Design and Automatic Generation of 60Gb/s Wireline Transceivers

Jaeduk Han

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2019-143
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-143.html

December 1, 2019

Design and Automatic Generation of 60Gb/s Wireline Transceivers

by

Jaeduk Han

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

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Elad Alon, Chair
Professor Vladimir Stojanovic
Professor Liwei Lin

Fall 2017

The dissertation of Jaeduk Han, titled Design and Automatic Generation of 60Gb/s Wireline Transceivers, is approved:

Chair _____ Date _____

_____ Date _____

_____ Date _____

University of California, Berkeley

Design and Automatic Generation of 60Gb/s Wireline Transceivers

Abstract

Design and Automatic Generation of 60Gb/s Wireline Transceivers

by

Jaeduk Han

Doctor of Philosophy in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Elad Alon, Chair

The continual expansion of Internet connectivity has raised data traffic substantially, increasing demands on high-bandwidth wireline communication systems. In contrast to the rapid increase in bandwidth that will be necessary, the allowable power consumption of high-speed transceivers remains relatively constant. Specifically, assuming the same total power budget as current designs, transceivers operating in the range of 50-60Gb/s must achieve 3-5pJ/bit efficiency to remain within the current total power window.

To address these trends and challenges, this thesis first reports a 60Gb/s receiver frontend in a 65nm CMOS process. Current integration combined with a cascode gate-voltage bias technique enables energy-efficient implementation of CTLE, FFE, and DFE circuits while operating near the speed limits of the technology. The addition of interleaved, offset-canceled deserializing samplers addresses the high gain requirement of adaptive error-samplers. The prototype 65nm CMOS receiver operates at 60Gb/s, consuming 173mW from 1.2V and 1.0V supplies.

To enhance the productivity and quality of the receive equalizer design by capturing the parasitic loading and wiring capacitance promptly and precisely, an automated flow that generates the equalizer schematic and layout from target specifications and technology parameters is developed. Utilizing the Berkeley Analog Generator, the equalizer generation flow is scripted, executed, and iterated automatically to produce an optimal design that meets for a given configuration.

In addition to presenting the equalizer design, the thesis also introduces a new baud-rate CDR scheme that leverages the current integration frontend and phase dithering. Correlation of the adaptive error sampler output with the phase dithering sequence indicates the direction of phase offset. The resulting baud-rate CDR saves power and complexity compared to an oversampling CDR by not requiring additional clock phases/deserializers. The implemented 65nm CMOS transceiver operates at 60Gb/s with an eye-opening of 30% UI and consumes 288mW while equalizing 21dB of loss at 30GHz over a 0.7m Twinax cable.

To My Family

Contents

List of Figures

List of Tables

Acknowledgments

I feel extremely fortunate to have a wonderful chance to study at UC Berkeley as a graduate student. I have received so much help from many individuals, and I am happily taking this page to extend my special thanks to them.

First of all, I would like to thank my advisor, Prof. Elad Alon, for his valuable advice on my study. The achievement I've made wouldn't be possible without his advice and support. Professor Alon never said no whenever I wanted to discuss with him and has been always willing to help his students to learn and make progress. He also has been a great teacher and mentor to follow. I was deeply inspired by his knowledge as well as his passion to research and teaching. He has been very supportive to his students in all aspects as well. Whenever I have personal concerns and hardships, he kindly encouraged me with warm remarks and guidance. I am a very lucky person to spend my time in his research group.

I also would like to thank Prof. Vladmir Stojanovic, Prof. Lewei Lin, Prof. Ali Niknejad, and Prof. Paul Wright for being on my thesis and qualification exam committee. They greatly improved the quality of my research and thesis with valuable feedback. I also want to thank Prof. Borivoje Nikolic and Prof. Seth Sanders for their kind advice on design automation and power electronics. I am also grateful to Prof. Deog-Kyoon Jeong of Seoul National University for his guidance and support on my research and future career.

My research has been continuously supported by various organizations. Among them, I would like to thank Bryan Casper and James Jaussi of Intel Labs for supporting resources and valuable discussions, Ken Chang, Yohan Frans, and Jay Lim of Xilinx, Frankie Liu, Ziad Shehadeh and Vincent Liu of Oracle Labs for their support on measurements. I also want to thank Prof. Patrick Chiang of OSU, Prof. Chih-Kong Ken Yang of UCLA, and Prof. Pavan Hanumolu of UIUC for useful discussions on the 60Gb/s serial link project.

I thank Berkeley Wireless Research Center staff, Ajith Amerasekera, Brian Richards, Fred Burghardt, James Dunn, Candy Corpus, and Yessica Bravo, for their tremendous help and support on my research. Without their assistance, it would not be possible to finish my study. In particular, I am grateful to Brian Richard for stimulating discussions and technical support on tapeouts, and Fred Burghardt for maintaining excellent lab environments.

All BWRC students are very talented and always willing to help out their colleagues. I thank them for their feedback, cooperation and friendship. Among them, I would like to especially thank Yue Lu, for his valuable feedback and help on my research. He has been the best person to work with, great mentor, and good friend. I also thank students and postdocs I've been working with: Nicholas Sutardja, Kwangmo Jung, Stevo Bailey, Angie Wang, Eric Chang, Zhongkai Wang, Woorham Bae, Paul Rigge, Amy Whitcombe, and Yongjun Li. Thanks also go to all BWRC and Berkeley students, postdocs, and visiting scholars I've spent my time with: Jiashu Chen, Lingkai Kong, Chintan Thakkar, John Crossley, Yida Duan, Namseog Kim, Jun-Chau Chien, Alberto Puggeli, Shingwon Kang, Jaehwa Kwak, Steven Callender, Shaoyi Cheng, Andrew Townley, Nathan Narevsky, Nai-Chung Kuo, Dongjin Seo, Pengpeng Lu, Bonjern Yang, Pi-Feng Chu, Sen Lin, Greg LaCaille, Seobin Jung, Taehwan

Kim, Nandish Mehta, Kyoungtae Lee, Krishna Settaluri, Emily Naviasky, John Wright, Bo Zhao, Han-Chih Yeh, Hokeun Kim, Taehoon Kim, Junho Son, Jinho Park, Donggeon Han.

Finally, I would like to thank my family: Jinseop Han, Jongsim Jeong, Yangja Kim, Kyoungjin Han, Yunrae Jo and Soonae Lee, for supporting me spiritually throughout my life in general. Above all, my deepest expression of appreciation goes to my wife, Minjeong Cho. Without her support, patience and love, none of my proud achievements could be made. Thank you my sweetheart.

Chapter 1

Introduction

1.1 Background

High-speed links have been widely used for providing wide-bandwidth wire-line connectivity between electronic components in various scales, ranging from on-chip to chip-to-chip and system-to-system interconnects. Figure 1.1 shows 2 representative examples that use high-speed links for wireline connectivity; computers and network switches. In both examples, massive data transfers are involved in communications between subsystems, and conventional general purpose digital I/Os would be very inefficient for the transfers in terms of both pin utilization and power consumption. By replacing general purpose I/Os in those systems with high-speed links optimized for multi-gigibit data transfer, their aggregate throughput can be greatly enhanced while consuming reasonable power and pin counts.

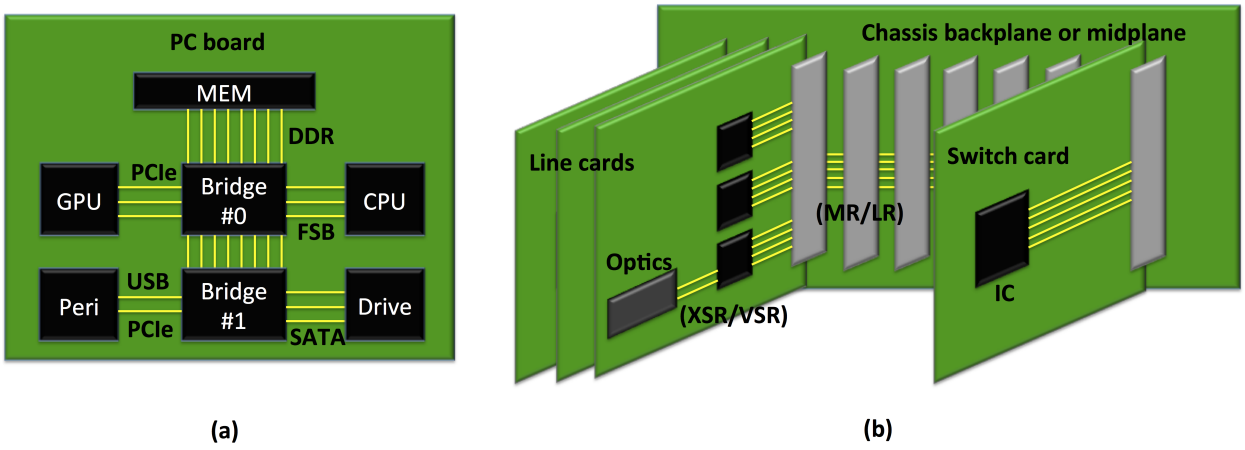


Figure 1.1: Examples of high-speed links (a) For computer applications (b) For network applications

Recently, the demand for high-speed links is increasing tremendously, as the rapid growth

of Internet connectivity (as shown in Figure 1.2) has required systems to become more distributed and data oriented. Figure 1.3 shows the recent technical trend from ISSCC[1][2], on the per-lane bandwidth of high-speed link standards. The trend reveals that average per-pin data-rates of the standards are doubled around every 4 years. In 2017, leading standards (such as the Common Electrical I/O (CEI) from Optical Internet working Forum (OIF)) are already reaching 56Gb/s per-lane bandwidth. Following the increasing trend, it is quite evident that more than 60+Gb/s per-lane bandwidth should be widely supported in a couple of years.

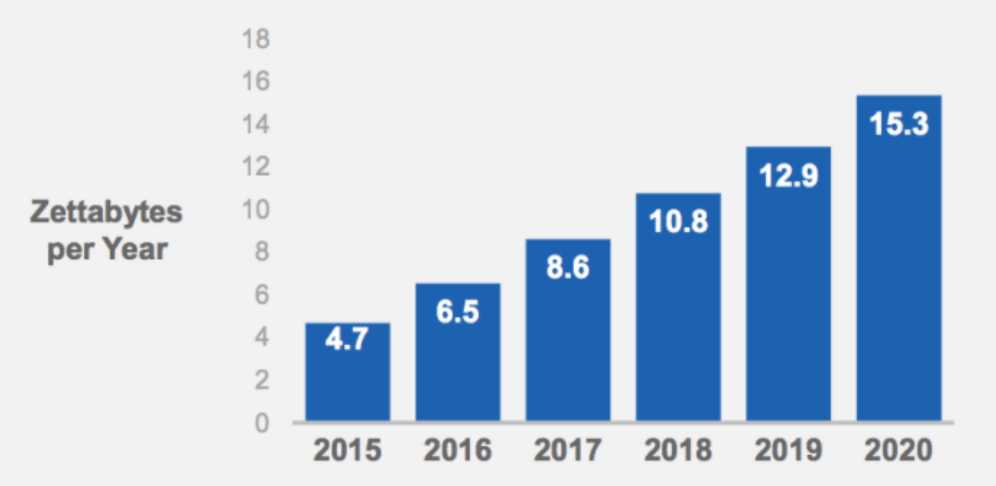


Figure 1.2: Global data center traffic growth (source: Cisco)

In contrast to this rapid bandwidth increase, the power consumption budgeted for these high-speed links remains relatively constant, as projected in the International Technology Roadmap for Semiconductors (ITRS)[3]. This is mainly because the link power budget is heavily constrained by severe thermal restrictions of the entire system, and increasing power consumption to achieve a higher data-rate is usually not a viable approach. Therefore, the energy efficiency of the high-speed links should be constantly improved as the data-rate scales up. Specifically, at the 60Gb/s data-rate, the link is expected to achieve 3-5pJ/bit energy efficiency, to keep the per-lane power consumption below 200-300mW/lane. While the bandwidth and power constraints are very challenging by themselves, to make matters worse, the scaling of the CMOS process is slowing down and the performance gain from scaling is not as dramatic as in the past. As a result, data-rates of recent high-speed link designs show an evident saturating trend below 40nm nodes (Figure 1.4[4]). Therefore, innovative design techniques that push the limits of electrical signaling are essential in meeting those bandwidth and energy efficiency goals in advanced CMOS processes.

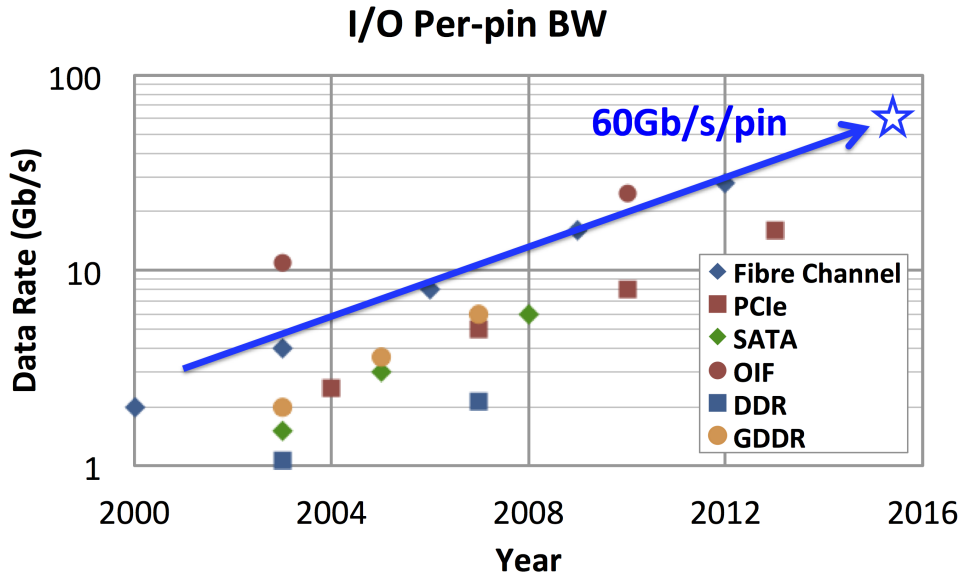


Figure 1.3: High-speed link bandwidth trends

1.2 Prior work

Channel Architecture

As the data-rate reaches up to the 60Gb/s range, channel improvements for ensuring good signal integrity becomes a critical issue. A typical example of electrical channels for backplane serial links is shown in Figure 1.5[5]. Since most wireline channels have a low-pass characteristic in the frequency domain, pulse responses measured from those channels tend to have larger dispersion at high data-rates (Figure 1.6). In addition to that, impedance discontinuities in complex PCB structures, such as packages and connectors, generate highly reflective transients in the pulse response. The resulting pulse responses, therefore, end up having significant inter-symbol interference (ISI) that contributes to higher bit error rates. In order to realize error-free transmissions at high data-rates, designers are constantly looking for ways to improve the channel architecture.

The midplane architecture in Figure 1.7(b) is an example of improving the channel and the signal integrity by reducing the physical distance between connected boards (and potentially reducing signal trace lengths). Combined with the orthogonal connectors shown in the same figure, this technique can provide universal connectivity between all cards without intermediate plane traces.

In such cases, the attenuation will be dominated by remaining traces on daughter cards. Further loss reductions can be achieved by replacing the traces with twinax cables, which have lower loss profiles than conventional backplane traces. The twinax cables are mounted

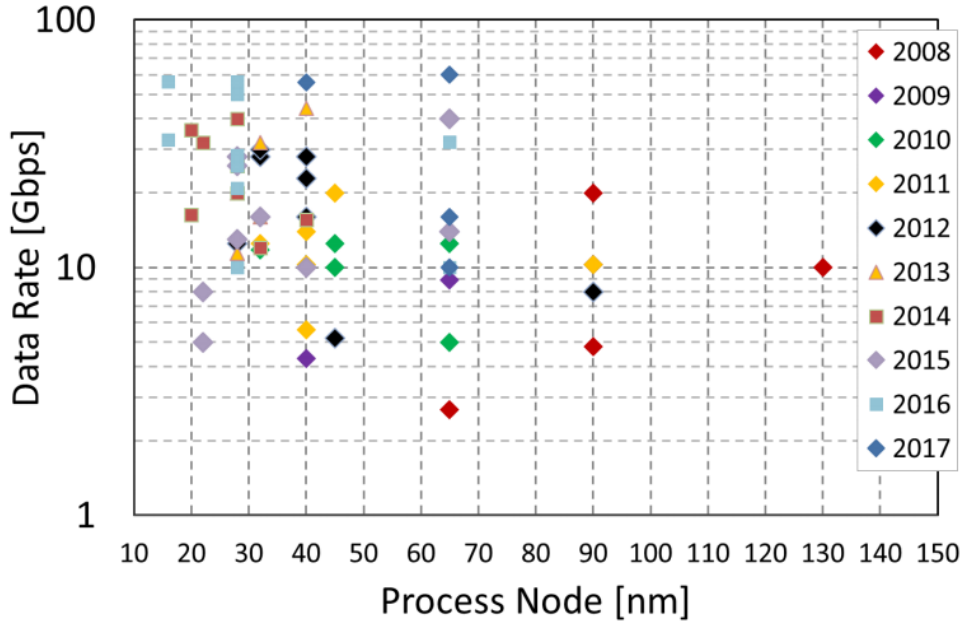


Figure 1.4: Data-rate vs. process node and year[4]

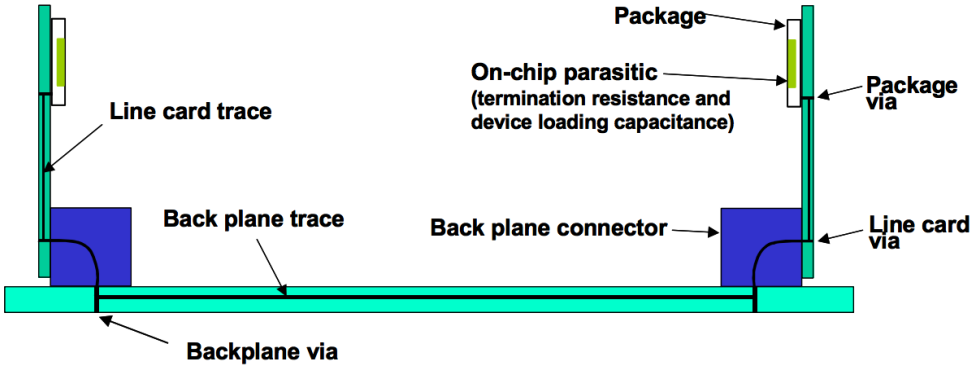


Figure 1.5: a typical high-speed backplane channel[5]

near transceiver chips to minimize residual backplane traces. Further improvements can be made by mounting cables on the top side of IC packages [6], and repeater ICs can be attached on the mounting points to completely isolate reflective structures from the channel, as shown in Figure 1.8.

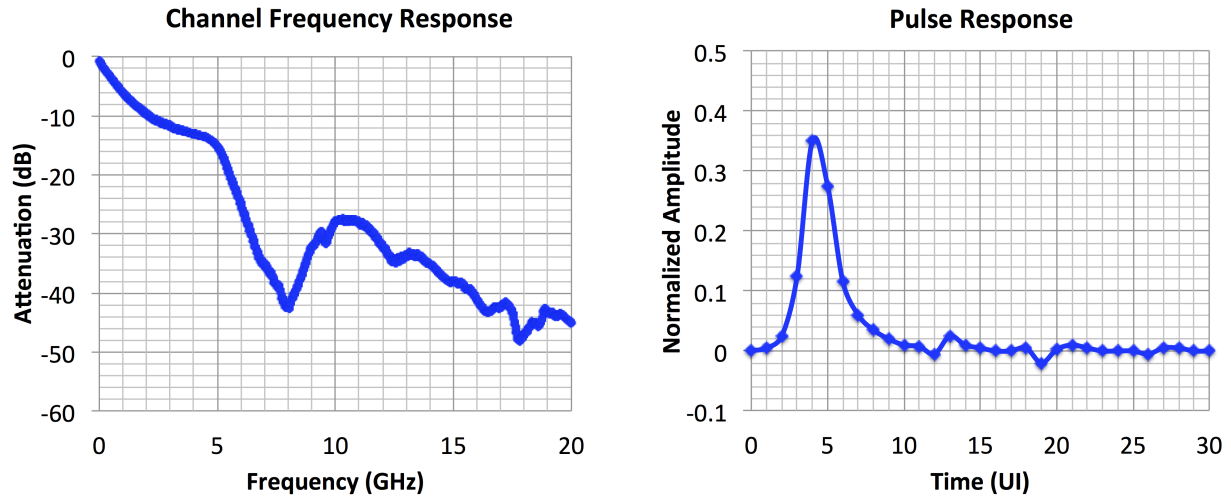


Figure 1.6: Typical backplane channel characteristics (a) Frequency domain (b) Time domain

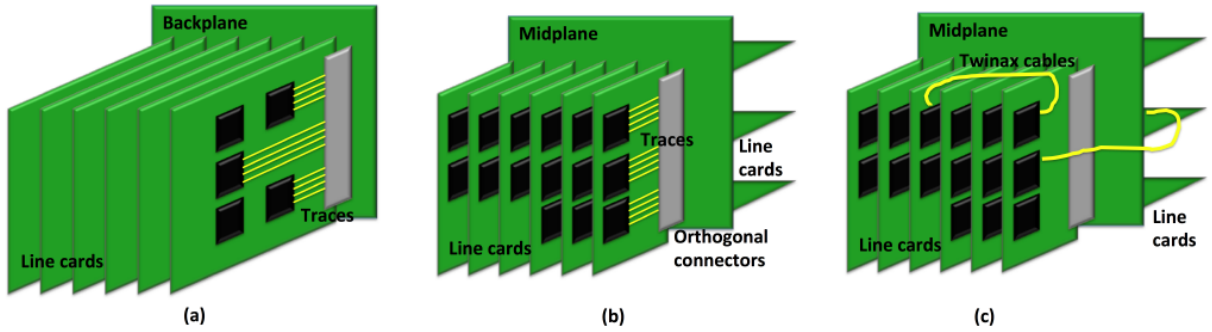


Figure 1.7: Evolution of backplane architectures (a) traditional backplane structure (b) midplane with orthogonal connectors (c) cabled backplane

Signaling schemes

Complex modulation schemes, such as the PAM4 or duobinary [7], have been widely explored in recent transceiver designs [8][9][10] to relax the baud-rate requirements. While PAM4 transceivers are capable of handling high-loss channels, most PAM4 designs require error coding and/or complex digital equalization due to a lower SNR and a higher impact of residual ISI and nonlinearity on eye opening, which typically leads to their power consumptions being higher than the aforementioned 3-5pJ/bit target.

On the other hand, conventional NRZ signaling still remains an attractive alternative. Despite its higher baud-rate requirement, NRZ signaling is more tolerant to non-ideal effects, such as residual ISI and nonlinearity. Therefore, NRZ often leads to reduced power by relaxing design constraints on the frontend signaling path. If the overall insertion loss remains

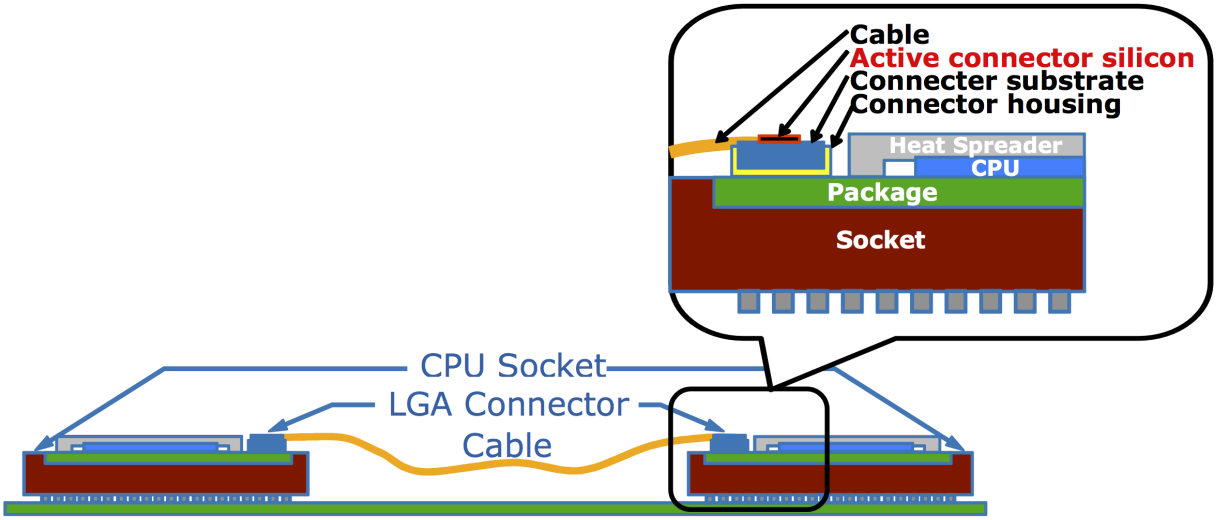


Figure 1.8: Flex interconnect[6] with active connectors (drawing from Bryan Casper, Intel)

within a reasonable range (<25dB), which presumably can be achieved by improving the channels themselves [6][11][12], the power consumption could become even smaller by relaxing the circuit complexity. Therefore, it is worth looking at NRZ signaling when exploring the energy efficiency limits of high-speed links, as well as investigating design techniques for implementing high baud-rate stages for NRZ signaling.

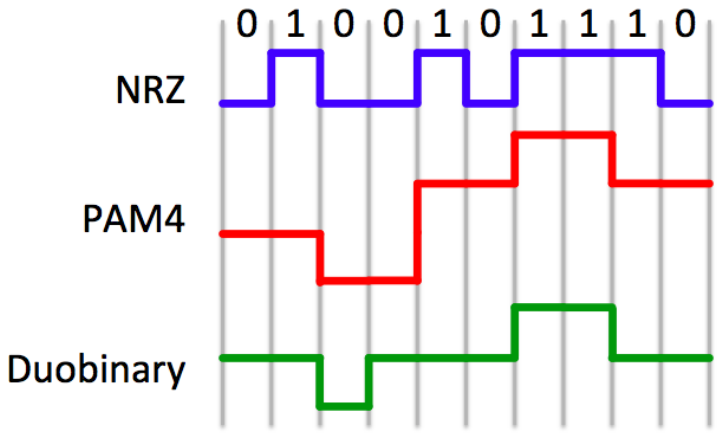


Figure 1.9: Comparisons of different signaling schemes (a) NRZ (b) PAM4 (c) duobinary

Equalizers

Although improvements in channel architecture greatly reduce inter-symbol interference, realistic channels still exhibit higher than a 20dB loss at a 60Gb/s data-rate, and they will still require equalization for error-free operation. If the amount of channel loss remains within a reasonable range (<25dB for NRZ and <15dB for PAM4), analog and mixed-signal equalizers are clearly preferred over digital approaches to address the ISI. Various types of analog and mixed-signal equalizers have been implemented, including continuous time linear equalizers (CTLE), feed-forward equalizers (FFE), and decision feedback equalizers (DFE).

CTLE(Figure 2.3) is commonly located in front of other equalizers and configured to cancel long-tail post-cursor ISI, because the CTLE parameters can readily span over 10-30 taps for long-tail cancellations, while DFEs require too many delay elements to have the same coverage.

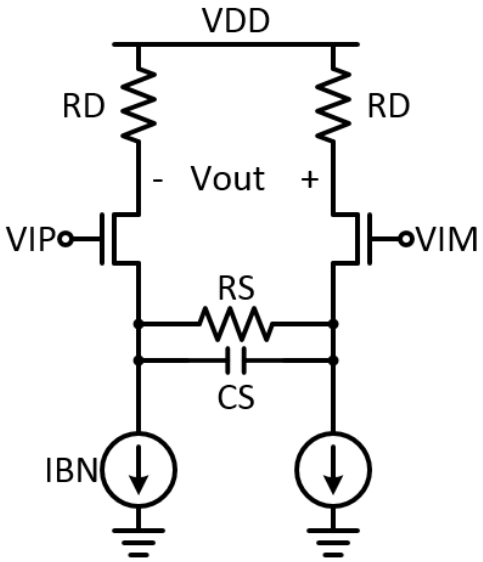


Figure 1.10: CTLE structure

On the other hand, DFE(Figure 1.11(a)) outperforms CTLE for cancelling major post-cursor ISI near the main cursor for following reasons: first, DFE summer taps don't have the linearity constraints present in CTLE, which implies a better energy efficiency. Second, DFE taps are more tolerant to noise, and the stage will have a higher output SNR than CTLE for large cancellation settings (while CTLE with high boosting settings will enhance the high frequency noise). Third, adapting DFE taps for irregular pulse responses is easier than CTLE, which leads to better cancellation performance in major post-cursor ISI.

However, pre-cursor ISI taps cannot be cancelled by DFEs (since they rely on a causal feedback loop), and typical CTLE transfer functions do not directly correct pre-cursor ISI. A feed forward equalizer (FFE, Figure 1.11(b)) [13][14] is therefore often used to handle these

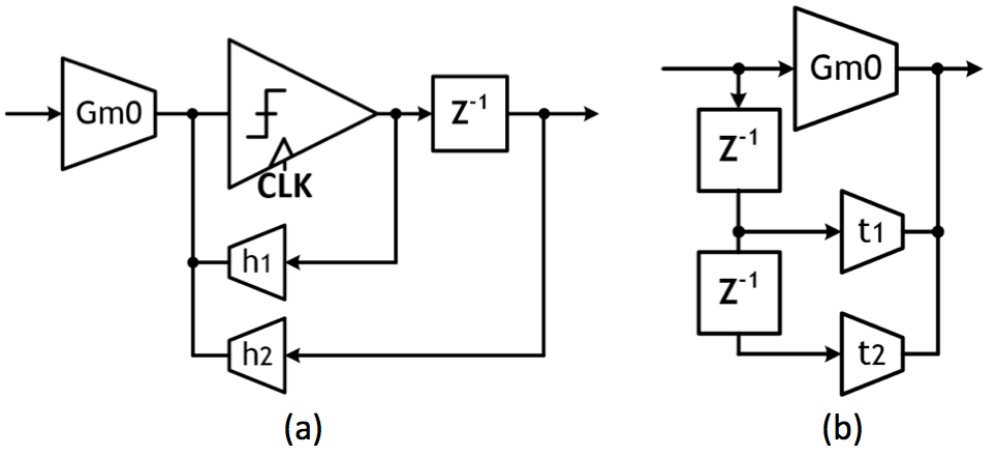


Figure 1.11: (a) DFE structure (b) FFE structure

pre-cursor ISI taps. FFEs are often realized at the transmitter side due to implementation simplicity. However, at the receive side, the underlying circuits are not required to drive a 50Ω termination, opening up the opportunity to save power (via techniques like current-integration) over implementing the equivalent function in the transmitter.

Consequently, using only one type of equalizer is not enough to cancel the whole ISI for most channels. In order to get the maximum equalization performance, multiple equalizers should be combined and set to address various types of ISI. In most cases, the most efficient equalizer configuration will be similar to this: pre-cursors - FFE, large nearby post-cursors - DFE, and long-tail post cursors - CTLE.

Substantial additional capabilities are needed beyond the basic equalization functions to enable robust operation over a broader variety of channels. For example, offset cancellation and tunable biasing circuitry are required to make the equalizer frontend tolerant to process variations and systematic offsets. In order to enhance channel coverage, equalization coefficients should also be re-configurable, ideally set by internal adaptation engines. Zero-forcing (ZF) and sign-sign least-mean-square (SSLMS) algorithms are widely used for implementing the equalizer adaption functions.

Considering the equalizers are usually placed at the frontend of the signaling path (which operates at the highest frequency in transceiver systems), one can easily find that improving the energy efficiency of the equalizer will be extremely critical for achieving the efficiency target (3-5pJ/bit at 60Gb/s). Various ways to implement the energy efficient equalizer stages have been proposed, and in this thesis, the current integration technique will be analyzed and evaluated.

CDRs

In order to compensate the phase skew between the received signal and the receiver's internal clock, a clock and data recovery (CDR) was included in the design. Dual-loop CDRs [15][16][17] are currently the most popular choice because they have the advantage of allowing separate loop bandwidth selection for clock generation and phase tracking. This loop bandwidth separation feature is especially useful when there is a need to reject the VCO noise (usually true if ring oscillators are used) by increasing the loop bandwidth of clock generation. In that case, the phase tracking bandwidth could still be set very low to reject any phase disturbances from residual ISI in incoming signals. Also, for multi-lane implementations (which are very common in practice), the clock generation loop can be shared across the whole lane by having a central (and single) PLL, which could be advantageous if the clock distribtution power is much smaller than the clock generation power.

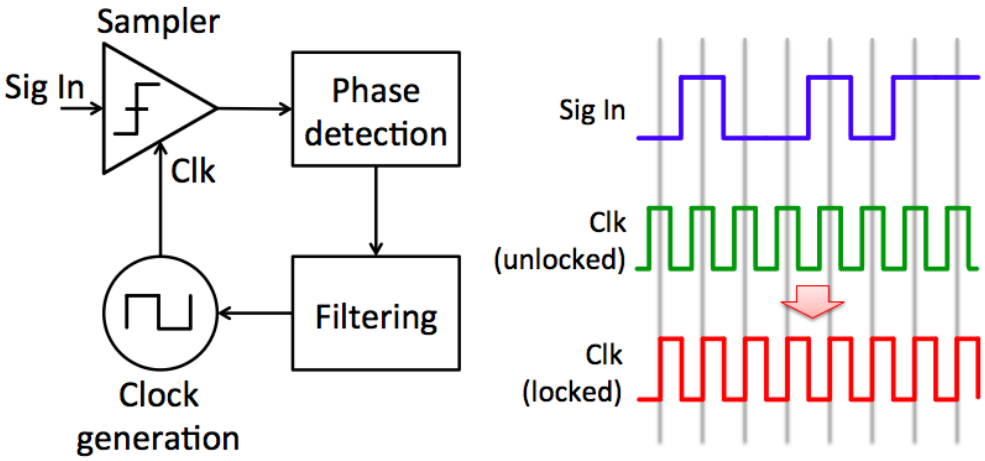


Figure 1.12: Clock and data recovery

On the other hand, the single loop CDR uses only one loop bandwidth parameter for clock generation and phase tracking, which also translates into having a dedicated oscillator per lane. These aspects make the single loop approach less attractive in transceivers at 10-40Gb/s/lane, however, the single loop approach is indeed a better solution for ultra-high data-rate applications for the following reasons. First, most of the 40+Gb/s links use an LC oscillator and a resonant buffer to generate a low jitter differential clock and to distribute it efficiently at a high frequency. Although dual-loop CDR's can reject VCO noise by having a higher bandwidth in the oscillator loop compared to the phase tracking (often implemented with a phase interpolator) loop, a dual loop CDR is not necessary in this scenario because LC oscillators for 40+Gb/s links already have relatively low-phase noise profiles. Second, the power overhead of distributing 20-30GHz clocks over multiple lanes (assuming half-rate clocking) would be substantial, while the physical dimensions of on-chip inductors for the 20-30GHz clock generation are compact, and hence, are easily fit into the area of a single

lane. Therefore, even in multi-lane implementations, once the clock rate is high enough that the physical area of the inductor is relatively small, a single loop CDR is an attractive option for high-speed applications.

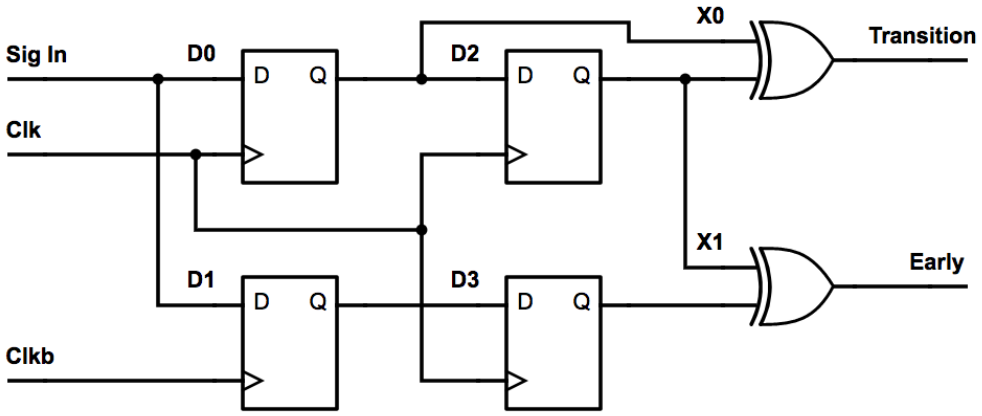


Figure 1.13: bang-bang phase detector

Turning now to the phase detection scheme, commonly adopted 2x oversampling CDRs using bang-bang phase detectors (Figure 1.13) capture phase difference information by sampling the edge signals and comparing them with neighboring data signals. This implies that the CDR requires additional samplers and clock phases for this edge sampling. In particular, 4-phase clock generation and distribution (instead of differential clock generation and distribution) and an additional edge sampler as well as deserializer, would be necessary for a half-rate operation (Figure 1.15). These additional samplers and clock phases can be very expensive at data-rates close to the limit of the process technology (40-60Gb/s). For this reason, alternative approaches such as baud-rate CDRs have been investigated [18], [19], [20], [21] to remove the need for edge sampling.

Transceiver integration

Several high-speed NRZ receivers [22][23][24] and transmitters [14] have been published recently and include various levels of equalization. These designs primarily focus on critical building blocks for equalization, implementing continuous-time linear equalization (CTLE), decision feedback equalization (DFE), and receive feed-forward equalization (FFE) circuits, demonstrating the ability to cancel inter-symbol interference (ISI) while operating close to intrinsic speed limits of the underlying technologies. In order to make a complete high-speed link system, those building blocks are integrated and connected by way of realizing full functionality with autonomous control loops. Recently published NRZ transceivers operating up to 56.5Gb/s with a 1-tap DFE and/or CTLE [25][26] in 28-40nm CMOS processes have been demonstrated for <20dB loss channels with energy efficiencies ranging from 4.4-11.96pJ/bit.

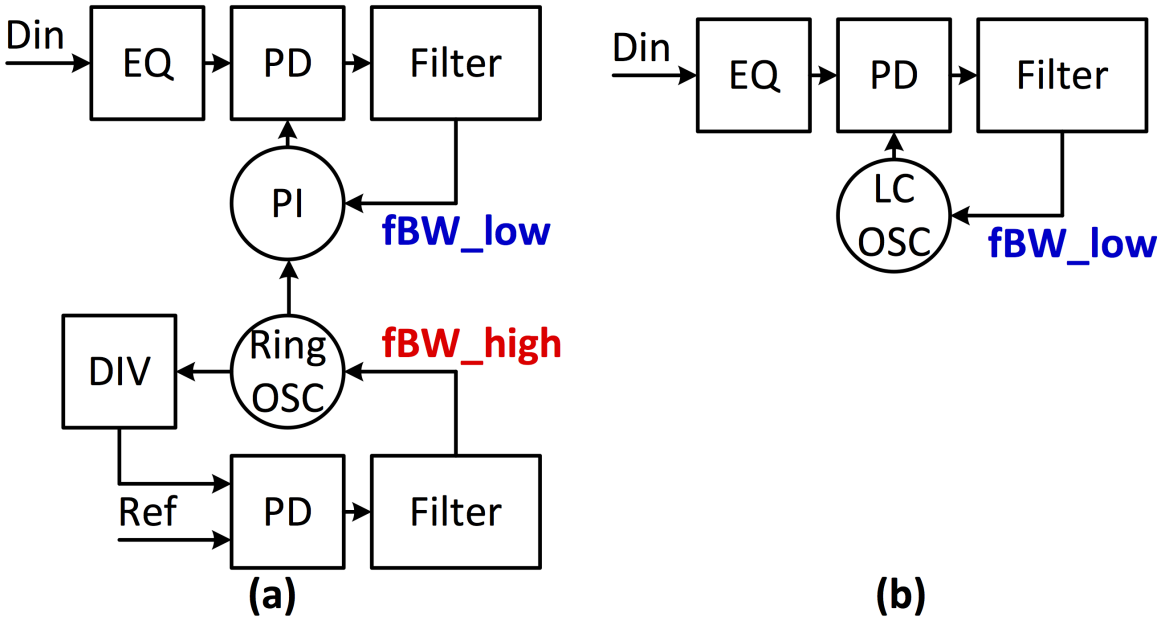


Figure 1.14: (a) dual loop CDR (b) single loop CDR

For PAM4 links, 2 transceivers were published in VLSIC2016[8] and ISSCC2017[10], both operating at 56Gb/s. [8] adopted ADC+DSP architecture to handle high loss channels, while [10] used the mixed-signal approach with 3-tap DFE. Both designs assume the use of error-coding to handle the SNR penalty in the PAM4 signaling, and their energy efficiencies range from 9.82-10.75pJ/bit.

1.3 Thesis Organization

With the motivation of researching high-speed, energy-efficient serial links, and the previous introduction of equalization and CDR techniques, this thesis aims to examine the design techniques to implement high-speed wireline transceivers that achieve 60Gb/s datarate with excellent energy efficiencies and equalization/CDR capabilities. In Chapter 2, a receiver front-tend with current-integrating CTLE+FFE+DFE equalizers will be presented, with cascode gate bias and clock delay techniques that enable complete 60Gb/s equalizer operation.

To demonstrate that all of these requirements can be fulfilled while meeting the overall power budget target, design techniques used to realize a 65nm 60Gb/s receiver frontend including CTLE, FFE, DFE, output slicers, and clock generation, as well as distribution circuits [27][28], are presented. Despite a substantial expansion in equalization capabilities (CTLE + FFE), the power consumption of the complete equalizer path remains essentially identical to our earlier 46mW DFE-only design [22].

Chapter 2 therefore begins by introducing the two key techniques that enable this ad-

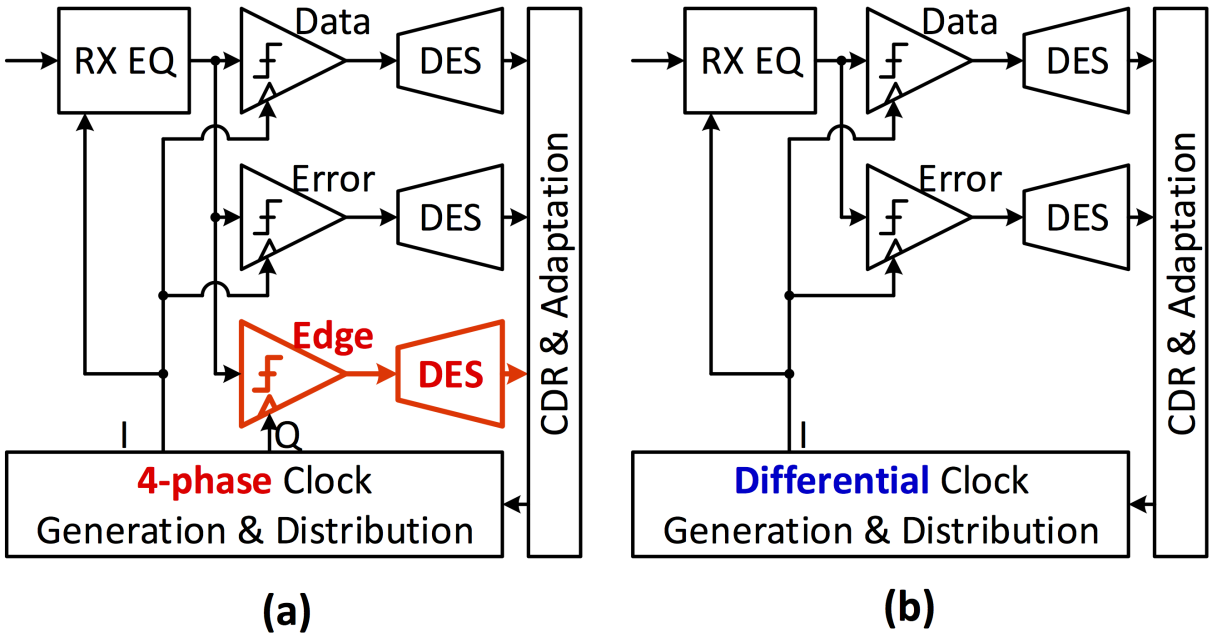


Figure 1.15: (a) 2x oversampling CDR (b) baud rate CDR

vancement - specifically, current-integration combined with the dynamic-latch based DFE architecture from [22] and cascode gate voltage bias as an efficient means of analog gain control for the FFE. It also describes the components necessary to complete the receiver frontend design: interleaved and offset-cancelled deserializing slicers to enable adaptive error sampling as well as LC-based clock generation/buffering. All of these techniques combined result in a receiver frontend supporting 60Gb/s while consuming 173mV from 1.2V and 1.0V supplies.

The 60Gb/s frontend circuitry is highly sensitive to physical and parasitic effects, making it difficult to optimize the design within reasonable time and effort. Multiple iterations through the entire design steps (schematic, layout design, and verification) are often involved in the high-speed link designs to capture parasitics precisely. Chapter 3 introduces the automated circuit generation flow that was applied to the 60Gb/s receive frontend equalizer to address this issue. By utilizing the circuit generation framework, the automated design flow captures layout dependent effects and produces the final design. Sizing and layout generation procedures are scripted and the entire design is completed by running the scripts sequentially, accompanied by multiple iterations to capture post-layout effects accurately, which are very expensive in traditional hand-driven flows. Power reduction from the automated flow (by taking more iteration steps) is analyzed and compared with a reference design.

Continuing in the direction of research presented in previous chapters and in order to demonstrate that a complete transceiver can be realized while meeting the bandwidth and power targets, chapters 4 and 5 present techniques to realize a 60Gb/s NRZ transceiver in a

65nm process that includes transmit FEE, CTLE, receive FFE, DFE, output slicers, clock generation distribution, adaptation, and robust baud-rate CDR [29]. Chapter 4 mainly focuses on the CDR implementation, and begins by briefly reviewing the previous works and challenges for implementing such baud-rate CDRs. The the chapter then describes the proposed baud-rate CDR in detail - specifically addressing phase wandering, the CDR logic, and a phase dithering technique.

Chapter 5 implements all missing blocks and integrates them to a single system. The chapter begins by briefly introducing the transmitter circuitry and the per-path adaptation and calibration loops necessary for finding optimal equalization coefficients. It also describes the measurement results, showing that the transceiver supports 60Gb/s under a 21dB loss channel while consuming 288mW, despite the 65nm process.

Finally, Chapter 6 summarizes the thesis with conclusions and future research directions.

Chapter 2

60Gb/s Energy-Efficient Equalization

2.1 Current Integrating Equalizers

Current integrating equalizers were originally proposed in [30] for implementing a low power DFE, and have been applied in a number of designs since then [30][31][32][33][34]. An extensive analysis of the current integrating DFE/FFE [31] reveals that the current-integration technique typically provides around 3x power saving compared to resistive loaded stages.

The reason why the current integration consumes smaller power is as follows. First of all, the current consumption of resistive-loaded stages(Figure 2.1(a)) is given by

$$I_d = \frac{1}{2} A V_{ov} C_L \omega_{bw} \tag{2.1}$$

where A is the gain, V_{ov} is the overdrive voltage, C_L is the loading capacitance, and ω_{bw} is the bandwidth of the stage.

For high-speed links, ω_{bw} is usually set by the settling error(e_{settle}) constraint $N_\tau = -2.3 \log_{10} e_{settle}$, then the equation 2.1 becomes

$$I_d = \frac{1}{2} A V_{ov} C_L \frac{T_b}{N_\tau} \tag{2.2}$$

where the T_b is the bit period. On the other hand, for current integration(Figure 2.1(b)), the current consumption becomes

$$I_d = \frac{1}{2} A V_{ov} C_L T_b \tag{2.3}$$

Even though 2.2 and 2.3 did not consider the self loading effect, they give a good estimate for comparison. Note that there is a factor of N_τ difference between the resistive loaded and current integration stages (3 for 5% settling error), which translates to a N_τ times smaller power consumption.

From the analysis, the current integrating scheme is substantially more energy efficient than conventional stages. Hence, we have applied the current integration scheme throughout

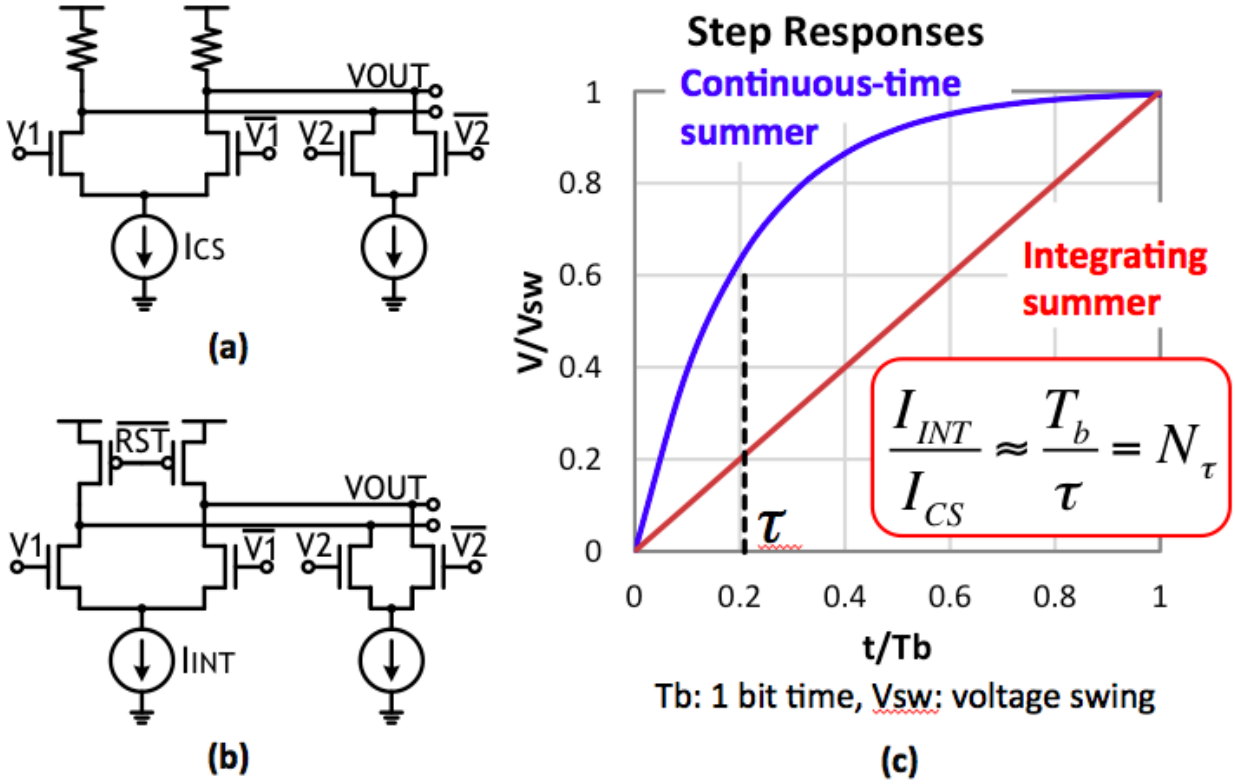


Figure 2.1: (a) Continuous time summer (b) Current integrating summer (c) Step responses

the entire equalizer chain except the final 1-tap dynamic-latch based DFE, which generates NRZ outputs for the DFE feedback.

The proposed receive equalizer architecture is shown in Figure 2.2. The source-degenerated current integrating CTLE's transfer function is tailored to mitigate long-tail ISI, and the circuit performs demultiplexing so that the following stages can operate at half-rate with wider integration time windows. Since the frontend CTLE integrates over the entire bit period, as described in [32], it can introduce 3.9dB of pattern-dependent high-frequency loss (typically one pre- and/or one post-cursor tap), but these taps are readily handled by the power-efficient current integrating FFE and DFE stages.

Figure 2.3 depicts the CTLE integrator structure. The input differential pair (M0, M1) is capacitively degenerated for linear equalization, while switches (M2-M5) are inserted between the input pair and output nodes to support the demultiplexing operation. Inspired by the cascoded sampling technique proposed in [35], the M2-M5 switches are alternatively turned on (by VCKC and /VCKC) to steer the differential input. For each sub-branch, after the integration phase, PMOS precharge devices reset the output nodes to VDD. Within the context of the overall multi-stage equalizer that contains multiple such current-integration stages, this resetting (return-to-zero or RZ) behavior must be carefully managed since the

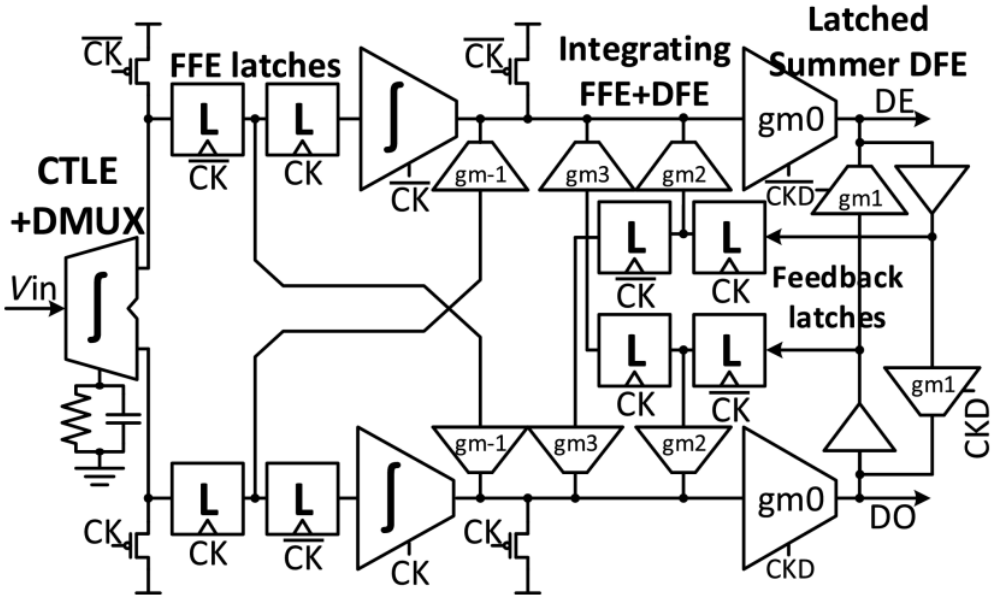


Figure 2.2: Block diagram of the 60Gb/s equalizer with CTLE, 2-tap FFE, and 3-tap DFE

output signal of one stage would not naturally be available to the next stage during its integration time.

Dynamic latches (L0-L3) are therefore inserted after the CTLE to convert the RZ waveforms to non-return-to-zero (NRZ) ones, extending the available signal window to be latched properly by following stages. These latches also generate UI delayed signals for the 2-tap FFE operation. The basic structure of FFE latches is inherited from DFE feedback latches in [22], but with increased input transistor overdrive voltages to ensure that the latches retain linear signal transfer characteristics.

It is worth noting that the resetting nature of current-integration stages leads to several further design challenges within the overall equalizer, particularly within the timing-critical final DFE stage. After examining the proposed approach for implementing FFE variable gain control, these issues and our solutions to them will be described next.

2.2 Integrating FFE Implementation

After the CTLE and delay latches, two half-rate current integrating 2-tap FFE and 3-tap DFE stages receive the UI spaced signals. These circuits were evolved in several respects from the architecture proposed in [22]. First, the power hungry continuous time summers for the 2nd and 3rd DFE taps were replaced with current-integration summers, and an FFE tap was added to each integrator in order to support the FFE functionality without introducing additional summers, as shown in Figure 2.4.

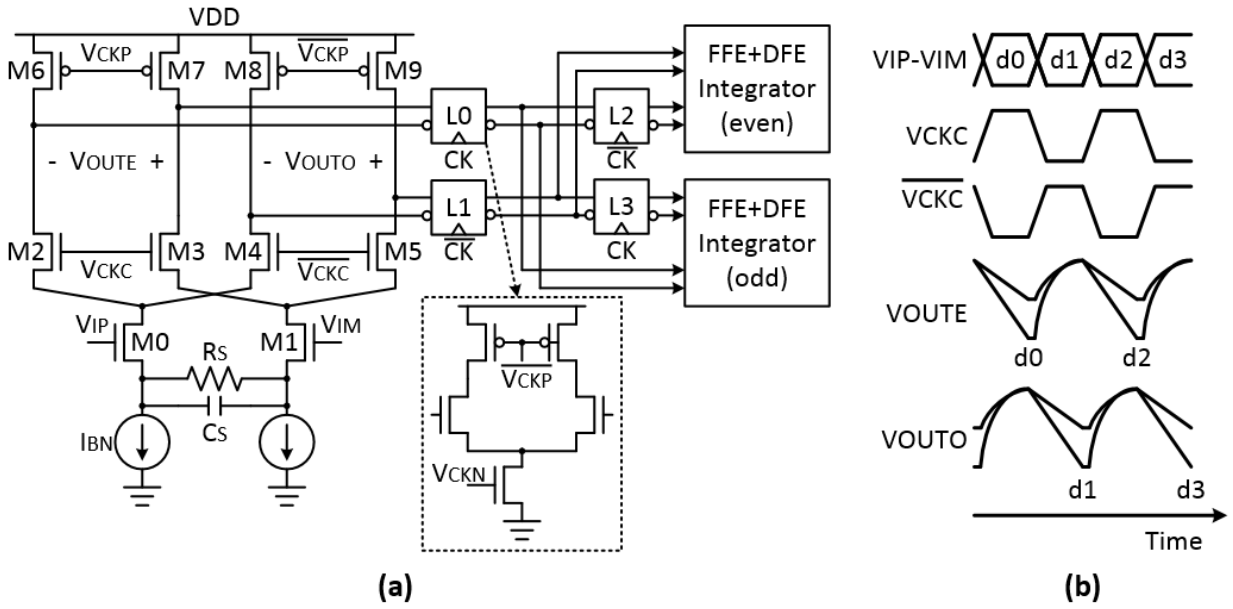


Figure 2.3: (a) Current integrating CTLE (b) Its timing diagram

The transconductances of both the FFE and DFE integration branches must be programmable in order to realize variable equalizer coefficients needed to support varying channel characteristics. Fig. 5 shows three potential methods to realize variable transconductance (i.e., tap weight). The conventional variable current bias control (Figure 2.5(a)) works well for the DFE integration paths [22], and is applied to the 2nd and 3rd DFE tap branches (gm2 and gm3 in Figure 2.4). However, this approach (bias current control) is unfortunately infeasible for the FFE because of the negative impact of reducing tail current on the linearity of the FFE stage's differential input pair. Specifically, as the gain of this stage is reduced (e.g. to compensate for channels with relative small pre-cursor ISI), the overdrive of the input pair will drop. The FFE tap input pair will therefore clip when presented with relatively large signals (which will almost certainly occur since the signal has not yet been fully equalized at this point in the chain), leading to non-linear signal distortion that cannot be directly compensated by the equalizers.

One potential method to realize variable gain while avoiding this linearity issue is to modulate the device sizing as the bias current scales, as shown in Figure 2.5(b). Specifically, multiple differential pair unit cells are configured in parallel and the number of cells turned on determines the variable gain. This scalable differential pair approach preserves the operating conditions of the input devices regardless of the gain setting, maintaining the linear characteristic. Alternatively, the transconductance can also be controlled by modulating the source degeneration resistance (Figure 2.5(c)). Unfortunately, both of these approaches suffer in power efficiency due to dynamic range and minimum device size limitations [31]. To illustrate the issue by way of example, let's assume that the tap requires 6 bits of gain

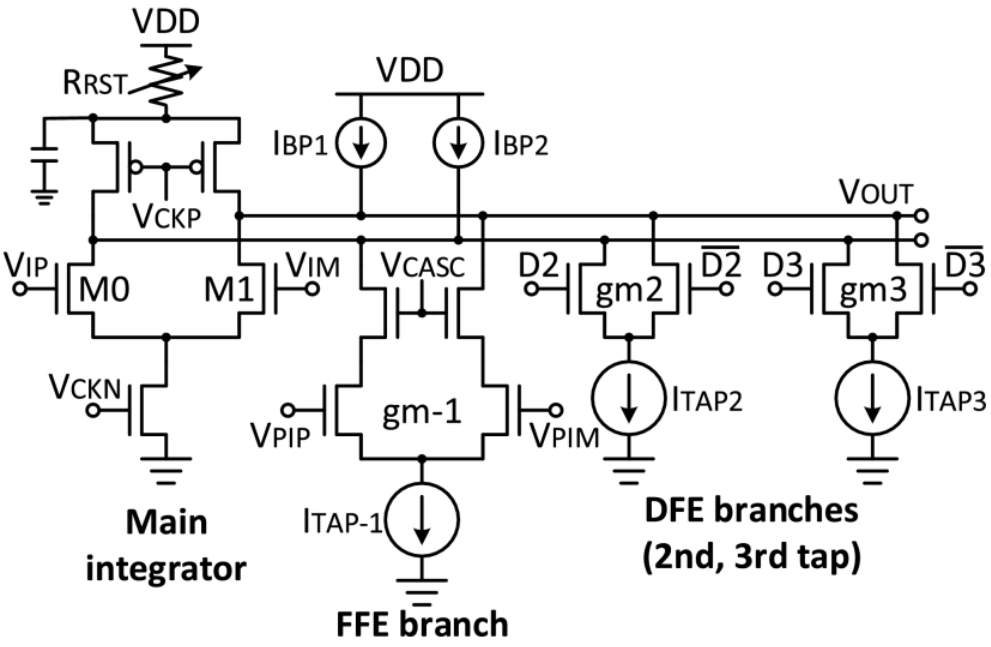


Figure 2.4: FFE+DFE integrator

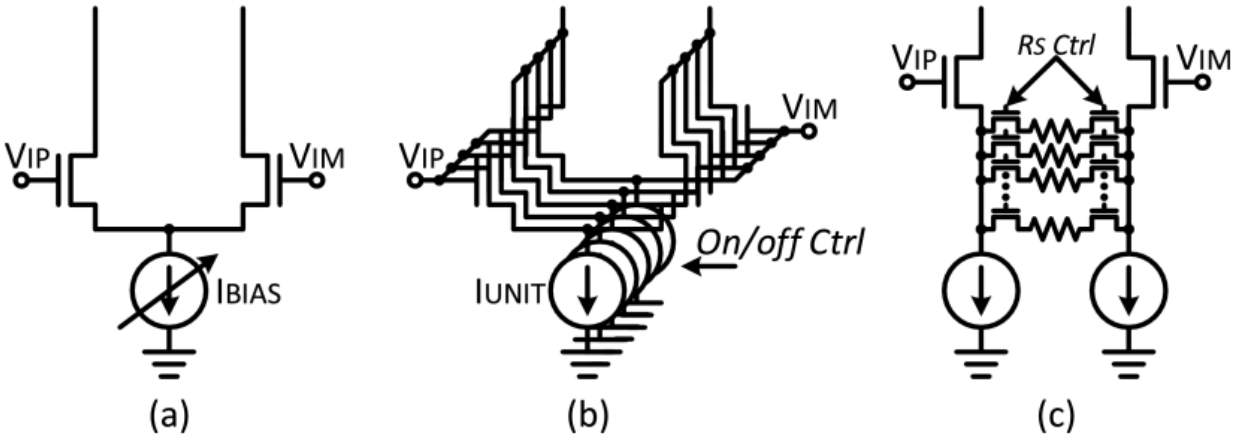


Figure 2.5: Various ways of implementing programmable gains

control range. This would require the input pair (or degeneration resistor) to be broken up in to 64 unit cells; considering that each unit cell cannot be made smaller than a certain minimum size, the total device width (and routing parasitics) associated with that many units typically substantially exceeds the width that would have been needed just to meet the required signal swing (gain)/bandwidth. For the variable width input pair this directly results in substantial power consumption overhead (since current and device width are directly related), while for the variable source degeneration, the extra parasitics at the source node

will force increased bias current in order to ensure that the resistive degeneration remains effective at the frequencies of interests. As a variant of Figure 2.5(c), the degeneration resistor array in Figure 2.5(c) may be replaced with one transistor with variable gate voltage bias to modulate its on-resistance. However, it should be noted that the transistor on-resistance (and the overall transfer function) becomes highly non-linear at low-gate bias, causing non-linear distortions, which are not desirable for FFE operation as indicated in the previous paragraph.

In order to simultaneously support a wide range of variable gain without suffering in terms of signal linearity or a direct increase in power consumption as the dynamic range in gain is increased, we propose a control scheme based on adjusting the gate bias of cascode devices embedded within the FFE stage (Figure 2.6(a)).

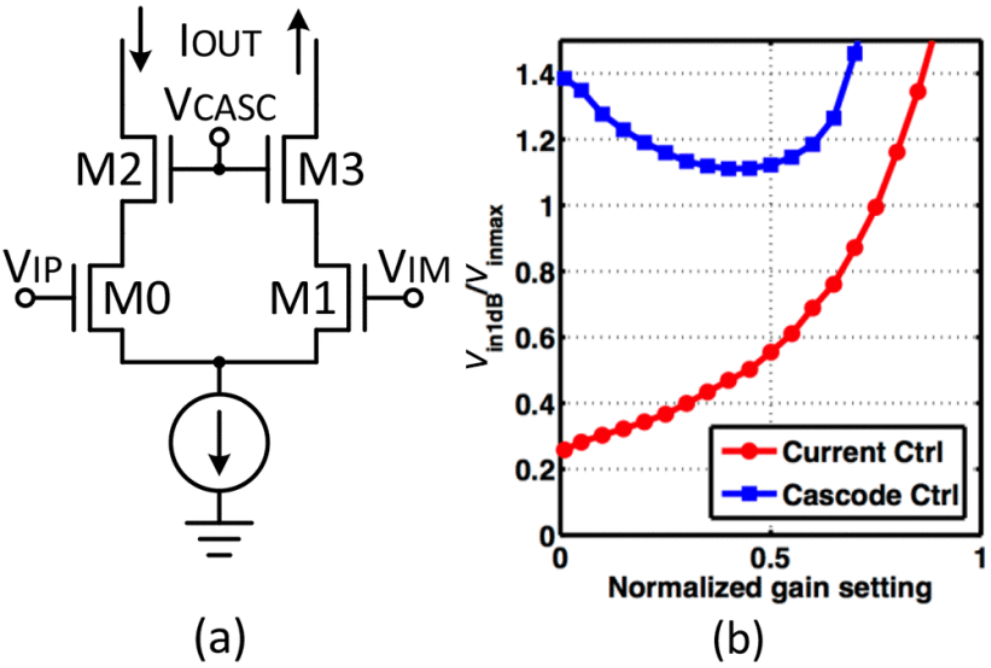


Figure 2.6: Variable cascode gate-voltage bias: (a) schematics and (b) linearity comparison with a variable tail current-control scheme. In (b), V_{in1dB} is the input voltage at which the gain drop by 1 dB from its target value, while Vinmax is the maximum input-swing expected to be fed into the FFE stage.

NMOS devices (M2, M3) are inserted between the input differential pair and output nodes, and the gate voltage of the cascode devices (which is set by a 6-bit voltage DAC) controls the gain of the cascoded differential pair. This cascode gate bias sets the drain voltage and output impedance of the input differential pair devices, thus setting the overall gain, but since the overdrive voltage of the input devices is largely preserved, the signal linearity (in terms of diff. input voltage to diff. output current) remains largely unaffected

by the gain setting. Figure 2.6(b) shows a linearity comparison between the variable current source and the variable cascode gate voltage bias schemes, highlighting the ability of the cascode control scheme to maintain linearity over the entire gain control range.

2.3 3-tap Hybrid DFE Design

As mentioned previously, since DFE taps don't need to maintain linearity in terms of their input to output signal characteristics (their inputs are sliced digital values in any case), current-integration branches for the 2nd and 3rd DFE taps use variable current sources for coefficient control. As also mentioned earlier, the FFE and DFE taps are summed in a single stage to reduce the number of stages, and the main integration branch is a clocked differential pair whose tail current is shut off during the reset phase to reduce static power consumption (Figure 2.4).

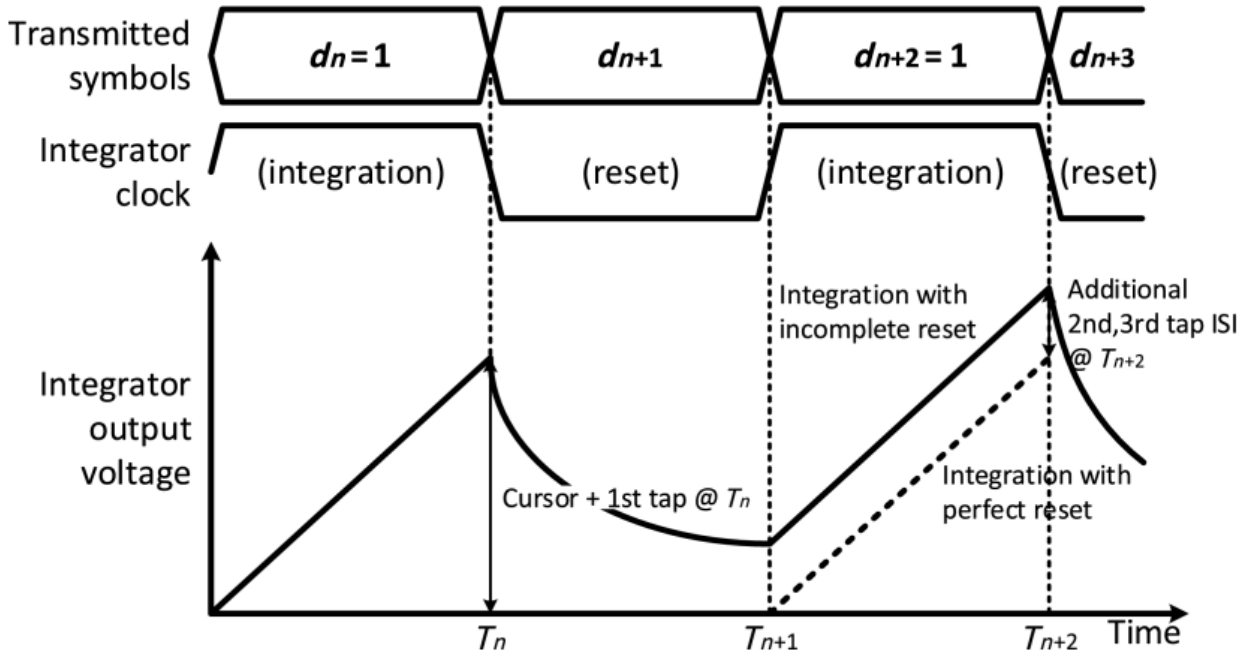


Figure 2.7: Illustration of incomplete reset settling resulting in postcursor ISI.

One of the key implementation issues to be addressed regarding the current integrating DFE is resetting the integrator. Among various resetting schemes in [31][32][33][34], we chose precharging PMOS loads to minimize the number of devices connected to the output nodes. Despite the strong gate drive of PMOS loads, in the 65nm process it was very challenging to achieve precise reset settling within the 16ps of reset time available for the half-rate 60Gb/s equalizer. Therefore, in this work we used relatively small reset switches that settle to only 85% of the final value (i.e., 2τ of settling) and relied on the DFE to compensate for the

resulting circuit-induced ISI. Specifically, this incomplete reset translates in to 2nd and 3rd tap ISI (Figure 2.7), which are readily handled by overdriving the current integrating DFE coefficients. This greatly relaxes the reset bandwidth constraints and regulates the power consumption by reducing the device sizes of the 30GHz reset network and their parasitic capacitances.

To further quantify the efficacy of this incomplete reset approach, we will next analytically examine the tradeoffs between reset device size and the required current of the current-integration stage. In [14] the power consumption of a current integrating DFE (Pint) depicted in Figure 2.8(a) at data-rate f_S, gain A, and load capacitance C_L was shown to be:

$$I_d = \frac{I_{nom} \cdot 2(1 + k_{reset})}{1 - \frac{A \cdot f_S}{\omega_{T,in}} \cdot \gamma(1 + k_{tap0} \cdot \frac{v_c}{V_i^*} \cdot \beta) \cdot (1 + k_{reset})} \tag{2.4}$$

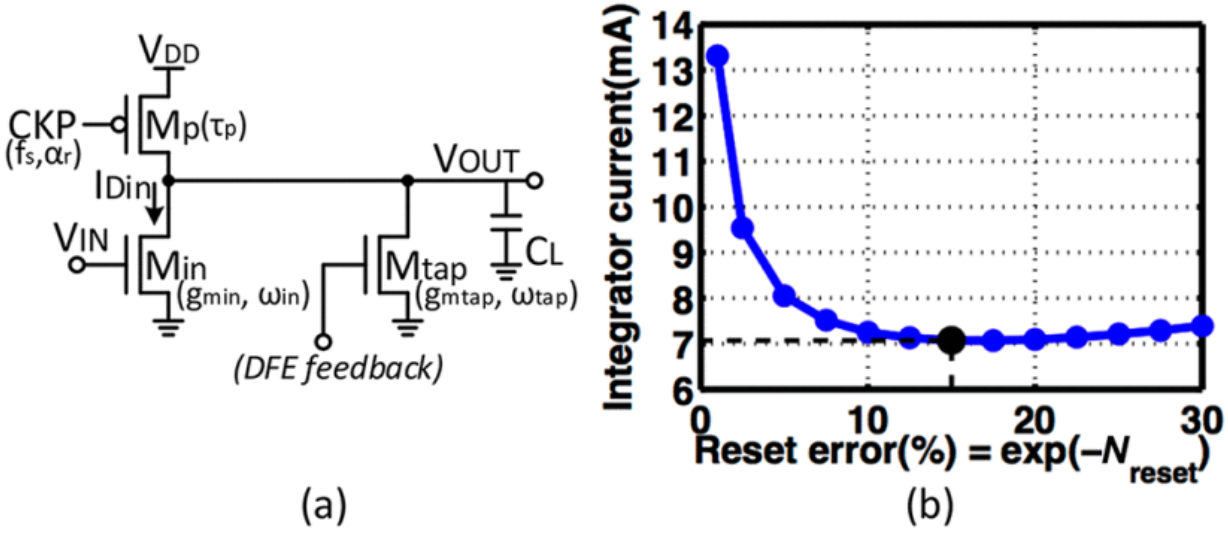


Figure 2.8: (a) Simplified current integrating DFE circuit diagram. (b) Integrator current consumption vs. target reset accuracy.

In the above equation, $I_{nom} = C_L \cdot V_i^* \cdot A \cdot f_S$ is the current consumption of a class-A amplifier without self-loading, $\omega_{T,in}$ is the transit frequency of the input transistor (Min), V_i^* is defined by $2I_{D,in}/g_{m,in}$ of the input transistor, v_c is the input cursor amplitude, k_{tap0} is the ratio between the cancelled ISI and the cursor, which is originally denoted by $k \cdot N_{coef}$ in [31], γ is the drain to gate capacitance ratio, $\beta = \omega_{T,tap}/\omega_{T,in}$ is the ratio between the transit frequency of the tap transistors and the input transistor, and k_{reset} is the factor by which the capacitance of the summing node needs to be increased to include reset capability. Instead of deriving k_{reset} in terms of digital fanout-of-4 delay, as in [31], the effects on the incomplete reset are captured by an equivalent RC network composed of the PMOS reset

device (Mp) and load capacitances at the output. We can then express the time constant of the reset network as:

$$\tau_{reset} = \tau_p \cdot \frac{1 + k_{reset}}{k_{reset}} = \frac{\alpha_r}{f_S \cdot N_{\tau,reset}} \tag{2.5}$$

where $\tau_p = C_{Dp} \cdot r_{op}$ is the time constant of PMOS resetting devices without additional loads, r is the fraction of each UI spent for resetting, and $N_{\tau,reset}$ is the number of time constants of exponential RC settling for the desired accuracy. k_{reset} is therefore given by:

$$k_{reset} = \frac{\tau_p}{\frac{\alpha_r}{f_S \cdot N_{\tau,reset}} - \tau_p} \tag{2.6}$$

It is also important to note that the tap strength has to be increased by $e^{-N_{\tau,reset}}$ to cancel the translated ISI from finite resetting, which limits the gain from this reset bandwidth relaxation technique. By combining 2.4 and 2.6, the normalized summer current requirement with different target accuracies can be derived, as shown in Fig. 8(b). From the plot, 85% settling accuracy is chosen for optimal power dissipation.

Another design issue regarding the PMOS precharging reset is that the output nodes experience significant common-mode (CM) fluctuations during the integration and reset phases. This causes several problems; first, during the integration phase, the substantial drop in CM can reduce the output resistance of the input devices (M0, M1 in Figure 2.4) of that stage, degrading gain and linearity [31]. In addition, especially for multi-stage equalizer implementations, excessive CM fluctuations can impact the operation of the following stages. These effects lead to the need for CM control schemes to reduce these fluctuations. In this design, a resistor DAC (R_{RST}) along with a bypass capacitor is inserted between VDD and the reset voltage node to create a virtual supply that regulates the reset voltage level, and pull-up current sources [34] connected to the output nodes maintain the CM level during integration. The combination of these two techniques regulates the output CM fluctuation and ensures proper operation of the following stage.

As in the original architecture from [22], the stringent timing latency requirements in the first post-cursor tap DFE feedback path justify the use of a dedicated summer for the 1st tap DFE. After the FFE+DFE integration summer, a separate dynamic-latch-based stage [22] provides the one-tap post cursor cancellation, as shown in Figure 2.9. Noting that the integrator output is a signal that ramps-up from differential zero, it is desirable to fine-tune the dynamic latch transparent window timing to best match the peak overall output of the integrator. This can be achieved by delaying the clock of the dynamic latch relative to that of the integrator.

Figure 2.10(a) shows simulated 1st tap DFE output waveforms with and without the latch clock delay. In the figure, the DFE output swing increases by 40% when the clock is delayed by 3ps. Since this delay is only 10% of the clock period, an RC passive network is chosen as the most efficient means of generating this delay, which can be readily utilized without suffering significant clock amplitude attenuation. One potential design issue involving the RC

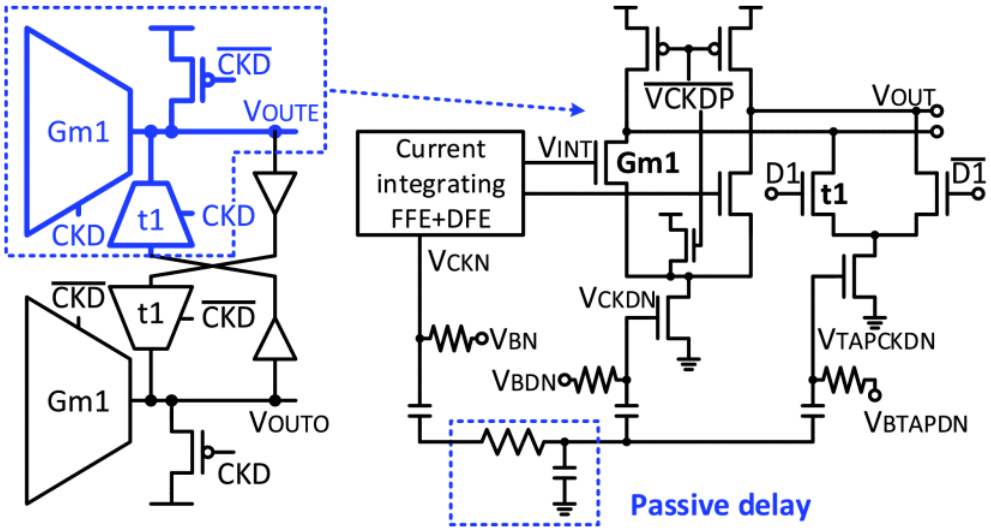


Figure 2.9: One-tap DFE structure and the associated clock-delay technique.

delay is the variability of the passive elements. Figure 2.10(b) shows the output amplitude with various delay settings (varying the resistance value): up to 20% delay variations result in only 5% variation in output amplitude.

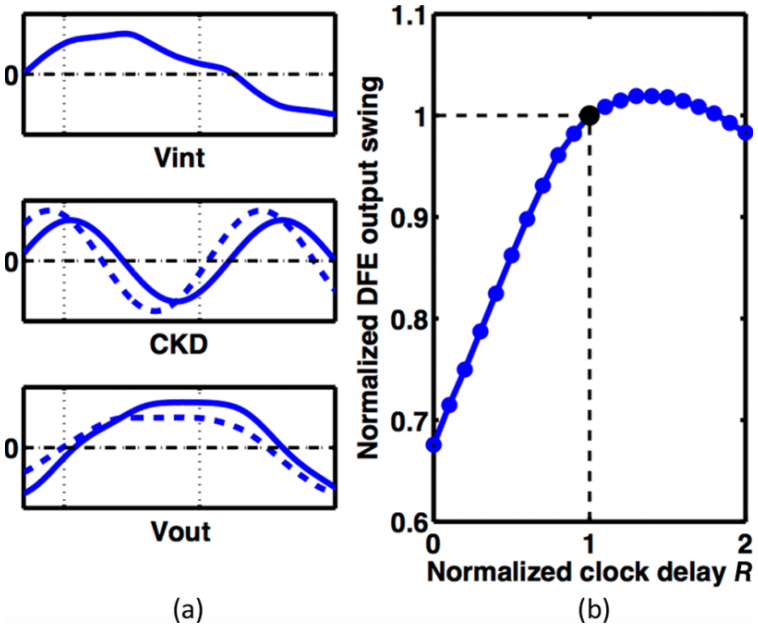


Figure 2.10: (a) DFE behavior without clock delay (dashed) and with clock delay (solid). (b) Clock delay versus DFE output-swing.

2.4 Sampling Backend Design

A complete receiver frontend must support CMOS levels in order to be compatible with the final digital backend processing/consumer of the data. In this particular architecture, the final DFE stage already offers a relatively large output swing since at least 250mV swing is required to force the DFE feedback tap currents to be fully steered [22]. Thus, further slicing of the DFE output to resolve the received symbol (data) might be easily done with a sampler stage with moderate gain (and likely without requiring offset cancellation). However, in addition to the data itself, the frontend must provide additional digitized signals; in particular, adaptive equalization requires digital signals representing the signs of the differences between the signal amplitude and the estimated cursor amplitude (error) [36][31][37], and the gain and bandwidth of the data/error sampling paths should be matched in order to avoid incorrect equalizer settings for the actual signal path.

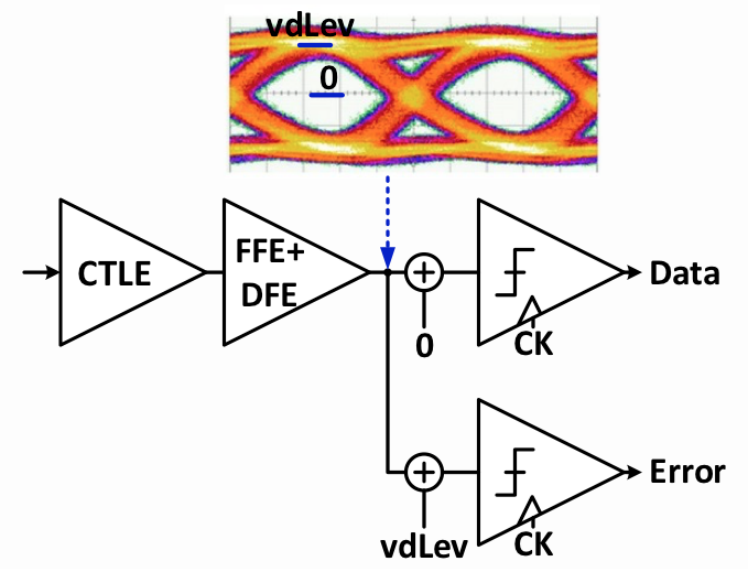


Figure 2.11: Data and error(adaptive) samplers

Since the adaptive loop will intentionally be adjusting the equalizer's (and its own) settings to minimize the error amplitude, the error-slicing path inherently has to sample very small signals(Figure 2.11). Thus, significant additional gain is required to support the error sampling operation no matter how wide of an output swing range is secured from the dynamic DFE latch stage. Therefore, clocked regenerative samplers [31][37][38][39] with offset cancellation capability are essential to meet this high gain requirement.

For this design, a StrongArm latch(Figure 2.12) with offset calibration in the pre-amplifier (preamp) stage [31] was selected due to its narrow aperture window and CMOS output levels. Furthermore, instead of pushing individual StrongArm latches to operate at the extremely high frequency, multiple latches are interleaved to achieve the overall 30GHz sampling rate.

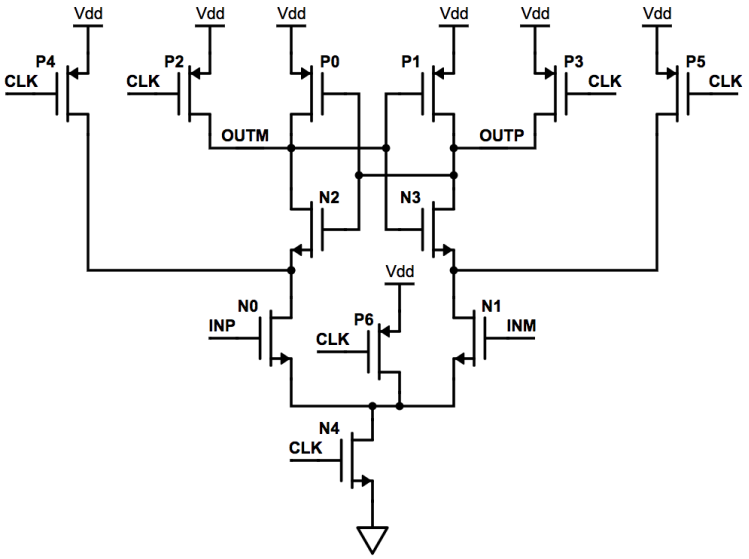


Figure 2.12: StrongArm latch for sampling operation.

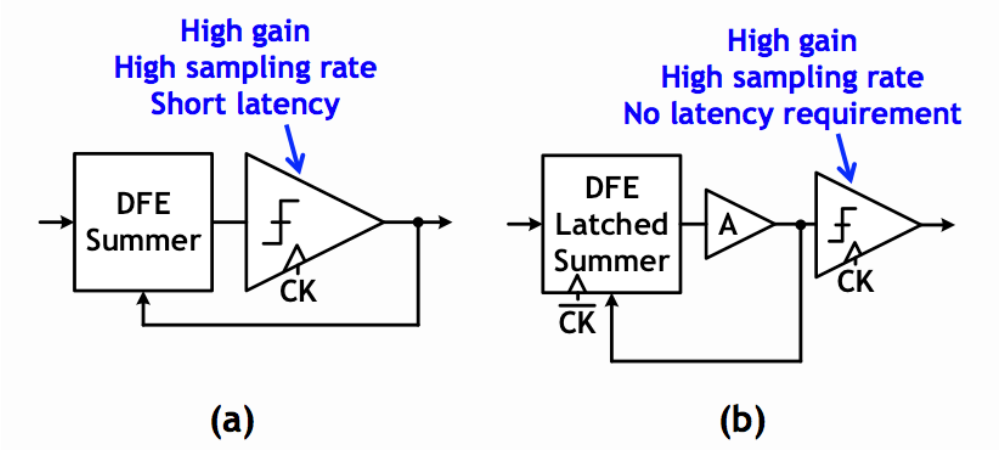


Figure 2.13: (a) Sampler design constraints for conventional DFE (b) for our DFE

This interleaving technique is possible since these samplers are outside of the DFE loop and hence do not have tight latency requirements (Figure 2.13). The relationship between the minimum detectable signal level and the sampling frequency (Figure 2.14) reveals that 4x interleaving provides sufficient sensitivity without adding too many samplers. Individual samplers are clocked by one of 7.5GHz quad phased CMOS level clocks, provided by clock dividers and a phase interpolator.

As hinted at earlier offset cancellation is required within the deserializing samplers to ensure sufficient sensitivity as well as to support an adaptive threshold for error-slicing. While adding the offset cancelling branches to preamp output nodes works for the offset cancellation

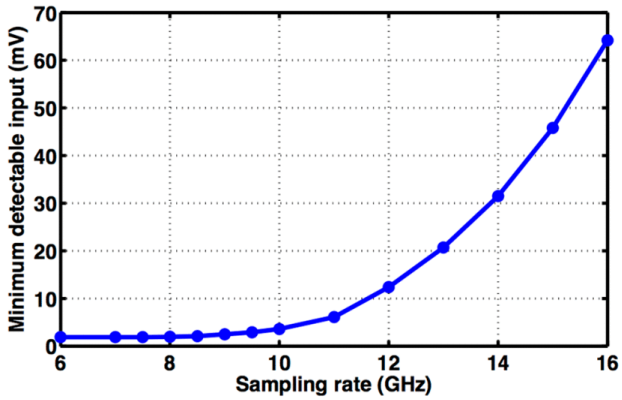


Figure 2.14: Sampling rate versus minimum sampler input-swing.

of individual samplers, it is not effective for the error-sampling purpose, which requires a large-scale decision threshold shift that poses a linearity requirement and substantial sideload to the preamp stage. In order to inject the cursor amplitude offset intentionally for the error-sampling, AC coupling capacitors are inserted between the DFE output and preamp input ports, and the DC operating points of the coupler outputs are controlled to set the decision thresholds. Figure 2.15 visually illustrates the flow of improvements on the sampler design and Figure 2.16 shows the resulting implementation of the sampling array with 4x interleaving, offset calibration, and AC coupling for the error sampling.

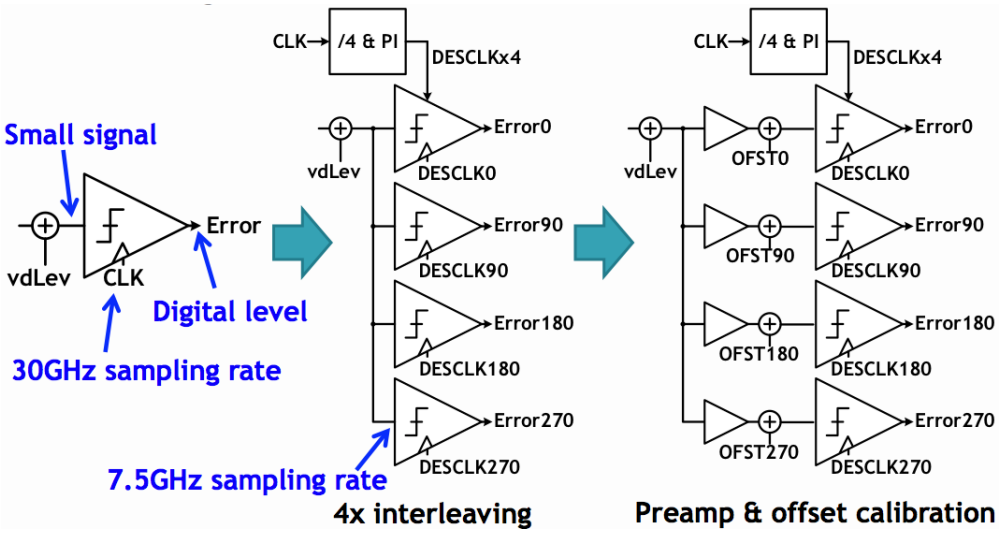


Figure 2.15: Evolution of output sampler design.

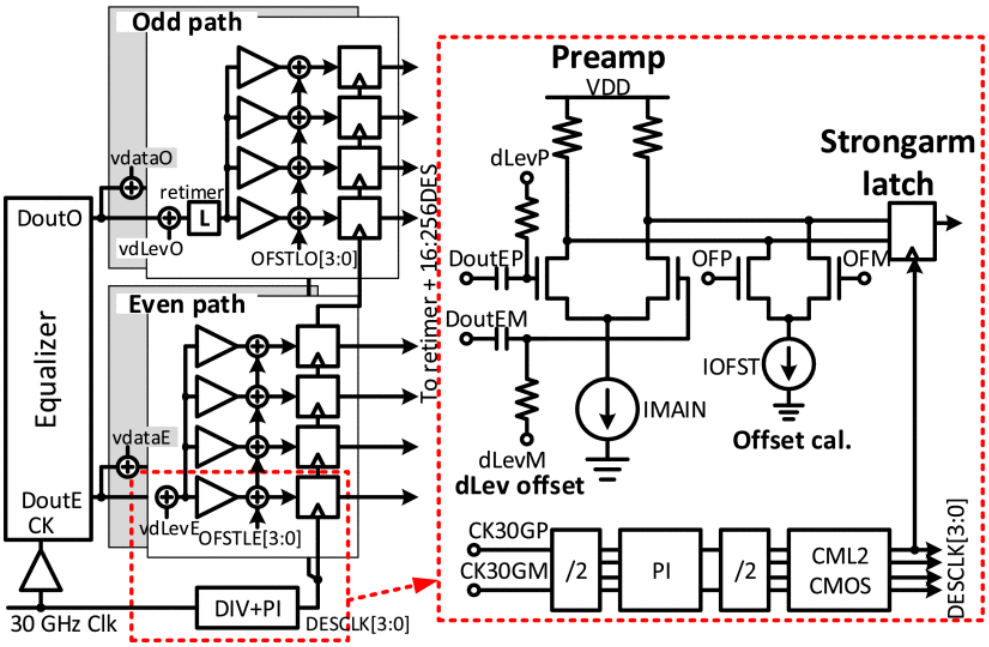


Figure 2.16: Interleaved deserializing slicers.

2.5 Measurement Results

The receiver frontend test-chip was designed and fabricated in a 65nm CMOS process (Figure 2.17). The receiver frontend occupies $0.16mm^2$ except the pad, ESD and t-coil area, and the equalizer core circuit occupies $0.012mm^2$; both the specific device sizes and the layout of all equalizer circuits (CTLE, FFE, and DFE) in Figure 2.2 were generated via the utilization of the analog circuit generator framework described in [40][41].

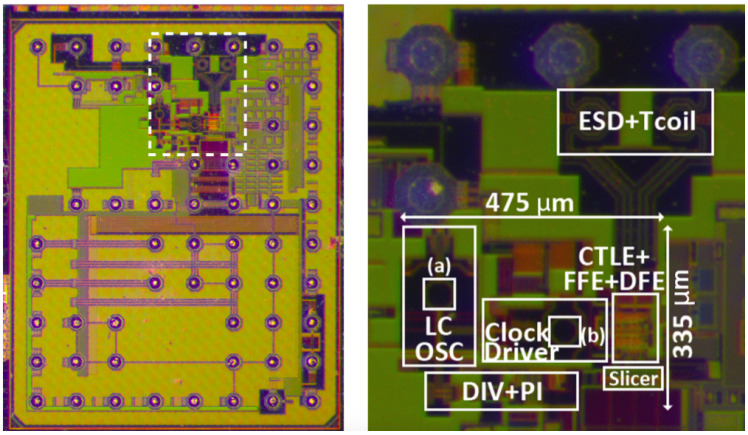


Figure 2.17: Die photo

The measurement setup used to characterize the test-chip is depicted in Fig. 16. The fabricated receiver chip was directly soldered to a PCB made out of Nelco 4000-13 material via flip-chip bumps to minimize parasitic loading from bonding and packaging structures. Since there was no 60Gb/s signal source available for the measurement, we re-used the pattern generator/channel-emulator circuit from the chip described in [22]. This band-limited transmitter with built-in programmable ISI generation provides differential PRBS7 signals with emulated channel profiles.

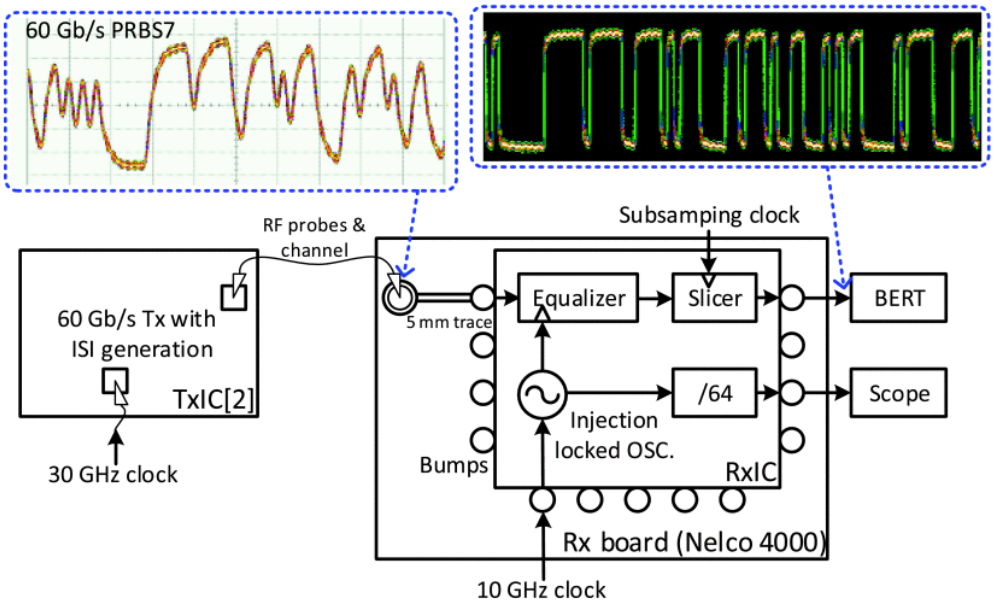


Figure 2.18: Measurement setup and measured waveforms. (All clock sources are synchronized).

As a first step in receiver frontend testing, the performance of the clock generator was characterized to determine whether the oscillator supports the desired operating frequencies. Fig. 17 shows the oscillation frequency with various band select settings. The LC oscillator covers a 25-30GHz frequency range and the receiver frontend operation was verified at the maximum frequency setting, which corresponds to 60Gb/s data-rate.

Following clock path characterization, a BER bathtub curve measurement was performed to test the equalization capability of the receiver frontend data path. Figure 2.20(a) and Figure 2.20(b) show the 60Gb/s pulse response and eye diagram measured at the input of the receiver frontend evaluation board. The total amount of ISI in the pulse response is calculated to be 1.54 times the cursor amplitude. With the oscillator injection locking enabled, a 10GHz clock generator (Keysignt E8267D) synchronized with a 30GHz transmitter clock source (Keysight E8257D) provides the injection clock with different phases for bathtub characterization. A Keysight 86130A BERT measures the BER of the reconstructed PRBS7 pattern from 1/128x sub-samplers, clocked by an external source (Keysight E8267D). Under

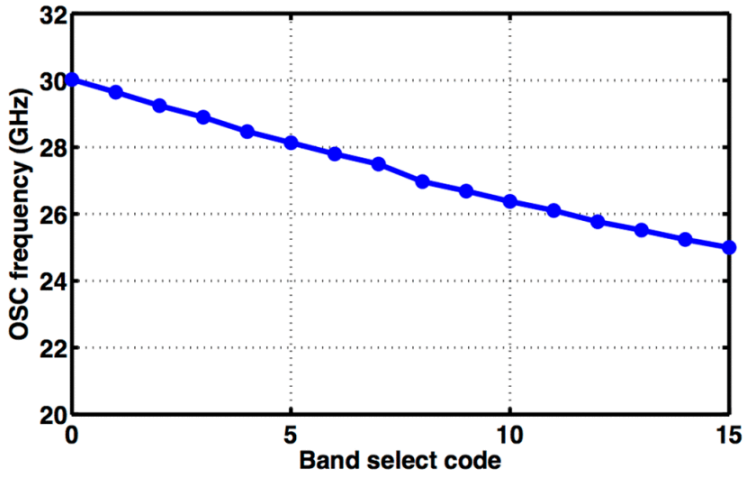


Figure 2.19: LC oscillator characterization curve.

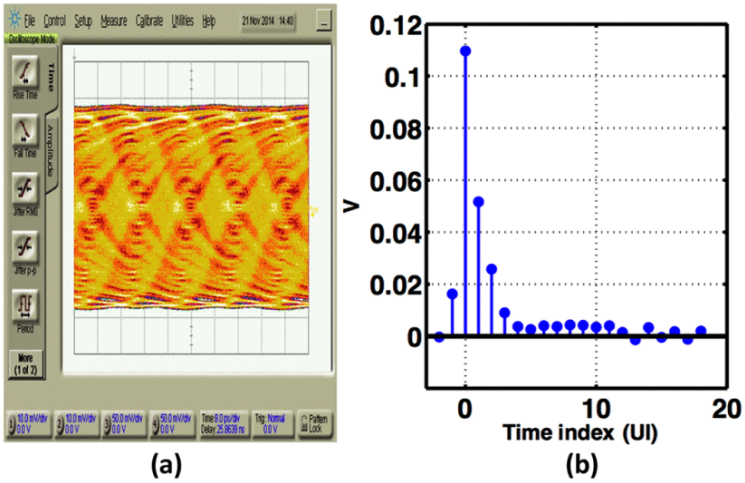


Figure 2.20: (a) Eye diagram at the channel output. (b) Estimated pulse response.

these conditions, the receiver frontend recovers the transmitted PRBS7 pattern and operates error free over 1e12 bits in the center region (Figure 2.21). The total power consumption of the design under these evaluation conditions is 173mW, 138mW from 1.2V, and 35mW from 1.0V supplies.

Table 2.1 compares the design with prior 56-80Gb/s equalizer designs. While incorporating the integrated the highest level of functionality/capability as compared to previous works, it achieves excellent power consumption in a mature 65nm technology. Specifically, despite the addition of CTLE and FFE, the equalizer core achieves nearly the same power consumption (48mW) as the previous 3-tap DFE design (46mW) [22]. The receiver frontend further supports 30GHz clock generation and distribution as well as the high-speed hard-

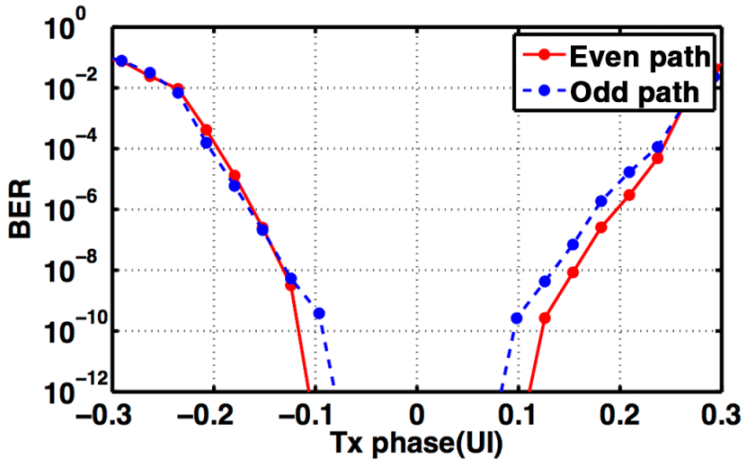


Figure 2.21: Bathtub-curve after equalization.

Table 2.1: Receiver Frontend Performance Summary

Reference	**[22]**	**[23]**	**[24]**	**This work**
Process	65nm CMOS	20nm CMOS	130nm SiGe	65nm CMOS
Data-rate (Gb/s)	66	56	80	60
Equalizer	3-tap DFE	Ext. 2-tap FFE CTLE 1-tap DFE	1-tap DFE	CTLE 2-tap FFE 3-tap DFE
VISI/VCURSOR or Channel loss (dB)	1.65	23dB	12dB	1.54
Power (mW)	46	177*	4000	173
Equalizer	46		1772◇	48
Deserializer	N/A		N/A	28
Clock generation	N/A	N/A	N/A	52†
Clock distribution	N/A		N/A	45
Efficiency (pJ/bit)	0.7	3.16	50	2.88

∗Includes equalizer, 4:16 DES, clock distribution.
◇Includes output buffer.
†LC oscillator + divider + PI.

ware necessary to support adaptive equalization and a baud-rate CDR, and despite being implemented in a substantially more mature process technology (65nm vs. 20nm), does so while dissipating slightly lower power and operating at slightly higher data rate than [23].

2.6 Conclusions

This chapter reports a 60Gb/s receiver frontend in 65nm CMOS technology. The receiver incorporates CTLE, FFE, and DFE equalizers, output slicing, and clocking blocks to demonstrate the possibility of serial link receiver operation operating near the frequency limits of 65nm CMOS technology. The energy-efficient equalization functionality is achieved by a current-integration technique combined by cascode gate voltage gain control for efficient RX FFE, passive clock delay between the current integrating and dynamic latch stages, and optimized reset device settling time/sizing. In addition to implementing the 60Gb/s equalizer, we focused on developing all high frequency data and clock paths for the complete receiver frontend operation, including interleaved deserializing slicers, an oscillator and clock dividers. The design achieves 60Gb/s operation with >0.2UI timing margin at 1e-9 BER (and error-free operation over 1e12 bits at the center of the eye) while consuming 173mW from 1.2V and 1.0V supplies in 65nm CMOS.

Chapter 3

Automatic Generation of 60Gb/s Equalizers

3.1 Overview

The receive equalizer is aimed to operate at 60Gb/s, near the speed limit of the 65nm CMOS technology. Therefore, it is expected that the equalizer performance is heavily affected by self loading and wiring capacitances (Figure 3.1), and the equalizer sizing will often require exhaustive parasitic extractions and post-layout simulations to be completed. Specifically, the equalizer frontend (Figure 2.2) is composed of multiple 30GHz signaling stages and feedback paths, and all transistors and wires involved with the high-speed paths need to be carefully designed to meet timing/bandwidth requirements for 60Gb/s operation. Usually, multiple iterations are involved with the design procedure.

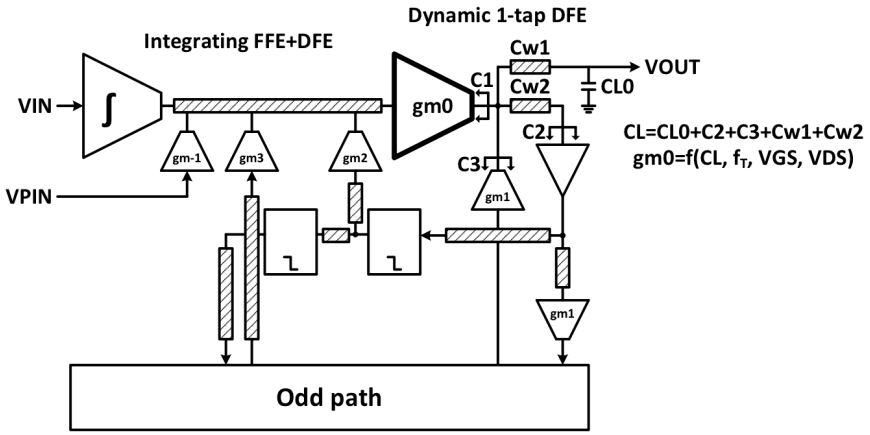


Figure 3.1: Wiring parasitic in DFE

However, when sizing and layout are done manually, even for the most advanced designs, going through multiple iterations for optimizing the sizing parameters requires substantial

effort and design time. As a result, high-speed link designs often rely on the designer's intuition and heuristics, and tend to be over-designed with high power consumption.

In order to expedite the flow and produce a more optimal design on time, this work adopts an automated design flow using the Berkeley Analog Generator (BAG). In [40] and [41], BAG is introduced as a tool that captures the designer's intent and methodologies, codifies them, and generates output designs for target specifications. This method is often called a generator-based design flow, since the designer will produce generators rather than instances, and the framework will create output instances by running the generators over multiple design specifications.

Compared to the traditional design methodology, the generator-based flow takes advantage of software-based approaches, such as reusability and reconfigurability. On top of that, since the generator code can run much faster than human-driven flows, designers can finish their tasks within a much shorter time period. Especially, feeding layout/parasitic parameters back to the sizing scripts (to capture post-layout effects) can be done automatically (which were previously done by human interactions between the circuit designers and the physical layout (or mask) designers) and greatly reduces the design time, as well as enables more iterations and verification to find optimal sizing parameters for the given specifications.

3.2 Berkeley Analog Generator (BAG)

The equalizer generator is built from the Berkeley Analog Generator (BAG), which consists of classes and functions for creating a schematic, layout, and testbench of the analog design, with extra helper functions that support the automation. Since the details of BAG are described in [40] and [41], this section briefly describes how BAG is structured, and the rest of the chapter focuses on explaining how to build the equalizer generator using BAG. The basic structure of BAG is shown in Figure 3.2. Generally, BAG takes 3 sets of parameters for the input-design specification, circuit topology, and device parameters.

- Design specification: items that describe the design to be generated in detail. Examples are gain, bandwidth, input/output range.
- Circuit topology: topological structure of the design, provided as a schematic template drawing.
- Device parameters: numerical parameters forming the characteristic and performance of devices in a certain technology. Usually provided as a lookup table (LUT).

By using those three classes of input variables and Python language, designers can build schematic, layout, and testbench generators, which produces generated schematics, layouts, testbenches (and related results), respectively. Those generated outputs can fed back to the generator code for further optimization or verification.

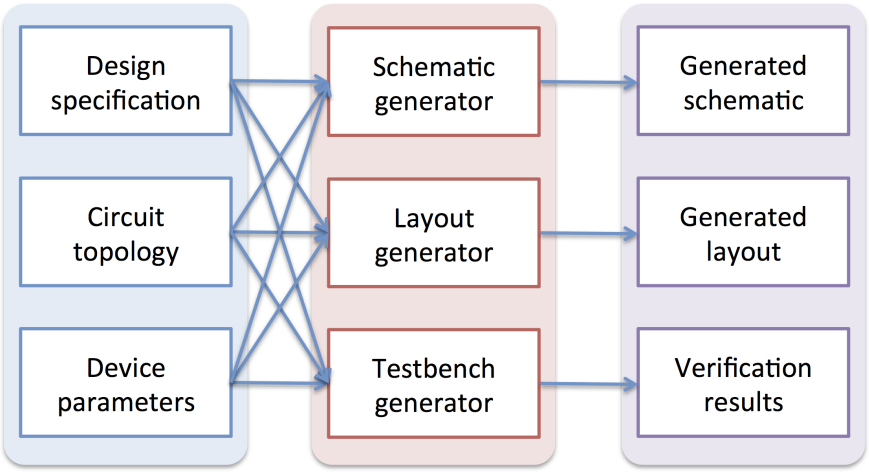


Figure 3.2: BAG Structure

It should also be noted that it is the designer's freedom to choose how to create such generators, since the BAG provides a framework and the API functions, rather than a pre-defined flow. Users can build their own generators for different preferences and applications in different ways.

The most complicated part is codifying the layout generator that generates a DRC/LVS clean layout. The initial version of the BAG utilizes PyCell technology[42] to implement the layout generator, and the PyCell provides a very convenient way of describing the layout with the design rules abstracted, especially for planar technologies.

3.3 Equalizer Generation

Basic Idea

While the details of the BAG framework is described in [40] and [41], this chapter focuses on explaining how BAG is utilized for high-speed link designs, and additional features are introduced on top of the basic framework. Especially, among various blocks in the high-speed link systems, we focused on automating the equalizer design because equalizers are located at the very beginning of the TX/RX high-speed signaling chain and operate at the highest frequency. Therefore, their pararasitic loading should be captured to avoid overdesign, and a significant portion of design efforts is spent on this task. Automating the process greatly expedites the design time and increases the chances to achieve a higher quality of results (QOR).

The automated flow is illustrated in Figure 3.3. As shown in the figure, the entire generation procedure is composed of running generator scripts of individual building blocks (with parameters derived in the initial step and previous generator scripts). The design

scripts of each block can be divided into three parts: sizing, schematic generation, and layout generation. In practice, block-level verification scripts are also run to check if each generation step is done correctly.

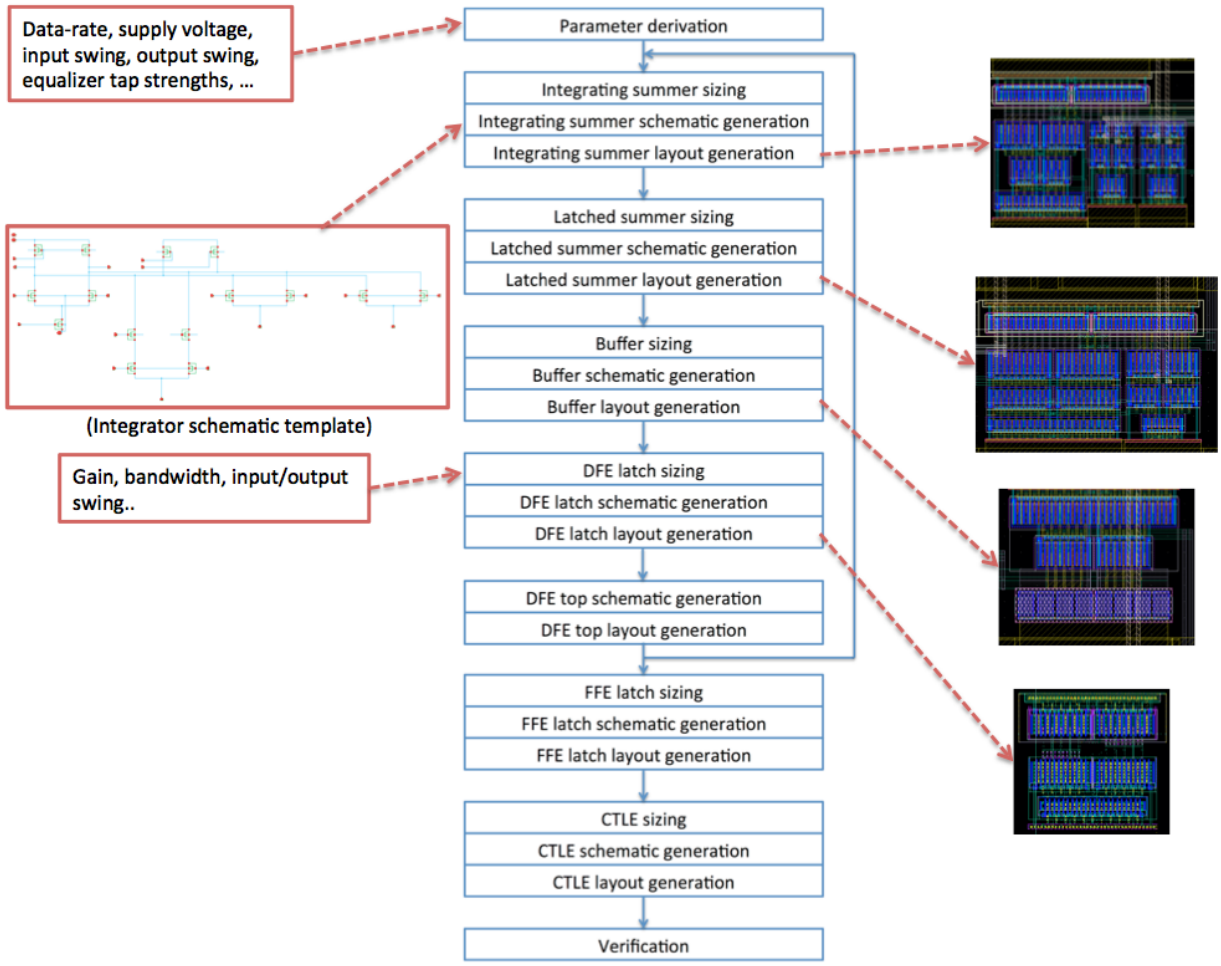


Figure 3.3: Equalizer generation flow

Noting that most stages in Figure 2.2 are linear, the equalizer system can be modeled as a consecutive chain of linear stages. As long as all the device and parasitic parameters are captured precisely, an equation based-sizing methodology[40] should work nicely, and the sizing scripts are written to find design values from a couple of simple equations.

One issue is that in many cases the device and parasitic parameters cannot be captured easily, especially from simulated results with schematic models. This is simply because in advanced processes, the geometry of devices heavily affects the performance of the transistor, which cannot be captured easily in schematic design phases. For example, the gate-source capacitance Cgs cannot be estimated only with schematics; actual layout needs to be generated

to extract its accurate gate capacitance. BAG handles this issue by storing the parameters of the unit-sized, extracted devices in a look-up table (LUT), and querying the parameters during the design phase [41]. Size_for() function is implemented in [41] for this purpose, and the function receives operating conditions (such as Vov, Vds, and Vds) of the transistors to be designed, as well as the device parameters (such as gm, cgg, and cdd). Since the LUT is consturcted from a post-extracted netlist of the unit transistor, the flow readily captures the device and parasitic parameters precisely.

By combining the parameters from the LUT with the design equations derived by the designers, the sizing parameters can be found. An example of the sizing process of the dynamic-latch (for FFE and DFE) is explained for reference, and the schematic is shown in Figure 3.4. Since it is a simple linear stage with a dominant pole in transparent phases, the design equation of the latch circuit is given by

$$I_{dynlatch} = \frac{A\omega V_{ov} C_L}{2(1 - \frac{\gamma A\omega}{f_T})} \tag{3.1}$$

where A is the gain, $\omega is the bandwidth$,V_{ov} is the overdrive voltage, C_L is the loading capacitance, γ is given by the ratio of the gate capacitance and the drain capacitance of the input pair, and f_T is the transit frequency of the input pair.

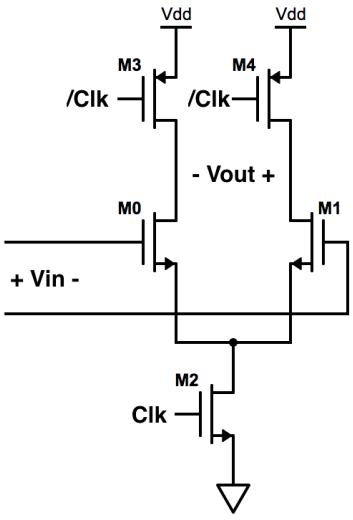


Figure 3.4: Dynamic-latch schematic

The key point is that the device parameters (such as V_{ov}, γ, and f_T) for the given operating conditions are found by fetching parameters from the preconstructed LUT [41].

Once $I_{dynlatch}$ is chosen, the sizes (widths) of the transistors can be found by

$$W = \frac{I_{dynlatch}}{Id_{unit}} W_{unit} \tag{3.2}$$

Where Id_{unit} is the drain current of the unit device that was used for building the LUT and W_{unit} is the width of the unit device. Since all those steps can be described in a formal way, its sizing script can be easily implemented.

The resulting pseudo code of the sizing script is as follows.

```
def size_dynlatch(vin, vout, vckn, vckp, vdd,
                  A, wbw, vov_in, CL):
    nin_db=nmos.size_for(vd=vout, vg=vin, vb=0, vov=Vov_in)
    ncs_db=nmos.size_for(vd=nin_db.vs, vg=vckn, vs=0, vb=0)
    pld_db=pmos.size_for(vd=vout, vg=vckp, vs=vdd, vb=0)
    ibias=A*vov_in*CL/(2*(1-nin_db.gamma*A/nin_db.ft))
    win=wunit*ibias/nin_db.id
    wcs=wunit*ibias/ncs_db.id
    wld=wunit*ibias/(-pld_db.id)
```

After finding out the sizes of all the devices, the sizing parameters are passed through the schematic and layout generation scripts. For the schematic generation, the generator calls update_lib_instance(), as described in [40], to produce a sized schematic by combining the schematic template with the sizing parameters. For the layout, PyCell technology can be utilized for generating a DRC clean-by-construction layout. An example pseudo code is shown below.

```
def generate_layout_dynlatch(l, win, wckn, wckp, nfin, nfckn, nfckp):
    min0=place_nmos(l=l, w=win, nf=nfin, xy=[0, 0])
    min1=place_nmos(l=l, w=win, nf=nfin, reference=min0, from=Right)
    mckn0=place_nmos(l=l, w=wckn, nf=nfckn, reference=[min0, min1], from=Bottom)
    mckp0=place_pmos(l=l, w=wckp, nf=nfckp, reference=min0, from=Top)
    mckp1=place_pmos(l=l, w=wckp, nf=nfckp, reference=min1, from=Top)
    generate_rail(VDD, VSS)
    connect(min0.drain, mckp0.drain) #out_m
    connect(min1.drain, mckp1.drain) #out_p
    connect(min0.source, mckn0.drain) #tail
    connect(min1.source, mckn0.drain) #tail
    connect(mckn0.drain, VSS)
    connect(mckp0.drain, VDD)
    connect(mckp0.drain, VDD)
```

The actual implementation of place_nmos, place_pmos, generate_rail, and connect functions can be done by running the primitive PyCell API (assuming that the iPDK and PyCell setups are supported) [42].

Parameter Searching and Interpolation

From the sizing flow, which was explained in the previous subsection, one can easily find that the performance and usability of the flow heavily depends on the device parameter being searched. While the size_for() function provides the core functionality for this, additional features have to be implemented for a further improvements. For example, the size_for() can receive potential differences between the transistor terminals (such as vgs, vds, vbs), but in many cases, those voltage differences are implicit or related to each other, and as a result, it is hard to figure out the arguments for running the size_for() function. In the dynamic-latch example (Figure 3.4), the tail voltage of the latch (which is the source voltage of the input devices) is unknown, whereas the gate and drain voltages can be found from the specification (Figure 3.5). This is because it is more natural to have the overdrive voltage of the input devices for the dynamic-latch design (which sets the maximum input voltage swing), as the source voltage will be indirectly determined by the overdrive voltage. Therefore, there should be an intermediate step that find out the potential differences (vgs, vds, vbs) from the absolute potentials (vg, vd, vov) given by the specifications.

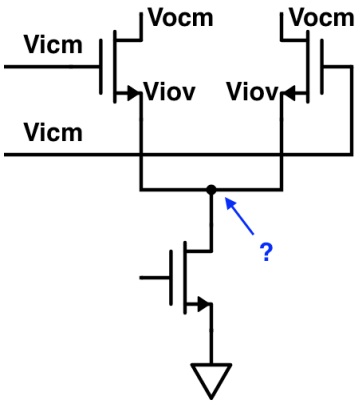


Figure 3.5: Tail structure of the dynamic-latch

In this work, a function called finddevparam() is implemented to serve for the intermediate layer. As shown in Figure 3.6, the finddevparam() receives the operating conditions and searching method for the device, and invokes the actual size_for() function with a parameter sweep to find the device parameters with the implicit conditions given.

More specifically, for the input devices of the dynamic-latch, the input and output common modes (v_{icm} and v_{ocm}) are given as the gate and drain voltage of the devices along with its overdrive voltage v_{iov}, while its source voltage is set to be an unknown with a sweep range. Then, the finddevparam() sweeps the source voltage and queries the size_for() function with $v_{ov} = v_{iov}$, $v_{ds} = v_{ocm} - v'_s$, $v_{bs} = -v'_s$, while the v'_s is swept from 0 to VDD. After that, the finddevparam() picks a point that gives $v_g = v_{icm}$ and that corresponds to the operating point of the input devices.

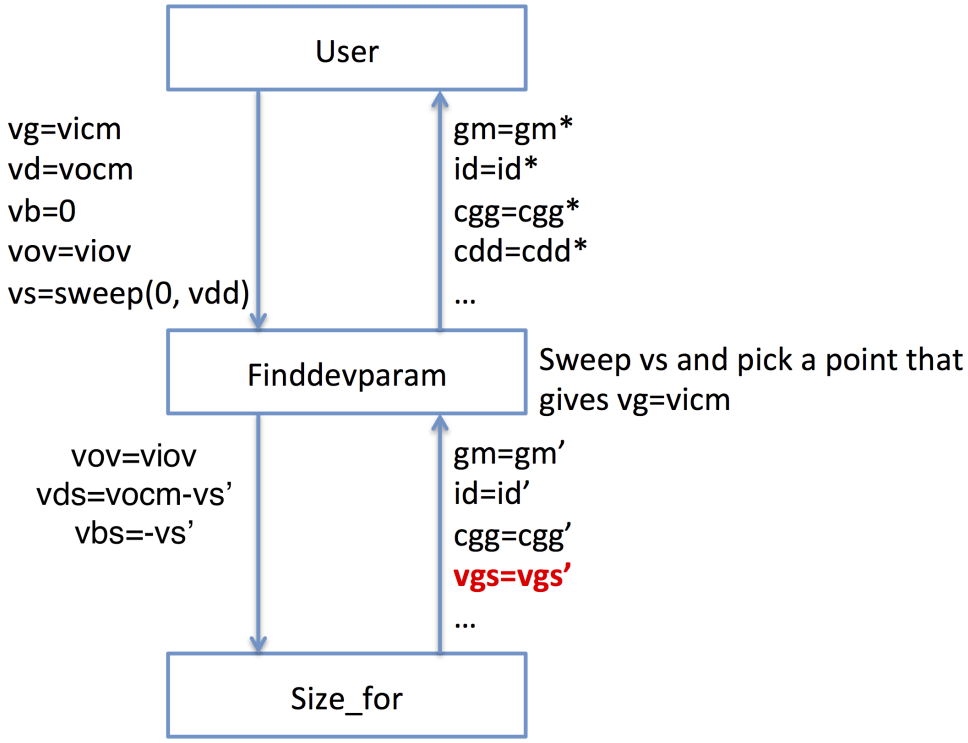


Figure 3.6: Device parameter searching method with implicit input parameters

The following pseudo code describes the finddevparam() function that supports this parameter searching feature.

```
def finddevparam(vg=None, vd=None, vs=None, vb=None, vov=None):
    params={'vg':vg, 'vd':vd, 'vs':vs, 'vb':vb, 'vov':vov):
    if vs==None: #vs sweep
        for vs in range(0, VDD, 0.001): #1mV sweep
            vds=vd-vs; vbs=vb-vs; vov=vov
            _dev_sweep=size_for(vds=vds, vbs=vbs, vov=vov)
            if _dev.vgs+vs ~= vg: #found the matched point
                _dev=_dev_sweep
    elif vg==None:
        vds=vd-vs; vbs=vb-vs; vov=vov
        _dev=size_for(vds=vds, vbs=vbs, vov=vov)
    elif vd==None:
        vgs=vg-vs; vbs=vb-vs; vov=vov
        _dev=size_for(vgs=vgs, vbs=vbs, vov=vov)
    elif vb==None:
        vgs=vg-vs; vds=vd-vs; vov=vov
        _dev=size_for(vgs=vgs, vds=vds, vov=vov)
```

```
elif vov==None:
    vgs=vg-vs; vds=vd-vs; vbs=vb-vs;
    _dev=size_for(vgs=vgs, vbs=vbs, vds=vds)
dev=_dev*W/Wunit #since size_for returns parameters for unit devices, the
    parameters should be scaled up by W/Wunit
return dev
```

Another issue regarding the parameter extraction is the finite resolution of the LUT. If the resolution of the LUT is not high enough, the size_for function fails to return output parameters or the output parameters are not accurate enough. In order to resolve this, parameters are interpolated from the 2 closet values in the sweep set, using the interp() function in numpy.

Circuit Sizing Flow

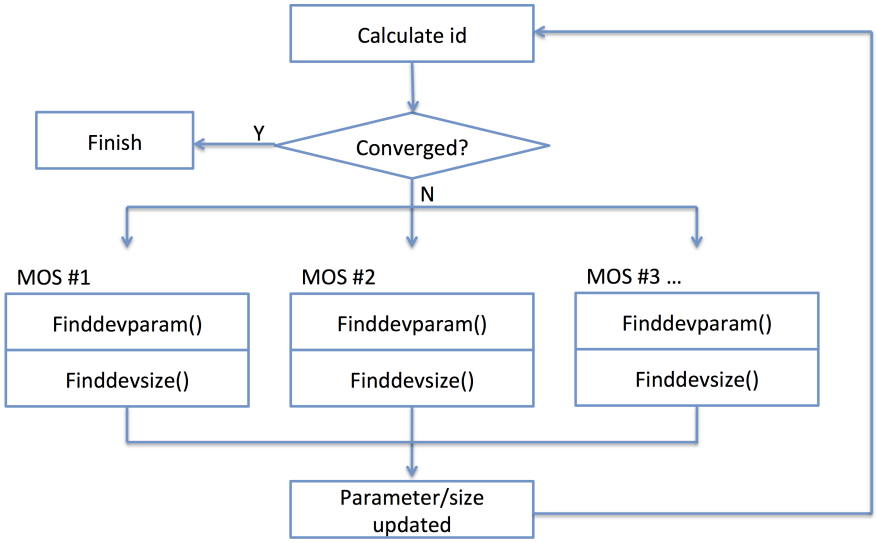


Figure 3.7: Iterative sizing flow

Once the device parameters are found, the device can be sized for target drain currents. For example, if the drain currents of the current device (used for finddevparam()) is i_d and the target current (found by a design equation) is $i_{d,target}$, the size of transistor is given by

$$W_{new} = \frac{I_{d,target}}{I_d} W_{old} \tag{3.3}$$

where W_{old} is the width of the transistor used for finddevparam(). The finddevsize() is implemented for this purpose. Those two functions compose the basis of the sizing scripts, and most of the sizing process can be done by calling the finddevparam() and finddevsize(),

repeatedly. Figure 3.7 is the formalized way for sizing an individual building block of the equalizer. In this way, sizing all the building blocks can be implemented in an unified way.

Simulation-based Sizing

While the equation-based scripts can achieve a higher accuracy by using the techniques introduced (extracting parameters from LUT, searching, and interpolation), there are some cases when the equation-based approach doesn't provide the best results, especially when there are nonlinear/nonideal effects that heavily affect the circuit behavior and increase the complexity of the design equations. In that case, an alternative sizing approach using simulations can be considered, as BAG also provides functions to launch simulations with testbenches automatically. The simulation-based sizing approach used for this work is illustrated in Figure 3.8. As shown in the figure, the sizing can be adjusted until the simulation results meet the target specifications, as shown in Figure 3.9.

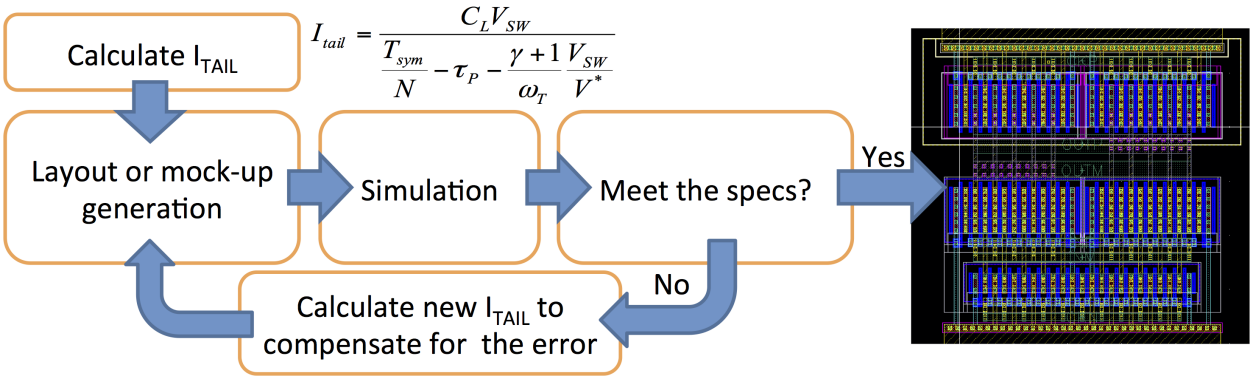


Figure 3.8: Simulation-based sizing

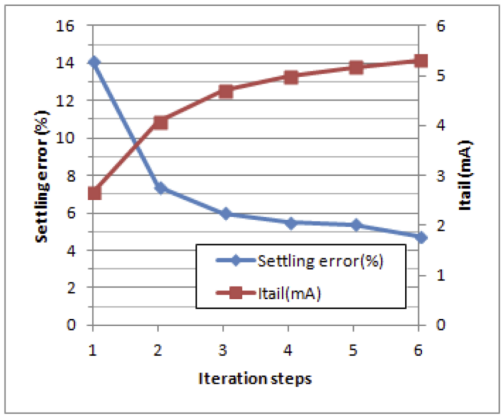


Figure 3.9: Number of iterations vs. simulated settling errors

3.4 Generator Results and Conclusions

By running the design scripts for the individual building blocks of the equalizer, the equalizer is generated, with sizings optimized for the given specifications. The generated layout of the DFE half circuit is shown in Figure 3.10.

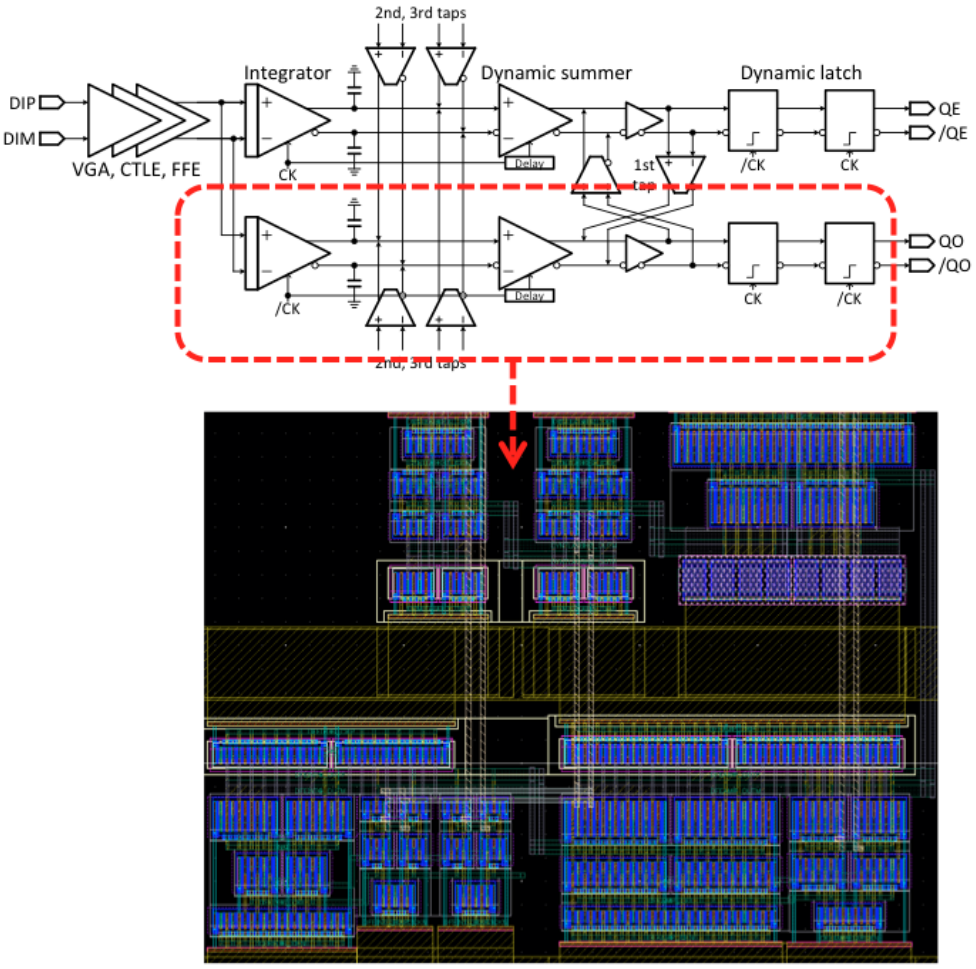


Figure 3.10: Generated DFE layout

As explained earlier, the layout is implemented using the PyCell API [42]. The benefit of optimization is analyzed by comparing the power consumption of the generated design with the previous 66Gb/s DFE [22], with the results summarized in Figure 3.11.

As shown in the figure, the power consumption of the DFE is reduced from 46mW to 32mW, which allows additional power budgets for the CTLE and the FFE implementation. While part of the power reduction comes from using the current integrating summation, the automation also contributes to the improvements by optimizing the sizing parameters

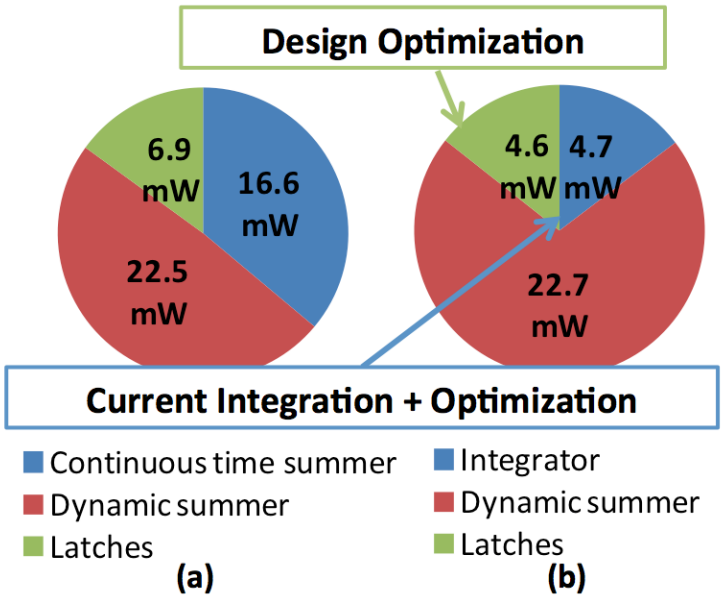


Figure 3.11: DFE power consumption comparison between the (a) reference design and (b) generated design

more than the previous manual design. On top of the power reduction, the generator greatly expedites the design flow (<1hr running time), which makes the design very flexible to any changes in specification and technology. This feature is very advantageous especially for producing designs in advanced technology, since a significant amount of effort is spent on handling parameter changes and parasitic effects for advanced CMOS processes.

Chapter 4

60Gb/s Baud Rate CDR

4.1 Clock Generation and Distribution

Clocking circuits need to be implemented on top of the signaling and equalization path for full transceiver operation (Figure 4.1). In high-speed serial links, 3 kinds of clocking circuits are used; clock generation, distribution, and CDR. For clock generation, an LC oscillator is utilized to generate a differential pair of 30 GHz clocks, and an auxiliary differential pair for 3x injection-locking for bathtub curve measurement.

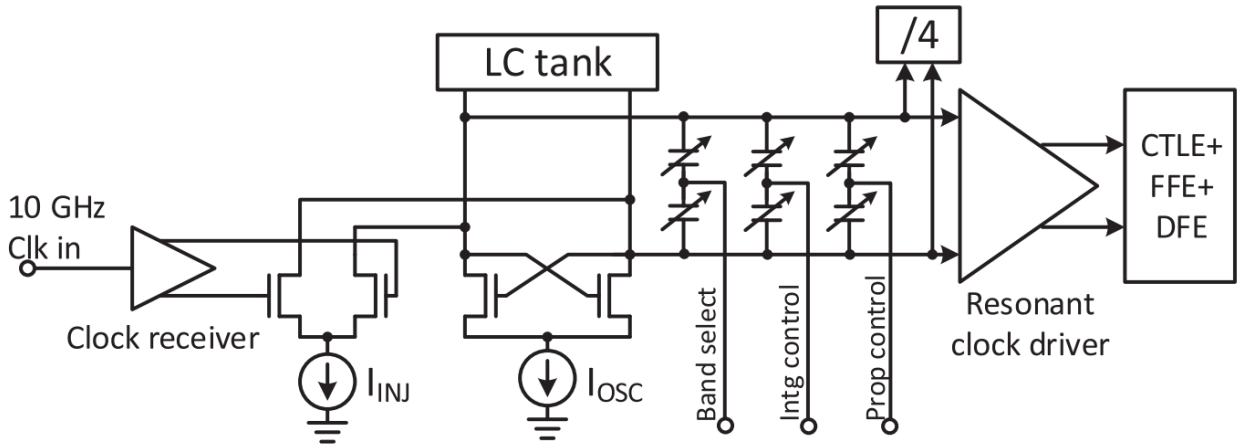


Figure 4.1: Clock generation and distribution circuits

The differential 30 GHz clocks need to be buffered before driving the clock ports of the equalizer frontend, which can consume substantial power at these high frequencies [23]. A resonant clock buffer (Figure 4.2(a)) is therefore utilized to provide clock signals to the equalizer frontend through an AC-coupled clock-distribution network with proper bias settings, as proposed in [22]. The clock buffer is a differential stage with a differential on-chip balun that resonates with its capacitive load from the rest of the network. A tunable capacitor array is

inserted to control the resonance frequency, as in the oscillator circuitry. Although these LC oscillator and clock buffer require inductors (baluns), the area occupied by these inductors is mitigated by the high operating frequency. Specifically, in this design, the inductor for LC oscillator and clock buffer occupies only 56 um × 56 um and 53 um × 53 um, respectively.

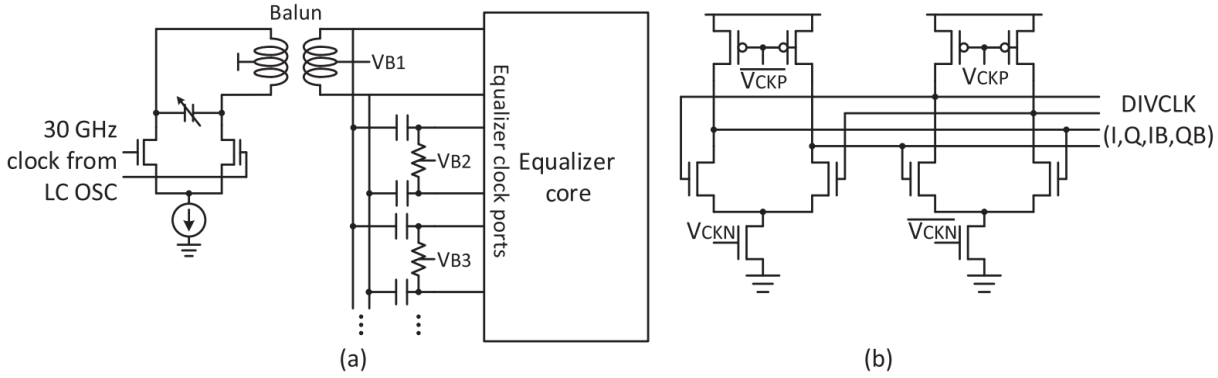


Figure 4.2: (a) Resonant clock buffer with clock distribution network. (b) 30 GHz clock divider

Subrate clocks have to be generated as well for the interleaved sampler and back-end functions, such as deserialization, adaptation, and CDR. The same dynamic-latch design used in the DFE feedback paths is reused to implement the 30 GHz clock divider (Figure 4.2(b)). Due to placement considerations and improved distribution of the capacitive-loading between the oscillator and the output buffer, the divider's input clock is driven directly by the oscillator rather than by the clock buffer. Following the 30 GHz divider, a phase interpolator is inserted to enable optimal placement of the deserializing clock phases.

4.2 Baud-rate CDR Design Considerations

The most commonly adopted baud-rate CDR design is based on the Mueller-Muller CDR (or MMCDR) [43]. While the MMCDR scheme will find a locking point that represents equal levels of precursor and first tap post cursor ISI, it is not guaranteed that this condition corresponds to an optimum point, or even to a unique point. This issue is denoted as phase wandering. In other words, an MMCDR relies on specific characteristics in the pulse response, and these conditions cannot be guaranteed over variations in channel environment or process. Consider for example the extreme case where the input pulse is a perfect square wave; the first postcursor tap will be equal in level to the first precursor tap over the entire width of the pulse (i.e., the entire UI), and hence the phase selected by the MMCDR is effectively free to wander anywhere (i.e., the design would not operate robustly). To avoid these kinds of issues, we set out to realize a baud-rate CDR with a unique locking point, and we describe our proposed approach in the following section.

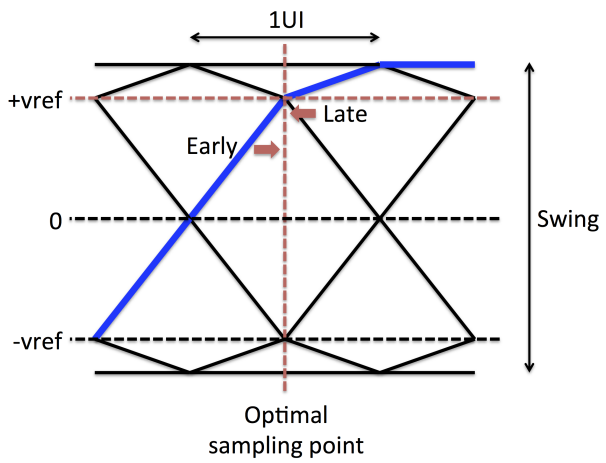


Figure 4.3: MMCDR phase detection scheme

4.3 Proposed Baud-rate CDR and Integration-and-reset Frontend

This design proposes to augment traditional baud-rate CDRs by utilizing an integration-and-reset frontend to create a unique locking point for the CDR. If the received signal is filtered by an integration-and-reset frontend, the frontend output will be maximized when the integration window perfectly overlaps with the incoming pulse, which translates to a single locking point, even when the input pulses have wide flat regions.

To demonstrate how the integrate-and-reset front-end helps to address the issue of non-unique locking, let's consider the expected value of the output of the front-end y(t) given a perfectly square input pulse (ranging from 0 to some amplitude):

$$E[y(t)] = \begin{cases} (V_a/T) \cdot [(t - T_0) + T] & \text{if } T_0 - T < t < T_0 \\ (V_a/T) \cdot [(T + T_0) - t] & \text{if } T_0 < t < T + T_0 \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where Va is the amplitude, T is the bit period, and T_0 represents the phase of the incoming pulse relative to the receiver. This equation (4.1) implies that the average output of the integration-and-reset stage has a triangular output waveform, which gives only one single peak at $t = T_0$. It also indicates that the integrator output offers the maximum sampling margin at phase T_0, since this is the point where the sampler will see the peak output voltage.

Figure 4.4 visually depicts the integration and reset operation. Noting that the integration happens from t to t+T and the sampling is done at t+T, there is no need for additional clock phases with T/2 spacing (i.e., the receiver is operating at baud-rate). Furthermore, in our receiver design, we already have the integration-and-reset stage in the frontend CTLE

[27][28] (which is denoted as intg0 in Figure 2.2); therefore, no additional hardware is required for this operation.

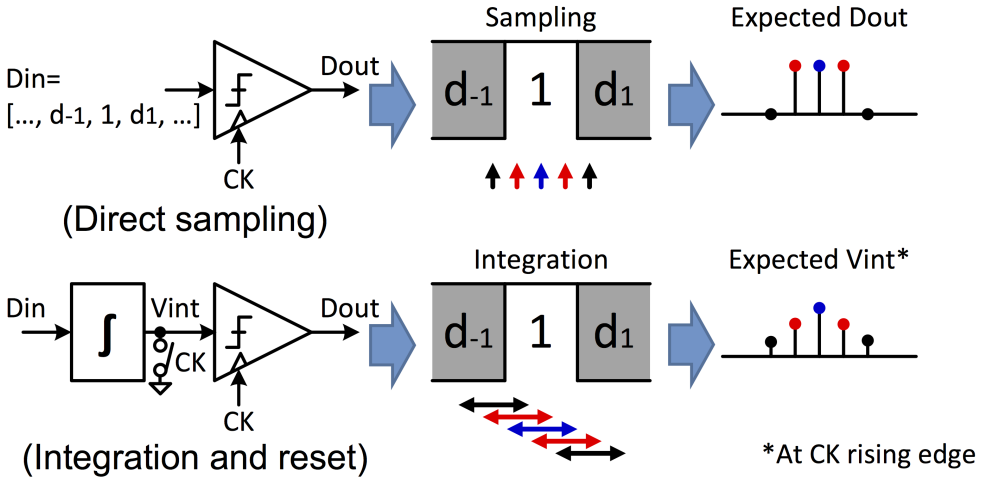


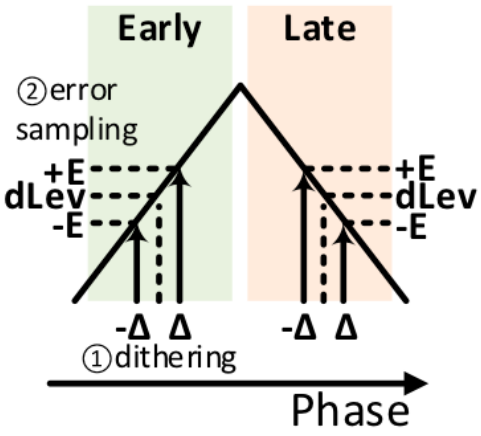
Figure 4.4: Integration and reset frontend behavior

The discussion above points to the idea that once we have applied the front-end integration and reset, we should implement the CDR so that it converges to the peak point of the (post integration) waveform. In order to lock to the phase at which the integrator reaches its peak expected output value, a signal that is highly correlated to the output amplitude is necessary as an indicator. Fortunately, the data level (dLev) tracking loop that is typically used for equalizer adaptation [37] can serve for this purpose as well.

The most straightforward way to implement the baud rate CDR using the ideas outlined above would be to adjust the receiver's phase setting (possibly in both directions), measure the cursor amplitude from the dLev output, and then move the phase in the direction that achieved a higher dLev value. However, since the CDR loop would then need to wait until the dLev loop converges and produces a correct value, this approach would limit the phase tracking bandwidth to be substantially slower than the dLev tracking bandwidth. Unfortunately, the dLev loop bandwidth is usually set to be very low in order to reject residual ISI.

Instead, the approach taken in this design decouples the CDR tracking bandwidth from the data level tracking bandwidth by directly using the output of error (adaptive) samplers (which from the standpoint of the dLev loop, represents the change in the data level) rather than the output of dLev tracking loop. Specifically, in order to find the setting with the maximum (post-integrator) data level, the integration and sampling phase is dithered by a controlled amount, and then the error sampler output is correlated with the digital dither signal (Figure 4.5(a)).

This approach achieves the desired result because the sign of the error sampler's output will be opposite to that of the dithering sequence if the integration phase is late, and the signs



Dithering	Error sampler output	PD output
-Δ	0	Early
Δ	1	Early
Δ	0	Late
-Δ	1	Late

(b)

Figure 4.5: (a) Phase dithering and error sampling for phase detector implementation (b) truth table of proposed phase detector

will be aligned if the integration phase is early. In other words, the phase detector's output is found by XOR'ing the error sampler output with the dithering sequence (Figure 4.5(b)). Additional data filtering is applied on top of the XOR operation, since the dLev tracking loop is configured to be activated by +1 symbols (and filtering out -1 symbols) in order to reduce the number of error samplers [37]. However, it should be noted that this data filtering does not degrade the pattern coverage of the proposed CDR scheme, since it is guaranteed that there is at least one +1 symbol whenever there is an NRZ data transition.

In order to elucidate how design decisions (such as dithering amplitude) should be made for this type of CDR, as well as what its limitations/tradeoffs are, we will next derive a small signal model for the baud-rate phase detector as a function of the input jitter (as well as the dither amplitude). First, the CDR should lock to the phase where the average of the phase detector output is zero, which can be expressed as the following condition:

$$P(E) = P(L) \tag{4.2}$$

where P(E) and P(L) are the probabilities of having Early and Late outputs from the phase detector, respectively. In the locked condition, the dLev loop will converge to the value that meets the following equation:

$$P(e) = P(-e) \tag{4.3}$$

where P(+e) and P(-e) are the probabilities of having positive error (+e in Figure 4.5(a)) and negative error (e in Figure 4.5(a)) outputs from the error sampler, respectively. In the

presence of the phase dither for the CDR [1], the probabilities in (4.2) and (4.3) are given by:

$$P(E) = P(E|-\Delta)\cdot P(-\Delta) + P(E|\Delta)\cdot P(\Delta) \tag{4.4}$$
$$P(L) = P(L|-\Delta)\cdot P(-\Delta) + P(L|\Delta)\cdot P(\Delta) \tag{4.5}$$

$$P(+e) = P(+e|-\Delta)\cdot P(-\Delta) + P(+e|\Delta)\cdot P(\Delta) \tag{4.6}$$
$$= P(L|-\Delta)\cdot P(-\Delta) + P(E|\Delta)\cdot P(\Delta) \tag{4.7}$$

$$P(-e) = P(-e|-\Delta)\cdot P(-\Delta) + P(-e|\Delta)\cdot P(\Delta) \tag{4.8}$$
$$= P(E|-\Delta)\cdot P(-\Delta) + P(L|\Delta)\cdot P(\Delta) \tag{4.9}$$

where $P(E|\Delta)$ and $P(L|\Delta)$ are the conditional probabilities of having Early and Late outputs from the phase detector when the integration phase is dithered by Δ, respectively. Since $P(E|\Delta) = 1 - P(L|\Delta)$, $P(E|-\Delta) = 1 - P(L|-\Delta)$, and $P(\Delta) = P(-\Delta) = 0.5$ (assuming that the dither is balanced), combining (4.4)-(4.8) gives the following locking condition:

$$P(E|-\Delta) = P(L|-\Delta) = P(E|\Delta) = P(L|\Delta) = 0.5 \tag{4.10}$$

This condition is illustrated in Figure 4.6(a), with the converged dLev (i.e., $dLev_0$) indicated by the thick dashed line. If the jitter follows a Gaussian distribution with standard deviation σ_j (and its probability density function is p(t), as in Figure 4.6(a)), then as shown in the figure, the phase τ_0 at which the integrator output Y(t) is equal to $dLev_0$ (i.e., $Y(\tau_0) = dLev_0$) can be found by:

$$Q(\frac{-\tau_0+\Delta}{\sigma_j}) - Q(\frac{\tau_0+\Delta}{\sigma_j}) = Q(\frac{-\tau_0-\Delta}{\sigma_j}) - Q(\frac{\tau_0-\Delta}{\sigma_j}) = 0.5 \tag{4.11}$$

Figure 4.7(a) shows the relationship between Δ/σ_j and τ_0/σ_j. Note that as Δ increases, τ_0 converges to Δ (and $dLev_0$ approaches to $Y(\Delta)$).

Let us next examine the case where the input data's phase has been temporarily shifted by τ; this will change the integrator's output waveform, but due to its (intentionally) low tracking bandwidth, dLev will remain stationary at $dlev_0$ (Figure 4.6(b)). In this case the probabilities become:

$$P(E|\tau) = P(E|-\Delta,\tau)\cdot P(-\Delta) + P(E|\Delta,\tau)\cdot P(\Delta) \tag{4.12}$$
$$= \frac{1}{2}[1 - Q(\frac{-\tau_0-\tau+\Delta}{\sigma_j}) + Q(\frac{\tau_0-\tau+\Delta}{\sigma_j}) + Q(\frac{-\tau_0-\tau-\Delta}{\sigma_j}) - Q(\frac{\tau_0-\tau-\Delta}{\sigma_j})] \tag{4.13}$$

[1] Note that in order to simplify the analysis, we do not explicitly include here the effects of random voltage variations (which would introduce dependence on the exact shape of the integrator output). However, the measured jitter tolerance results presented later match fairly closely (within 20%) with the predictions from the simplified analysis.

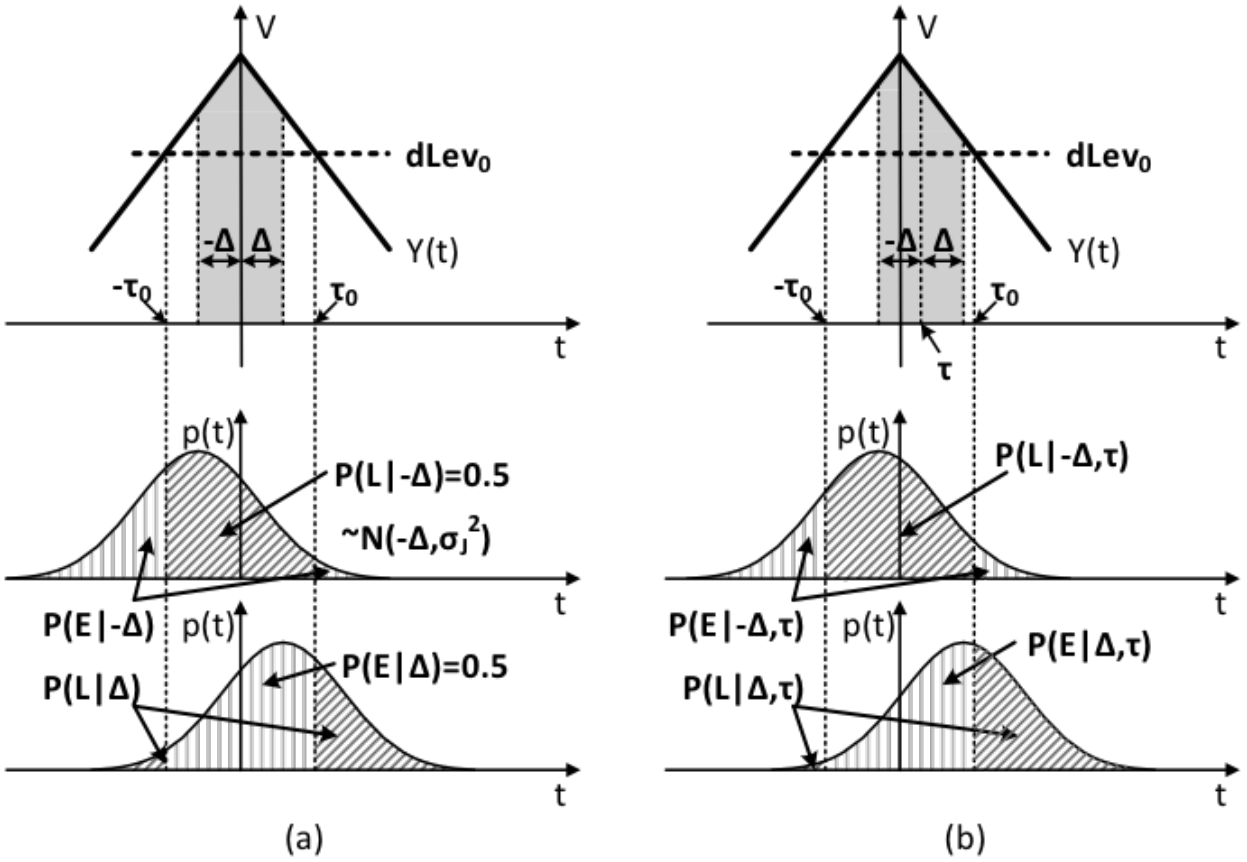


Figure 4.6: Phase detector statistics under dither (Δ), random jitter (σ_j), and input phase (τ), (a) in the locking condition (b) when there is a phase shift by τ

$$P(L|\tau) = P(L|-\Delta,\tau)\cdot P(-\Delta) + P(L|\Delta,\tau)\cdot P(\Delta) \tag{4.14}$$

$$= \frac{1}{2}[Q(\frac{-\tau_0-\tau+\Delta}{\sigma_j}) - Q(\frac{\tau_0-\tau+\Delta}{\sigma_j}) + 1 - Q(\frac{-\tau_0-\tau-\Delta}{\sigma_j}) + Q(\frac{\tau_0-\tau-\Delta}{\sigma_j})] \tag{4.15}$$

The expected value of the phase detector output is thus:

$$PDout = TD\cdot FD\cdot[P(E|\tau) - P(L|\tau)] \tag{4.16}$$

$$= TD\cdot FD\cdot[Q(\frac{\tau_0-\tau+\Delta}{\sigma_j}) - Q(\frac{-\tau_0-\tau+\Delta}{\sigma_j}) + Q(\frac{-\tau_0-\tau-\Delta}{\sigma_j}) - Q(\frac{\tau_0-\tau-\Delta}{\sigma_j})] \tag{4.17}$$

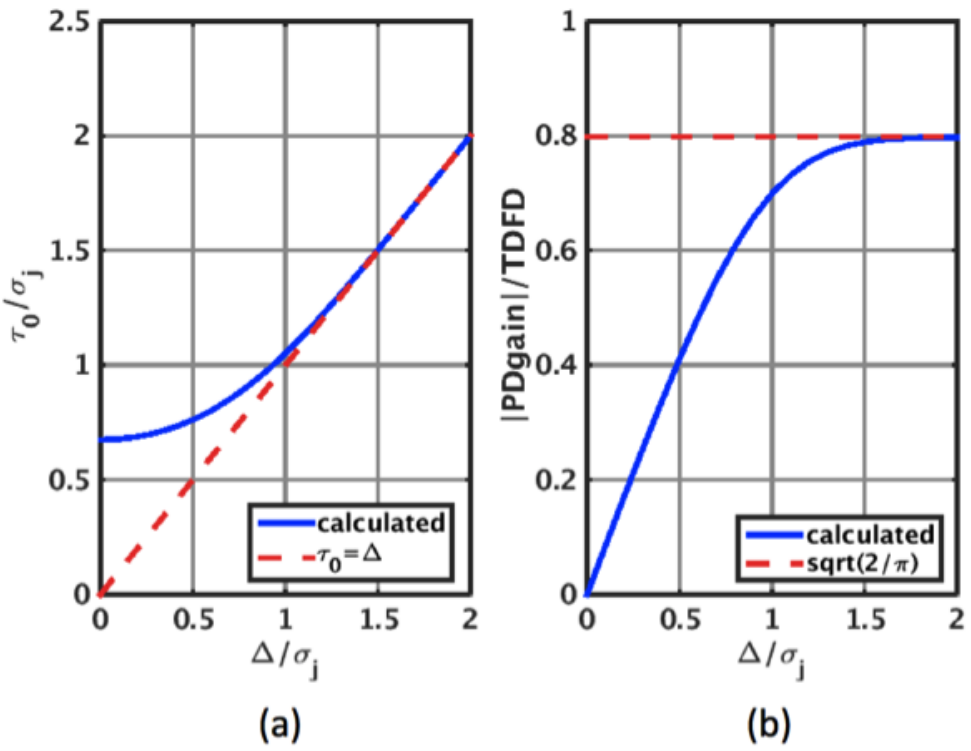


Figure 4.7: (a) Relationship between Δ/σ_j and τ_0/σ_j in (9) (b) phase detector gain from (14)

where TD is the transition density (0.5 for PRBS), and FD is the data filtering density (0.5 for +1 filtering). The small signal PD gain is therefore given by finding the derivative of (4.16) at =0, which for Gaussian input jitter evaluates to:

$$PDgain = TD \cdot FD \cdot \sqrt{\frac{2}{\pi}}(e^{-\frac{(\Delta+\tau_0)^2}{2\sigma_j^2}} - e^{-\frac{(\Delta-\tau_0)^2}{2\sigma_j^2}}) \tag{4.18}$$

If the dithering amplitude Δ increases, τ_0 approaches Δ (as indicated in Figure 4.7(a)), and (4.16) and (4.18) can be approximated as:

$$PDout = TD \cdot FD \cdot [P(E|\tau) - P(L|\tau)] \tag{4.19}$$

$$= TD \cdot FD \cdot [Q(\frac{-\tau + 2\Delta}{\sigma_j}) + Q(\frac{-\tau - 2\Delta}{\sigma_j}) - 2Q(\frac{-\tau}{\sigma_j})] \tag{4.20}$$

$$PDgain = TD \cdot FD \cdot \sqrt{\frac{2}{\pi}}(e^{-\frac{2\Delta^2}{\sigma_j^2}} - 1) \tag{4.21}$$

(4.16)-(4.21) reveal the dependence of the CDR's loop gain (and hence tracking bandwidth) upon the dithering amplitude Δ. Specifically, from (4.18), a higher Δ gives a larger

phase detector gain (also as shown in Figure 4.7(b)), and from (4.16), the phase detector output diminishes to zero as the instantaneous phase shift becomes much larger (by 3X) than the dithering amplitude Δ (because all Q values in (4.16) converge to one). However, increasing Δ gives a narrower horizontal eye margin due the (intentional) pseudo-randomized dithering operation. In this work, 2% dithering amplitude is selected to have a negligible impact on the horizontal eye opening, while still supporting MHz range[2] tracking capability.

4.4 Control Loop Implementation

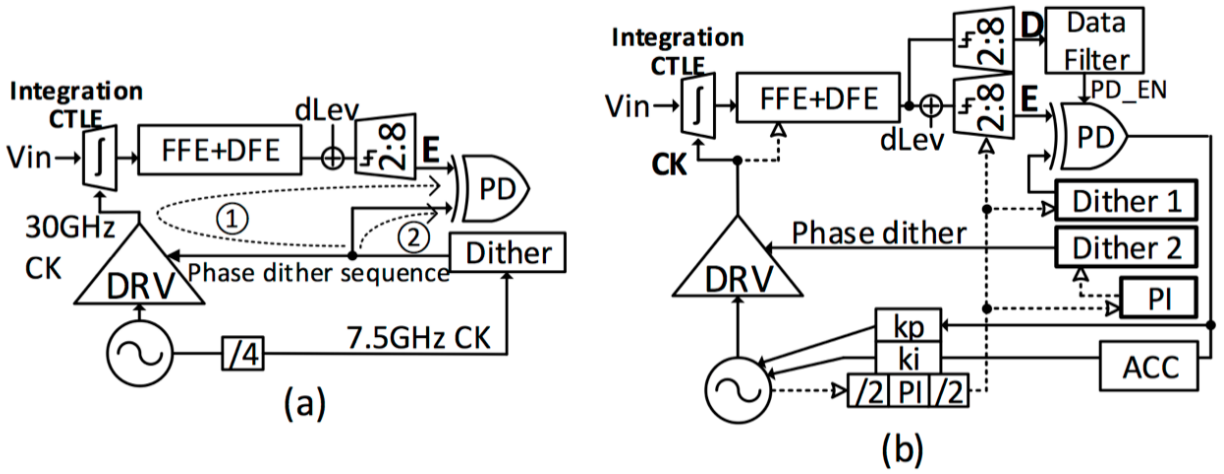


Figure 4.8: (a) Intrinsic path delay difference between the received signal modulated by the dither sequence (1) and the dither sequence itself (2) (b) Proposed CDR architecture to compensate the path mismatch by using 2 separate dither generation and a phase interpolator

Figure 4.8(a) shows the signaling and clocking hardware associated with the dithering and phase detector operation. Note that there is a latency difference between the two inputs to the phase detector; the dithering sequence (1) and the signal modulated by the dithering sequence (2).

Separate dither generators that produce the same sequence but with a time offset are used to compensate for the latency difference. Specifically, one dither generator is input directly to the phase detector, while the other modulates the actual phase used in the high-speed data-path. Altering the seeds of these two dither generators (which are just PRBS sequences) enables alignment with step sizes of 8-UI, and shown in the figure, an additional

[2] For practical values of dither amplitude Δ, the achievable absolute bandwidth of this type of CDR will be lower than e.g. a 2x over-sampled design. However, since most 50+Gb/s standards continue to require CDR tracking bandwidth only in the MHz range which is easily achievable at these sample rates and with reasonable values of Δ sacrificing the maximum achievable CDR bandwidth for energy-efficiency is typically a favorable trade-off.

phase interpolator enables finer 1-UI steps within this 8-UI window. If there is x UI of timing mismatch between these two path, the effective gain of the phase detector will be scaled by a factor of x/8 (since the dither generators are triggered by the 8-UI period/7.5GHz clock); the 1-UI step phase interpolator therefore limits the reduction in phase detector gain to 12.5%.

As shown in Figure 4.8(b), the CDR loop is then closed by applying additional filtering and feeding the signal back to control the clock generator. The resulting CDR and clocking path hardware are shown in Figure 4.9.

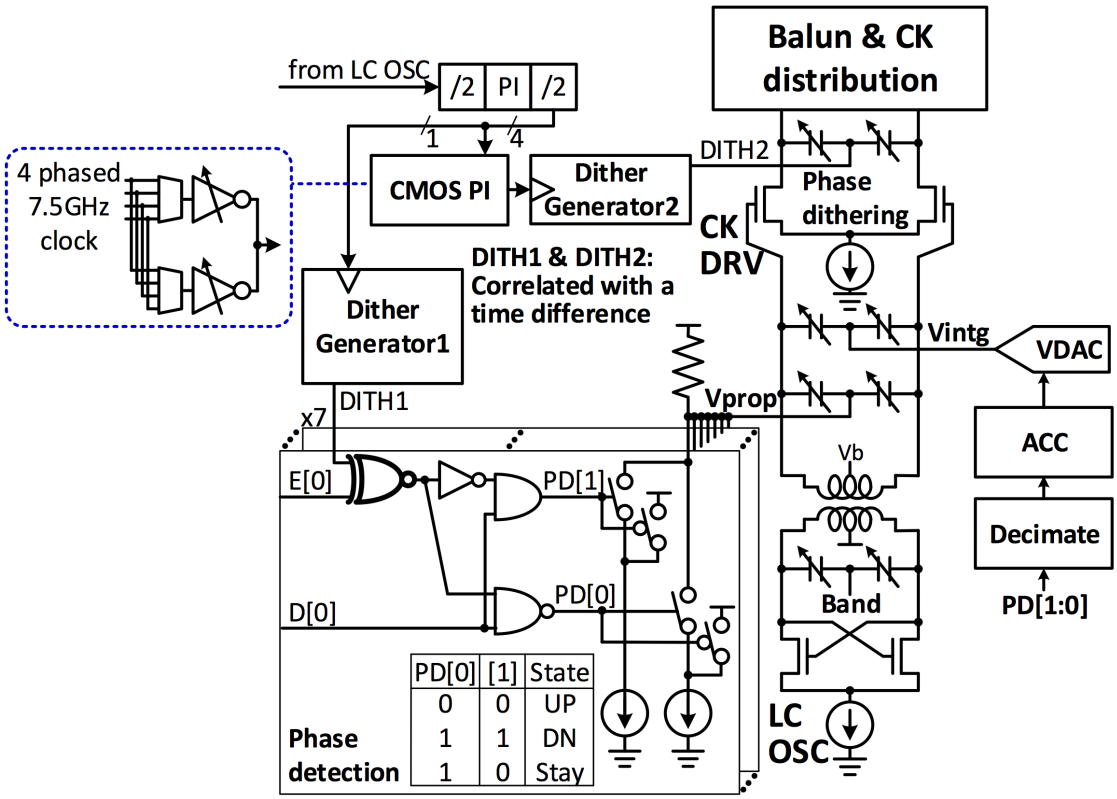


Figure 4.9: CDR implementation

The dithering operation is done by modulating the load capacitance of the clock driver, and the dithering capacitor is chosen based on simulated results with s-parameter models of the inductor and clock distribution network that are extracted from an electromagnetic simulator [44] to check if the dithering capacitance meet the dithering level specification (which is also set by the system level requirements such as the jitter tracking bandwidth and horizontal eye opening).

In this work, the dithering capacitor is constructed as a differential capacitive DAC (CDAC) array with 7.5fF unit capacitor cells for programmability. The capacitance modulation circuit supports up to 2% dithering amplitude, as shown in the simulated characteristic (Figure 4.10).

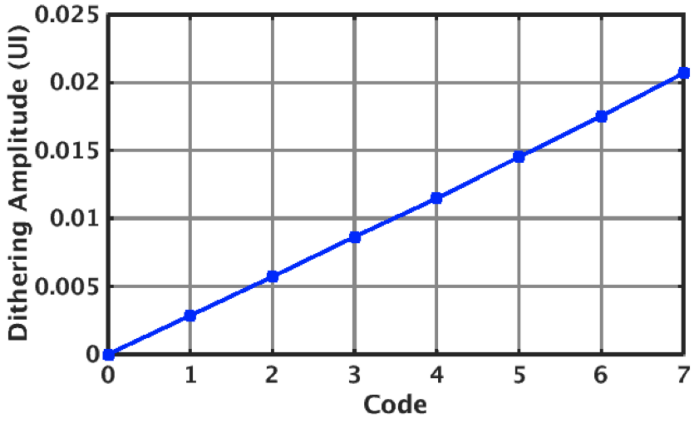


Figure 4.10: Dither strength code vs. simulated dithering amplitude

While the integral path of the CDR is implemented in the digital domain (which operates on deserialized data) with an accumulator, in order to minimize its latency, the proportional path is implemented in the 7.5GHz (UI/8) domain with analog summation.

Chapter 5

60Gb/s Transceiver Integration

5.1 Transmitter Design

Since transmitters are generally terminated to match the channel impedance and hence drive relatively low impedance, removing a substantial amount of ISI is generally more expensive than doing so at the RX. However, since the delay elements within a TX FFE can simply consist of digital latches (as opposed to analog latches in a mixed-signal RX FFE), expanding the number of ISI taps covered by the TX is generally less costly than doing so at the RX. The transceiver balances these competing considerations by designing the TX FFE to include 3 taps (as opposed to the RX FFE's 2 taps), while having the TX cancel only a small fraction of the pre-cursor ISI.

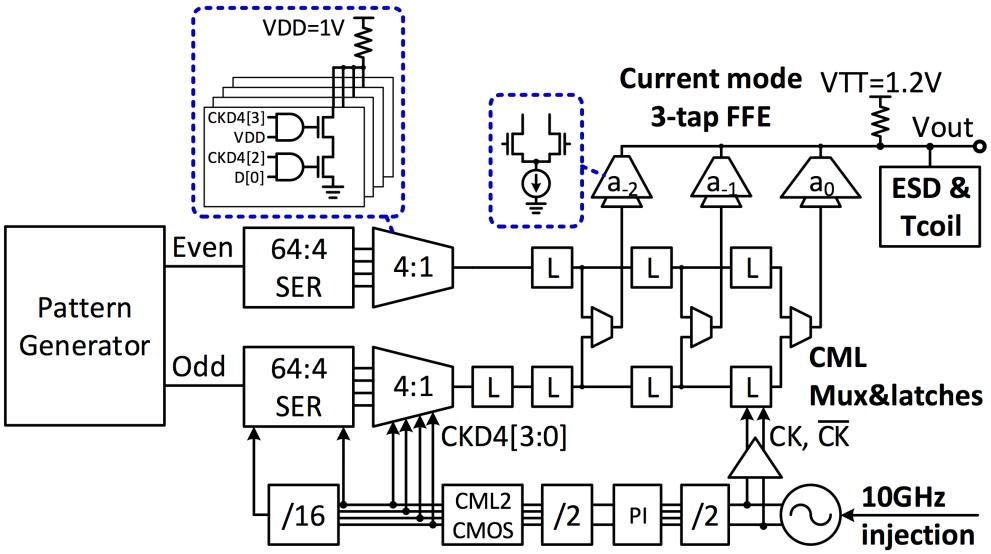


Figure 5.1: Transmitter architecture

The transmitter's block level architecture in Figure 5.1 shows the three phases of serialization used in this design to achieve a serialization ratio of 128:1. A 64:4 CMOS shift register-based serializer and a 4:1 mux-based serializer [13] make up the first two phases of serialization in each (odd and even) path. Shift register based serializers were selected for the lower-frequency portion for their simplicity, and because of the lower frequency, the absolute power penalty of this choice (as compared to a multiplexer based design) is relatively minimal. Following the 4:1 mux-based serializer, current mode logic (CML) muxes combined with CML latches and drivers (Figure 5.2) make up the final 2:1 serialization and 3 taps of equalization necessary for achieving the desired equalization at the data-rate.

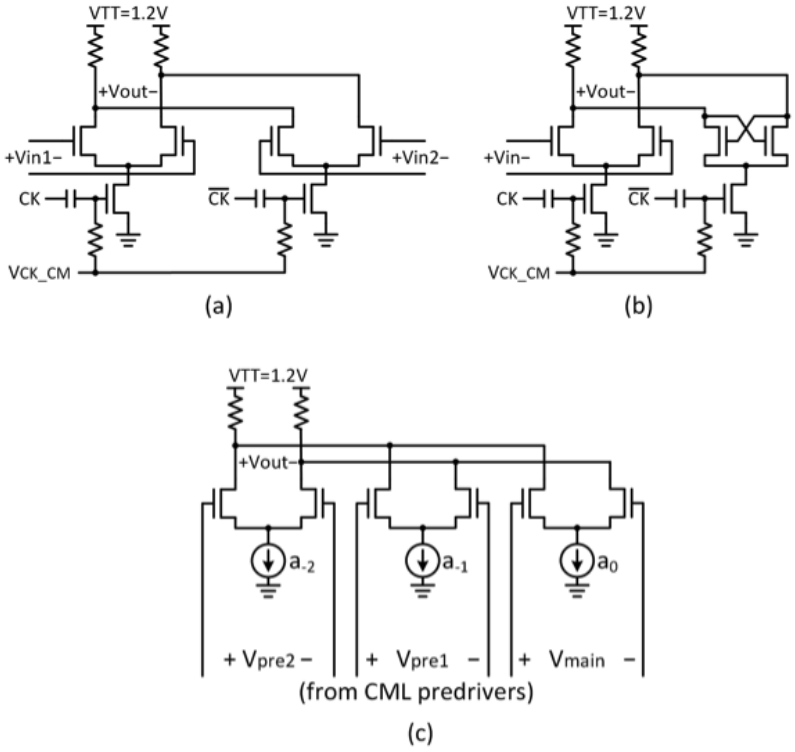


Figure 5.2: Transmitter frontend circuits (a) CML MUX (b) CML latch (c) 3-tap FFE

In order to provide 30GHz and sub-rate clocks for the transmitter operation, an injection-locked LC oscillator followed by a different resonant buffer and divider circuits is used; these have a similar structure to the clock generation and distribution circuits in the receive side, originally presented in [28], with minor differences in device sizing and bias currents due to different loading and swing requirements. The 10GHz injection locking nodes are always turned on and driven by a differential 10GHz reference clock for phase locking. The divider chain consists of a dynamic-latch based 30GHz divider, a phase interpolator, and a 15GHz CML divider and following CMOS divider stages.

5.2 Backend Adaptation and Offset Calibration

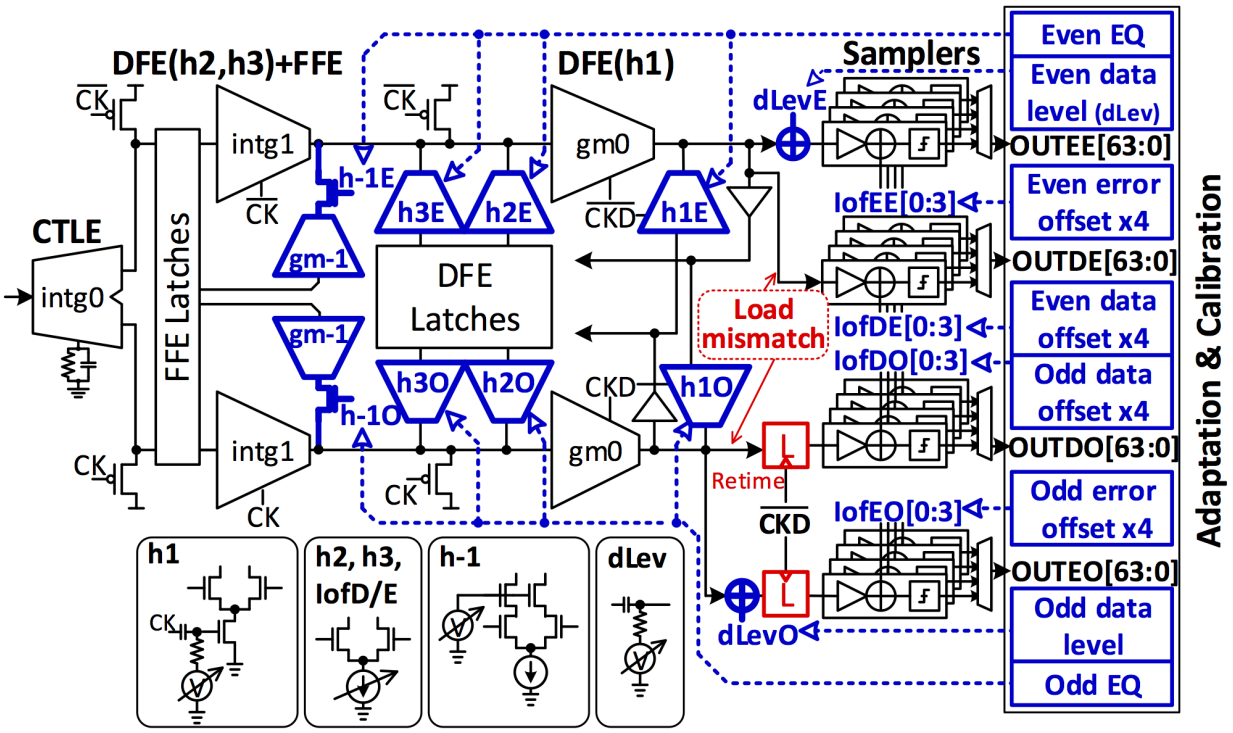


Figure 5.3: RX datapath architecture with adaptation, offset calibration, and data level tracking loops

In order to support various channel characteristics, the sign-sign-least-mean-square (SS-LMS) algorithm [37][45] is used for adapting DFE coefficients and the zero-forcing (ZF) algorithm [46] is used for FFE coefficients. As in any adaptation engine, the equalizer coefficients should solely be determined by the channel response, and there should only be one optimal value for each tap. However, since our design incorporates a retiming latch only in the odd path, loading and bandwidth mismatches are present between the even and odd paths; note that the prior summer stages (gm0 in Figure 5.3) would have to be substantially overdesigned in order to suppress these mismatches.

Instead of incurring a power penalty by overdesigning the summers, dedicated loops for each path (as shown in Figure 5.3) are used to handle these mismatches along with any other mismatches due to bias DAC variations, parasitic capacitors, and clock skew. As a result, the even and odd path will each have their own separate adaptation loops and therefore converge to different equalizer coefficients. In total, 26 adaptation and offset calibration loops (8 for FFE+DFE, 2 for data level tracking (dLev), and 16 for the samplers' offset calibration) are implemented on chip, as shown in Figure 5.3. While this introduces additional hardware, it should be noted that these loops are in the lower frequency domain, and their power

consumption is trivial compared to the power overhead necessary to handle these factors by upsizing and/or adding additional high frequency stages to the signaling path. The final transceiver block diagram is shown in Figure 5.4.

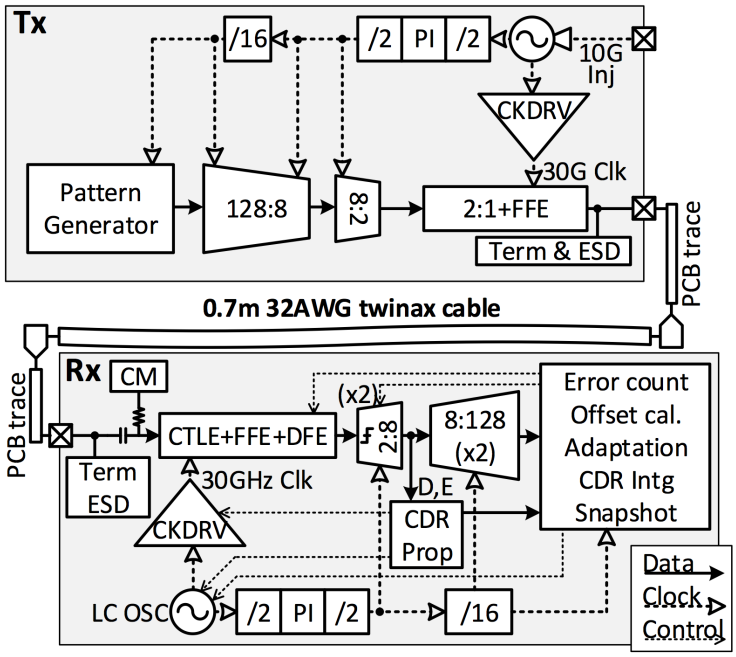


Figure 5.4: Transceiver Architecture

5.3 Measurement Results

The transceiver test-chip was designed and fabricated in a 65nm CMOS process (Figure 5.5). The entire transceiver occupies $2.48mm^2$ (TX: $0.45mm^2$, RX: $2.03mm^2$).

Figure 5.6 shows measurement setup for the test chip and channel environment. In order to maintain signal integrity and minimized unwanted reflections and/or additional losses from parasitic elements, a 0.7m differential 32AWG twinax cable is directly soldered on the testing board made out of Megtron6-5670 material. The chip is also directly attached to the board via flip-chip bumps to minimize parasitic loading.

Figure 5.7 shows the measured S21 of the channel in the configuration shown in Figure 5.6(a). The insertion loss at 30GHz is 21dB, without significant reflections from complex package and connector structures [6]. Figure 5.8 shows the associated pulse response and eye diagram, measured by Keysight 86118A sampling heads (Figure 5.6(b)).

The transmitter is configured to generate a 60Gb/s $2^7 - 1$ PRBS pattern, and a Keysight E8267D is used to generate a 10GHz reference clock to injection lock the transmitter's

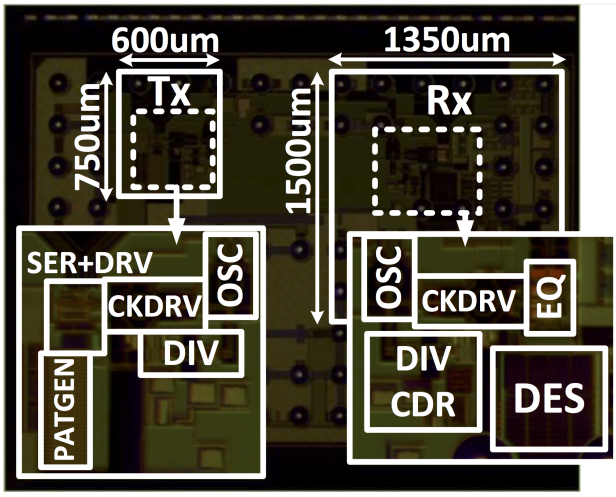


Figure 5.5: Die photo

30GHz LC oscillator. Significant ISI from channel loss and a completely closed eye diagram are observed, corroborating the measured ratio between ISI and cursor of 2.9.

Following channel characterization, on-chip eye diagrams and a BER bathtub curve were measured to test the adaptive equalizer (Figure 5.6(c) and Figure 5.9). Notably, the design achieves $> 0.3UI$ opening at 1e-12 BER for both the odd and even paths. After the CDR is turned on, the RX clock is locked to the TX clock, and their divided-by-64 clocks are plotted in Figure 5.10.

Figure 5.11 shows the measured output of the dLev tracking loop and phase detector output while sweeping the receiver's clock phase. The CDR loop is designed to converge to the zero-crossing point of Figure 5.12(b), where the converged dLev output (and the cursor amplitude) is close to its maximum value.

Figure 5.13 shows the measured jitter tolerance curve. As expected from the bathtub curve, the jitter tolerance curve confirms 0.3UI of margin at high frequency, and 3MHz of tracking bandwidth. The transceiver power breakdown is shown in Table 5.1, and Table 5.2 shows a performance summary and comparisons between various state-of-the-art transceivers. Particularly considering the 65nm process technology, the RX efficiency highlights the benefits of the energy efficient adaptive signaling path and the proposed baud-rate CDR.

5.4 Conclusions

Chapter 4 and chapter 5 described design techniques enabling the realization of an efficient (less than 5pJ/bit) and complete 60Gb/s NRZ transceiver in 65nm CMOS technology. The transceiver incorporates CTLE, FFE, and DFE, output slicing, clocking, equalizer adaptation, and a baud rate CDR, allowing the design to operate with a 21dB loss channel.

Table 5.1: Tranceiver power breakdown

	Item	Power (mW)
TX	TxFFE, 2:1SER	63
	128:2SER, Pattern Generator	20
	OSC, CKDRV, DIV4	51
	DIV16	18
	TX Total	152
RX	CTLE, RXFFE, DFE	46
	2:128DES, CDR, Adaptation, Error Count	17
	OSC, CKDRV, DIV4	55
	DIV16	18
	RX Total	136
Transceiver Total		288

Table 5.2: Transceiver Performance Summary

Reference	[2] Frans 2016	[3] Peng 2017	[13] Lee 2015		[14] Shibasaki 2016	**This work**
Modulation	PAM4	PAM4	PAM4	NRZ	NRZ	**NRZ**
Process	16nm	40nm	40nm	40nm	28nm	**65nm**
Data-rate (Gb/s)	56	56	54.1-56.8	55.5-56.5	56	**60**
Channel loss (dB)	25 -	24 -	N/A	N/A	18.4 -	**21** **2.9***
Equalizer	3-tap TXFFE CTLE DSP	3-tap TXFFE CTLE 3-tap DFE	CTLE	CTLE	2-tap TXFFE CTLE 1-tap DFE	**3-tap TXFFE** **2-tap RXFFE** **CTLE** **3-tap DFE**
SERDES ratio	1:32	1:64	1:16(TX) 1:2(RX)	1:8	1:32	**1:128**
Adaptation	Y	N	N	N	Y	**Y (per-path)**
Eye opening	N/A	25% @ 1e-9	N/A	N/A	28% @ 1e-9	**35% @ 1e-9** **30% @ 1e-12**
TX Power (mW)	140	200	290	450	104.7	**152**
RX Power (mW)	370	382	420	220	141.7	**136**
Tot. Power (mW)	550◇	602†	710	670	246.4‡	**288**
TX Efficiency (pJ/bit)	2.5	3.57	5.17	8	1.87	**2.53**
RX Efficiency (pJ/bit)	6.61	6.82	7.5	3.93	2.53	**2.26**
Total Efficiency (pJ/bit)	9.82	10.75	12.67	11.96	4.4	**4.8**

*VISI/VCURSOR is measured from a probing setup with additional <2dB loss @30GHz
◇ DSP power not included, 40mW clocking power,
† 20mW clocking power
‡ Clocking power is amortized over 2 lanes

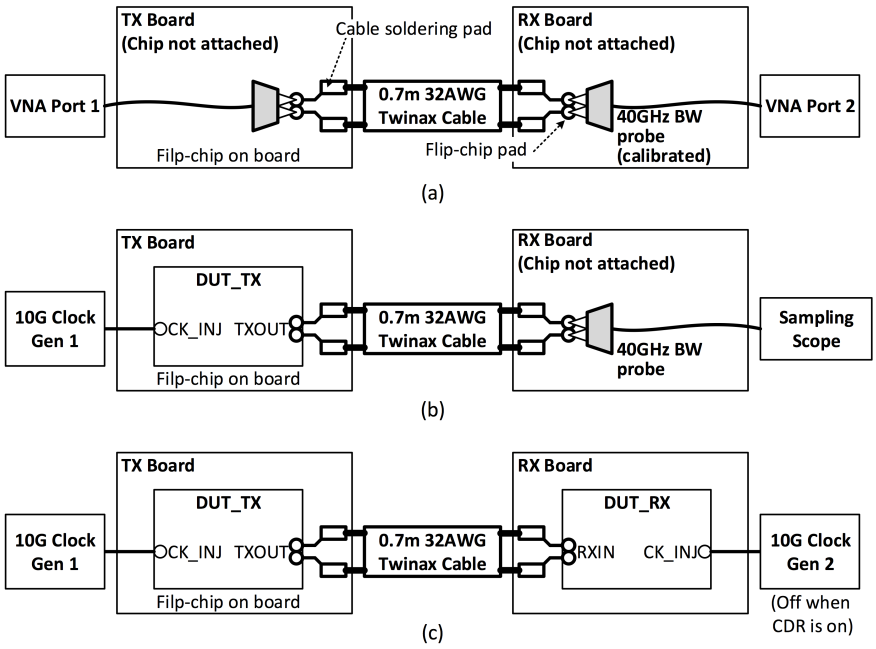


Figure 5.6: Measurement setup for (a) channel frequency response (b) pulse response (c) equalizer and CDR characterizations

An energy efficient and robust baud-rate CDR is realized by making use of an integrating frontend and phase dithering circuits. The design achieves 60Gb/s operation with $> 0.3UI$ opening at 1e-12 BER (and error-free operation over 1e13 bits at the center of the eye), while consuming 288mW and occupying $2.48mm^2$.

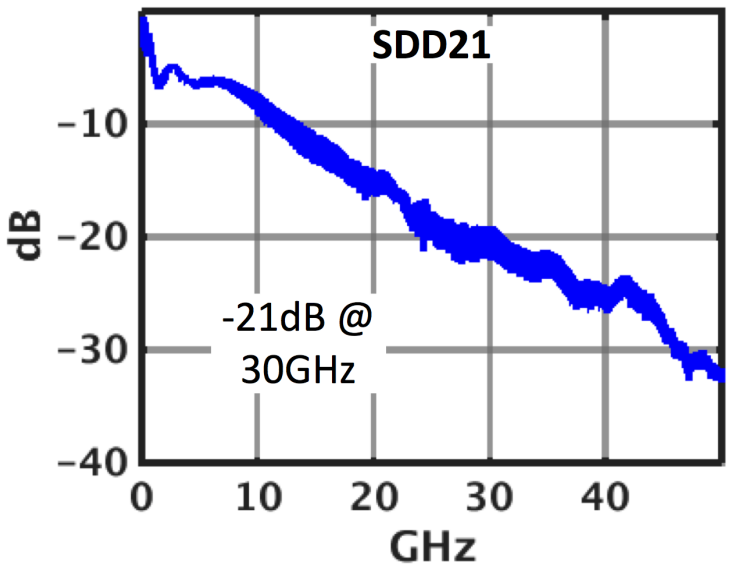


Figure 5.7: Measured channel frequency response

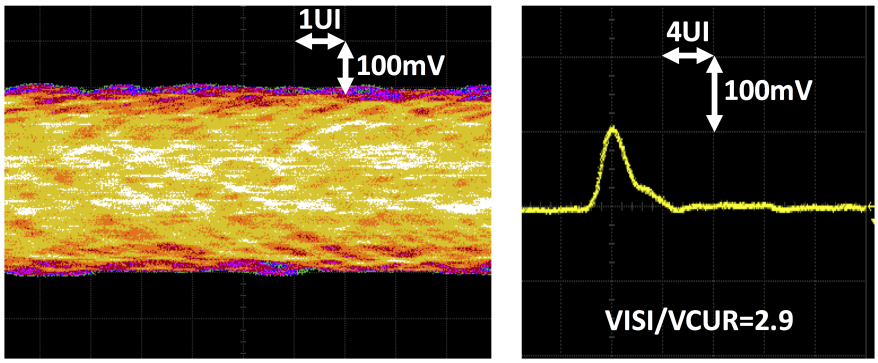


Figure 5.8: Transmitter and channel characterizations (a) eye diagram and (b) pulse response

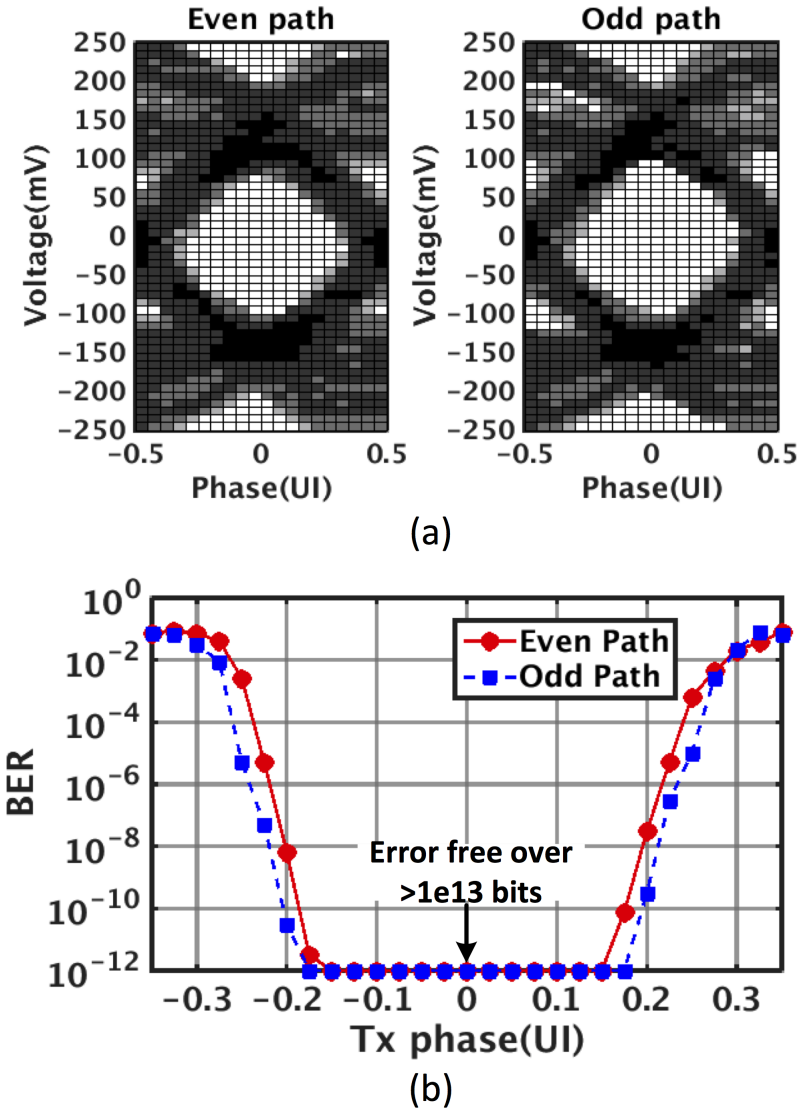


Figure 5.9: Equalizer measurements (a) on-chip eye diagram (b) bathtub curve

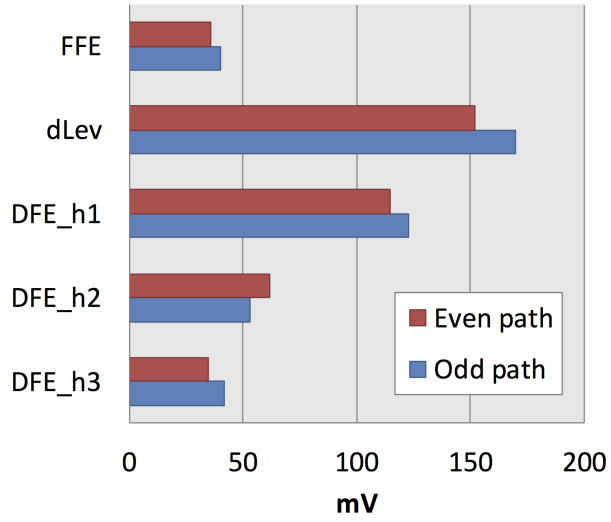


Figure 5.10: Converged equalizer coefficients (referred to the final DFE output)

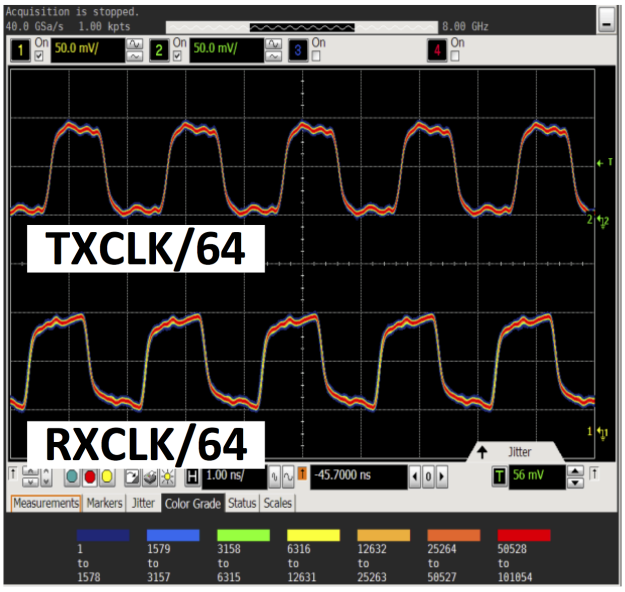


Figure 5.11: Measured divided clock outputs

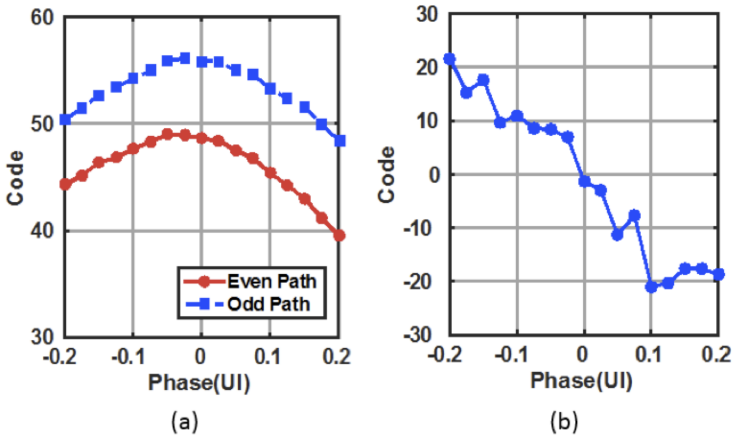


Figure 5.12: (a) Measured dLev values (b) subsampled phase detector output

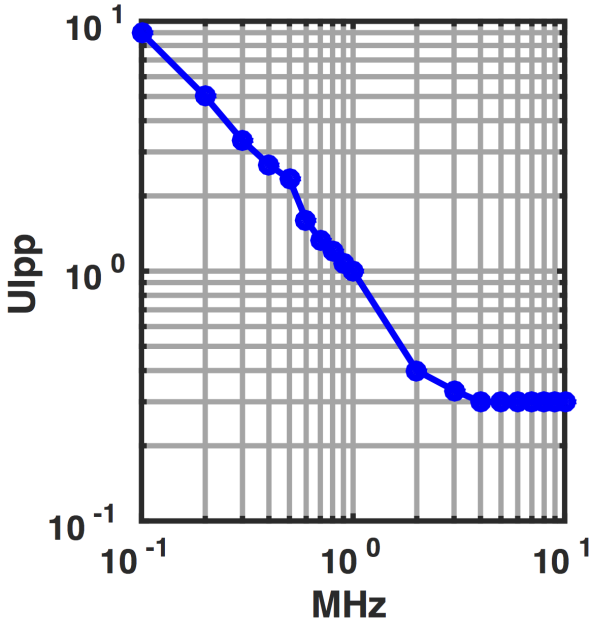


Figure 5.13: Measured jitter tolerance

Chapter 6

Conclusions

6.1 Thesis Summary

As the rapid growth of internet connectivity requires systems to be more distributed and data oriented, the demand for high-bandwidth wireline communication systems continues to increase. Industrial standards have responded to this trend by increasing the data-rate of chip I/Os, doubling per-pin data-rate around every 4 years. Following this trend, 60Gb/s chip-to-chip transceiver systems will soon (within 2-3 years) need to be widely supported. While the bandwidth continues to increase rapidly, the budgeted power consumption for these high-speed transceivers remains relatively constant, which implies that improving energy efficiency is a must.

Various types of equalizers have been used to cancel ISI from band-limited channels, however, their high-operating frequency requires the equalizer designs to be highly optimized. By using the current integration technique, the proposed receive equalizer design can be made with an improved overall energy efficiency compared to previous designs. The current integration technique is combined with cascode sampling, clocked integration, and variable cascode bias, to meet the dedicated requirements for the CTLE, DFE, and FFE, respectively. The implemented receiver frontend operates error free over $> 1e12$ bits at 60Gb/s, occupying $0.16mm^2$, and consuming 173mW.

Since the equalizer frontend is operating close the speed limit of 65nm CMOS technology, the design tends to be highly sensitive to parasitic and wiring capacitance. Traditionally this requires multiple, manual iterations between design steps, increasing the design time significantly. This work improves the productivity and captures the parameters more precisely for better sizing, by utilizing an analog circuit generation framework (BAG). By scripting the sizing and layout generation processes using BAG, A design that meets requirements for 60Gb/s operations with smaller power consumption than the previous work was easily generated.

An energy-efficient clock and data recovery (CDR) scheme helps the transceivers to achieve the target power consumption by reducing clocking power overheads. In this work,

the frontend current integration stage for CTLE is combined with the integration phase dithering to implement a robust baud-rate CDR. The correlation of the adaptive error sampler output and the phase dithering sequence indicates the direction of a phase offset, and the resulting baud-rate CDR saves power and complexity compared to an oversampling CDR by not requiring additional clock phases nor deserializers.

Finally, challenges for the full transceiver completion are addressed, by implementing a 60Gb/s FFE transmitter and automated adaptation and calibration loops. The per-path adaption and per-sampler offset calibration loops readily address the BER degradation due to variability and non-ideal factors. The fabricated 65nm CMOS transceiver operates at 60Gb/s with an eye-opening of 30% UI and consumes 288mW while equalizing 21dB of loss at 30GHz over a 0.7m Twinax cable.

Bibliography

[1] S. G. Narendra, L. C. Fujino, and K. C. Smith, "Through the looking glass - the 2015 edition: Trends in solid-state circuits from isscc," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 1, pp. 14–24, winter 2015.

[2] K. Chang, G. Zhang, and C. Borrelli, "Evolution of wireline transceiver standards: Various, most-used standards for the bandwidth demand," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 4, pp. 47–52, Fall 2015.

[3] ITRS. (). Itrs 2012 updates (assembly packaging), [Online]. Available: `http://www.itrs.net/Links/2012ITRS/Home2012.htm`.

[4] D. C. Daly, L. C. Fujino, and K. C. Smith, "Through the looking glass – the 2017 edition: Trends in solid-state circuits from isscc," *IEEE Solid-State Circuits Magazine*, vol. 9, no. 1, pp. 12–22, winter 2017.

[5] V. Stojanovic and M. Horowitz, "Modeling and analysis of high-speed links," in *Proceedings of the IEEE 2003 Custom Integrated Circuits Conference, 2003.*, Sep. 2003, pp. 589–594.

[6] B. Casper, G. Balamurugan, J. E. Jaussi, J. Kennedy, M. Mansuri, F. O'Mahony, and R. Mooney, "Future microprocessor interfaces: Analysis, design and optimization," in *2007 IEEE Custom Integrated Circuits Conference*, Sep. 2007, pp. 479–486.

[7] T. De Keulenaer, J. De Geest, G. Torfs, J. Bauwelinck, Y. Ban, J. Sinsky, and B. Kozicki, "56+ gb/s serial transmission using duo-binary signaling," eng, in *DesignCon 2015, Proceedings*, vol. 10TH-3, Santa Clara, CA, 2015, pp. 1–23.

[8] Y. Frans, M. Elzeftawi, H. Hedayati, J. Im, V. Kireev, T. Pham, J. Shin, P. Upadhyaya, L. Zhou, S. Asuncion, C. Borrelli, G. Zhang, H. Zhang, and K. Chang, "A 56gb/s pam4 wireline transceiver using a 32-way time-interleaved sar adc in 16nm finfet," in *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, Jun. 2016, pp. 1–2.

[9] J. Lee, P. C. Chiang, and C. C. Weng, "56gb/s pam4 and nrz serdes transceivers in 40nm cmos," in *2015 Symposium on VLSI Circuits (VLSI Circuits)*, Jun. 2015, pp. C118–C119.

[10] J. Im, D. Freitas, A. Roldan, R. Casey, S. Chen, A. Chou, T. Cronin, K. Geary, S. McLeod, L. Zhou, I. Zhuang, J. Han, S. Lin, P. Upadhyaya, G. Zhang, Y. Frans, and K. Chang, "6.3 a 40-to-56gb/s pam-4 receiver with 10-tap direct decision-feedback equalization in 16nm finfet," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb. 2017, pp. 114–115.

[11] H. Braunisch, J. E. Jaussi, J. A. Mix, M. B. Trobough, B. D. Horine, V. Prokofiev, D. Lu, R. Baskaran, P. C. H. Meier, D. H. Han, K. E. Mallory, and M. W. Leddige, "High-speed flex-circuit chip-to-chip interconnects," *IEEE Transactions on Advanced Packaging*, vol. 31, no. 1, pp. 82–90, Feb. 2008, ISSN: 1521-3323.

[12] M. Mansuri, J. E. Jaussi, J. T. Kennedy, T. C. Hsueh, S. Shekhar, G. Balamurugan, F. O'Mahony, C. Roberts, R. Mooney, and B. Casper, "A scalable 0.128 x2013;1 tb/s, 0.8 x2013;2.6 pj/bit, 64-lane parallel i/o in 32-nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 12, pp. 3229–3242, Dec. 2013.

[13] 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*, Feb. 2013, pp. 38–39.

[14] 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 technology," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 8, pp. 1903–1916, Aug. 2015.

[15] 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 Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2885–2900, Dec. 2006.

[16] P. Upadhyaya, J. Savoj, F. T. An, A. Bekele, A. Jose, B. Xu, D. Wu, D. Turker, H. Aslanzadeh, H. Hedayati, J. Im, S. W. Lim, S. Chen, T. Pham, Y. Frans, and K. Chang, "3.3 a 0.5-to-32.75gb/s flexible-reach wireline transceiver in 20nm cmos," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, Feb. 2015, pp. 1–3.

[17] B. Zhang, K. Khanoyan, H. Hatamkhani, H. Tong, K. Hu, S. Fallahi, K. Vakilian, and A. Brewster, "3.1 a 28gb/s multi-standard serial-link transceiver for backplane applications in 28nm cmos," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, Feb. 2015, pp. 1–3.

[18] T. Shibasaki, T. Danjo, Y. Ogata, Y. Sakai, H. Miyaoka, F. Terasawa, M. Kudo, H. Kano, A. Matsuda, S. Kawai, T. Arai, H. Higashi, N. Naka, H. Yamaguchi, T. Mori, Y. Koyanagi, and H. Tamura, "3.5 a 56gb/s nrz-electrical 247mw/lane serial-link transceiver in 28nm cmos," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan. 2016, pp. 64–65.

[19] F. Spagna, L. Chen, M. Deshpande, Y. Fan, D. Gambetta, S. Gowder, S. Iyer, R. Kumar, P. Kwok, R. Krishnamurthy, C. c. Lin, R. Mohanavelu, R. Nicholson, J. Ou, M. Pasquarella, K. Prasad, H. Rustam, L. Tong, A. Tran, J. Wu, and X. Zhang, "A 78mw 11.8gb/s serial link transceiver with adaptive rx equalization and baud-rate cdr in 32nm cmos," in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, Feb. 2010, pp. 366–367.

[20] C. Thakkar, S. Sen, J. E. Jaussi, and B. Casper, "23.2 a 32gb/s bidirectional 4-channel 4pj/b capacitively coupled link in 14nm cmos for proximity communication," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan. 2016, pp. 400–401.

[21] V. Balan, J. Caroselli, J. G. Chern, C. Chow, R. Dadi, C. Desai, L. Fang, D. Hsu, P. Joshi, H. Kimura, C. Y. Liu, T.-W. Pan, R. Park, C. You, Y. Zeng, E. Zhang, and F. Zhong, "A 4.8-6.4-gb/s serial link for backplane applications using decision feedback equalization," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1957–1967, Sep. 2005.

[22] Y. Lu and E. Alon, "Design techniques for a 66 gb/s 46 mw 3-tap decision feedback equalizer in 65 nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 12, pp. 3243–3257, Dec. 2013.

[23] T. Shibasaki, W. Chaivipas, Y. Chen, Y. Doi, T. Hamada, H. Takauchi, T. Mori, Y. Koyanagi, and H. Tamura, "A 56-gb/s receiver front-end with a ctle and 1-tap dfe in 20-nm cmos," in *2014 Symposium on VLSI Circuits Digest of Technical Papers*, Jun. 2014, pp. 1–2.

[24] 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 Journal of Solid-State Circuits*, vol. 49, no. 2, pp. 452–470, Feb. 2014.

[25] J. Lee, P. C. Chiang, and C. C. Weng, "56gb/s pam4 and nrz serdes transceivers in 40nm cmos," in *2015 Symposium on VLSI Circuits (VLSI Circuits)*, Jun. 2015, pp. C118–C119.

[26] T. Shibasaki, T. Danjo, Y. Ogata, Y. Sakai, H. Miyaoka, F. Terasawa, M. Kudo, H. Kano, A. Matsuda, S. Kawai, T. Arai, H. Higashi, N. Naka, H. Yamaguchi, T. Mori, Y. Koyanagi, and H. Tamura, "3.5 a 56gb/s nrz-electrical 247mw/lane serial-link transceiver in 28nm cmos," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan. 2016, pp. 64–65.

[27] J. Han, Y. Lu, N. Sutardja, K. Jung, and E. Alon, "A 60gb/s 173mw receiver frontend in 65nm cmos technology," in *2015 Symposium on VLSI Circuits (VLSI Circuits)*, Jun. 2015, pp. C230–C231.

[28] ——, "Design techniques for a 60 gb/s 173 mw wireline receiver frontend in 65 nm cmos technology," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 871–880, Apr. 2016.

[29] J. Han, Y. Lu, N. Sutardja, and E. Alon, "6.2 a 60gb/s 288mw nrz transceiver with adaptive equalization and baud-rate clock and data recovery in 65nm cmos technology," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb. 2017, pp. 112–113.

[30] M. Park, J. Bulzacchelli, M. Beakes, and D. Friedman, "A 7gb/s 9.3mw 2-tap current-integrating dfe receiver," in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, Feb. 2007, pp. 230–599.

[31] 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 Journal of Solid-State Circuits*, vol. 49, no. 11, pp. 2588–2607, Nov. 2014.

[32] T. O. Dickson, J. F. Bulzacchelli, and D. J. Friedman, "A 12-gb/s 11-mw half-rate sampled 5-tap decision feedback equalizer with current-integrating summers in 45-nm soi cmos technology," in *2008 IEEE Symposium on VLSI Circuits*, Jun. 2008, pp. 58–59.

[33] B. Kim, Y. Liu, T. O. Dickson, J. F. Bulzacchelli, and D. J. Friedman, "A 10-gb/s compact low-power serial i/o with dfe-iir equalization in 65-nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 12, pp. 3526–3538, Dec. 2009.

[34] R. Bai, S. Palermo, and P. Y. Chiang, "2.5 a 0.25pj/b 0.7v 16gb/s 3-tap decision-feedback equalizer in 65nm cmos," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, Feb. 2014, pp. 46–47.

[35] Y. Duan and E. Alon, "A 12.8 gs/s time-interleaved adc with 25 ghz effective resolution bandwidth and 4.6 enob," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 8, pp. 1725–1738, Aug. 2014.

[36] J. E. Jaussi, G. Balamurugan, D. R. Johnson, B. Casper, A. Martin, J. Kennedy, N. Shanbhag, and R. Mooney, "8-gb/s source-synchronous i/o link with adaptive receiver equalization, offset cancellation, and clock de-skew," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 80–88, Jan. 2005.

[37] V. Stojanovic, A. Ho, B. W. Garlepp, F. Chen, J. Wei, G. Tsang, E. Alon, R. T. Kollipara, C. W. Werner, J. L. Zerbe, and M. A. Horowitz, "Autonomous dual-mode (pam2/4) serial link transceiver with adaptive equalization and data recovery," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, pp. 1012–1026, Apr. 2005.

[38] M. J. E. Lee, W. J. Dally, and P. Chiang, "Low-power area-efficient high-speed i/o circuit techniques," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1591–1599, Nov. 2000.

[39] 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)*, Sep. 2011, pp. 1–4.

[40] J. Crossley, A. Puggelli, H. P. Le, B. Yang, R. Nancollas, K. Jung, L. Kong, N. Narevsky, Y. Lu, N. Sutardja, E. J. An, A. L. Sangiovanni-Vincentelli, and E. Alon, "Bag: A designer-oriented integrated framework for the development of ams circuit generators," in *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2013, pp. 74–81.

[41] J. W. Crossley, "Bag: A designer-oriented framework for the development of ams circuit generators," PhD thesis, EECS Department, University of California, Berkeley, Dec. 2014. [Online]. Available: `http://www2.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-195.html`.

[42] Ciranova. (). Cironova pycell studio, [Online]. Available: `http://www.synopsys.com/Tools/Implementation/CustomImplementation/Pages/pycell-studio.aspx`.

[43] K. Mueller and M. Muller, "Timing recovery in digital synchronous data receivers," *IEEE Transactions on Communications*, vol. 24, no. 5, pp. 516–531, May 1976.

[44] S. Kapur and D. E. Long, "Modeling of integrated rf passive devices," in *IEEE Custom Integrated Circuits Conference 2010*, Sep. 2010, pp. 1–8.

[45] B. S. Leibowitz, J. Kizer, H. Lee, F. Chen, A. Ho, M. Jeeradit, A. Bansal, T. Greer, S. Li, R. Farjad-Rad, W. Stonecypher, Y. Frans, B. Daly, F. Heaton, B. W. Garlepp, C. W. Werner, N. Nguyen, V. Stojanovic, and J. L. Zerbe, "A 7.5gb/s 10-tap dfe receiver with first tap partial response, spectrally gated adaptation, and 2nd-order data-filtered cdr," in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, Feb. 2007, pp. 228–599.

[46] Y. Hidaka, W. Gai, T. Horie, J. H. Jiang, Y. Koyanagi, and H. Osone, "A 4-channel 1.25 x2013;10.3 gb/s backplane transceiver macro with 35 db equalizer and sign-based zero-forcing adaptive control," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 12, pp. 3547–3559, Dec. 2009.