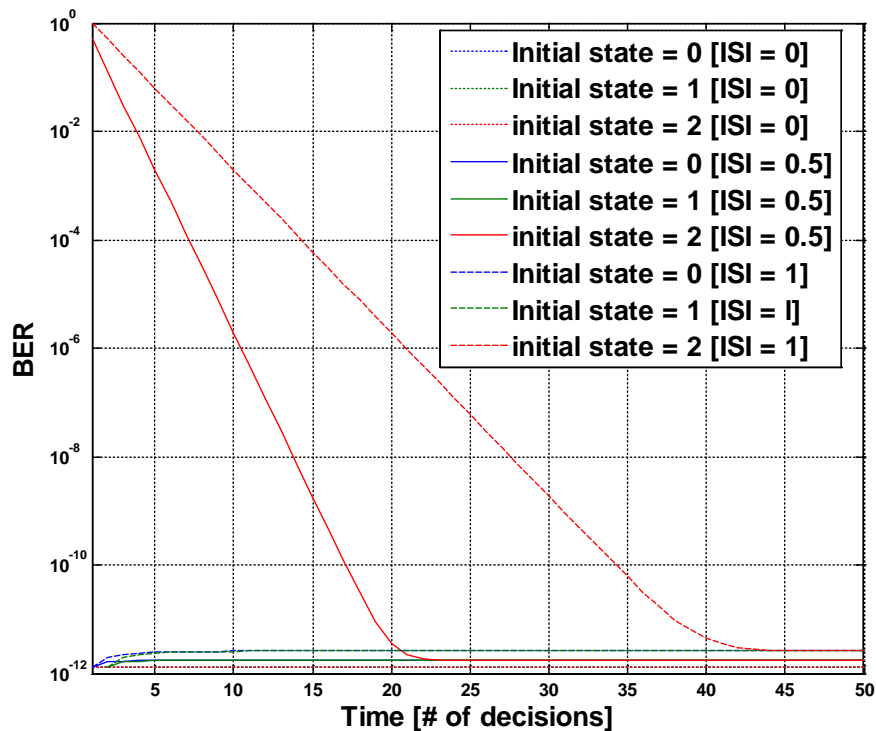


Figure C.4: BER vs. time with different initial states

## C.2 A Brief Summary for Multi-Tap DFE

As shown before, the required number of states and transition matrix size to completely capture the error-propagation behavior is  $3^N$  and  $3 \times 3^N$ , respectively, where  $N$  represents # of DFE taps. As an example, Figure C.5 shows a 2-tap DFE that requires  $3^2 = 9$  states with  $3 \times 3^2 = 27$  transitions and a 3-tap example can also be found in [68]. While it is possible to write programs to automatically figure out the state transition diagram and

probabilities, the # of states and transitions still grows exponentially with # of taps resulting in a large computational resource requirement.

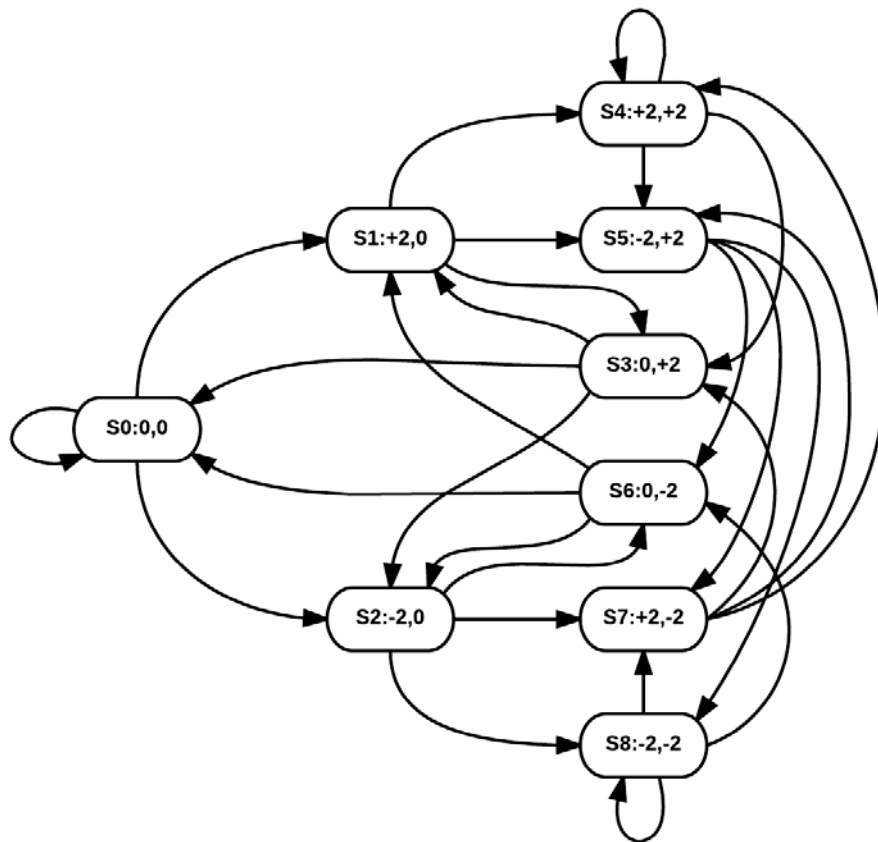


Figure C.5: 1st-order discrete Markov chain model for a 2-tap DFE

Since often times a higher/lower bound number of the error propagation probability would be sufficient to estimate its impact to the system, people have come up with different model reduction methods to simplify the STD ([68], [69]). Probably the most straight-forward model reduction is to just differentiate between error and non-error states. Called 1<sup>st</sup>-order reduction ([68]), this technique unifies  $e=\pm 2$  states into one state such that the requires states and # of transitions can be reduced to  $2^N$  and  $2 \times 2^N$  respectively. Taking again the 1-tap case as an example, the new states X0 and X1 corresponding to without and with decision errors are shown in Figure C.6, where only 2 states and 4 transition probabilities are needed ([9]).

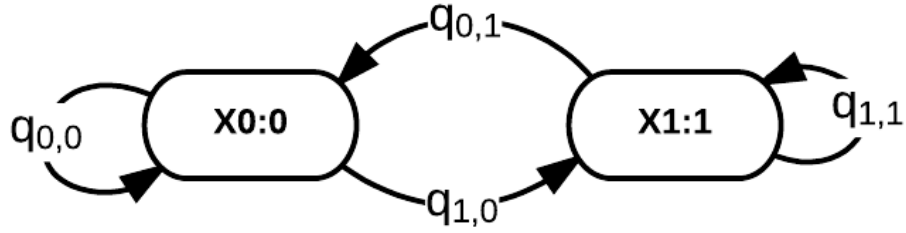


Figure C.6: 1st-order reduction of STD for the 1-tap DFE example

The new transition probabilities are essentially the weighted sum of the different branches in the old STD:

$$q_{1,0} = p_{1,0} + p_{2,0} = 2p_{1,0} \quad (\text{C.16})$$

$$q_{0,0} = 1 - q_{1,0} \quad (\text{C.17})$$

$$q_{1,1} = \frac{1}{2}(p_{1,1} + p_{1,2}) + \frac{1}{2}(p_{2,1} + p_{2,2}) = p_{1,1} + p_{2,2} \quad (\text{C.18})$$

$$q_{0,1} = 1 - q_{1,1} \quad (\text{C.19})$$

with the new transition matrix expressed as:

$$\overline{Q} = \begin{bmatrix} q_{0,0} & q_{1,0} \\ q_{0,1} & q_{0,0} \end{bmatrix} \quad (\text{C.20})$$

While this reduction substantially reduces the size of the transition matrix to potentially save computational resources, the exact probabilities still need to be computed to fully capture the error propagation behaviors.

Since knowing an upper bound of the error propagation problem would be enough most of the time, the probabilities for only the worst-/best-case paths can be computed to reduce the evaluation time. As an example in [68] - which aligns the observations we found in the previous section - the worst-case probabilities for each state can be identified as the ISI cancelation values to give the minimum eye-opening. With such a simplification, the transition probabilities for each state no longer needs to be calculated as the weighted sum of probabilities with different error vectors as shown before. It is sufficient to use only the probability from the error vector that results in the minimum eye-opening.

Further STD reduction is also achievable given other constraints. As another example, in [69], a 2<sup>nd</sup>-order reduction is introduced to further reduce the STD to only (N+1) states. The criterion used for state definition in this method is to consider the states as the required number of consecutive correct decisions to return to the error-free state. Again, as for finding error bounds, the transition probability for each state should be calculated with the worst-/best-case identification to save time and efforts.

While there have been numerous other publications on finding the upper bound for DFE error propagations with similar model reduction techniques ([70]–[72]), the conclusions always end up stating that with enough SNR budget, the steady-state BER

won't be affected by taking DFE's error propagation effect into account. Of course the tighter the bound we can find the better estimation on the SNR requirement we can get.