SYTHESIZABLE MIXED-SIGNAL BUILDING BLOCKS FOR OPEN SOURCE HIGH SPEED SERIAL LINKS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Sung-Jin Kim
June 2021

This dissertation is online at: http://purl.stanford.edu/sm295qx9833

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Mark Horowitz, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Boris Murmann

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Priyanka Raina

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Dedication

For my parents, EunSang and JungSook

Acknowledgments

First and foremost, I would like to thank Professor Mark Horowitz, who is a truly inspirational advisor, for inviting me to Stanford. Not only has Mark taught me the value of simplicity, but he inspired me to keep challenging various topics of my interest in fields of analog circuit design, design productivity, and open-source hardware. I would like to thank him for his tolerance, patience in letting me explore different fields, for always making me understand and interpret the true nature of things, and great technical guidance. I have tried to learn every bit I could from him and am immensely grateful for all the lessons, which led me to become an independent researcher during my Ph.D. years at Stanford.

I was fortunate to work with Dr. Byongchan Lim. He was not hesitant to give me the right feedback whenever needed, and helped me a lot to settle down to the VLSI group. I would like to thank him for his concerns and sacrifices.

I have had the pleasure of working with great collaborators and discussing various topics with many friends, not limited to academic ones. Among others, I appreciate Zachary Myers and Steven Herbst for the fruitful discussions with me.

I would also like to thank the organizations that supported this research: DARPA POSH and Stanford SystemX.

Finally, I would not be where I stand without the everlasting love of my parents, EunSang and JungSook, in Korea. I appreciate their emotional support and considerations for this journey.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 High-Speed Serial Links

The advent of the Internet, coupled with advances in IC fabrication technology, has dramatically changed our lives. We are always connected to the cloud to easily access, and share information and a variety of electronic devices make us more productive. The amount of data transferred through the Internet continues to grow exponentially as shown in Figure 1.1(a) [1]. In 2017, 120 exabytes of data was transferred per month, whereas today, only 4 years later, the traffic has grown to 320 exabytes, about 3 times more than 2017. The ever-growing Internet traffic requires appropriate network communication to support it, which is manifested by the data rate road map of the Ethernet standard shown in Figure 1.1(b) [2].

On the other hand, as integration density and processing bandwidth keep increasing, a system on a chip (SoC) can support fancier applications, such as machine learning, AI, and high performance computing. However, these new applications create demand for higher I/O throughput to connect the chip to memories and peripherals. Figure 1.1(c) [3] is the data rate road map of PCIe, which is one of the most popular standards for high-speed peripherals, showing a thirst for higher I/O throughput. Figure 1.1(d) [4] clearly shows the importance and demand for high-speed wireline data links, which constitute about a quarter of today's entire silicon IP market.

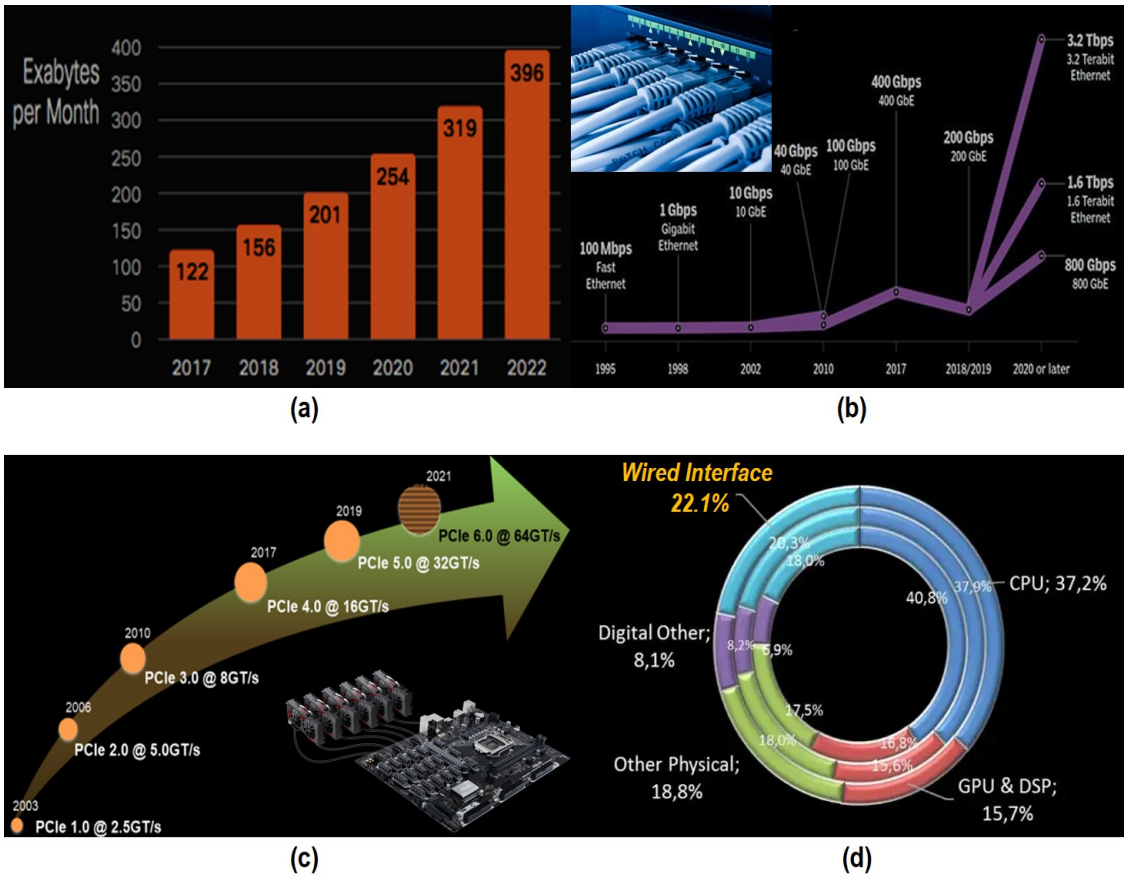


Figure 1.1: (a) Global internet traffic. (b) Data rate road map of Ethernet. (c) Data rate road map of PCIe. (d) IP revenue by category from 2017 to 2019 (2017 is the inner ring).

These standard communication protocols are defined by the open systems interconnection (OSI) model [5], which consists of seven abstraction layers. The physical layer, or PHY, is the layer that directly interacts with a physical media (channel) and is responsible for the transmission and reception of unstructured raw data between a device and the medium. For the rest of this dissertation, the term "link" refers to the PHY layer of a wireline communication link.

Compared to today's complex functions on an SoC, the function of the I/O seems extremely simple: its only purpose is sending ones and zeros through a wire and successfully receiving them. However, when we want to make the throughput of the I/O

tens of gigabits per second (Gbps), this task is no longer simple. The transmitted signal experiences frequency-dependent energy loss due to the inherent band-limited characteristic of the physical channel, resulting in severe distortion of the received waveform. Significant effort goes into developing better communication channels, ranging from more advanced printed circuit boards (PCBs), chip packages, and connectors. Meanwhile, most high-speed serial links today employ multiple signal processing techniques to overcome the bandwidth limitations of legacy channels.

Until about a decade ago, the architecture of high-speed serial links had converged into a "golden architecture," also known as the mixed-signal architecture, shown in Figure 1.2(a) [6][7][8]. Continuous-time linear equalizers (CTLEs), variable gain amplifiers (VGAs), feed-forward equalizers (FFEs), and decision-feedback equalizers (DFEs) are commonly employed to compensate for the effect of channel loss. Dual-loop clock and data recovery (CDR) with an oversampling phase detector (PD) is the most popular approach to achieve symbol synchronization, and a phase-locked loop (PLL)-based clock generator with a phase interpolator (PI) usually provides the optimal sampling clocks for the PD.

Although the mixed-signal serial link architecture has worked well and supported a variety of commercial standards so far, the analog-to-digital converter (ADC)-based serial link architecture has recently gained traction, due to its many advantages [9][10][11]. ADC-based links can enjoy the benefit of CMOS technology scaling by processing most of the signal in the digital domain, as shown in 1.2(b). This digital signal processing (DSP) approach not only enables more sophisticated equalization and symbol detection techniques, but is also robust to process, voltage, and temperature (PVT) variations and much easier to reconfigure. Moreover, it is a natural fit for error correction channel coding schemes [12][13], which are a crucial driver to push the data rate up to 100 Gbps, or even higher. For these reasons, the majority of high-speed links for modern standards adopt the ADC-based architecture as a default option.

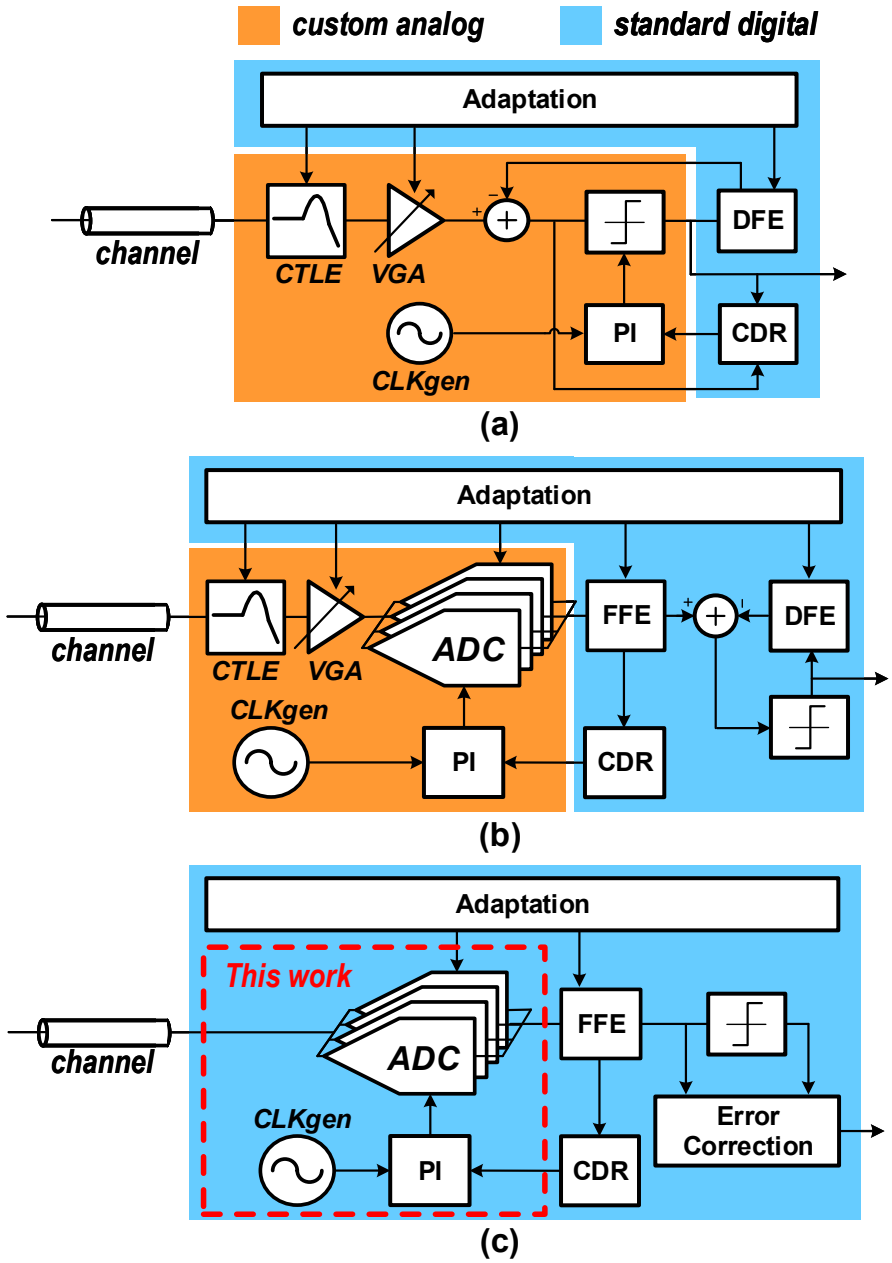


Figure 1.2: (a) Conventional mixed-signal receiver architecture. (b) Conventional ADC-based receiver architecture. (c) Proposed open-source synthesizable PHY architecture.

1.2 Open Source PHY

Generally, design and validation of a high speed serial link require lots of engineering effort, since its performance tightly depends on sophisticated analog and mixed-signal (AMS) circuit blocks. While digital blocks use a set of electrical rules for formal equivalence checking to ensure the Boolean functions of circuits, there are no such universal rules for the AMS blocks – understanding the rules for dedicated circuits separates novice and experienced AMS designers. Besides, while digital designs are synthesized into standard gates and then laid out using an automatic placement and routing (PnR) tool, AMS blocks rely on a transistor-level custom design flow, which requires a significant amount of non-recurring engineering (NRE) cost. An even bigger problem is that AMS blocks are often the bottleneck for full-chip verification, since they usually require more precision than what is needed for the rest of the system, and hence severely affect the turn-around time (TAT) of product developments.

This dissertation proposes a synthesizable and portable architecture of the AMS building blocks of an ADC-based serial link along with an automated design platform. The proposed architecture enables the link to be entirely described by a hardware description language (HDL) using a minimal amount of analog precision as shown in Figure 1.2(c). This design is intended to be released to the public with all required design collateral so that its users, even those who do not have deep analog backgrounds, can produce their own PHY easily and efficiently. This platform is called the Open-Source PHY.

The proposed receiver architecture does not contain front-end amplifiers (CTLE, VGA) because they are tricky to implement in a synthesizable fashion. Moreover, it does not rely on DFE because it can be problematic for timing closure. Instead, the proposed architecture adopts a novel MLSD-based error correction technique to compensate for the performance degradation due to the absence of amplifiers and DFE. The error correction, or equalization, is beyond the scope of this dissertation: interested readers may refer to Appendix B.

1.3 Dissertation Organization

Chapter 2 provides the background about ADC-based links to introduce the main components of this link design, including key error sources. It also gives the performance requirement of the link. Chapters 3, 4, and 5 describe the design of the ADC, PI, and clock generator respectively. Each starts with its own brief review of background work to introduce prior art. Chapter 6 presents the design flow and physical implementation for the proposed AMS blocks. Chapters 7 presents a 20 Gbps receiver front-end, including all the proposed AMS blocks, with experimental results. Chapter 8 summarizes this dissertation and proposes possible future work.

Chapter 2

Background on ADC-Based Links

This chapter reviews the fundamentals of ADC-based links. The objectives of this chapter are two-fold. The first objective is to provide insight into what factors influence the performance of the link and how the link performance can be estimated. This will provide the foundation for the later chapters. The second objective is to motivate the specifications of each building block for a given target performance.

Sections 2.1 and 2.2 define the performance of a link and review key error sources. Section 2.3 introduces common approaches that estimate the performance of the link, and the target specifications of each building block are discussed in Section 2.4.

2.1 Bit Error Rate

Ideally, a high-speed serial link should be designed to be error-free even in the "worst-case" scenario. However, given the random nature of the noise in active devices, PCBs, and other link components, bit errors will occur when the link operates for a sufficiently long time, and it is natural and necessary to characterize the performance of the link statistically, especially when the timing budget keeps shrinking with increasing signaling speed.

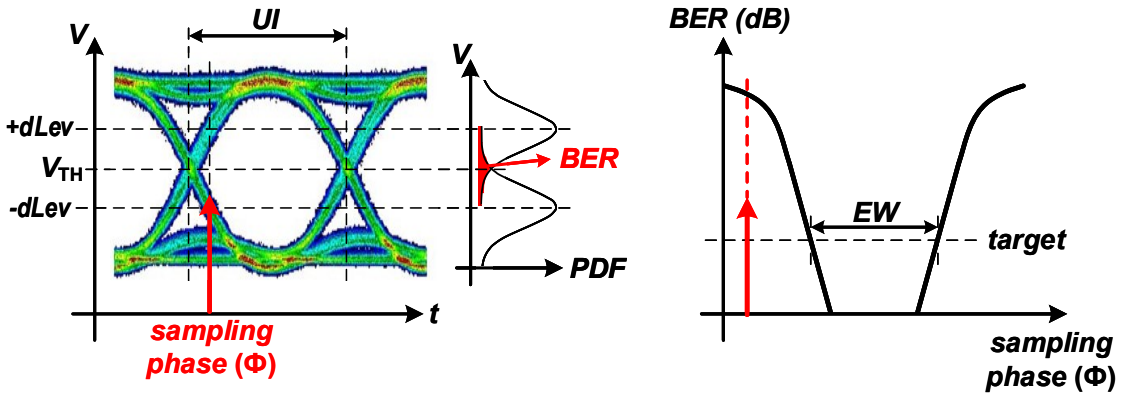


Figure 2.1: Typical eye diagram of a high-speed serial link and corresponding BER bathtub curve.

Bit error rate (BER), defined as the ratio of the erroneous bits received to the total number of the bits transmitted, is a statistical performance indicator of a communication channel. It indicates the mean time to a bit error for a link operating at a given speed. BER results are often presented in the bathtub curve format, which is closely related to the well-known eye diagram. For example, Figure 2.1 shows an eye diagram generated from a received waveform. From the eye diagram, the probability distribution function (PDF) of the received signal at a given sampling phase can be extracted. The BER is calculated by integrating the PDF that sits over a decision boundary (V_{TH}) based on the maximum likelihood decision criteria of a bi-modal signal. A time-domain BER bathtub curve represents the BER with respect to the sampling phase swept over one unit interval (UI) of the eye. The eye width (EW), also know as horizontal margin (HM), of a link is defined as the maximum allowable range of the sampling phase that can achieve a BER lower than a given target. A voltage-domain BER bathtub curve along with eye height (EH) can be defined in a similar way by sweeping the decision boundary for a given sampling phase.

2.2 Error Sources of ADC-Based Links

There are a variety of error sources in ADC-based links that affect the BER performance of link as shown in Figure 2.2. They reside in both the voltage domain and the time domain and can be either bounded deterministic or unbounded random error. While a sequence of bounded deterministic error, such as simultaneous switching output (SSO) noise, is correlated in time and its magnitude is bounded, a sequence of the unbounded random error, such as thermal noise, is uncorrelated in time and its probability distribution is nonzero even for a very large magnitude. Understanding these error sources is one of the most important steps in designing a competitive link system.

In this section, these major error sources of ADC-based links are investigated with their characteristics, such as probability distribution and power spectral density, and impacts on link performance. We first discuss errors caused by physical channel impairments, V_{ISI}, in Section 2.2.1 and then, we briefly describe input-referred thermal noise of the RX, V_n, in Section 2.2.2. Next, we analyze characteristics of ADC-induced errors, V_q, including quantization, non-linearity, and time-interleaving,

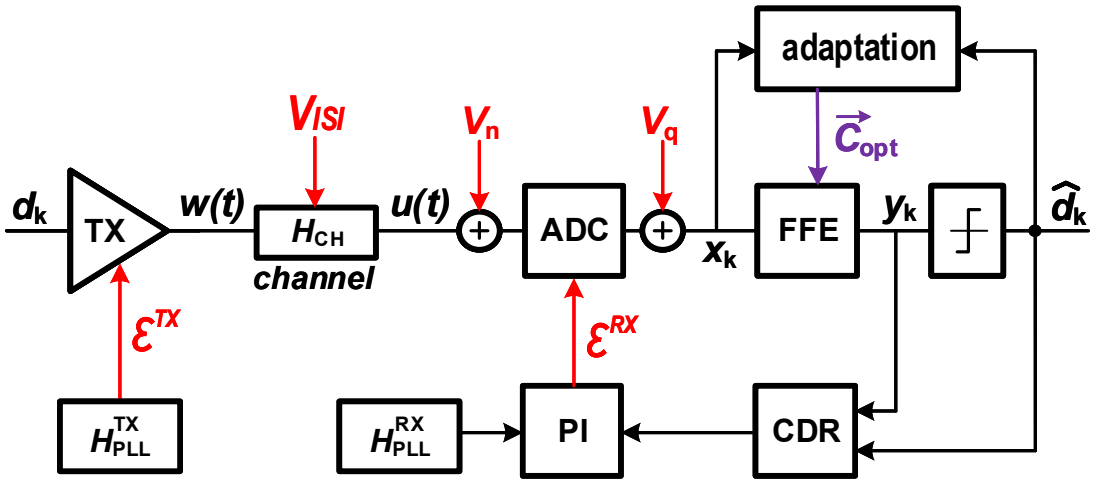


Figure 2.2: Major error sources in ADC-based links.

in Section 2.2.3. The impact of timing jitter of both transmitter (TX) and receiver (RX), ε_{TX} and ε_{RX}, is covered in Section 2.2.4. Finally, we address the impact of a linear equalizer after the ADC associated with its coefficient adaptation, $\vec{C}_{opt}$, in Section 2.2.5.

2.2.1 Channel ISI (V_{ISI})

The channel of an electrical link refers to all physical media between the TX and RX, including copper wire traces on chip packages, PCBs, sockets, connectors, and balls/pins of the package on the chip. As the frequency of the transmitted signal from the TX goes up to multi-GHz range, the channel starts to behave as a lossy transmission line and the signal dissipates its energy mainly due to the conductor/dielectric loss of the transmission line. The conductor loss, also known as the skin effect, is caused by the fact that higher frequency components of current tend to flow near the surface of a conductor, and hence experience the higher effective resistance of a small cross-sectional area. The effective resistance of a copper wire contributed by the skin effect is [14]:

$$R_{eff}(f) \sim \frac{K}{2W}\sqrt{f}$$

where W is width of the wire and K is a constant.

The source of the dielectric loss is the energy loss in the dielectric surrounding the transmission line, and it is also a function of the signal frequency [14]:

$$L_D \sim \frac{\pi\sqrt{\varepsilon_r}}{c} f \tan\delta$$

where c is the speed of light, ε_r and is the relative permittivity of the media, and $\tan\delta$ is the loss tangent. The dielectric loss of a transmission line is usually specified with only the loss tangent, which strongly depends on the insulator material.

Moreover, an impedance discontinuity in the channel reflects a portion of energy back to the opposite direction. In long backplane channels, impedance discontinuities occur at several distinct points, such as connectors, via stubs, and TX/RX termination, causing multiple bounces of the signal, which can persist for a relatively long

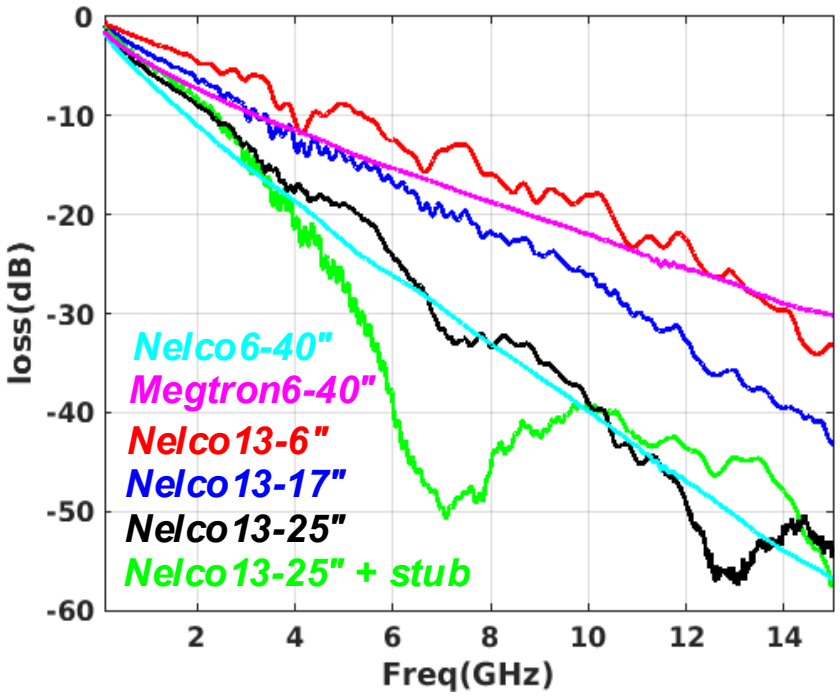


Figure 2.3: Frequency-dependent energy loss of different channels (S_{21}).

time. It complicates the link architecture since significant hardware resources are required to compensate for it.

Figure 2.3 shows a scatter parameter (S-parameter) plot, which is widely used to define the energy transfer characteristic of a system, of a number of different channels [15]. Despite the wide variety of channel characteristics, these channels are mostly low-pass which implies that a nice narrow pulse at the input of the channel will be attenuated and dispersed at the output of the channel, as shown in Figure 2.4(a). The dots in the plot indicate the signal values sampled by a receiver. The portion of the signal smeared out over the symbol boundary interacts with neighboring symbols and may cause errors, as shown in Figure 2.4(b). This interaction is called inter-symbol interference (ISI). In Figure 2.4(b), the received signal that corresponds to a "101" pattern transmitted from the TX is corrupted by ISI and leads to a wrong decision.

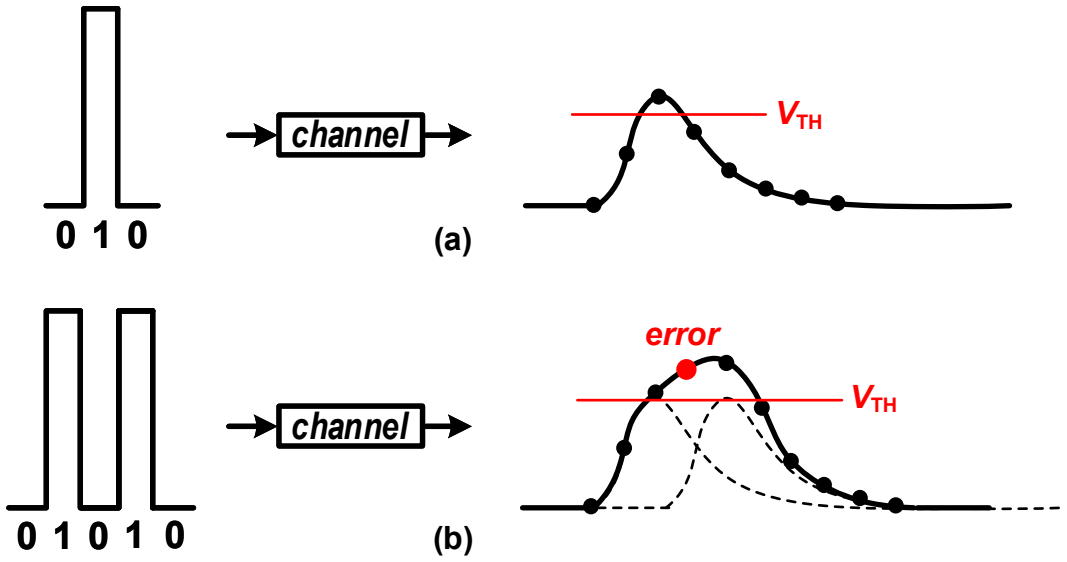


Figure 2.4: Effect of a band-limited channel on the received waveform. (a) Single-bit response of the channel. (b) Bit detection error caused by inter-symbol interference.

It is obvious that the effect of ISI gets worse as the symbol period decreases. ISI is one of the most significant error sources limiting the performance of high-speed links and is modeled as V_{ISI} in Figure 2.2. However, since it is deterministic, as long as we know the characteristics of the channel, the effect of ISI can be minimized (equalized) at the cost of additional hardware and power. There exist a variety of equalization techniques, but they will not be addressed here, since, in the proposed link, they are implemented as a sophisticated bit checker in the digital back-end, which can be ported easily (Appendix B). The proposed link architecture assumes no TX equalization and only uses a linear digital FFE at RX (Section 2.2.5) without any analog equalizers and DFEs.

2.2.2 Thermal Noise (V_n)

Thermal noise, which is random and unbounded, has traditionally been neglected in high-speed links since it was assumed that the thermal noise of resistors and transistor devices is very small when compared to the signal magnitude. However, with increased data rate, these noise sources are emerging as potentially critical components, since the system collects more noise power as its bandwidth increases. One of the main sources of the thermal noise is the 50Ω termination resistor. The device noise from RX circuitry also adds several dB of noise figure to the resistor noise level. The additional noise strongly depends on the RX architecture. For example, the input-referred noise of analog amplifiers (CTLE, VGA), comparators in an ADC, and kT/C noise in interleaving samplers all contribute to thermal noise. Since these thermal noise components can generlly be considered as independent and following zero-mean Gaussian distributions, they are usually modeled as a single random variable, V_n, as shown in Figure 2.2.

2.2.3 ADC Error (V_q)

The ADC can be viewed as a non-linear mapping from a continuous-time and continuous-amplitude input to a discrete-time and discrete-amplitude output. We divided the analysis of the errors introduced by the ADC, which is modeled as V_q in Figure 2.2, into two parts: voltage quantization errors and time interleaving errors. Let's begin with the quantization error in a single ADC slice.

An ADC with N-bit resolution samples an analog voltage at a given sampling instant and maps it to one of 2^N possible digital representations. For example, Figure 2.5 shows the static transfer characteristic between the analog input and the digital output of an ideal 3-bit ADC along with a corresponding quantization error profile. For an analog input within a full-scale range (FSR) of the ADC, the quantization error is bounded by $\pm\Delta/2$, where Δ is a unit step size (or least significant bit (LSB)) of the ADC determined by $FSR/2^N$.

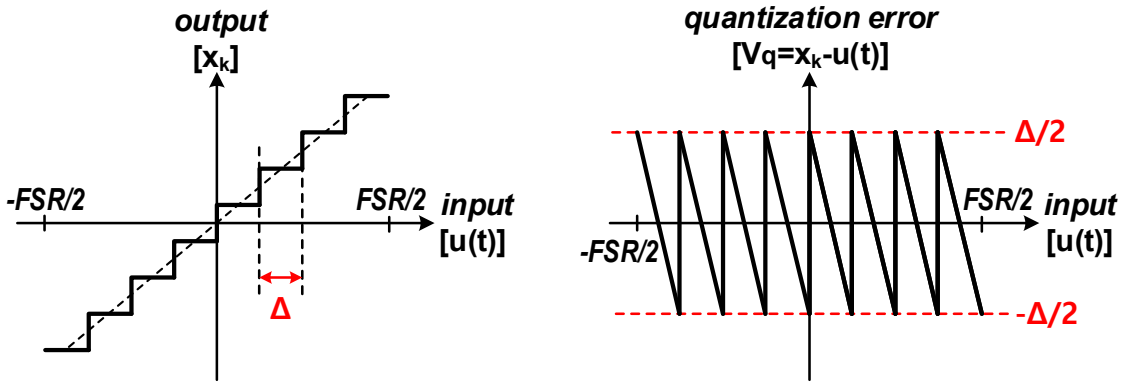


Figure 2.5: DC transfer characteristic of a 3-bit ADC and corresponding quantization error profile (V_q).

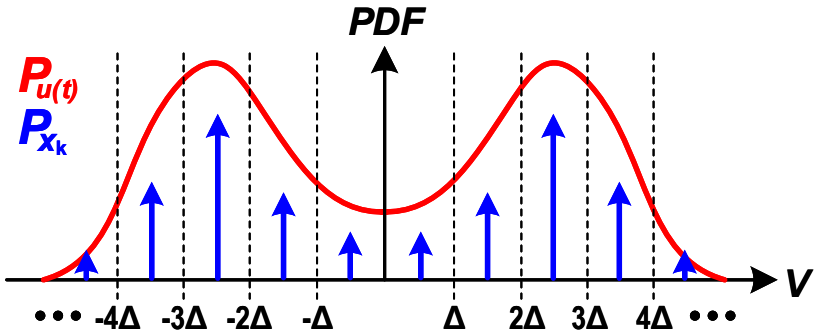


Figure 2.6: Typical PDFs of the input (red) and output (blue) of an ADC.

It is obvious that the distribution of the output of the ADC is discrete while that of the input is continuous and these have a deterministic relationship. Figure 2.6 shows a sketch of typical input/output PDFs of the ADC. Since each analog input voltage within a quantization interval is rounded to the nearest allowable discrete level of output, the strength of the delta functions centered at each quantization interval is identical to the integration of the input PDF over the interval. This quantization process is mathematically equivalent to the convolution between the input PDF (P_u)

and a uniform PDF bounded by $\pm\Delta/2$ ($P_{\sqcap}$), followed by sampling in the PDF domain.

$$P_x(v) = (P_u(v) * P_{\sqcap}) \sum_n \delta(v - n\Delta) \tag{2.1}$$

Quantization noise is normally modeled by adding a uniform noise as shown in Figure 2.7. In general, the quantization process is not the same as adding independent noise, since quantization adds an error that is deterministically related to the signal being quantized. By comparing the output of the ADC with the uniform noise model, we will derive, using the concept of characteristic function (CF), the condition on the input distribution when the approximation is valid [16][17]. The CF of a random variable α, Ψ_α, is defined as a Fourier transform of its PDF, i.e.,

$$\Psi_\alpha(w) = \int_{-\infty}^{\infty} P_\alpha(v) e^{-jwv} dv = E\{e^{-jwv}\} \tag{2.2}$$

Note that the statistical moments of α, such as mean, mean square, mean cube, etc., can be determined by taking derivative of the CF at the origin [17]. The p^{th} order moment is:

$$E\{\alpha^p\} = \frac{1}{j^p} \frac{d^p \Psi_\alpha(w)}{dw^p}\bigg|_{w=0} \tag{2.3}$$

The CF of the quantizer output calculated from Eq. 2.1 is:

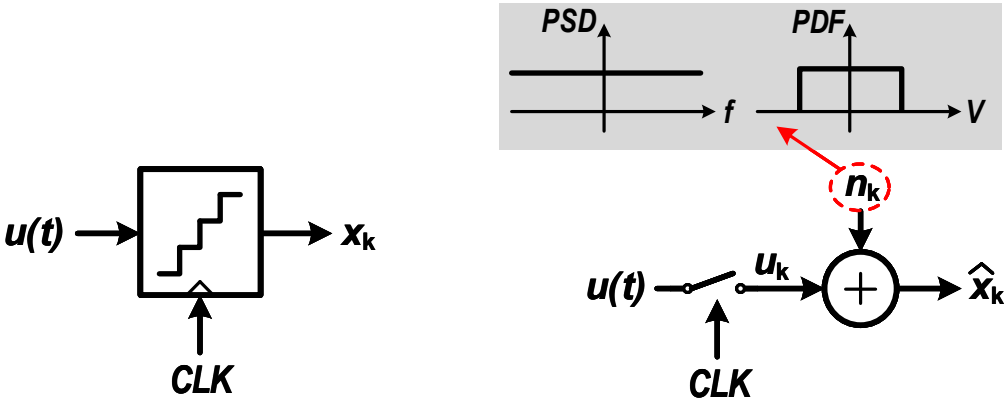


Figure 2.7: Comparison between quantization and the addition of independent noise.

$$\Psi_x = \sum_{k=-\infty}^{\infty} \Psi_u(w + kw_o)\text{sinc}\left(\frac{\Delta(w + kw_o)}{2}\right) \tag{2.4}$$

where Ψ_u is the CF of the original input PDF, and $w_o = 2\pi/\Delta$ which represents the "spatial" frequency of the input PDF that changes so rapidly it completes a full cycle within the quantization bin. Note that dimension of w is V^{-1}.

Figure 2.8 shows a sketch of the CF of the quantizer for an arbitrary input PDF and a comparison to the case of the additive independent noise. The PDF area sampling mechanism of the quantizer results in the repetition of the CF at the integer multiples of w_o and, very similar to the Nyquist's sampling theorem, the aliasing among the replica CFs happens unless Ψ_u's spatial bandwidth is narrower than w_o (-w_o/2 to w_o/2). This condition is easily met in most ADCs for high-speed link applications, since the input distribution is generally smooth, and the bins are small

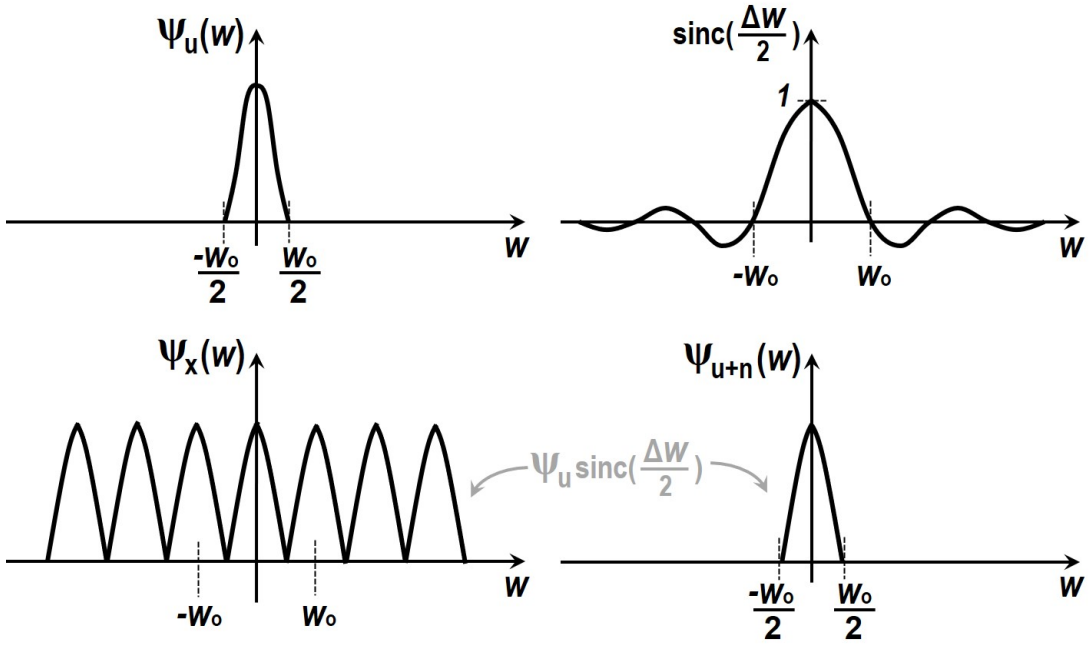


Figure 2.8: Comparison of the CF of quantizer output with the case when the quantizer is modeled by an additive independent noise.

enough, so the highest spatial frequency is smaller than w_o. Interestingly, all statistical moments of x_k and $\hat{x}_k(= u_k + n_k)$ still correspond exactly when the CF of the input is band-limited to $2w_o$ (-w_o to w_o), since there is still no overlap at w_o=0, i.e.,

$$\begin{aligned} \text{if} \quad \Psi_u(w) = 0, \qquad & \text{for} \quad |w| \geq w_o \\ E\{x^p\} = E\{(u+n)^p\} \qquad & (p = 1, 2, 3...) \end{aligned} \tag{2.5}$$

This implies that the quantizer in a system can be replaced by an additive independent noise source for the purpose of moment calculation when Eq. 2.5 is satisfied.

The distribution of the quantization error, $P_{V_q} = P_{x-u}$, can be derived by referring to Figure 2.6 again. The probability for the ADC to have a particular quantization error, ϵ, is the sum of the probabilities for the input to be ϵ away from each output level. In other words, the PDF of the quantization error can be constructed by chopping the input PDF into multiple strips whose width is Δ, and stacking and adding them as illustrated in Figure 2.9. i.e.,

$$P_{V_q}(v) = \left(P_u(v) * \sum_{n=-\infty}^{\infty} \delta(v + n\Delta) \right) \text{rect}(\frac{v}{\Delta}) \tag{2.6}$$

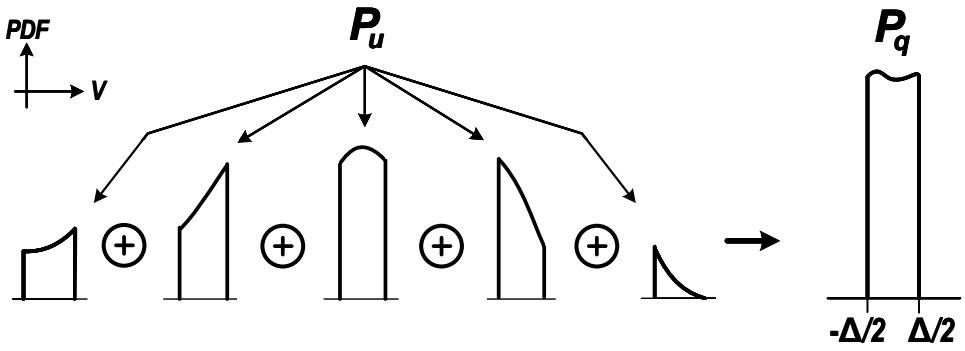


Figure 2.9: Construction of the PDF of quantization error from the input PDF.

where rect(x) is a rectangular function which has a value of unity over the range $|x| \leq 1/2$ and zero elsewhere. It can be seen that, intuitively, the quantization error converges to a uniform distribution if the input PDF is smooth and flat enough within the quantization intervals or, equivalently, Δ is small enough compared to the slope of the input PDF. To get an analytical expression, the CF of the quantization error can be calculated from 2.6.

$$\begin{aligned} \Psi_{V_q}(w) &= \left(\Psi_u(w) \sum_{k=-\infty}^{\infty} \delta(w - kw_o) \right) * \operatorname{sinc}(\frac{w}{w_o}) \\ &= \sum_{k=-\infty}^{\infty} \Psi_n(kw_o) \operatorname{sinc}(\frac{w}{w_o} - k) \end{aligned} \tag{2.7}$$

The PDF of the quantization error can be obtained by taking an inverse Fourier transform of Eq. 2.7.

$$P_{V_q}(v) = F^{-1}\{\Psi_{V_q}(w)\} = \begin{cases} \frac{1}{\Delta} + \frac{1}{\Delta} \sum_{k \neq 0} \Psi_u(kw_o) e^{-jkw_o v} & |v| < \Delta/2 \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

That is, the PDF of the quantization error has a flat top with additional ripples depending on the CF of the input. Note that satisfying the CF's band limit criterion of Eq. 2.5 also eliminates all the ripple terms and guarantees the uniform distribution of the quantization error. Proving the band-limitness for an arbitrary input distribution is not straightforward.

So far, we assumed the ADC to have uniform quantization steps. Unfortunately, however, most real-world ADCs have non-uniform size of steps mainly due to inherent device mismatch. This makes the transfer characteristic of the ADC non-linear and distorts the signal. Figure 2.10 shows two common definitions of the ADC's static non-linearity. Differential non-linearity (DNL) is defined as the deviation of the step size from its ideal size for each quantization interval. Integrated non-linearity (INL) is

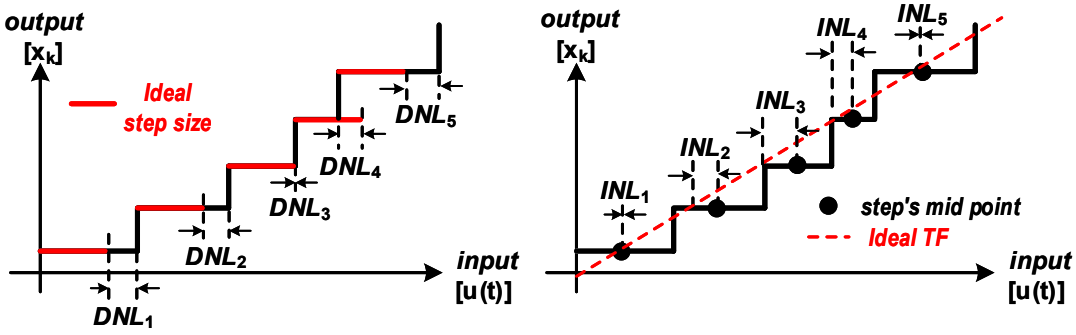


Figure 2.10: Definition of DNL and INL.

a measure of how much the transfer function of the ADC deviated from a straight line. Both non-linearities are measured in LSBs. The impact of this non-uniform step size on the PDF of the quantization error is obvious. The stack and add operation of Eq. 2.6 should be performed with different sizes of quantization intervals corresponding to the non-uniform steps. Since the source of the DNL/INL strongly depends on the detailed circuit implementation, it is challenging to get a closed-form expression of the PDF of the quantization error (V_q) for a general non-linearity.

These non-linearities also affect the spectral characteristics of the ADC. Figure 2.11 shows the magnitude of the output FFT of an 8-bit ADC with odd-order non-linearity. The INL generates extra spurious tones at integer multiples of the fundamental input frequency. The contribution of the DNL increases the noise floor and is usually indistinguishable from the quantization/thermal noise. Signal-to-noise and distortion ratio (SNDR) is defined as the power ratio between the fundamental signal (at f_o) and all the other signals.

$$\text{SNDR} = \frac{X^2(f_o)}{\sum_{f \neq f_o} X^2(f)}$$

where $X(f)$ is the Fourier transform of x_k, the output of the ADC. One effective way

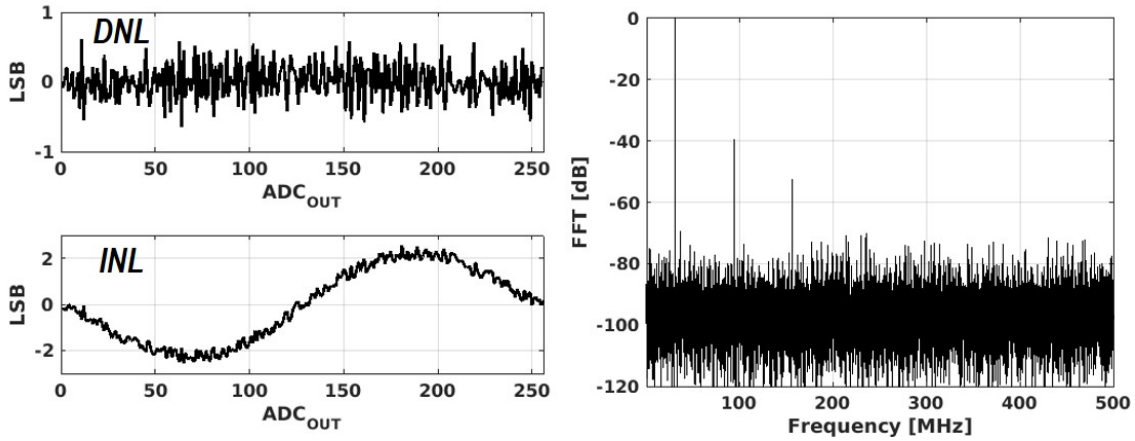


Figure 2.11: DNL/INL and 65,536-point FFT of an 8-bit ADC for a 31.4MHz sinusoidal input.

to represent the performance degradation caused by the impairments of the ADC is to consider it as an ideal ADC with lower "effective" resolution. If the SNDR is equated to the SNR of an ideal ADC with N_{eff} bit of resolution for a sine wave whose magnitude is A, the signal power would be $A^2/2$, and the quantization noise power is $(\Delta/\sqrt{12})^2$. If we assume that the input sine wave sweeps the FSR of the ADC, $2A = FSR = 2^{N_{eff}}\Delta$, $\Delta = 2A/2^{N_{eff}}$, i.e.,

$$\text{SNDR} = \text{SNR}_{eff} = \frac{A^2/2}{(\frac{2A/2^{N_{eff}}}{\sqrt{12}})^2} = \frac{3}{2}\,2^{2N_{eff}}$$

$$N_{eff} = \frac{1}{2}\,log_2(\frac{2}{3}\,\text{SNDR}) = \frac{\text{SNDR}(dB) - 1.76(dB)}{6.02(dB)}$$

The effective number of bits (ENOB), N_{eff}, is a compact and powerful metric that can measure the overall performance of an arbitrary ADC and is one of the most widely used metrics in the mixed-signal design field. However, it only contains information about the aggregated power of all impairments, and does not tell anything about the distribution of the errors, or relationship among them, which are very important in the context of the high-speed serial links. Therefore, a link designer should carefully

consider the validity or limitation of using ENOB for a given application.

The other dimension of ADC-induced error consists of time-interleaving errors. Time-interleaving is a widely-used technique to create a high-speed ADC using many slower ADCs. This design uses M ADC slices, where each slice operates at a different phase of the sampling clocks. The sampling clocks are M times slower than the target sampling rate. Ideally, the characteristics of the sub-ADC channels should be identical and the phase shift between sampling clocks should be exactly $1/M$ of the sampling clock, which is the desired sampling rate. However, in reality, there exist mismatches among the channels, such as offset, gain, and timing skew in clock distribution, as shown in Figure 2.12(a)(b). These mismatches add deterministic errors to the system

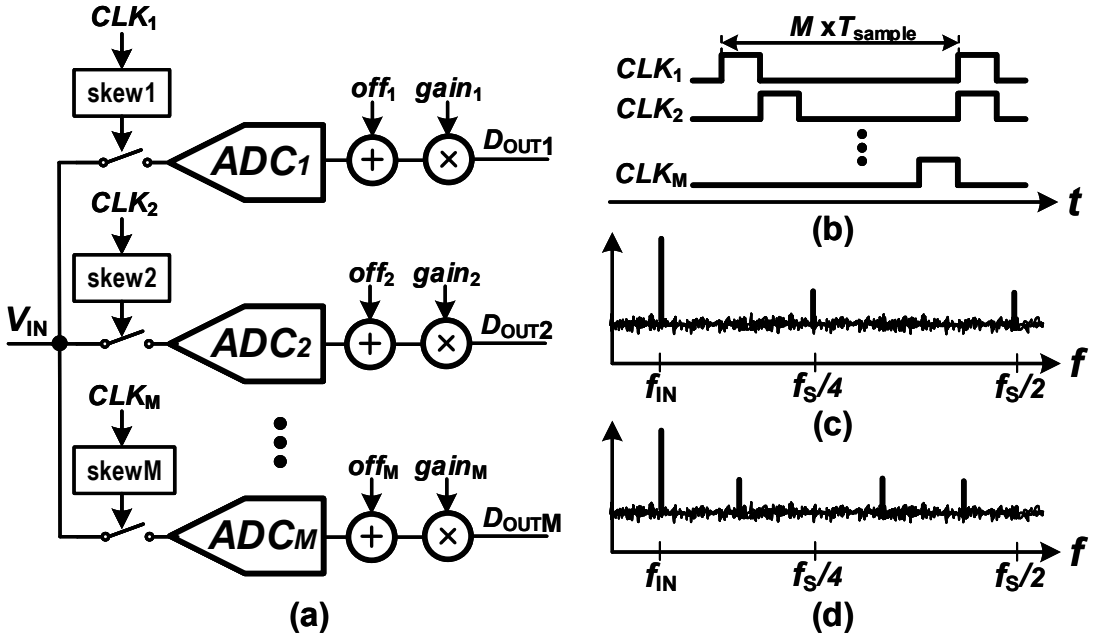


Figure 2.12: (a) Inter-channel mismatches in a time-interleaved ADC. (b) Sampling clock phases (c) Impact of offset mismatch. (d) Impact of gain mismatch and clock skew.

and may significantly degrade the performance of a time-interleaved ADC. Figure 2.12(c)(d) shows the impact of the mismatches on the spectral performance of a 4-channel ADC [18]. The offset mismatch is uncorrelated to the input frequency (f_{IN}) and causes spurious tones at integer multiples of the interleaved clock frequency (f_S/M).

$$f_{spur}^{(\text{offset})} = k \times \frac{f_S}{M}, \qquad k = 1, 2, ..., M$$

The effects of gain mismatch and clock phase skew depend on the input frequency and cause spurious tones at integer multiples of the interleaved clock frequency away from the input frequency.

$$f_{spur}^{(\text{gain,skew})} = k \times \frac{f_S}{M} \pm f_{IN}, \qquad k = 1, 2, ..., M$$

These mismatch-induced error sources in a time-interleaved ADC are hard to be merged into a single closed form variable, V_q in Figure 2.2. Fortunately, for most modern high-speed link applications, the input signal is constrained, so as long as the offset, gain, and timing mismatches are small enough, the inter-channel mismatches in the time-interleaved ADC are usually calibrated after quantization using DSP, based on long-term averaging [10][19] or per-channel FFE [20][21].

2.2.4 Jitter ($\varepsilon^{TX}, \varepsilon^{RX}$)

Jitter refers to the timing uncertainty of a periodic event, usually rising and/or falling edges of a clock signal. There exist several different definitions of jitter depending upon its application and property of interest [22]. In the context of high-speed serial links, jitter is typically considered as a zero-mean random variable whose magnitude is defined as the absolute deviation of clock edges from their ideal positions (a.k.a. absolute jitter). The TX clock jitter, ε^{TX}, and the RX clock jitter, ε^{RX}, are mutually correlated by the CDR, up to the CDR loop bandwidth, although they originate from different sources.[1] Each jitter source is correlated with itself as well

[1]The link is assumed as a plesiochronous system [23] in this dissertation.

(i.e., its frequency spectrum isn't white) since it is usually colored by a feedback clock generator such as phase-locked loop (PLL). The jitter sources contain both unbounded random component (RJ) and bounded deterministic component (DJ) depending on the clock generation and distribution scheme. The "Dual-Dirac" is a widespread method [24] to approximately model the distribution of the jitter sources by summing two identical Gaussian PFDs with different means, where the difference between the two means captures DJ, and the standard deviation of the individual Gaussian PDFs represents RJ.

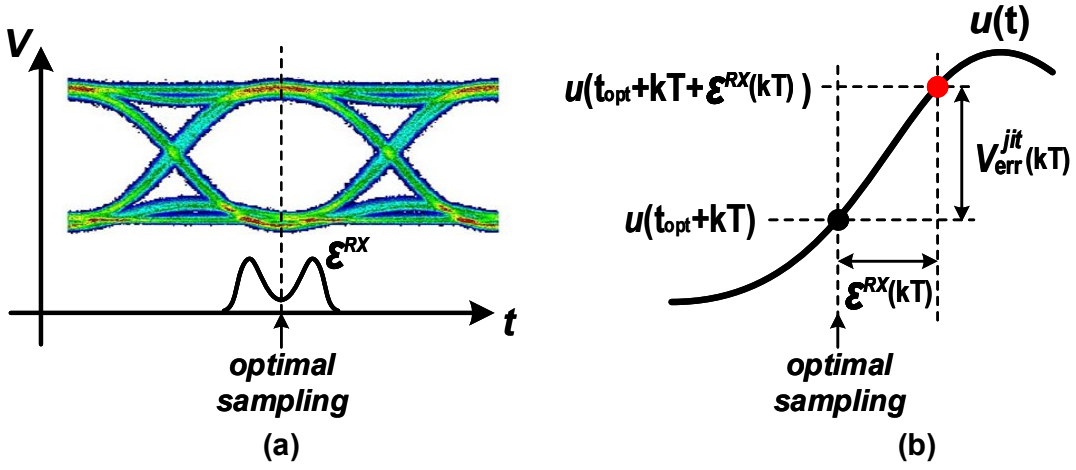


Figure 2.13: Effect of RX jitter on (a) eye width (EW) and (b) output of an ADC, using a dual-Diral jitter model.

The effect of RX jitter on link performance is straightforward. It degrades the timing margin of the sampled eye (EW) by dithering the optimal sampling position determined by the CDR, as shown in Figure 2.13(a). Figure 2.13(b) illustrates the RX jitter-induced voltage error when an ideal ADC is assumed to clearly see the effect. If the input voltage waveform of an ADC is smooth (differentiable) and $\varepsilon^{RX} \ll 1/f_{max}$, where f_{max} is the maximum frequency component of the input, the voltage error in the sampled output is:

$$V_{err}^{jit}(kT) \simeq u'(kT)\varepsilon^{RX}(kT)$$

For an arbitrary input signal, timing jitter affects the signal-to-noise ratio (SNR) of the ADC, which can be derived as [25],

$$\mathrm{SNR}_{jit} = 10\,log\left(\frac{\sigma_u^2}{-r_u''(0)\,\sigma_{\varepsilon^{RX}}^2}\right) \tag{2.9}$$

where σ_u^2 and $\sigma_{\varepsilon^{RX}}^2$ are the power of the input signal and the sampling jitter respectively, and $r_u''(x)$ denotes the second derivative of the autocorrelation function of the input. It can be seen that a slowly varying signal is less affected by sampling jitter, since it has a somewhat flat autocorrelation near zero. Note that Eq. 2.9 is reduced to the famous form of $20\,log(1/2\pi f_u \sigma_{\varepsilon^{RX}})$ when the input is a single tone, $u(t) = Asin(2\pi f_u t)$. The effect of sampling jitter on an ADC's output may be negligible when the eye is opened enough and the ADC samples its input near the flat region of the eye, as shown in this example. However, since the ADC generally samples an unequalized input, the sampling jitter-induced voltage noise does affect a system's performance margin. Again, the metric derived in Eq. 2.9 may not be accurate enough to estimate the BER of a link, since SNR does not take the distribution of the jitter into account, but only includes its total power.

The effect of TX jitter is more complicated than that of RX jitter since it interacts with the channel. While the low-frequency component of TX jitter can be regarded as equivalent to RX jitter, the high-frequency component of TX jitter is hard to analyze with the LTI system assumption, since the energy of the transmitted pulse is modulated by jitter [26][27]. The impact of TX jitter on link performance is becoming more significant as data rates increase. Without rigorous modeling and analysis of jitter, a system will not work properly or is prone to be over-designed. We will address practical solutions for incorporating jitter in Section 2.3.

2.2.5 Impact of FFE

If a passive channel is assumed to be a linear time-invariant (LTI) system, it is fully characterized by its impulse response. The received signal at the output of the channel, $u(t)$, for a transmitted bit sequence, d_k, can be constructed by a linear

combination of shifted single bit responses of the channel, $p(t)$, which is a convolution between the impulse response and transmitted waveform corresponding to a single pulse, i.e.,

$$u(t) = \sum_k d_k p(t - kT)$$

where k is the index of the transmitted symbols and T is the symbol duration. In the absence of random noise and jitter, the received signal sampled by the RX, u_m, is just $u(t)$ evaluated at $mT + \alpha$, where α represents the phase shift between TX and RX clocks. This means that we can represent $p(t)$ by $p_m = p(mT + \alpha)$. When undistorted bits were supposed to arrive the largest component of p_m is called "main cursor," and non-zero entries of p_m before and after the main cursor are called "pre cursor" and "post cursor," respectively, as shown in Figure 2.14 (a). u_m is determined by the transmitted bit sequence and the cursors of p_m.

$$u_m = \sum_{k=-N_{pre}}^{N_{pst}} d_k p_{m-k}$$

where N_{pre} and N_{pst} are the number of the pre- and post-cursors, respectively.

In our design, the FFE is a linear digital filter with a finite number of taps, whose output, y_k, is a weighted sum of the quantized samples from the ADC, $x_k (= u_k + V_q)$, as shown in Figure 2.2.

$$y_k = \sum_i C^{(i)} x_{k-i}$$

The weighting factors, or coefficients, of the FFE, $C^{(i)}$, are set appropriately so that they can eliminate the channel-induced deterministic error (ISI) in the quantized samples. The optimal coefficients of the FFE are determined using adaptive feedback, as shown in Figure 2.2, since the channel characteristics are not generally stationary.[2]

[2]Channel characteristics can be varied by external variables such as temperature, physical force applied to a PCB (or cables), and aging.

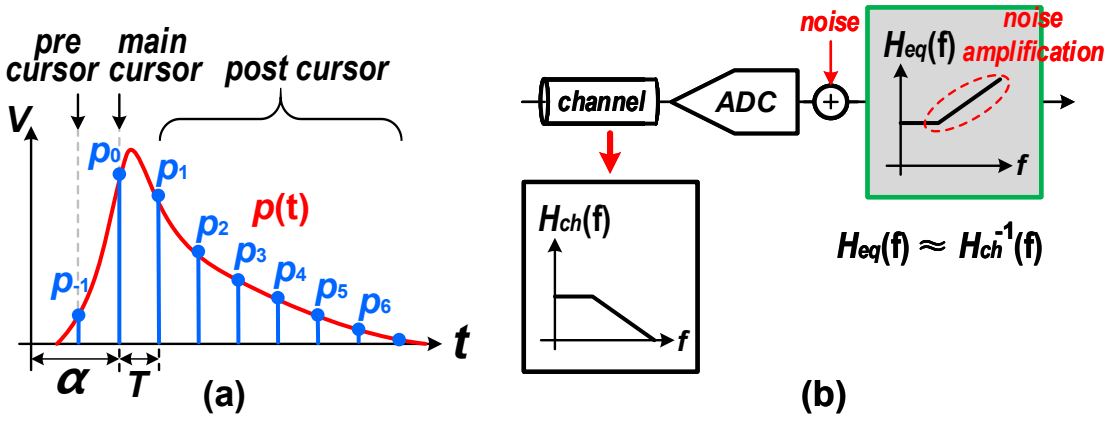


Figure 2.14: (a) Single bit response and cursor of a channel. (b) Zero forcing equalizer and noise amplification.

If one ignored noise in the system, the optimal coefficients would be a filter that performs an inverse function of the channel response, resulting in a flat frequency response overall. This approach is called "zero-forcing" [28]. The zero forcing FFE minimizes the contribution of the ISI, however, at the same time, magnifies high frequency noise in the signal as shown in Figure 2.14 (b).

To limit the noise amplification, most link designs use minimum mean square error (MMSE) for FFE adaptation [28]. The coefficients of the FFE are set so that the average power of the error, $E\{e_k\}$, is minimized, where the error is defined as the difference between the equalized signal and the desired level of the signal, i.e.,

$$e_k = y_k - dLev = \left(\sum_{i=-N_{pre}}^{N_{pst}} C^{(i)} x_{k-i} \right) - dLev \tag{2.10}$$

where $dLev$ denotes the desired level of the received signal shown in Figure 2.1, k is the time index, and i is the tap index. Since both the residual ISI and the noise contribute to the error are minimized, MMSE will leave some residual ISI if removing it causes a larger noise to be introduced. The optimal coefficients under the MMSE

criteria are obtained through a gradient descent algorithm,

$$C_{k+1}^{(i)} = C_k^{(i)} - \mu \frac{\partial E\{e_k^2\}}{\partial C_k^{(i)}} \;\Rightarrow\; C_k^{(i)} - \mu \frac{\partial e_k^2}{\partial C_k^{(i)}} \qquad (i = -N_{pre}, ..., N_{pst}) \tag{2.11}$$

Since the expectation of e_k^2 in Eq. 2.11 is not practical to get in real-world applications, it is usually approximated by its instantaneous value, often called as least mean square (LMS) [29] or stochastic gradient descent. By plugging Eq. 2.10 into 2.11,

$$C_{k+1}^{(i)} = C_k^{(i)} - 2\mu e_k x_{k-i} \qquad (i = -N_{pre}, ..., N_{pst})$$

Accurately estimating the final error distribution/spectrum at the input of the slicer including the impact of the FFE is not straightforward. Figure 2.15 shows an example of a simple case where the input of the FFE, x_k, does not contain any errors except the quantization error. The output of the FFE, y_k, is a weighted sum of N_{FFE} $(= N_{pre} + N_{pst} + 1)$ random variables, $\{x_k^{(j)} | j = 1, 2, ..., N_{FFE}\}$, each of which is a j-sample delayed version of the x_k, where j denotes the position of the taps.

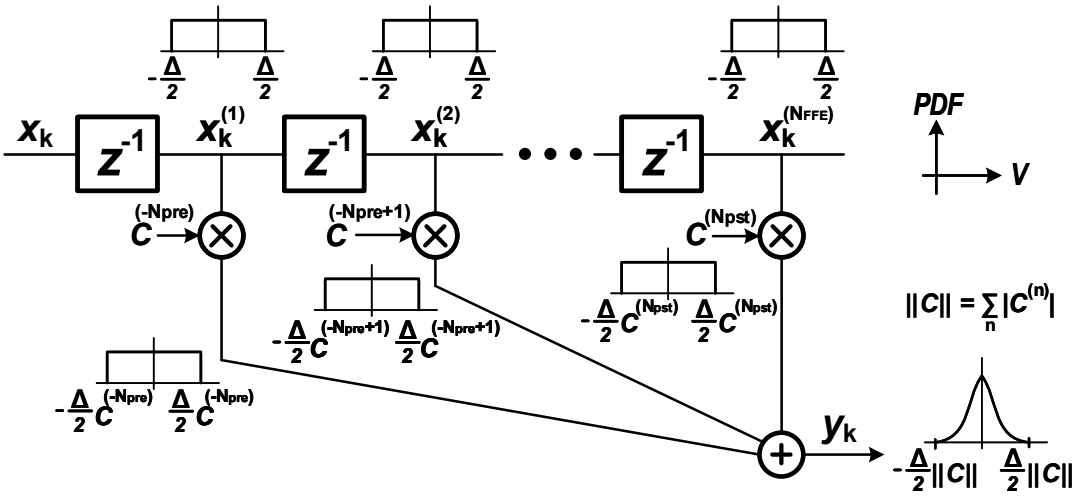


Figure 2.15: PDF modulation by FFE when the input signal only contains quantization error.

Once the PDF of the x_k is known, and the spectrum of it is white (uncorrelated), the error distribution of the y_k can be found by a weighted convolution of the PDF of x_k. Unfortunately, however, identifying the accurate PDF of the x_k, including the contributions of all the error sources mentioned in the previous sections, is a challenging task. Moreover, the overall PDF of the y_k is no longer a simple weighted convolution of the x_k, since the x_k are correlated due to jitter. Note that, in ADC-based links, not only the TX jitter but also the RX jitter cannot be decoupled from other voltage domain errors, which is not the case in conventional mixed-signal links. In other words, in mixed-signal links, the error distribution at the input of the sampler determines the BER of the link, since the sampler samples the signal, and at the same time, slices the signal. Therefore, the error distribution at a given sampling phase has nothing to do with the sampling jitter, i.e.,

$$P_y(x) = P_{y|\varepsilon^{RX}}(x|\varepsilon), P_{\varepsilon^{RX}}(\varepsilon)$$

However, in ADC-based links, the FFE located between the sampler (ADC) and the slicer couples the sampling jitter to the error distribution, making the analysis even more challenging. Accurate modeling of the impact of the FFE is usually done by a transient simulation, as we will discuss in Section 2.3.2.

2.2.6 Other Error Sources

Multi-lane serial links typically suffer from density constraints which cause significant inter-channel interference (ICI), also known as crosstalk between channels. It has been neglected in traditional links, since ISI usually predominates over ICI. However, the impact of the ICI, especially near-end cross talk (NEXT), is getting more critical, since the magnitude of the victim signal at the input of the RX is severely attenuated as the data rate increases.

Typical differential signaling with current-mode TX drivers can be well approximated by an LTI system. However, voltage-mode TX drivers with low swing for lower power consumption [30] may suffer from non-linearity; this can be a critical error source, especially for multi-modal signaling, such as PAM4.

Simultaneous switching output (SSO) noise, which is typically negligible in differential signaling, is one of the major bottlenecks in designing high-speed single-ended signaling systems such as GDDR5 [31]. Predicting system performance degradation due to the SSO noise requires accurate models for power distribution network (PDN).

Although these error sources may have considerable contribution depending on the application they are neglected or assumed to be suppressed by a properly designed system, in order to focus on the goal of this dissertation.

2.3 BER Estimation Frameworks

Creating noise and jitter specifications for various sub-components of a link is critical to achieving a target BER. Improperly budgeted link designs may overly stress one particular sub-component, lowing the overall system yield or increasing power dissipation. In the design phase, link systems may rely on a full-wave time-domain simulation involving a long sequence of random data. However, this "bit-by-bit" transient simulation is impractical for system performance prediction since the simulation time required to validate the worst case eye is prohibitively long. For example, for a link with an expected BER of 10^{-12}, the input data sequence should be at least one trillion bits long, and preferably many times longer, in order to get a reasonable statistical measure. Unfortunately, worst case analysis methodologies, such as peak distortion analysis [32] and peak jitter analysis [33], usually result in severe overdesign. For these reasons, today's link simulation tools employ either statistical eye generation or limited transient simulation with curve extrapolation. These two BER estimation frameworks are briefly introduced next.

2.3.1 Statistical Eye Estimation

Given that the transmitted bits are equally probable and independent to one another (i.i.d.), the PFD of residual ISI can be calculated by performing convolution of all individual residual ISI's PDF contributed by each bit. Cursors of an equalized pulse response with the linear equalization (FFE), p_n^{res}, are modeled by convolving

the equivalent equalizers' impulse responses with the channel pulse response. The probability mass functions (PMFs) of each bit weighted by the residual ISI will be two delta functions at 50% probability but positioned at $\pm p_n^{res}$.

$$\begin{aligned} P_{ISI_{res}}(v) &= P_{bit}^{(-N_{pre})} * P_{bit}^{(-Npre+1)} * \cdots * P_{bit}^{(N_{pst}-1)} * P_{bit}^{(N_{pst})} \\ P_{bit}^{(n)}(v) &= \frac{1}{2}(\delta(v - p_n^{res}) + \delta(v + p_n^{res})) \end{aligned} \tag{2.12}$$

where v denotes a random variable in voltage domain. Once a PDF of this bounded and deterministic component of error source (ISI) is obtained, the effect of random voltage error sources, such as thermal noise and quantization error, are included by convolving their PDFs with the PDF of ISI. Since the sampled single bit response, p_m is a function of the sampling phase (α), a family of PDFs is obtained when the sampling phase is swept over the range of 1UI and a statistical eye can be created from the PFD contour.

Incorporating the effect of jitter into this framework is not straightforward, unlike the case of the voltage-domain error sources. For instance, performance degradation caused by TX jitter is more severe than that caused by RX jitter due to jitter coloring by the passive channel as discussed in the previous section. Furthermore, certain jitter components could interact in the system, so it is inaccurate to treat them as independent variables. The majority of implementations of statistical link simulators assume a white Gaussian jitter and ignore the effect of TX jitter [34][35], or treat it like RX jitter [36], both of which are problematic approaches as data rates continue to scale higher.

The statistical simulation technique described in [26][27], called LinkLab, resolves many of these issues. The channel is modeled with its step response to incorporate pulse energy modulation effect by the high frequency component of the TX jitter as

shown in Eq. 2.13.

$$\begin{aligned} u(t) &= \sum_{k}(d_k - d_{k-1})s^{res}(t - \varepsilon_k^{TX} - kT) \\ u_m &= \sum_{k=-N_{pre}}^{N_{pst}}(d_k - d_{k-1})s^{res}(\varepsilon_m^{RX} - \varepsilon_k^{TX} + (m-k)T) \\ &\approx \left(\sum_{k=-N_{pre}}^{N_{pst}} d_k p_{m-k}^{res}\right) + n^{TX} + n^{RX} \end{aligned} \tag{2.13}$$

where $s_{res}(t)$ is the equalized residual step response of the channel. The jitters are mapped into equivalent voltage noise (EVN), n^{TX} and n^{RX}, using first order Taylor expansion. The channel coloring effect is captured by calculating the power spectral density of the EVNs with their autocorrelation matrix. However, this approach still relies on the assumption of Gaussian PFD of the jitters to estimate the overall distribution of error.

The statistical eye framework is one of the most popular ways to predict the BER performance of a link system without the need for long transient simulation. It may allow us a quick estimation of the link performance in the design exploration stage, but it is usually hard to get accurate results, unless we completely model the error sources and their complex interactions in the system. More importantly, although the framework is relatively mature for conventional mixed-signal links, it is tricky to be directly applied to the ADC-based links. The noise transfer of the FFE and the transient behavior of the interaction between the FFE adaptation loop and the CDR loop is challenging to be included in the framework. Thus, an alternative BER estimation framework is introduced in the next section.

2.3.2 Limited Transient Simulation with Curve Extrapolation

The BER for a given sampling phase is measured through a transient simulation by direct comparison between the transmitted bit from the TX and the sliced bit

by the RX only for a limited number of bits [37]. The test bench usually contains functional models of the channel and analog blocks to run the transient simulation. A BER bathtub curve, generated by sweeping the sampling phase, is extrapolated down to predict the low error probability realm. The Q-function is widely used to map a Gaussian function to a linear function which is defined as [24],

$$Q(x) = \sqrt{2}\,\mathrm{erf}^{-1}(1 - \mathrm{BER}(x))$$

where erf(x) is an error function defined as $2/\sqrt{\pi}\int_0^x e^{-t^2}\,dt$. The raw bathtub curve obtained by the limited transient simulation is linearly extrapolated in the Q-function domain and an inverse Q-function is applied to get a final bathtub curve as shown in Figure 2.16.

This framework is based on two underlying assumptions. The first is that, although the overall error distribution at the input of the slicer somewhat is complex, the extreme far-end unbounded portion of it eventually converges to a Gaussian tail no matter what the distribution of the individual errors and their interactions are. The second is that the total number of samples to be simulated should be long enough to capture all bounded portions of the errors. This assumption is often jeopardized when the pulse response has a long tail, which is mainly caused by signal reflection due to a poorly matched termination.

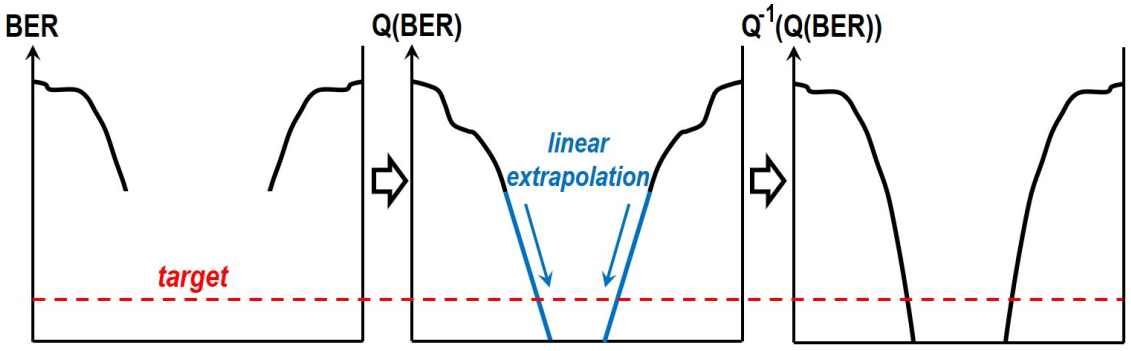


Figure 2.16: Curve extrapolation based on a raw results from the limited transient simulation.

The biggest advantage of transient simulation is that the accurate effect of the error sources with any spectral characteristic and PDF can be captured, including their complex interactions in the system. The effect of the FFE's noise amplification and the interaction between the coefficient adaptation loop and the CDR can be included in the framework as well. Moreover, especially for the case of synthesizable links, as claimed in this dissertation, the SystemVerilog description of the design used for synthesis and PnR can be directly reused in this BER estimation test bench.

2.4 Design Targets

To prove the feasibility of the proposed Open-Source PHY architecture, a 20 Gbps, differential NRZ, plesiochronous [23][38] receiver is designed and tested in this dissertation. PCIe Gen4 standard [3], which supports up to 16 Gbps of data rate with 10^{-12} of BER, is referred as a benchmark to determine the target specification of the proposed clock generator. The PCIe Gen4 defines maximum total jitter (TJ) of TX clock as 12.5ps (0.2UI), and accordingly, target RJ and DJ of the clock generator are set to 0.4ps-rms and 4ps-pkpk, respectively.[3]

The output phase of the PI should be monotonic with respect to its control input to ensure a stable convergence of the phase tracking loop. The step size of the PI should be designed small enough so that it can minimize the skew among the time-interleaved sampling clocks of the ADC. The target maximum step size of the proposed PI is determined as 1ps based on an assumption that a per-channel FFE scheme (Appendix A) is employed at the back-end.

Given the jitter specification and the maximum phase step size of the PI, the target resolution of the ADC is determined via a limited transient simulation with curve extrapolation. 20M samples of pseudo random bits are transmitted through a channel with -26dB loss at Nyquist frequency and the BER of the link is calculated by MATLAB. The TX is assumed to be a rail-to-rail voltage-mode driver without FFE, and is also assumed to be terminated by 50Ω resistance. Figure 2.17(a) shows the

[3]For 10^{-12} of target BER, the peak to peak value of a Gaussian jitter is roughly 14 times larger than its rms. i.e., $TJ_{target} = 14\,RJ_{rms} + DJ_{pkpk} \sim 10ps$ (0.2UI)

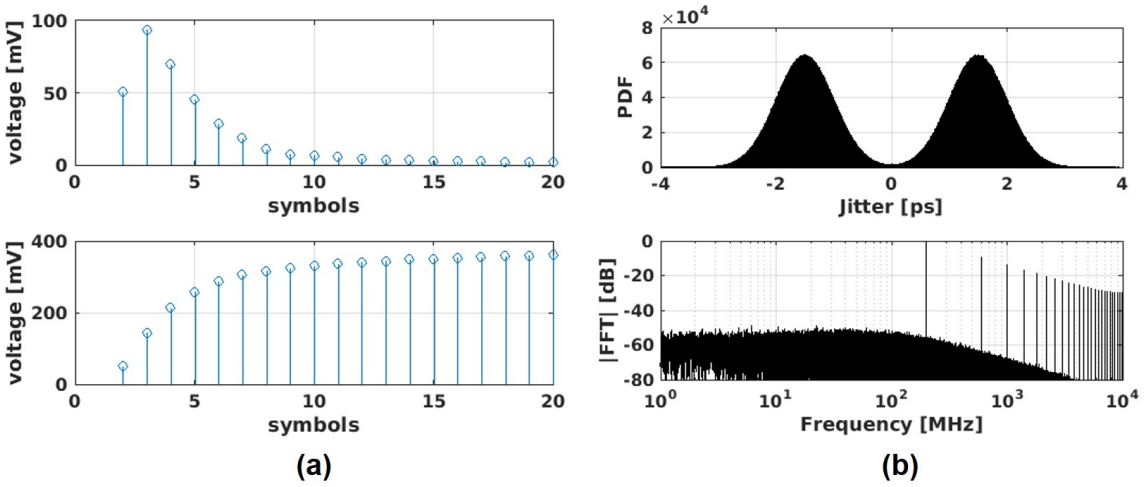


Figure 2.17: (a) Sampled single-bit response and step response of the channel used in the BER estimation and (b) PDF and spectrum of the transient jitter sequence added to the transient simulation.

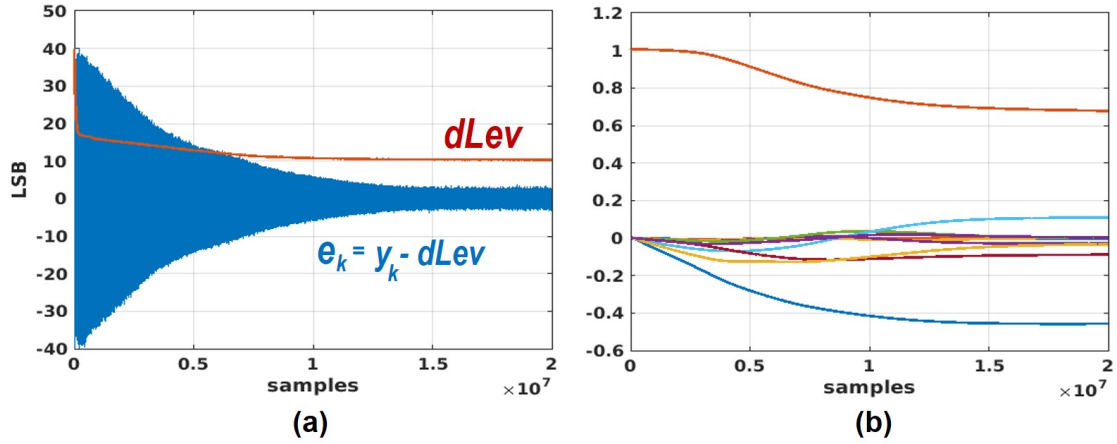


Figure 2.18: The sign-sign LMS adaptation. Transient behavior of (a) the desired signal level (dLev) and corresponding error, (b) the coefficients.

single bit response and the step response of the channel used in the simulation. The transient sequences of the TX and RX jitter are generated as shown in Figure 2.17(b). They are modeled as a dual-Dirac distribution colored by a first order low pass filter with 50MHz of bandwidth with 3ps of DJ and 0.5ps of RJ. The step response of

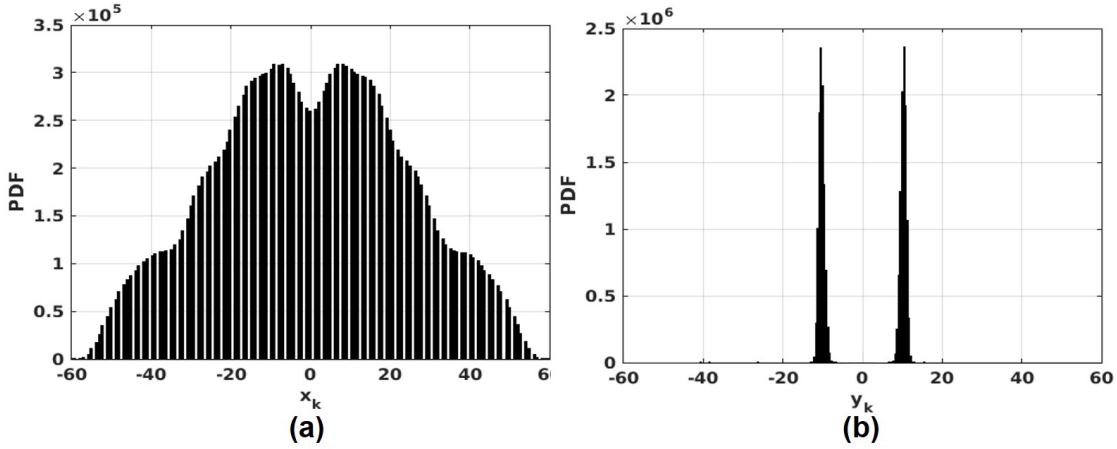


Figure 2.19: Sample distribution (a) before FFE and (b) after FFE.

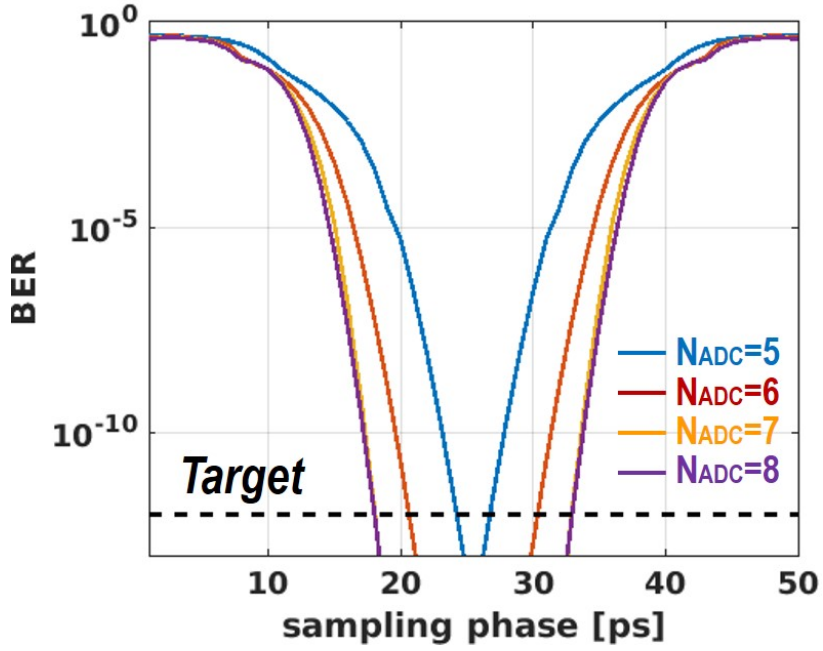


Figure 2.20: BER estimation results with respected to ADC resolution.

the channel is 50x oversampled and linearly interpolated to calculate the unequalized input samples of the ADC with incorporating the effect of both TX and RX jitter. The thermal noise added to the samples is assumed to be 1.5mV-rms according to

the SPICE simulation results of the RX front-end, which will be discussed in Section 3.3. The number of taps of the FFE was set to be 11 and sign-sign LMS adaptation was used.

Figure 2.18 shows the transient behavior of the dLev, error, and coefficients of the FFE, and Figure 2.19 shows the distribution of samples before (x_k) and after (y_k) the FFE.

The BER bathtub curves with respect to the ADC resolutions are shown in Figure 2.20. The ENOB target of the proposed ADC is determined as 5-6 bits, which allows 0.1-0.2 UI of EW, based on an assumption that the INL of the ADC will be calibrated by the post mapping with a lookup table (Section 7.1).

2.5 Summary

We introduced background on ADC-based high-speed serial links, and analyzed characteristics of error sources that affect link performance. Based on that analysis, design specifications for the proposed synthesizable AMS building blocks for the open-source PHY, namely the ADC, PI, and clock generator, were determined using the limited-time transient BER extrapolation method. Table 2.1 shows summarized specification of the AMS blocks proposed in this dissertation. In the next chapters, we will walk through the design and operating principles of each building block.

ADC	**Sampling Rate**	**ENOB**	
	20GS/s (1.25GS/s x16)	**> 5 bit**	
PI	**Frequency**	**Step size**	
	5GHz	**< 1ps**	
Clock Generator	**Frequency**	**RJ**	**DJ**
	5GHz	**< 0.4ps**	**< 4ps**

Table 2.1: Target specification of the proposed AMS blocks.

Chapter 3

Synthesizable Analog-to-Digital Converter

One of the biggest challenges in ADC-based links is to design a high-speed ADC. Since the requirements for the ADC have already been described in Section 2.4, we focus on the ADC design itself. The most popular architecture of the ADC, which is proven in commercial link products, is Flash or Successive Approximation Register (SAR). However, these are not necessarily the best choice for our synthesizable/portable design target. We have thought about what is the simplest possible architecture that gives us reasonably good performance and, at the same time, is easy to design using a conventional digital synthesis/PnR flow; our answer is a single slope ADC. The single slope ADC can be considered as a two-step data conversion: a voltage-to-time converter (V2T) converts an input voltage to a time delay first, and then the generated time delay is quantized by a following time-to-digital converter (TDC). It has been traditionally considered suitable for low speed applications, such as imagers [39], since it takes an order of $2^{N_{bit}}$ steps to perform an N_{bit} time-to-digital conversion. There were several researches that tried to boost the conversion rate up to GHz range, however, they tended to require high-precision circuitry, such as fine-grained multi-phase clock generators [40][41][42][43] and high resolution TDCs [44], which are difficult to implement by the automated digital design flow. To resolve this issue, we adopt a stochastic TDC [45] as the timing quantizer of the proposed ADC. It

can achieve high enough timing resolution without using any precision circuitry, even in high mismatch environments, such as an automatic PnR, at a cost of additional hardware redundancy.

In this chapter, we first review the conventional ADC architectures and point out what difficulties are likely to arise, in the context of high-speed links, as we attempt to push the architectures into synthesizable/portable design. Next, we introduce the proposed ADC architecture with its implementation and operating principles. Finally, we discuss time-interleaving of the ADCs along with its potential issues and solution.

3.1 Prior Art

This section reviews ADC architectures that have been reported in the literature and discusses their pros and cons. First, conventional topologies and their operating principles are briefly reviewed. We then look at a new paradigm leveraging the stochastic nature of devices to tackle problems in conventional architectures. Finally, given our portability goal, we end by reviewing several recent efforts to achieve a synthesizable/portable implementation.

3.1.1 Conventional ADCs

ADCs can be classified into two major categories: Nyquist-rate ADCs and over-sampling ADCs. The latter, mainly sigma-delta, run at a sampling speed much higher than the required frequency band to shape noise out of the frequency band of interest. They are widely used for audio systems or high precision measurement, which usually require high-resolution and low bandwidth, but hardly used for the high-speed serial link application and hence, are not covered in this section. The Nyquist-rate ADCs sample the input signal at the same rate as digitized values are generated. The major Nyquist-rate ADC topologies are flash, SAR, and pipelined, which are outlined below.

Flash

The architecture of a flash ADC is sketched in Figure 3.1. The input voltage is directly compared with reference voltages by a number of comparators. The reference voltages for each comparator are generated by a resistive ladder. All comparators with a reference voltage lower than the input voltage will trigger high and vice versa which will eventually generate a thermometer-coded representation of the input. Finally, digital logic following the comparators converts the thermometer code into a binary output.

The biggest advantage of the flash architecture is its high conversion rate and low latency. The maximum conversion rate is limited by the speed of the digital logic and the regeneration delay of the comparator. The high conversion rate requires fewer interleaving channels to achieve multi-GHz sampling rate and the low latency enables a high bandwidth in a closed loop design. However, it tends to be power inefficient

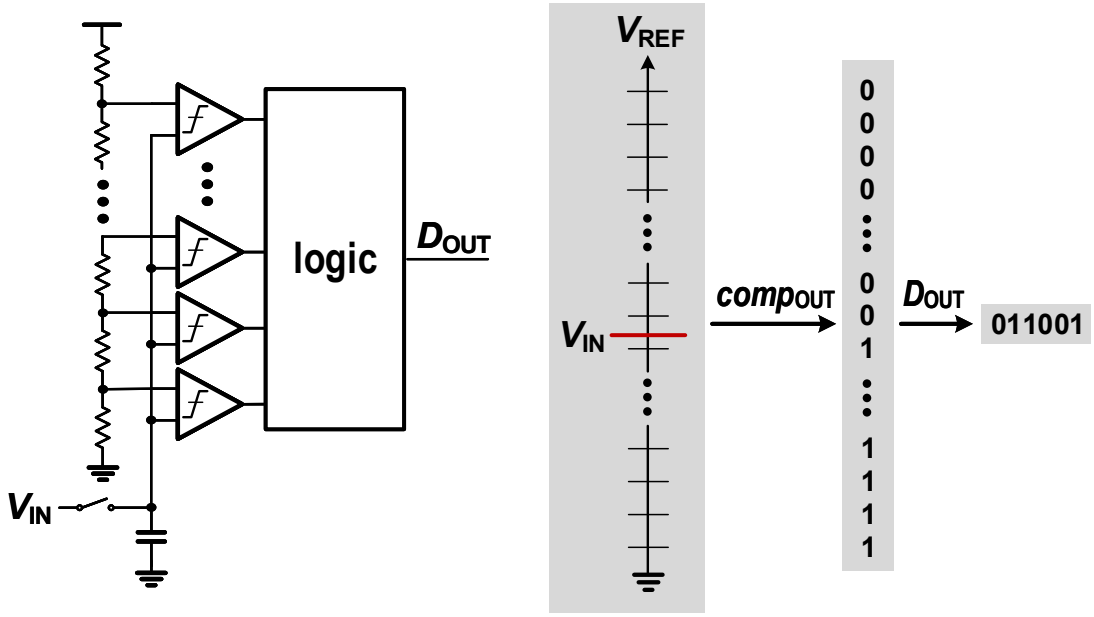


Figure 3.1: Conventional flash ADCS.

for high resolution, since the required hardware grows exponentially with the resolution. The majority of reported flash ADCs have a resolution less than 6bit. The flash ADCs have regained some interest due to the imminent shift from 2-level to 4-level pulse-amplitude modulation (PAM) signaling in high-speed data links. The time-interleaved flash design of [46] operates at 20 GS/s while maintaining outstanding power efficiency. Key to maintaining high efficiency in flash ADCs is to identify a proper offset calibration/mitigation scheme and to minimize the circuit complexity as much as possible. The design of [47] introduced a technique that generates extra decision levels using dynamic interpolation at the comparators' regenerative nodes. Instead of designing flash ADCs with near perfect thresholds, [48] proposes to adaptively control the decision levels to minimize the system's bit error rate, which is the ultimate specification of interest.

SAR

A SAR ADC converts one bit at a time by executing a binary search, starting with the most significant bit (MSB) and ending with the least significant bit (LSB). Figure 3.2 shows a conceptual schematic of a basic SAR ADC. It consists of track and hold (T/H), a voltage comparator, a digital-to-analog converter (DAC), and digital logic. It essentially forces the difference between inputs of the comparator to zero via a successive approximation. At the start of the conversion, the input voltage is sampled

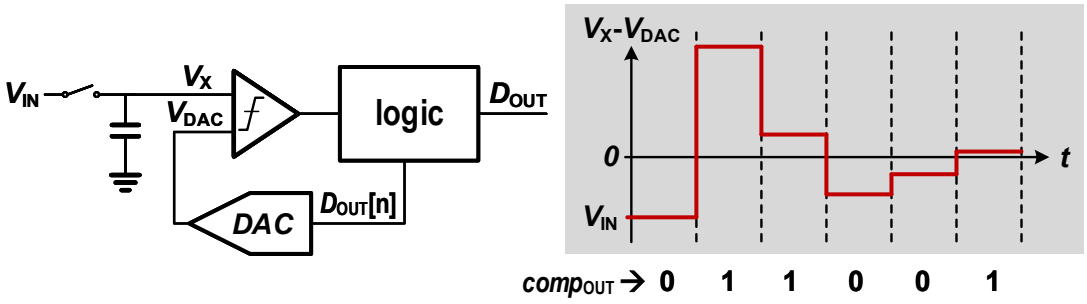


Figure 3.2: Conventional SAR ADC.

and the comparator makes its first decision, which represents the MSB. Based on the outcome of this decision, the DAC moves up or down the comparator's input for the next decision. This process repeats with progressively smaller steps (according to the binary weights) until all bits are resolved. The DAC, in conjunction with the T/H, is usually implemented by switching a capacitive array (CDAC). A classical N bit SAR ADC is approximately N times slower than the flash ADC due to its sequential operation, but only takes N comparator decisions to determine the output, compared to $2^N - 1$ for the flash ADC, resulting in higher energy efficiency. It can be designed to operate in a fully dynamic manner, which means that it does not consume static currents that would be required to operate opamps. Moreover, the unit capacitor of the CDAC, usually implemented by metal fringing capacitance, can be made as small as 100 aF (to save area and switching energy), and this still yields very accurate weighting in mass production in modern technology. The main application drivers in the past were in telephony and relatively low-speed instrumentation and measurement. By 2000, this situation had not changed significantly, and a number of alternative ADC topologies emerged to address high-end audio requirements (oversampling ADCs) as well as video and high-speed communication throughputs (folding and pipelined ADCs). After a significant rise in R&D activities on SAR ADCs over the past 20 years, this building block now stands as an attractive solution for a much wider performance range than ever anticipated.

The SAR ADCs discussed in the literature in recent years show great versatility and range from ultra low-power to ultra high-speed designs (using time interleaving). To see this, contrast the 10-bit 200 kS/s converter of [49] to the 10-bit 2.6 GS/s time-interleaved SAR ADCs that can digitize the entire cable TV spectrum [50], and finally to the 8-bit 90 GS/s part of [51]; All use very similar circuitry in their converter core.

Pipeline

A pipelined ADC consists of a number of equal or similar stages, one of which is shown in Figure 3.3. Each pipeline stage contains a DAC, a sample and hold (S/H), a summing node, and a residue amplifier (denoted by A). The input voltage is compared against the mid-point voltage and the residue is formed by subtracting the mid-point voltage if the input is in the lower half. The residue is amplified and sampled onto the next stage. Since all stages run concurrently, the pipelined ADC produces a new output in each clock cycle, trading latency for speed. The construction of these blocks typically involves an opamp or a similar circuit that is relatively slow and does not benefit as much from technology scaling as the components used in the flash or SAR architecture. Therefore, even though the pipelined ADC needs only one clock cycle per conversion, that clock cycle is longer than the cycle in a flash ADC or the sub-cycles in a SAR ADC. Over time, and with aggressive technology scaling, this disadvantage has grown in significance and it is getting harder to find examples in modern high-speed serial links. In applications which require high resolution-bandwidth product, such as wireless receivers, the pipelined ADC is still compelling option [52].

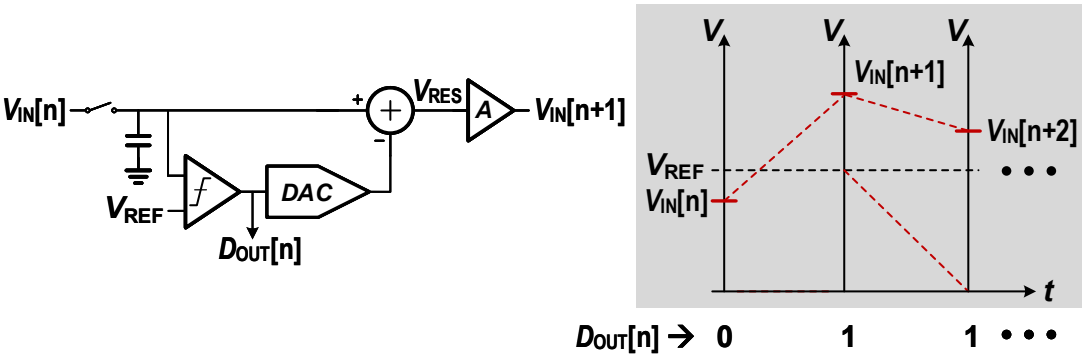


Figure 3.3: Conventional pipelined ADCs.

3.1.2 Stochastic ADCs

While digital circuits benefit from CMOS device scaling in terms of speed and power consumption, due to decreased parasitic capacitance and supply voltage, analog circuits suffer from higher process variability. This usually requires the analog circuits to consume a large silicon area and high power in order to reduce device mismatch [53], or use digital calibration techniques at a cost of design complexity [46] [47] [51] [52]. An alternative approach is to leverage the inherent randomness of the process rather than trying to compensate for it. This approach stochastically quantizes an input voltage.

The stochastic ADC is a way of implementing an ADC rather than a fixed architecture of ADC. Figure 3.4 shows an example of a flash ADC using stochastic quantization [54]. In a classic flash ADC, the reference voltages are set precisely, usually by a resistor string, such that threshold of each comparator is uniformly spaced by one LSB. In the real world, however, there are random offsets in each comparator and in the reference generation network. In a stochastic flash ADC, on the other hand, the threshold is not precisely set by design, but rather left to be random. The

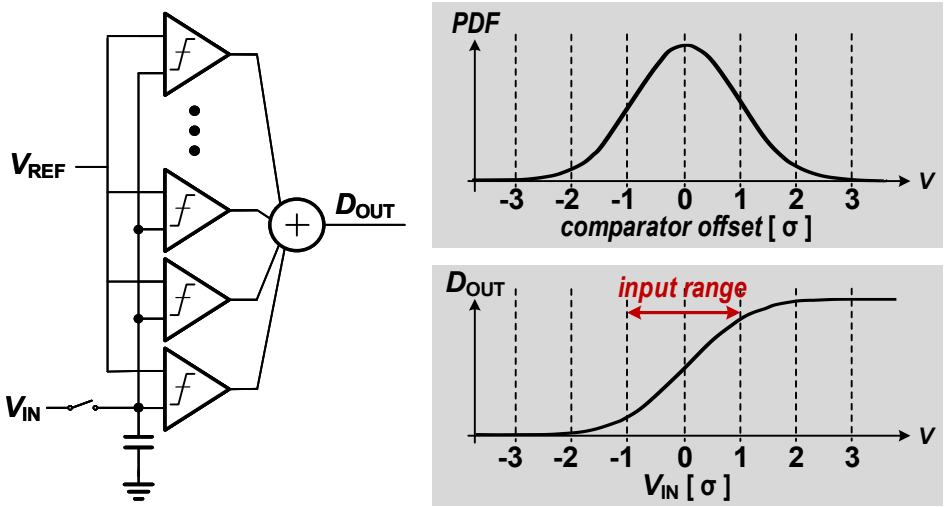


Figure 3.4: A stochastic flash ADC.

offset of the comparator can be approximated as a static random variable following a Gaussian probability density with a mean of zero and a variance inversely proportional to its device size. Although outputs of the comparators are not expected to have any order (unlike the thermometer code), the total number of comparators that trigger high is still monotonic with the input. An adder tree sums up all ones from comparators and produces the final binary output of the ADC. The total number of comparators should be large enough such that the transfer function of the ADC resembles the cumulative density function (CDF) of the random offset to the desired degree. Hence, to achieve N bit effective resolution, the stochastic flash ADC usually requires redundant comparators much more than 2^N-1.

One of the most critical drawbacks of the stochastic flash ADC is that its transfer characteristic is nonlinear due to the Gaussian distribution of the random offset. In other words, its linear input range is determined by the standard deviation of the offset distribution. This range might not be large enough even though minimum-sized comparators are used. In addition, the large number of comparators may easily cause bandwidth issues due to excessive input capacitance. 63 comparators are used in [54]

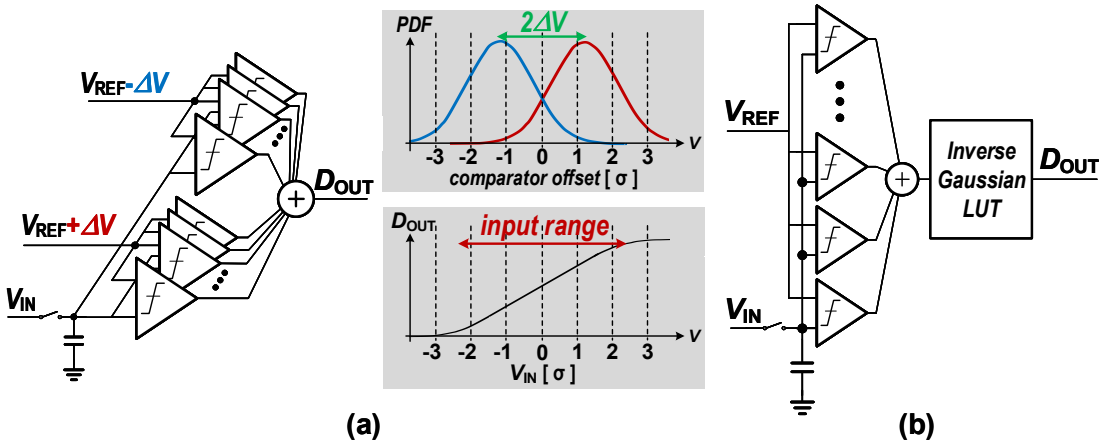


Figure 3.5: Common approaches to improve the input range of a stochastic flash ADC. (a) Multiple groups of comparators with different explicit offset voltages. (b) Inverse Gaussian post mapping.

to achieve 3.7 bits of ENOB at 1.5GS/s with only 50mV of input range, and 1,152 comparators which were chosen out of 7,680 after fabrication in [55] to achieve around 5 bits of ENOB for 18MS/s with 140mV of input rage. To improve the input range of stochastic ADCs, comparators can be separated into a few sub-groups, each of which has a different deterministic offset value, ΔV, as shown in Figure 3.5(a) [56][57][58]. Since the distribution of offset in each sub-group has the same variance but different means, the overall distribution can be close to uniform, at least near the center, as long as the ΔV is well set. Another approach, which was used in [59], employed a digital post processing to linearize the transfer characteristic via a look-up table based inverse Gaussian as shown in Figure 3.5(b). These approaches may enable the stochastic ADC to have a somewhat wider linear input range for the given number of comparators, however it is still not power efficient enough, especially for high-speed link applications.

Concept of the stochastic data conversion can be applied to time domain quantization as well. A time domain comparator ($COMP_T$) is implemented by a latch-based arbiter, or simply a flip-flop, and a number of $COMP_T$s are connected to inputs as

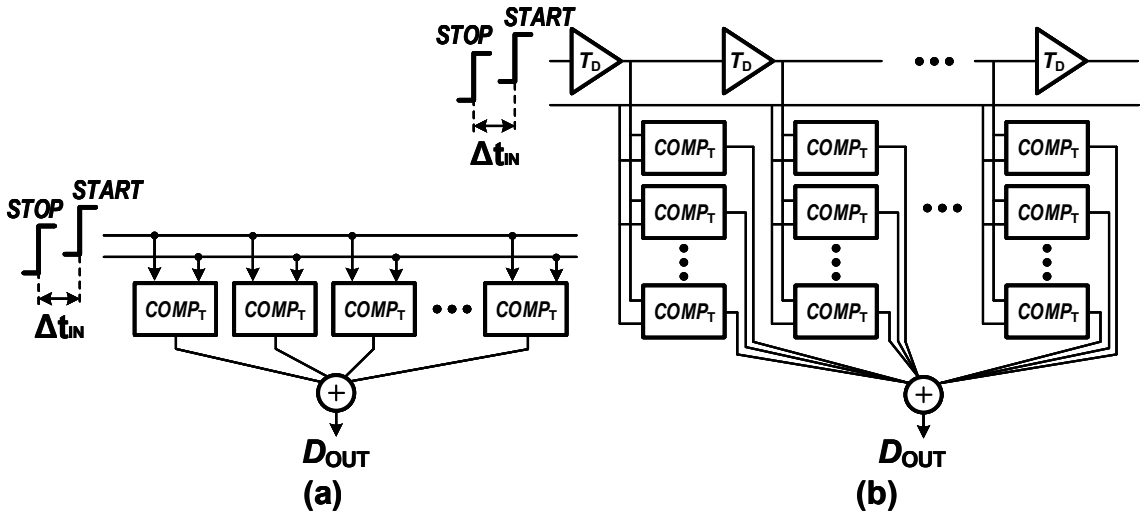


Figure 3.6: Conventional implementation of stochastic TDC.

shown in Figure 3.6(a)[60]. The time difference between the inputs, Δt_{IN}, is stochastically quantized by inherent random offset mismatch among the $COMP_T$s, t_{OFF}, that follows a Gaussian PDF. 127 $COMP_T$s are divided into three groups and each group has a different mean of the offset so that the overall linear input range is increased in [60] to achieve 7-bit resolution for 80MS/s with 20mW of power consumption.

The number of the Gaussian PFD groups can be maximized by adding a delay line and redundant $COMP_T$s as shown in Figure 3.6(b). The random offset mismatch among the $COMP_T$s sets the standard deviation of PDF of t_{OFF} and unit delay of the delay line determines the separation of each PDF group. Ito claimed this architecture in [61] but no silicon results are reported.

Drawbacks of these approaches are that the performance of the quantization strongly depends not only on the PVT variation but also on the target sampling rate. Effective resolution of the quantizer is degraded as the unit delay of the delay cell increases, or FSR decreases (due to higher data rate). These are critical hurdles for achieving the synthesizable/portable implementation of data converter. We will discuss a timing quantizer, that is immune to both the PVT variation and the data rate scaling, using stochastic modulo phase wrapping in Section 3.6.

3.1.3 Synthesizable ADCs

Recently, several attempts to design an ADC using only standard digital gates dedicated to synthesizable/portable implementations have been reported. A pseudo-differential StrongARM latched comparator, which is the most commonly used type of comparator in high-speed ADCs, can be implemented using two cross-coupled NAND gates as shown in Figure 3.7.

Although this is synthesizable, it is generally hard to use it in conventional ADCs, since the design space of device sizing is too narrow. Besides, the input common mode range has to be limited to make sure that the PMOS transistors connected to inputs are weak enough. However, it can be a very nice fit for stochastic ADCs, because the minimum-sized comparators should be used to maximize the spread of random offset, and the linear input range is narrow. The NAND-based comparator is applied to a

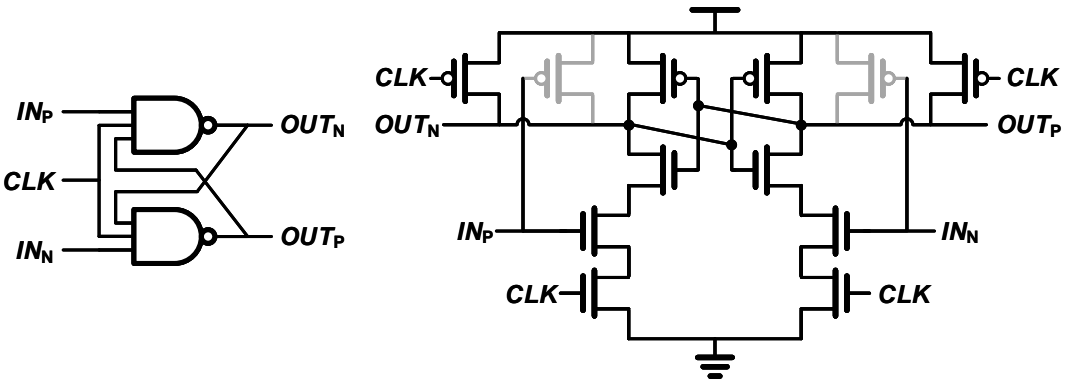


Figure 3.7: A latched comparator based on logic gates (NAND).

stochastic flash ADC in [57] and [62]. They achieved 5.2 bits of ENOB for 320MS/s with 400mV of input range and 5.7 bits of ENOB for 210MS/s with 140mV of input range respectively.

A SAR ADC can be implemented as well with the NAND-based comparator. An OAI22 gate in the standard library is used in [63] to implement a resistive DAC (RDAC), as shown in Figure 3.8. Unit cells on the bottom act as input-output shorted inverters to generate a reference voltage and unit cells on the top push up or pull down the voltage depending upon their control input. It achieved 5.4 bits of ENOB at 500kS/s with 400mV of input range. An RDAC using an inverter array and a T/H stage using power-gating switch cells were implemented in [64] to achieve 4.5 bits of ENOB at 10MS/s with 300mV of input range.

Unfortunately, these approaches tend to be vulnerable to PVT variation and random mismatches among the unit cells. This makes the size of the unit cells large and hence, severely limits their bandwidth. The next section presents the proposed synthesizable ADC architecture based on a stochastic time-to-digital converter (TDC) that is immune to the PVT variation and random mismatches of the unit cells.

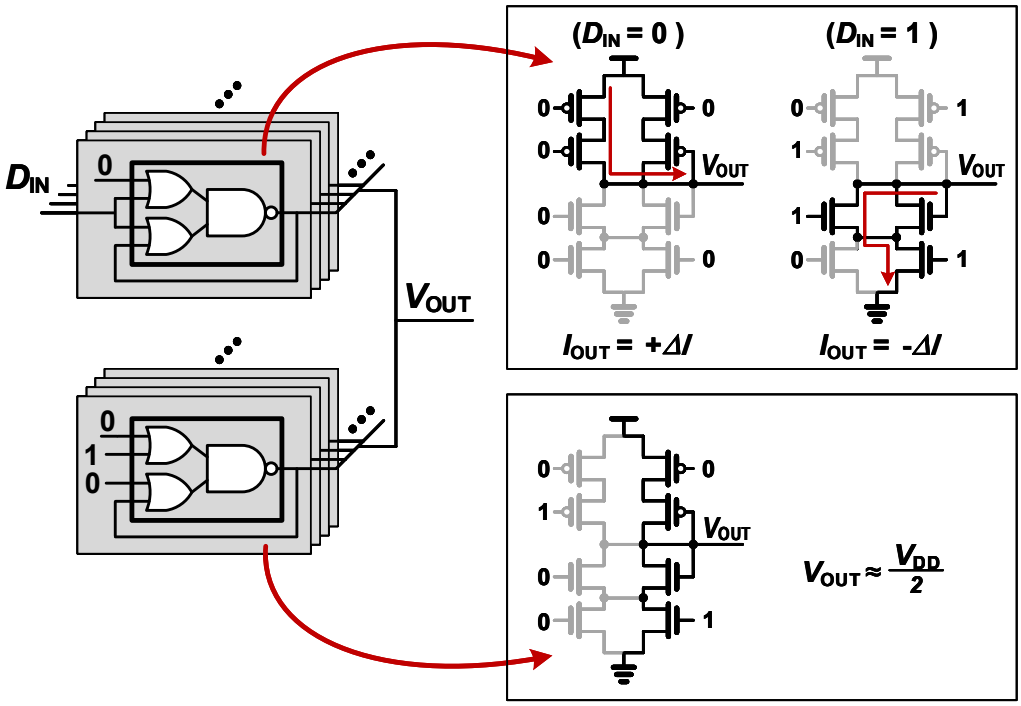


Figure 3.8: A resistive DAC based on logic gates (AIO22).

3.2 Proposed Architecture

Stochastic quantization is a very attractive way to achieve a synthesizable and portable design of an ADC since it does not fight against mismatches, but rather takes advantage of them. However, the conventional voltage-domain stochastic quantizers suffer from limited input range, due to the Gaussian nature of the mismatch, and excessive input load as discussed in the previous section.

These issues can be solved if the quantization is performed in the time or phase domain. In the voltage domain, since the mismatch distribution of each unit element (parallel comparators) has an identical mean, explicit offsets with accurate reference voltages should be added to the individual distributions to make the overall distribution close to a uniform distribution. While, in the phase-domain, the mismatch distribution of each unit element (cascaded delay cells) is naturally separated apart from each other by inherent propagation delay of the unit elements. Moreover, thanks

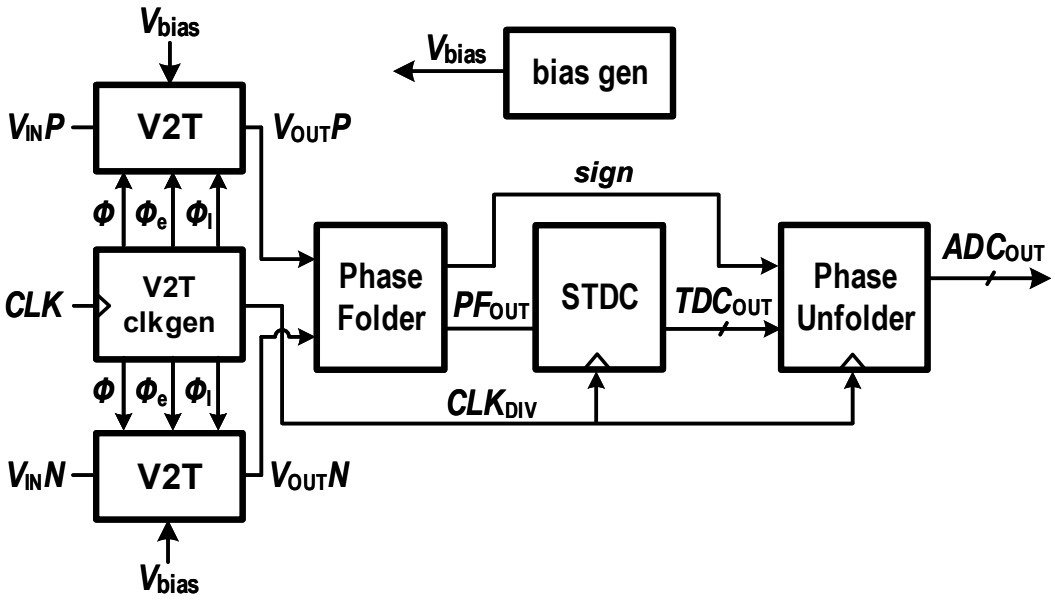


Figure 3.9: Proposed synthesizable ADC based on a stochastic TDC.

to the 2π-modulo nature of signals in the phase domain, the separation among the individual distributions can be interpolated into smaller than typical propagation delay of the delay cells. Therefore, the phase-domain stochastic quantizer is ideally suited for synthesis and can achieve better performance than the voltage-domain stochastic quantizers with a given amount of redundancy. One potential issue of the phase-domain quantizer may be the linearity of the initial domain conversion from voltage to phase. However, in high-speed serial link applications, the required ADC resolution is only around 6 bits, which makes the linearity requirements easy to meet.

Figure 3.9 shows the architecture of the proposed stochastic ADC. It consists of two voltage-to-time (V2T) converters, a V2T clock generator, a bias generator, a phase folder (PF), a stochastic time-to-digital converter (STDC), and a phase unfolder. The V2Ts encode the differential input voltage to the time difference between their output pulses, $V_{OUT}P$ and $V_{OUT}N$; the PF folds these two pulses into an unsigned pulse (PF_{OUT}), along with a sign bit ($sign$). The STDC quantizes the width of PF_{OUT} and the phase unfolder generates a final signed binary output of the ADC.

The bias generator and the V2T clock generator provide the required bias voltage and sampling clock phases for the V2T, respectively. The next sections describe the detailed implementation and operating principles of these building blocks.

3.3 Voltage-to-Time Converter

Each V2T first connects its input voltage, V_{IN}, to C_S when Φ is high, as shown in 3.10(a). Since input of the NOR gate is pulled to VDD, V_{OUT} goes low and VDD-V_{IN} is stored on the capacitor. When Φ falls, this voltage (V_C) is ramped down toward ground by a current source as shown in 3.10(b). A digital buffer generates an edge when V_C crosses its logic threshold voltage. The first gate of that buffer is a 3-input NOR whose inputs are tired together to position the threshold below VDD/2. The maximum input voltage is limited to VDD/2 so that V_C is guaranteed to initially be above this switching point. The switch cells in the design consist of a simple CMOS transistor pair and the same cell is used for the T/H stage of the time-interleaving receiver (Section 7.1). The current source consists of unit current cells that are

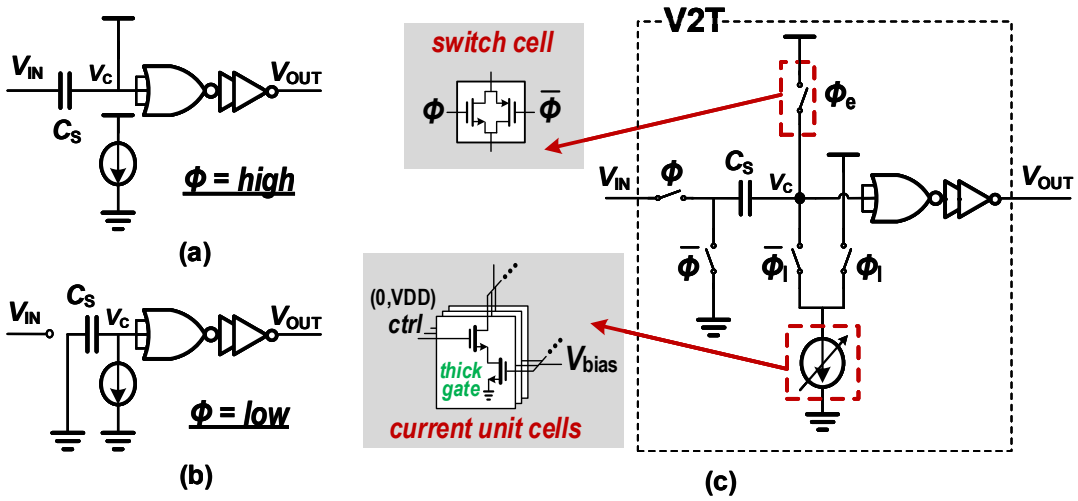


Figure 3.10: Proposed voltage to time converter.

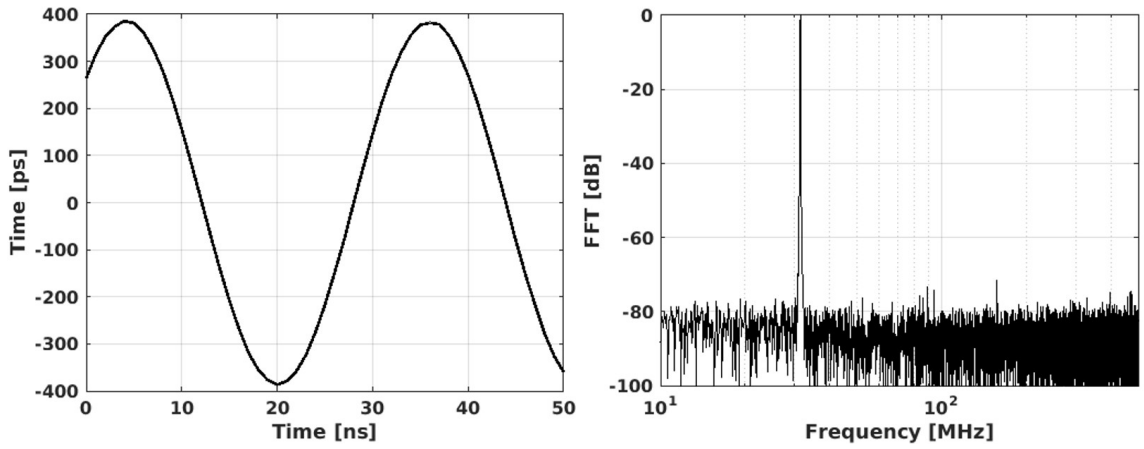


Figure 3.11: SPICE noise simulation results of the V2T. (a) Transient output time difference of the differential V2Ts and (b) its spectrum.

implemented by stacked thick-gate NMOS transistors to achieve sufficient output resistance. The larger threshold voltage of the thick-gate devices also enables us to use the supply voltage (logic core VDD) as a bias voltage of the cascode transistors, simplifying the design. The cascode transistor is turned off when its control is low and saturated when the control is high. These unit cells are in turn controlled by the output of a synthesizable bias generator which is shown in Section 3.5. The gain of the V2T can be digitally adjusted by enabling/disabling the unit current cells that are summed to produce the discharge current. The gain should be adjusted so the minimum V_{IN} (0V) produces a falling edge at the output of the buffer just before $\bar{\Phi}$ falls and the sampling phase begins again. Thus, optimal performance occurs when $\bar{\Phi}$ is much longer than Φ. The bottom plate of the capacitor is sampled by a slightly earlier phase, Φ_e, to minimize errors due to the input-dependent charge injection, and the current source initiates the discharging by a slightly later phase, Φ_l, to ensure a stable settling time for the charge redistribution. Throughout the entire design described in this chapter, the switch cell and the current cell in the V2T are the only parts that require conventional analog design efforts to create two new standard cells.

Figure 3.11 shows the SPICE noise simulation results of the V2T. A 400mV-pkpk, 31MHz sine is sampled by a 1GHz clock and an 8,192-point FFT is done for

the differential output of the V2T. SNR is 47dB and the corresponding input-referred noise is about 1.1mV-rms.

3.4 Phase Folder

The PF converts the two rising edges generated from the V2Ts into a single pulse whose width is the time difference between the input edges. The implementation of the PF is similar to that of a phase-frequency detector (PFD) in a charge-pump PLL, as shown in Figure 3.12. The output of the PF, PF_{OUT}, includes a small offset delay, D_{OFF}, which is intentionally added to ensure a reliable pulse width even when the differential input is zero. While this added delay makes the output linear and robust, it must be removed by the phase unfolder stage (Section 3.7) to generate the actual output of the ADC. A latch-based arbiter produces a sign bit depending on the lead-lag condition of the input edges. The signal waveforms of the V2T and the PF with

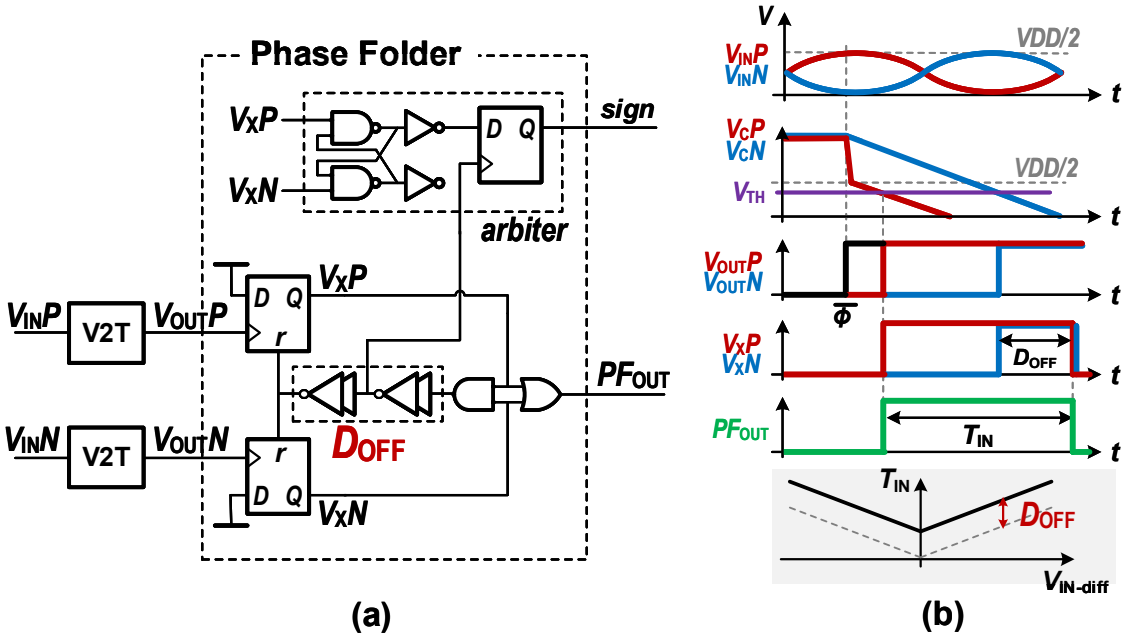


Figure 3.12: (a) Proposed phase folder. (b) Waveforms and transfer characteristic of the V2T and phase folder.

their overall transfer characteristic are shown in Figure 3.12(b). The pulse duration of PF_{out} determines the FSR of the STDC that follows. In the implementation described in Chapter 7, the maximum pulse duration of PF_{OUT} is slightly larger than 400ps out of 800ps of the sampling clock period, and the minimum pulse duration is slightly less than 100ps (i.e., D_{OFF}). Since the folding (and unfolding) feature doubles the dynamic range, the effective signal range of the following STDC is 75% of its full-scale (2*(400ps-100ps)/800ps). PVT variation of the minimum pulse duration might slightly change the absolute input range of the STDC, but does not affect the differential signal range and quantization results because it is a common-mode deviation.

3.5 Bias and Clock Generator

Using 4-input NAND gates and tying one of their inputs to ground provides a controllable resistance, or controllable current source (depending on the output voltage), and a 2-input NAND gate, with one input shorted to its output and the other input is tied to VDD, creates a diode-connected NMOS. With these NAND gates, a DAC is implemented as shown in Figure 3.13. Minimum-sized 4-input NAND gates

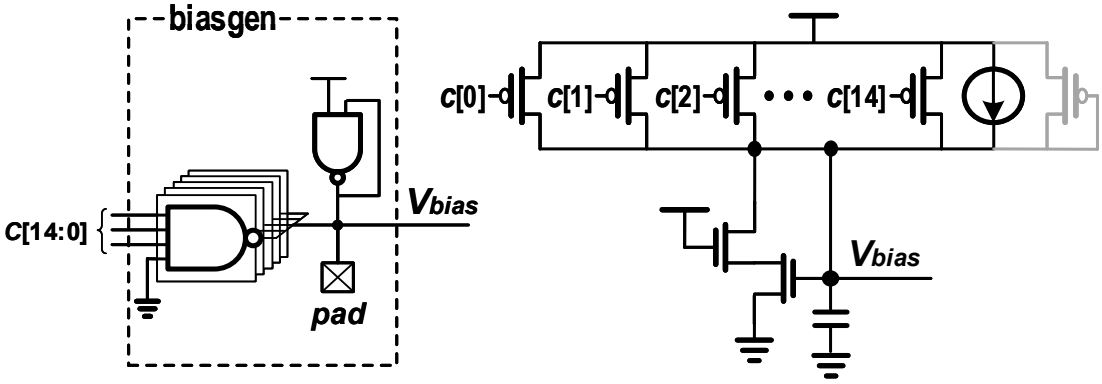


Figure 3.13: Proposed bias voltage generator and its transistor-level equivalent circuit.

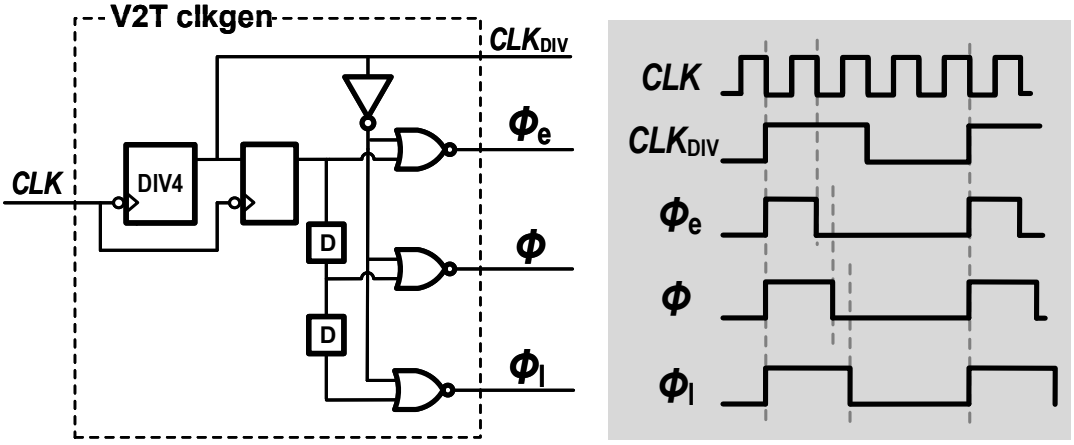


Figure 3.14: V2T sampling clock generator.

and a 12x 2-input NAND gate are used in our design for a nominal output voltage of 0.6V. Although this DAC is non-linear and inaccurate, it is enough to serve as a bias voltage generator for the V2T as long as the output node, V_{bias}, is fairly decoupled from noise. Attaching standard digital decoupling capacitor (DCAP) cells to the output node is not an efficient approach, since the capacitance of the standard DCAP cell is reduced for voltages lower than VDD, and because it is hacky to use the cell for a node isolated from the power rail from a layout perspective. Instead, standard analog PAD cells are attached to the output node to take advantage of their huge parasitic capacitance.

The V2T clock generator provides various clock phases required for the intended operation of the ADC. The unit uses a 5GHz input clock, CLK, which is divided by four to provide the input clock of the STDC, CLK_{DIV}(1.25GHz). This clock is used to generate a sampling clock for the V2T with a 25% sampling phase and a 75% ramping phase. The early/late versions of the sampling clock are generated using small delay cells, denoted as **D** (two NAND gates) in Figure 3.14. The V2T clock generator is the most timing-critical block in the proposed open-source link architecture and detailed implementation strategies for it are discussed in Chapter 6.

3.6 Stochastic Time-to-Digital Converter

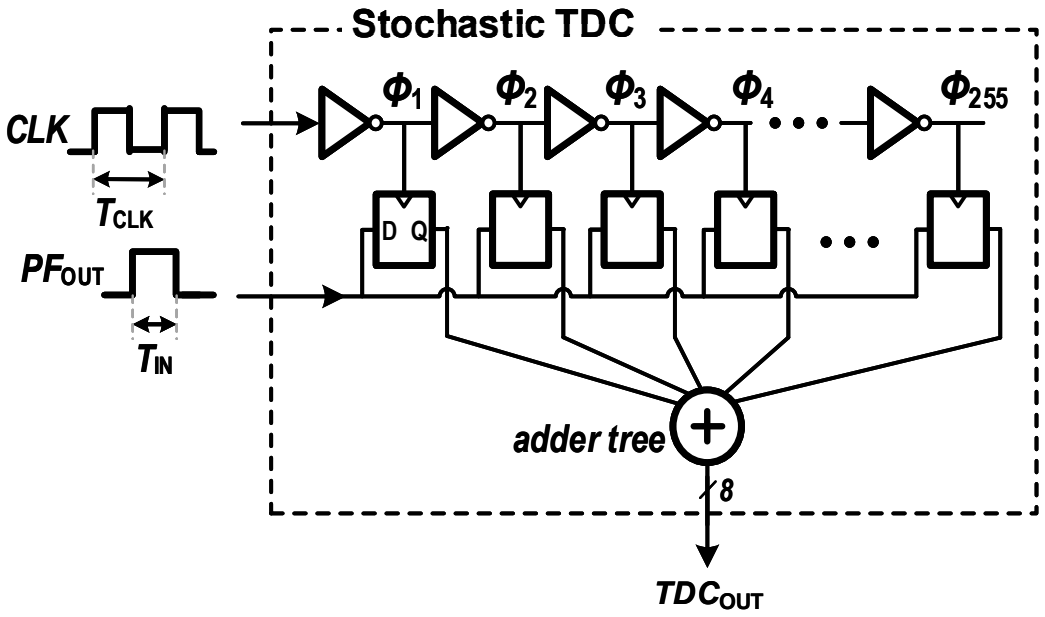


Figure 3.15: Proposed synthesizable stochastic time-to-digital converter

The STDC quantizes the width of PF_{OUT} (T_{IN}) by counting the number of delayed clock edges it contains. These clock edges are generated from 256 non-precise unit inverters, and an adder tree generates the binary result, as shown in Figure 3.15. A divided clock from the V2T clock generator (1.25GHz) propagates through the inverter chain, generating a distribution of clock edges that is quasi-uniform in the phase domain because the period of the clock is uncorrelated to the delay of each inverter as illustrated in Figure 3.16. The phase-domain position of the n-th generated clock edge can be represented as:

$$t_{\Phi_n} = \sum_{k=1}^{n} D_{inv}[k] \ (mod\ T_{CLK}) \quad (n = 1, 2, \ldots, N_{inv})$$

where $D_{inv}[k]$ is the delay of the k_{th} unit and N_{inv} is the total number of units in the chain. Automated PnR and inherent device mismatches, as well as jitter and PVT

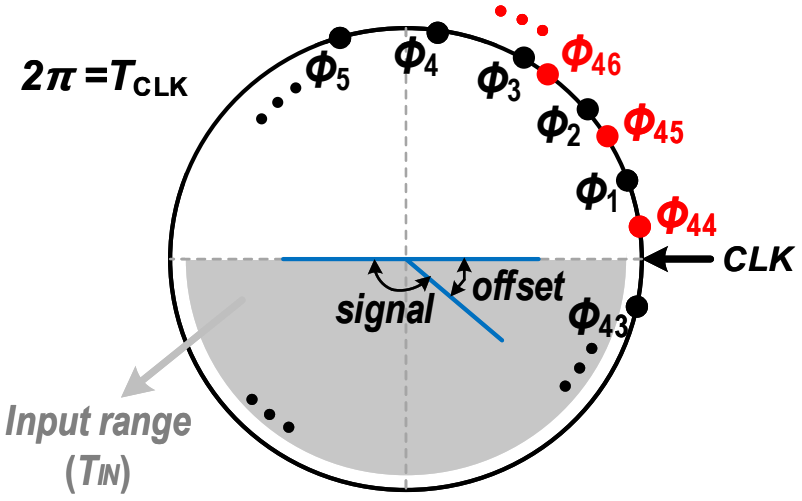


Figure 3.16: Phase domain distribution of the generated clock edges. This figure ignores the signal inversion (phase shift) that occurs at each stage.

variation, cause both static and dynamic uncertainty of the position of the edges:

$$
\begin{aligned}
D_{inv}[k] &= D_{inv0}[k] + J_{inv}[k] \\
D_{inv0} &\sim \mathcal{N}(D_{inv\_TYP}, \sigma_{D_{inv}}) \\
J_{inv}[k] &\sim \mathcal{N}(0, \sqrt{k}\sigma_{J_{inv}})
\end{aligned}
$$

where D_{inv_TYP} is the typical delay of the inverter units, $\sigma_{D_{inv}}$ is the standard deviation of static delay mismatch among the inverter units, and $\sigma_{J_{inv}}$ is the rms jitter contributed by an inverter unit. While the static mismatch among units is modeled as a Gaussian distribution, that model is not necessarily accurate because variation due to automated PnR can dominate the static mismatch (9.5-15.2ps in an extracted simulation).

Fortunately, the performance of this design is not sensitive to the exact statistics of static mismatches because of the effective randomization caused by the modulo-2π

operation.

Figure 3.17(a) shows the impact of static mismatch on the performance of the STDC when $N_{inv} = 256$ and the rms jitter of an inverter is 0.1ps, which is obtained from a transient noise simulation of the proposed STDC. As expected, the STDC shows poor performance when there is no mismatch at all and T_{CLK} is an integer multiple of the typical inverter delay. In this case, the generated edges periodically overlap with each other and the effective number of edges is reduced to T_{CLK}/D_{inv_TYP}. The presence of random static delay mismatch, combined with the modulo nature in phase generation, eliminates the periodicity and breaks the correlation between T_{CLK} and unit delays so that the STDC can achieve better performance. Moreover, the result of the STDC is not affected by the amount of static delay mismatch or the typical delay of the unit cell, because the quantization is not affected by individual edge positions or the relationship among the edges, but only by the total number of edges within a given time window. This is counterintuitive in comparison to conventional TDCs.

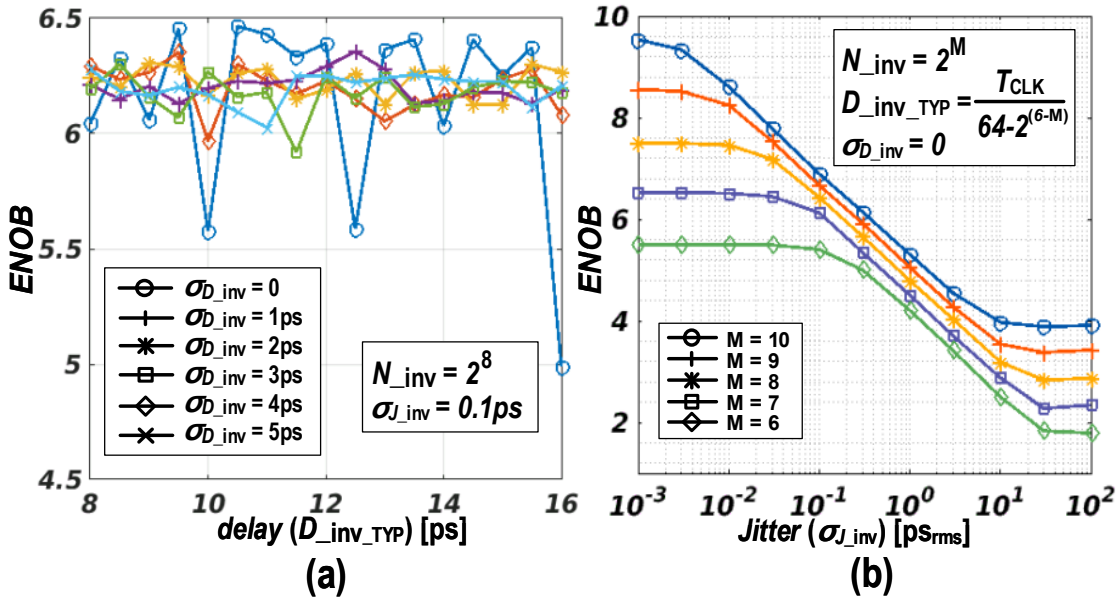


Figure 3.17: Simulation results of the STDC (a) ENOB with respect to the rms jitter of an inverter and the total number of inverters (b) ENOB with respect to the typical delay of inverters and mismatch among inverters.

For the conventional TDCs, resolution (LSB step size) is proportional to the minimum gate (inverter) delay, which scales with process but is independent on the clock speed. However, the FSR of the TDC is inversely proportional to the clock speed. As a result, the ENOB of a conventional TDC is enhanced with process scaling but degraded at a higher data rate. In contrast, the resolution of the proposed STDC is independent of the gate delay, as long as the total number of inverters stays the same, but proportional to the clock speed, because it basically quantizes the ratio between an input time difference to the clock period by the total number of delay units. As a result, the ENOB of the STDC stays almost the same against both process and data rate scaling. Due to this immunity against device mismatch, PVT variation, and data rate, the proposed STDC matches the needs of an open-source design and an automated PnR flow.

Figure 3.17(b) shows a simulation of the impact of jitter on the performance of the STDC when T_{CLK} is 800ps. In this simulation, zero static delay mismatch is assumed, and all the generated edges are uniformly distributed over a unit cycle of the input clock to see the isolated effect of jitter. The performance achieved by increasing the number of delay cells is bounded by jitter: while more cells create more clock edges, increasing potential resolution, the jitter of these additional edges also increases. When this accumulated jitter is considered, each t_{Φ_n} becomes a probability distribution with a sigma proportional to $\sqrt{n}$. As a result, the overall probability distribution of all N_{inv} generated edges is the modulo-summation of N_{inv} individual

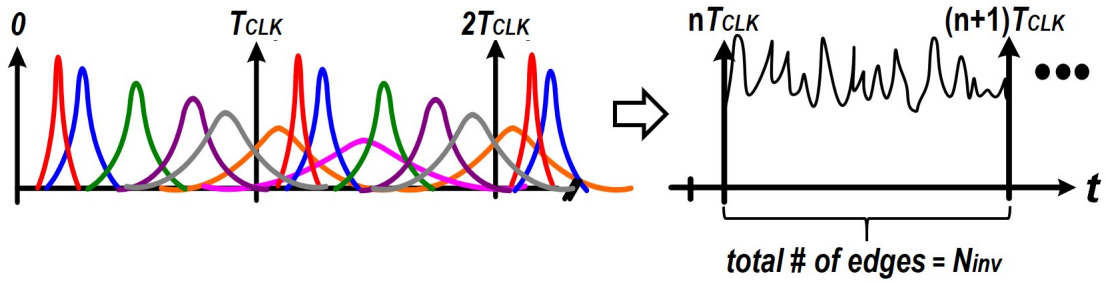


Figure 3.18: Modulo summation of PDF of accumulated jitters.

probability densities as shown in Figure 3.18. If the overlap between these distributions is small, counting the total number of generated edges that exist within a given input time difference, as the STDC does, will yield a result with low variance and a resolution proportional to N_{inv}. As the amount of jitter increases, the overall density of the generated edges converges to a uniform distribution. In this regime, the variance of the measurement becomes large: effectively, each clock edge becomes a weighted coin toss, and the measurement (D_{OUT}) for an input time difference (T_{IN}) follows a binomial distribution, independent of the amount of jitter:

$$\binom{N_{inv}}{D_{OUT}} (T_{IN}/T_{CLK})^{D_{OUT}} (1 - T_{IN}/T_{CLK})^{(N_{inv} - D_{OUT})}$$

In this case, the required number of units to achieve N-bit performance is $2 \cdot 4^N$ [55], which is too inefficient in terms of power and area for most applications.

3.7 Phase Unfolder

The phase unfolder removes the offset delay added by the PF (D_{OFF}) and converts the unsigned value back to a signed value according to the sign decision from the PF, as shown in Figure 3.19. Unfortunately, the offset delay is PVT-dependent and must be actively corrected. The average code density at zero will be higher than that of its neighboring codes if the estimation of D_{OFF} is too high and vice versa. Thus, the absolute value of the offset delay that should be subtracted can be calculated through a background adaptation loop, based on the histogram of output codes from the ADC (ADC_{OUT}), as shown in Figure 3.20. This offset correction takes advantage of the fact that ISI of a band-limited channel and the limited consecutive identical digits (CID) coding of most high-speed link standards make the output code density of the ADC fairly uniform near zero. In this design, 2^{20} samples are collected to calculate the histogram, which is enough for most modern serial link standards.

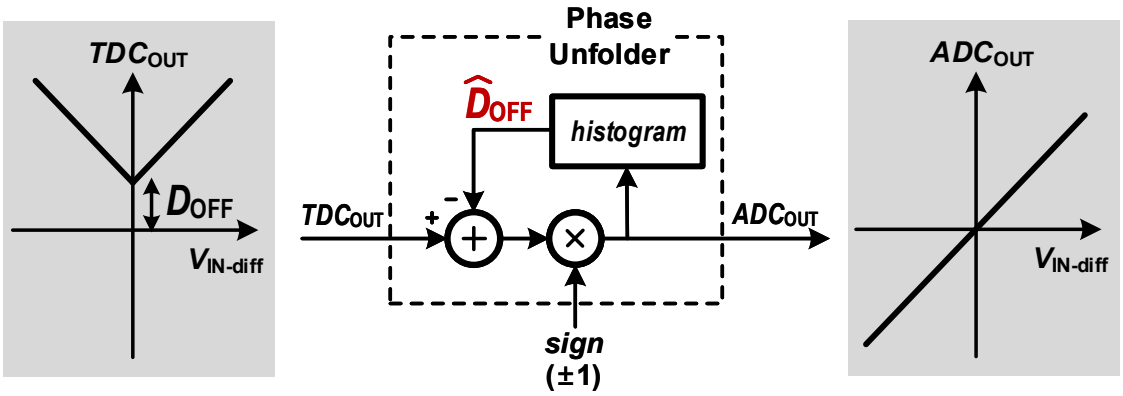


Figure 3.19: Proposed phase unfolder.

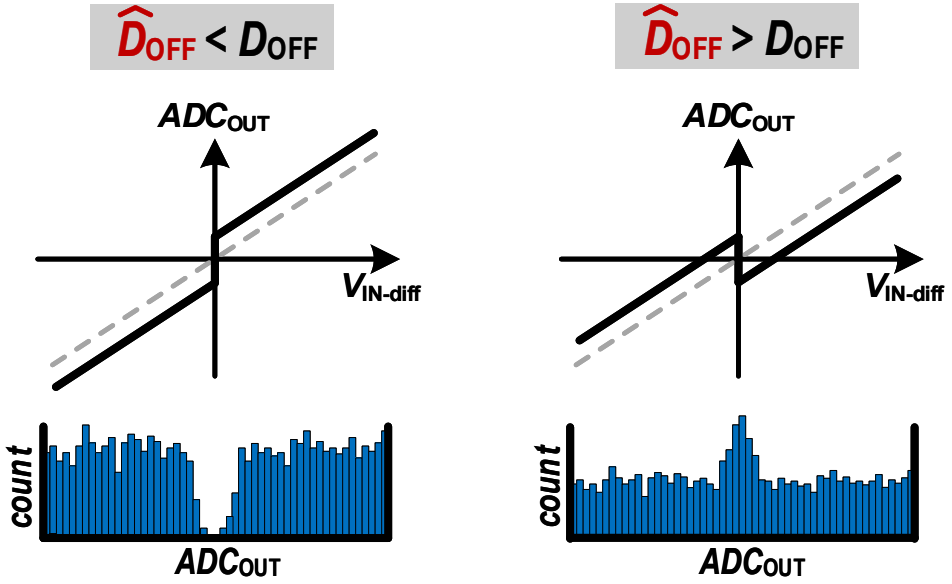


Figure 3.20: Folding offset estimation based on output histogram.

3.8 Time-Interleaving ADC

Sixteen stochastic ADCs are time-interleaved to support a 20 GS/s conversion rate. Four groups of ADCs, each of which consists of four ADC slices, are connected to individual switches (SW0-SW3) without inter-stage buffers, forming a two-stage passive track-and-hold (T/H) structure (front-end switches in each ADC slice act as the second-stage) as shown in Figure 3.21(a). 5GHz quadrature clocks drive the first stage switches.

Although appropriate timing constraints are set to minimize mismatches throughout the design, as shown in Figure 2.12, they may not be sufficient to achieve the level of performance required by users. The fundamental approach of this work is to build an architecture which performs reasonably in the mismatch-vulnerable circumstances of automated PnR. Since the STDC is inherently immune to mismatch, it could be designed simply focused on power and area. The V2T suffers from the mismatch, but

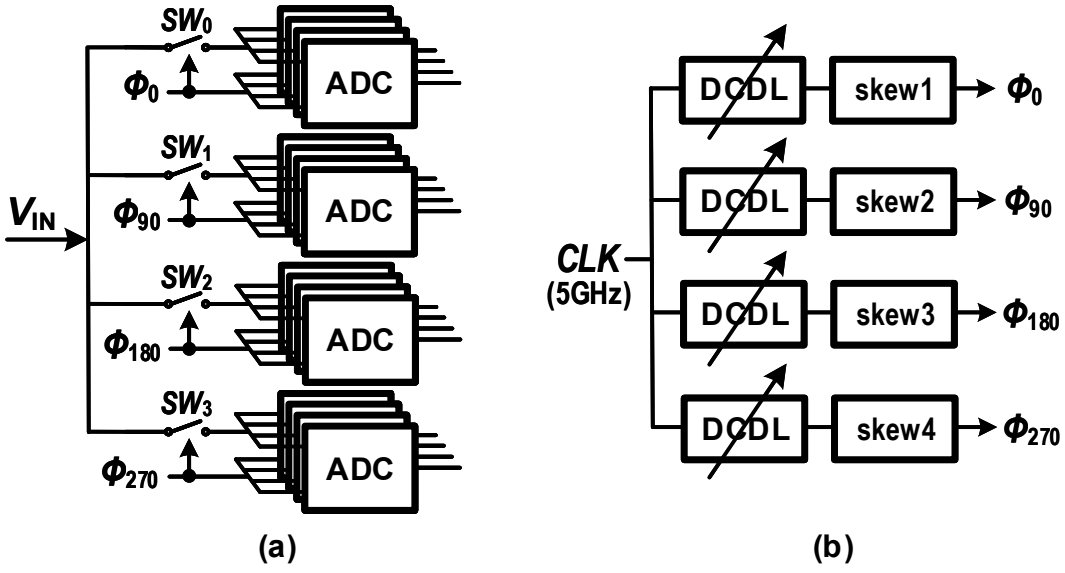


Figure 3.21: Time-interleaved ADC.

the resulting gain and offset errors can be corrected through digital back-end processing. For our time-interleaved ADC, the effect of mismatch between channels can be considered as a part of the physical channel, given that this work is intended for use in a high-speed link. Because of the high data rate, linear equalizers or FIR filters after the ADC have to be parallelized, and that allows us to customize those filters for each of the ADC slices with minimal overhead (Appendix A). However, the error induced by skew among the quadrature sampling clocks is non-linear and hard to correct by linear equalizers unless the error is small enough. Thus, the inter-channel sampling clock skew of the time-interleaved ADC is actively calibrated by a set of digitally controlled delay lines (DCDLs) as shown in Figure 3.21(b), and described in the next chapter.

3.9 Summary

We proposed a synthesizable architecture for a high-speed ADC using V2Ts followed by a stochastic TDC. The stochastic TDC is immune to PVT variation and static delay mismatch among the unit cells at a cost of hardware redundancy. By folding the timing information generated by the V2Ts before the quantization and unfolding it back after the quantization, the effective input dynamic range of the ADC is doubled. The offset delay of the phase folder (D_{OFF}) is estimated through a histogram-based background tracking loop and subtracted from the quantized output to achieve offset-independent unfolding. Throughout the entire design, all parts of the circuit are synthesized into standard digital logic gates and laid out by automatic PnR, except two custom-designed cells: the switch cell and the unit current cell in the V2T. A 16-channel time-interleaved ADC is implemented by a two-stage passive T/H network. Gain and offset mismatches among the ADC slices can be considered as a slice-dependent physical channel and will be calibrated by a per-channel FFE adaptation algorithm (Appendix A). In the next chapter, we will discuss the phase interpolator (PI) used to cancel the inter-channel sampling clock skew and track the optimal sampling phase for the ADC.

Chapter 4

Synthesizable Phase Interpolator

The PI generally refers to a circuit that shifts phase (edge) of an input signal by an amount set by its control code. The definition and expected characteristic of a PI depend on its application. While it can be regarded as a digital-to-time converter (DTC) in an application like fine delay generation [65] [66], it can be used as a digital-to-phase converter (DPC) or a phase rotator in applications such as a phase tracking loop and a fractional frequency division. The latter definition is more appropriate for the context of the high-speed links and hence, will be used throughout this dissertation.

There exists an intrinsic time delay between TX and RX since they are physically apart from each other and, in general, the RX does not know the delay *a priori* due to variation of environmental conditions. Moreover, in many standards, the reference clock frequency of the RX can be slightly different from that of the TX. Thus, the RX usually needs to have a CDR loop to achieve symbol synchronization by generating an optimum sampling clock phase that tracks the incoming data. The dual loop CDR with an internal PI (Figure 4.1(b)) dominates modern high-speed links due to its advantage over the PLL-based single loop CDRs (Figure 4.1(a)). The dual loop CDR decouples noise bandwidth for clock generation (core loop, wideband) and phase tracking (peripheral loop, narrowband) so as to minimize total jitter in the system. The PI in the peripheral loop converts the digital information from the loop

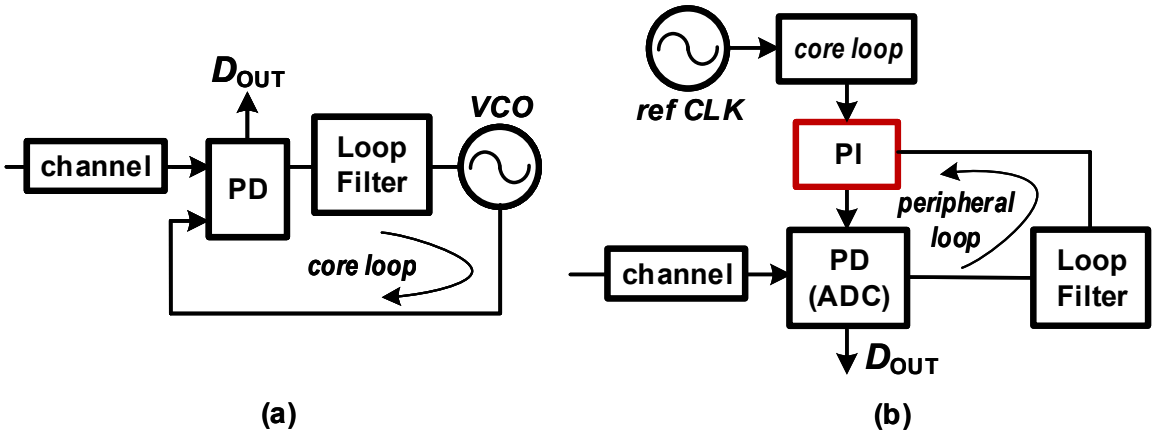


Figure 4.1: Conventional implementation of CDR loop. (a) A PLL based single loop CDR. (b) PI based dual loop CDR.

filter into the analog phase of the recovered clock by rotating the output phase of the core loop. The recovered clock samples the data (using an ADC) in order to generate timing error information via the PD. The error information is scaled and accumulated by the loop filter to close the CDR loop. The PI also serves as the fine DCDLs in our design, dedicated to canceling skew among quadrature sampling clocks of the time-interleaved ADC shown in Figure 2.12(a). The PI must be monotonic and the phase step size of the PI determines the loop gain of the CDR and the accuracy of the clock skew calibration.

In this chapter, we first review the conventional approaches to designing the PI with their drawbacks in the synthesizability and portability point of view, and then we propose a fully synthesizable PI architecture that can achieve sub-picosecond resolution.

4.1 Prior Art

The phase blender is a key building block of the phase interpolator. It takes two input edges and generates an output edge between the two inputs. Its position is

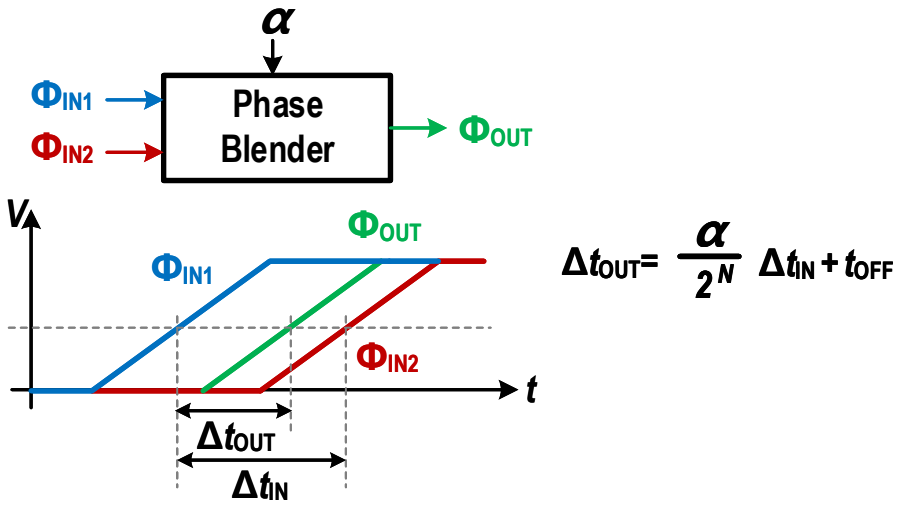


Figure 4.2: General definition of the phase blender.

controlled by a digital input (α) as shown in Figure 4.2. The output usually contains additional constant offset delay, t_{OFF}.

The phase blender is typically implemented by current mode logic (CML) as shown in Figure 4.3(a). Each input triggers a current source, and outputs of the sources are tied together. The strength of each source is set by the digital input, and generally, the sum of currents from both sources is held constant. The adjustable current is often created by changing the number of unit current cells connected in parallel [67][68][69][70]. Figure 4.3(b) shows an example of a CMOS implementation of the phase blender. The output of two branches of parallel tri-state inverters, each of which is triggered by the inputs, are tied together [71]. The current ratio between the branches is controlled by setting the number of active inverters in each branch. Figure 4.3(c) shows another variation of the phase blender [72][73][74]. It sequentially resolves the final output phase with cascaded unit blender cells with low resolution, just like a SAR ADC. Although this cascasded architecture may make some sense theoretically, it is impractical in real-world applications since delay errors, mainly due to random mismatch of devices and interconnects, are accumulated through the stages. The allowable resolution that guarantees monotonicity is very limited, especially when the design is synthesized and laid out with an automated PnR tool.

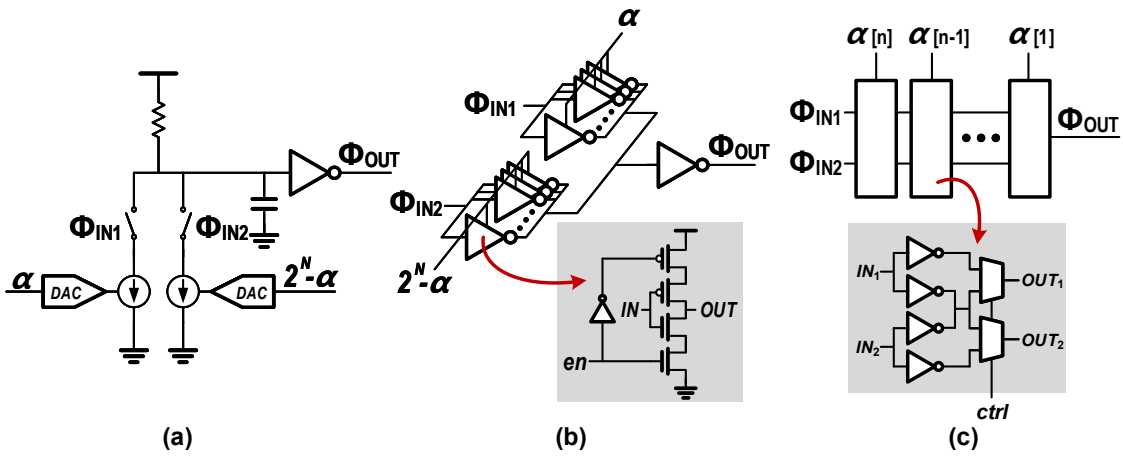


Figure 4.3: Implementation examples of the phase blender. (a) CML. (b) CMOS (c) cascaded.

Usually, the PI (specifically, a phase rotator) is implemented in a two-step fashion. Multi-phase clocks are generated from an input clock first, and then a phase selector chooses two adjacent phases for the following phase blender to achieve finer granularity, as shown in Figure 4.4. Although it is easy to generate the multi-phase clocks in the case when a ring oscillator-based clock generator is used, tight jitter requirements of modern high-speed serial link standards prohibit using a ring oscillator (issues with clock generation will be discussed further in chapter 5). LC oscillator-based clock generators are usually followed by a CML frequency divider to generate the multi-phase clocks [75][76][77], but this approach requires a faster oscillator than the PI requires. Explicit multi-phase generators like a delay-locked loop (DLL) [78] or an injection-locked oscillator (ILO) [79][80][81] are also commonly used, but they tend to significantly increase system complexity.

One potential issue of the phase blender is its failure to ensure sufficient linearity over PVT variation. Sidiropoulos showed that the ratio of the RC time constant at the blender output (output slope) and the phase spacing between inputs (Δt) has a strong influence on its linearity [82]. Weinlader extended that research to show that

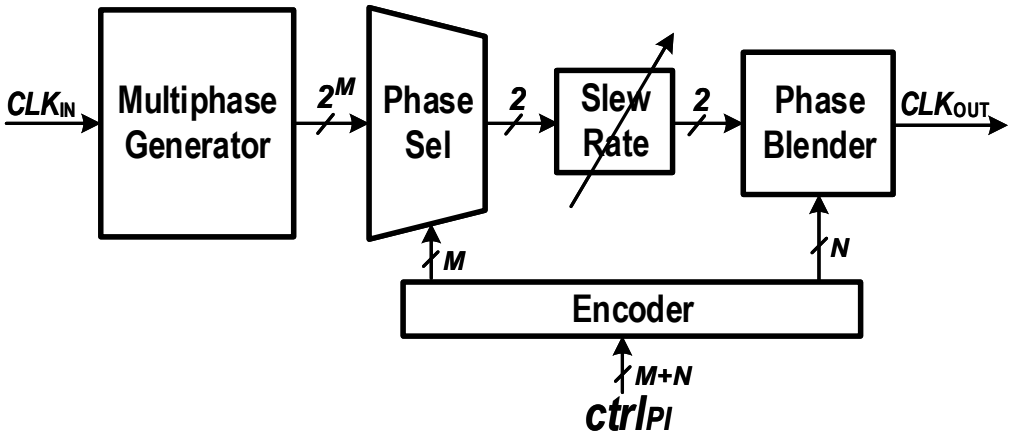


Figure 4.4: Conventional PI implementation.

there is also a dependency on the slew rate of Φ_{IN1} and Φ_{IN2} (input slope) [83]. To minimize nonlinearities, signal slope at both inputs and output and the input phase spacing should meet the following condition:

$$slope\{\Phi_{IN1,2},\ \Phi_{OUT}\} \geq \Delta t \tag{4.1}$$

In the conventional PIs shown in Figure 4.4, spacing between two input phases of the phase blender is changed with respect to data rate of the link (period of CLK_{IN}). Thus, to satisfy Eq. 4.1, slew rate control techniques using passive filters [84][85] [86] or segmented buffers that adjust driving strength [87] along with dedicated calibration loops are being widely used. However, the slew rate control not only makes the system susceptible to power supply-induced jitter but also obstructs the fully synthesizable implementation of a high resolution PI. The next section proposes a fully synthesizable PI architecture that can achieve sub-picosecond resolution with neither close loop multi-phase generation nor slew rate control.

4.2 Proposed Architecture

The proposed fully synthesizable PI takes an input clock, CLK_{IN}, and generates an output clock, CLK_{OUT}, whose phase is shifted with respected to its phase control code, $ctrl_{PI}$, as shown in Figure 4.5. A delay chain designed with unit delay cells (denoted as **D**) using two inverters generates 32 delayed phases of the input clock (Φ_n). A mux network selects one out of 16 odd phases (${\Phi_{2n-1}}^M$) for the first input of a 4-bit phase blender (ϕ_{odd}) and one out of 16 even phases (${\Phi_{2n}}^M$) for the second input of the phase blender (ϕ_{even}). Although the open loop logic delay chain is the simplest option to generate the multi-phase signals of the input clock, it is rarely used for the phase rotator application since phase relationships between the generated clocks and an arbitrary input clock are not well defined and change due to the PVT variation of the delay cells. We added some extra circuitry to tackle this issue. First, because the

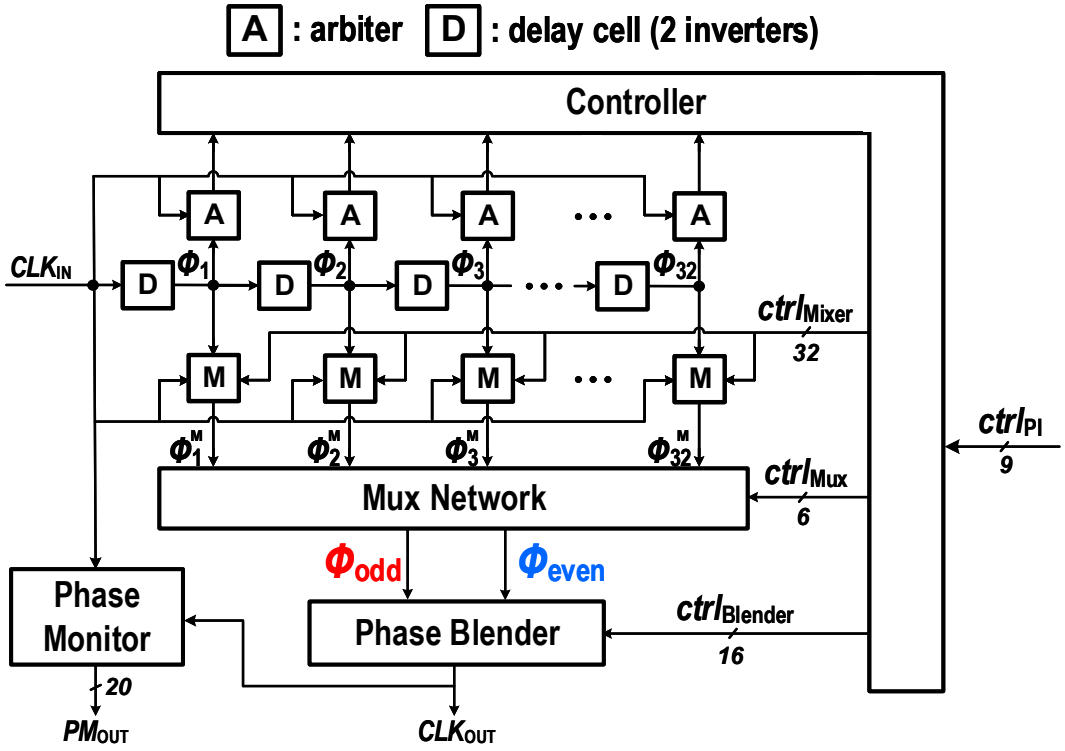


Figure 4.5: Proposed fully-synthesizable PI slice based on a digital phase blender.

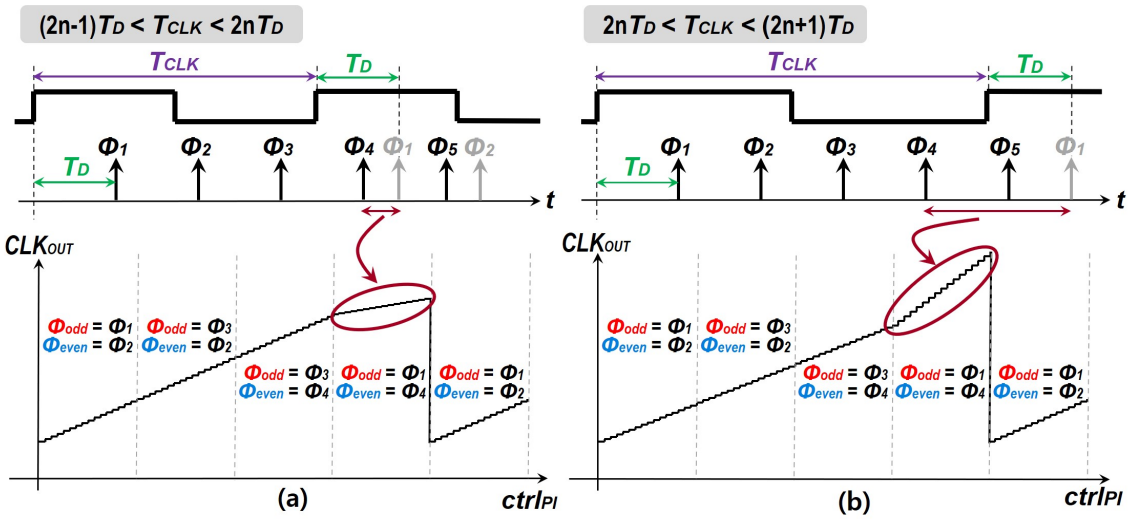


Figure 4.6: Transfer characteristic of the PI. (a) When the quantized period is odd. (b) When the quantized period is even.

PI should be able to operate over a range of input clock frequencies, arbiters (denoted as **A**) are attached to every node of the delay chain to find (quantize) the number of delay units in a clock cycle (T_{CLK}). According to the quantized period, Q_{peri}, the mux network switches the selected phase back to the very first phase, Φ_1, whenever the accumulated phase crosses the period boundary (phase rotation boundary).

There is one more issue we need to address. We need to interpolate between an odd and an even phase. Since Φ_1 is odd, when the selected phase crosses the rotation boundary, we need to interpolate between the even phase nearest to the input clock phase and Φ_1. For example, if the period of the input clock is larger then $(2n-1)T_D$ and smaller than $2nT_D$, where T_D is delay of a delay unit, the phase step size near the rotation boundary is smaller than the nominal step size of the PI, as shown in Figure 4.6(a). On the other hand, if the period of the input clock is larger than $2nT_D$ and smaller than $(2n+1)T_D$, phase step size near the rotation boundary is larger than the nominal step size of the PI as shown in Figure 4.6(b). In this case, input spacing of the phase blender can be as large as $2T_D$. To minimize the worse case phase step size, phase mixers (denoted as **M**) are connected to every node of the delay chain.

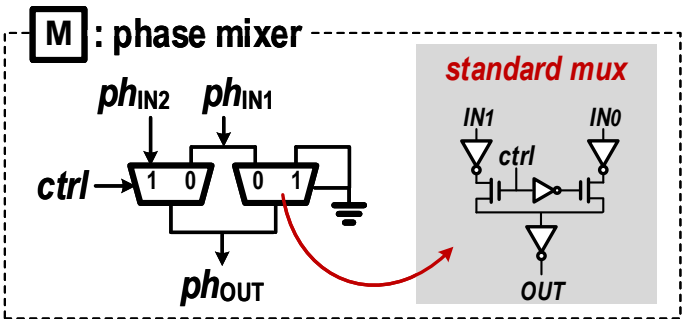


Figure 4.7: Phase mixer (1b phase blender).

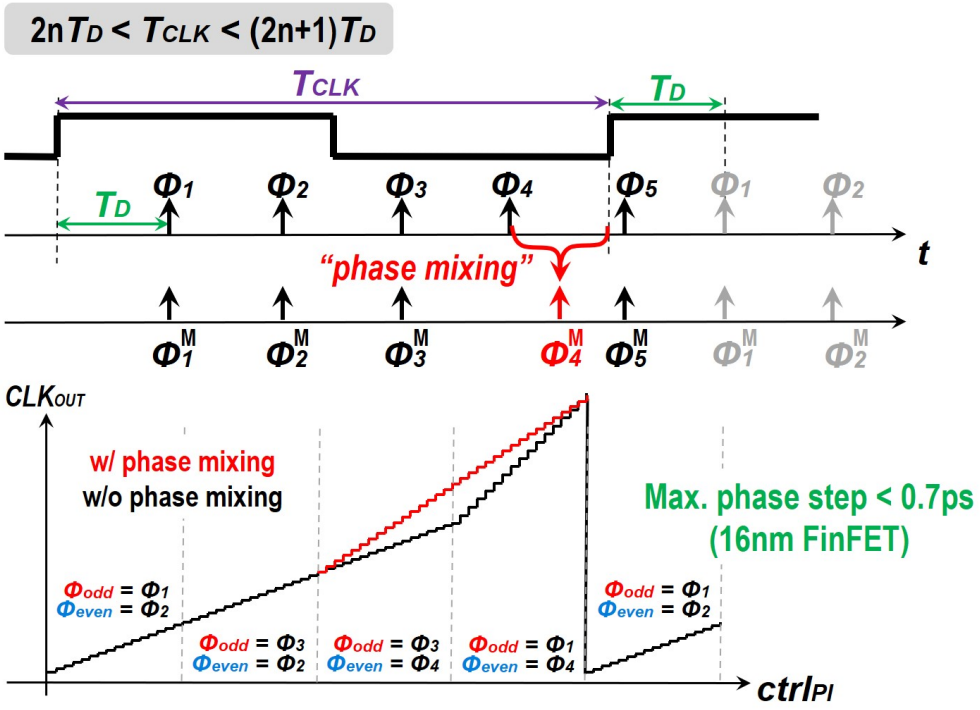


Figure 4.8: Concept of the phase mixing and expected transfer characteristic of the PI.

The phase mixer can be simply implemented using two standard muxes, as shown in Figure 4.7, so that it generates an average phase between its two input phases (1-bit phase blender). The mixer cells are controlled by a digital controller based on the

quantized period. The one mixer at the phase rotation boundary blends its input edge with the input clock edge to generate an average edge between them, while the rest of the mixers act as buffers to achieve monotonic phase rotation across this edge as shown in Figure 4.8. The proposed phase mixing reduces the worst case phase step of the mux network to $1.5\,T_D$, which leads to $\frac{1.5\,T_D}{16}$ for the final phase step of the proposed PI.

4.3 Mux Network and Phase Blender

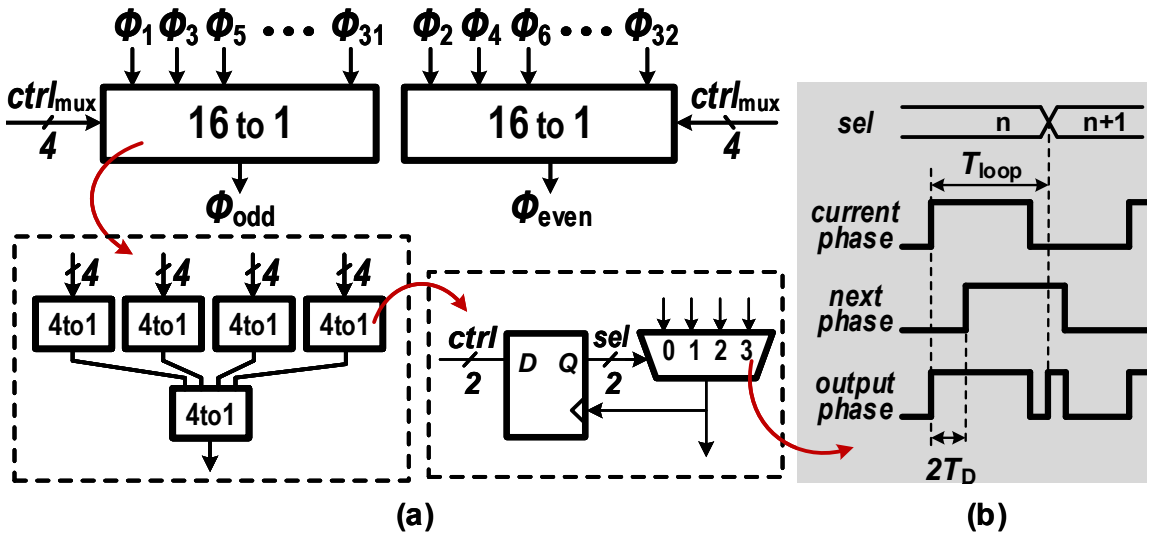


Figure 4.9: Proposed mux network

The mux network consists of two 16-to-1 muxes that cover odd and even phases, respectively, and the 16-to-1 mux consists of two stages of 4-to-1 muxes each of which is implemented by a 4-to-1 standard mux cell and two flip-flops, as shown in Figure 4.9(a). Selection bits of the 4-to-1 mux encoded by the controller are retimed by its output phase to prevent a large phase glitch when the output phase is switched as shown in Figure 4.9(b). We assume the mux control word, $ctrl_{Mux}$, only changes by $\pm$ 1.[1] The glitch-free operation of the 4-to-1 mux is guaranteed as long as the delay

[1]This is true because of the limited bandwidth the CDR loop in most of high-speed serial links.

of the retiming loop, T_{loop}, is kept less than half of the input clock period, i.e.,

$$2\,T_D < T_{loop} = T_{mux} + T_{FF}^{(ck-q)} < \frac{T_{clk}}{2} = 100ps$$

The lower bound of the loop delay does not have much meaning since $D_{mux} \geq T_D$ and $D_{FF}^{(ck-q)} > T_D$ generally holds regardless of technology nodes.

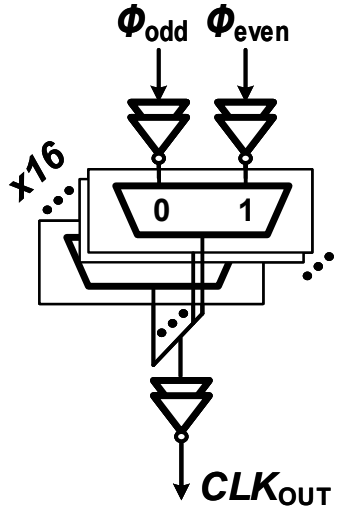


Figure 4.10: Proposed phase blender.

The phase blender is implemented by 16 shorted muxes as shown in Figure 4.10. The output current of each mux, which is merged at a shared output node, is controlled by a thermometer code encoded by the controller. The input and output of the phase blender are buffered by fan-out-of 4 (FO4) inverter stages to drive muxes and to isolate the shared output node from the rest of the circuits. Since the maximum time difference between the two input phases of the phase blender is limited to $1.5T_D$ by the mux network, and slope of signals in a FO4 delay line is usually larger than $1.5T_D$, it naturally satisfies the criteria in Eq. 4.1. Therefore, the phase blender in the proposed PI does not require explicit slew rate control, which makes the design simpler and fully synthesizable.

$\pm$ 1 of $ctrl_{Mux}$ is equivalent to $\pm$ 16 of $ctrl_{Blender}$.

4.4 Mismatch Calibration and Controller

Unfortunately, the monotonicity of the PI may be jeopardized by automated PnR if the static mismatch among accumulated path delays, including wire interconnects, exceeds the unit delay of the delay chain, T_D. To prevent this problem, we measure this delay mismatch by adding an extra arbiter to the input of the phase blender and an adjustable delay element to all inputs of the mux network as shown in Figure 4.11. For instance, if the difference between accumulated delay of the red path and

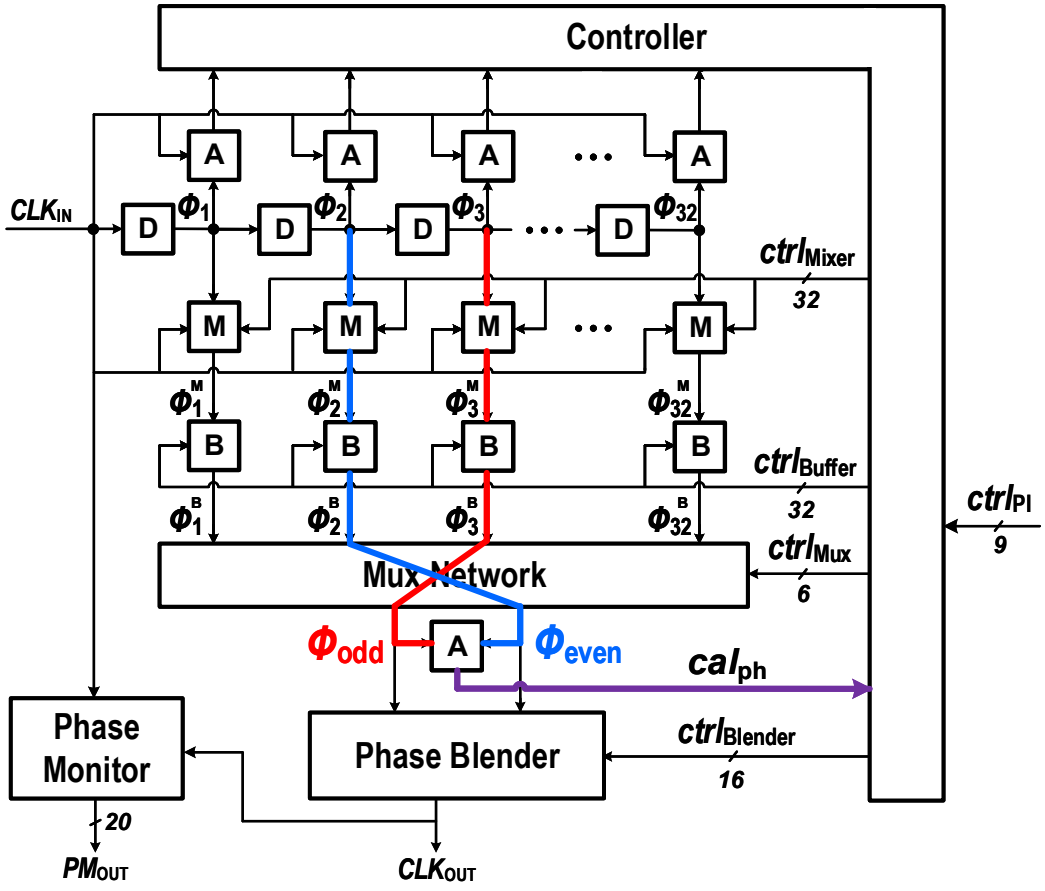


Figure 4.11: Foreground delay mismatch calibration with an error detecting arbiter and delay adjustable buffers.

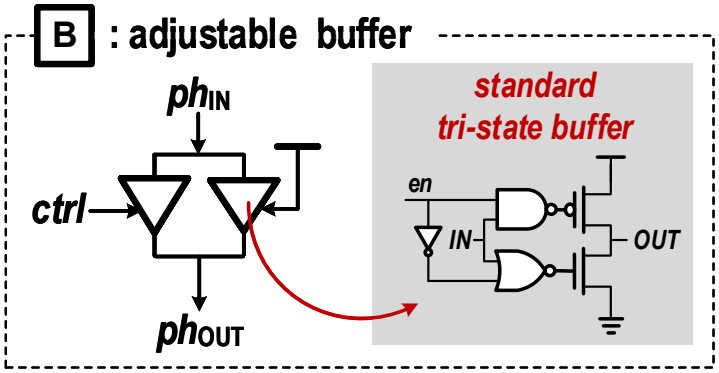


Figure 4.12: Proposed adjustable delay buffer.

the blue path is greater than T_D, ϕ_{even} can lag ϕ_{odd}. In this case, the delay of the red path is tweaked by an adjustable delay buffer cell (denoted as **B**). This mismatch calibration is done for all 32 delay paths, once after fabrication. The larger the adjustable range/resolution the buffer supports, the more robust against mismatches the PI is, at the cost of extra power consumption. We chose a 1-bit adjustable buffer cell implemented by two parallel standard tri-state gates, as shown in Figure 4.12, for our PI design.

The PI controller takes a 9-bit phase control code ($ctrl_{PI}$) and a 32-bit vector generated by arbiters attached to every node of the delay chain (arb_{OUT}) and controls all the rest of the blocks in the PI, as shown in Figure 4.13. The input clock period decoded from the arb_{OUT} determines the maximum allowable control code of the mux network, *max_ctrl_mux*, and which mixer cell should be enabled. It also normalizes the phase control code so that the gain of the PI stays the same regardless of its input clock frequency. 5 MSBs and 4 LSBs of the normalized phase control code are used to control the mux network and the phase blender, respectively. The control code of the mux network is sub-divided by muxing stages and by odd/even part of the network to achieve the glitch-free phase switching.

Figure 4.14 illustrates a simplified strategy for the glitch-free mux control where the orange, dashed line describes the rotation boundary, and the red line indicates

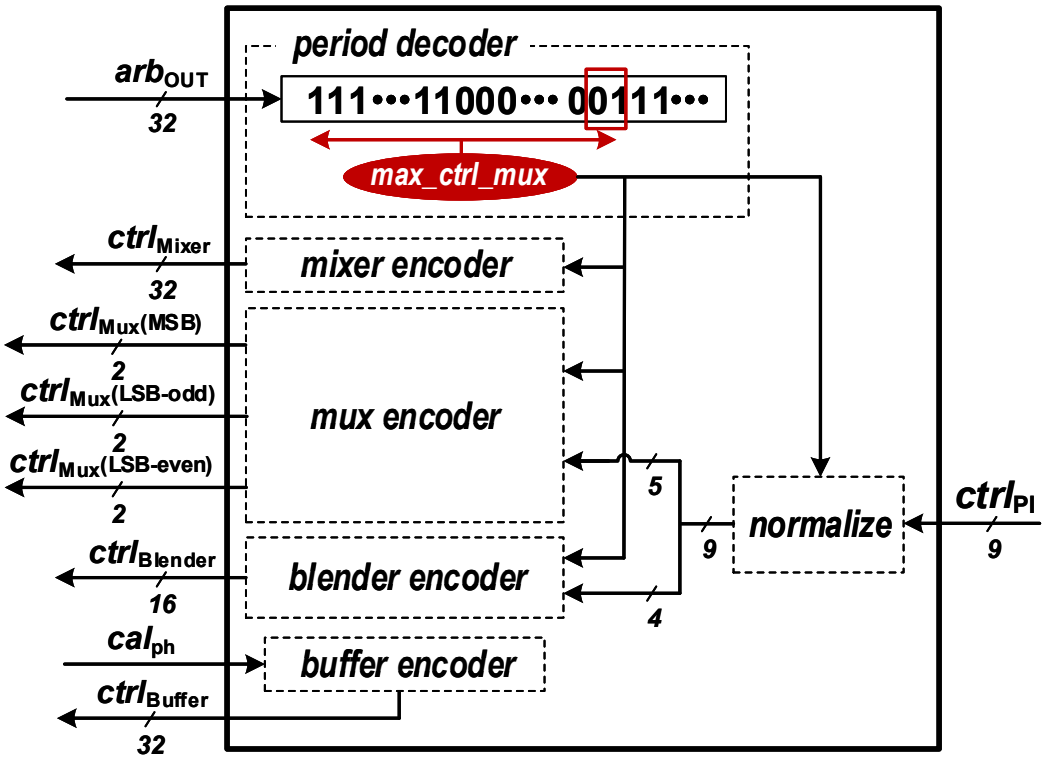


Figure 4.13: PI controller.

the connection of the muxes for the currently selected phase.

When the currently selected phase is coming from a mux (M1) and is also at a boundary of two muxes (M1 and M2), the control code of the neighboring mux (M2) is forced such that its output phase is the closest to the boundary (the green line) in order to minimize the magnitude of possible phase transition (case (a)). The boundary is affected by the *max_ctrl_mux* as well when it comes to considering the phase rotation that happens whenever the output phase crosses the period boundary (case (b)). If the currently selected phase is at a boundary of a mux (M3) and, at the same time, at the period boundary, control codes of both the first mux (M1) and the neighboring mux (M2) are forced such that their output phases are the closest to the boundaries (case (c)). This setting of the unselected mux inputs ensures that

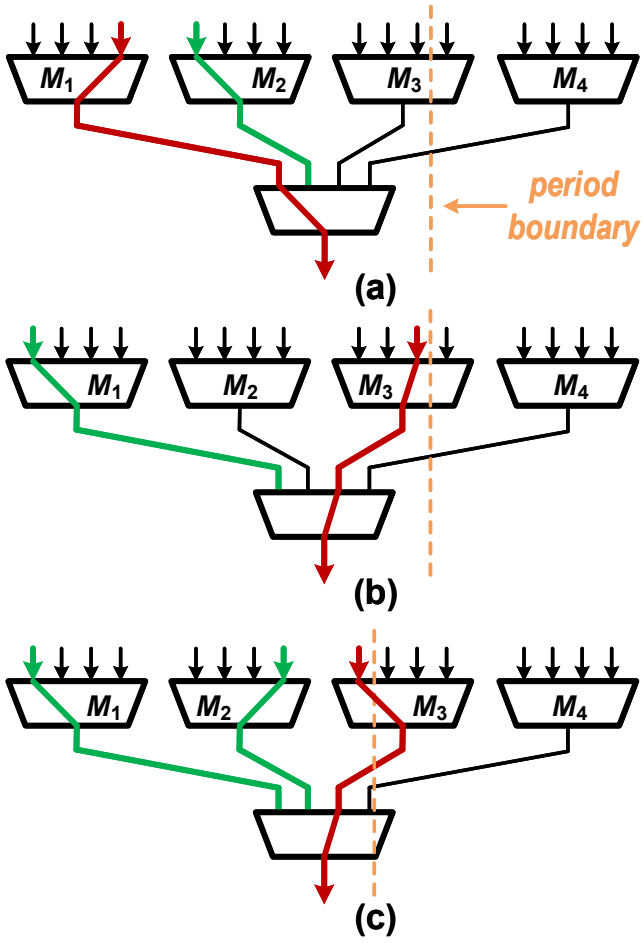


Figure 4.14: Mux control strategy to guarantee glitch-free operation of PI (only odd part is shown).

crossing mux boundaries only requires the change of one mux control step.

The phase blending direction of the phase blender should be changed depending on the selected phases by the mux network. For example, it blends the phases in the forward direction when the mux network selects Φ_{2n-1} and Φ_{2n} as ph_{odd} and ph_{even}, respectively, while in the backward direction when the selected phases are Φ_{2n+1} and Φ_{2n}, respectively.

4.5 On-Chip Phase Monitor

We integrated an on-chip phase difference monitor with the PI to measure an accurate static characteristic of the PI. It quantizes the delay between input (CLK_{IN}) and output (CLK_{OUT}) with respect to control codes, based on the edge detection with asynchronous sampling [88]. An external test clock signal with a known frequency, CLK_{ASYNC} is sampled by the input and output edges, respectively, and the difference between the sampled values is monitored as shown in Figure 4.15. If the test clock is asynchronous with CLK_{IN}, then its rising edge will occur uniformly distributed over the clock cycle and we know the average number of edges in the cycle. Thus if the test edges are monitored over a long enough period of time, the static time difference between input and output of the PI can be accurately measured. A similar technique is used in the proposed clock generator as well, and the detailed operating principle with design requirements is discussed in Section 5.2.

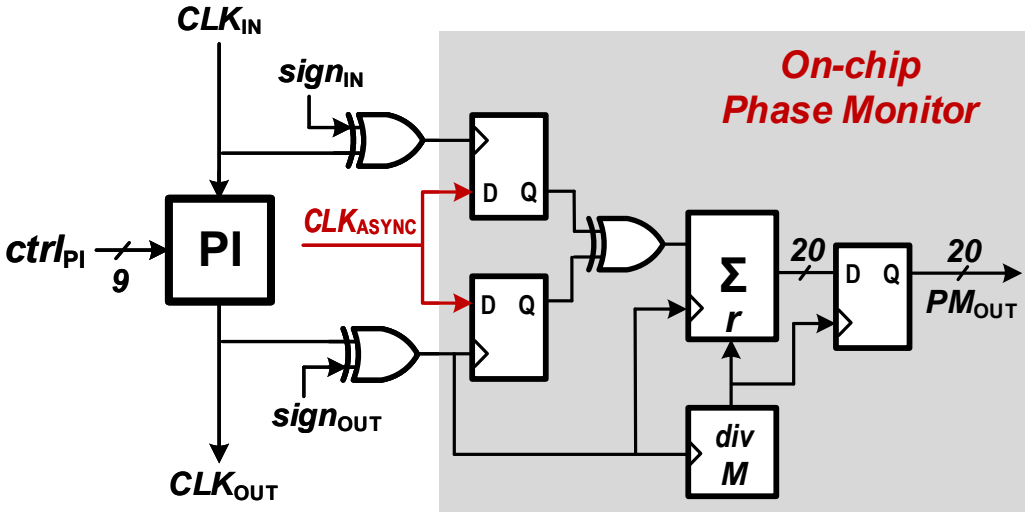


Figure 4.15: Proposed on-chip phase monitor.

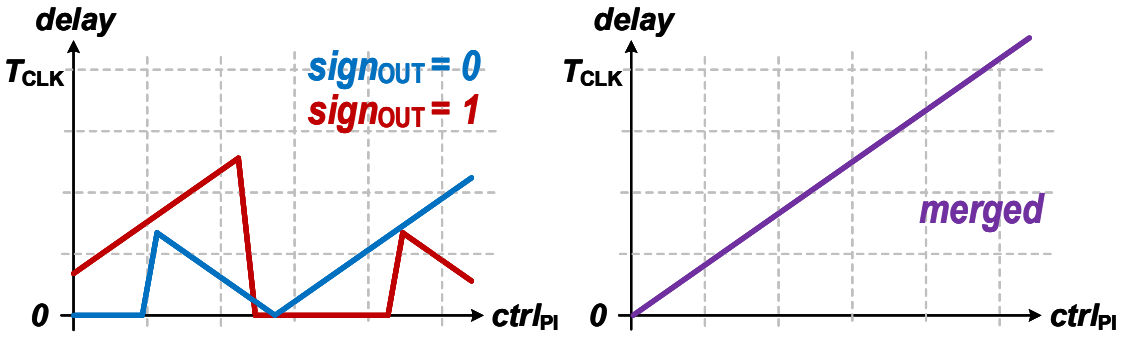


Figure 4.16: Proposed input phase flipping to avoid measurement errors due to the edge pulling.

One challenge in the fine timing measurement is that the measured results may be inaccurate when the two input edges are close to each other in both time and spatial domain. One transition affects the timing of the other transition mainly due to physical coupling paths between them such as supply, substrate, and parasitic capacitance of devices. This effect is often called "edge-pulling" [89]. To mitigate the effect, our design conditionally inverts the input edges with an XOR gate. The measurement is performed twice, once for the in-phase output clock and once for the out-phase output clock, and the final result is obtained by selectively merging the two results as shown in Figure 4.16.

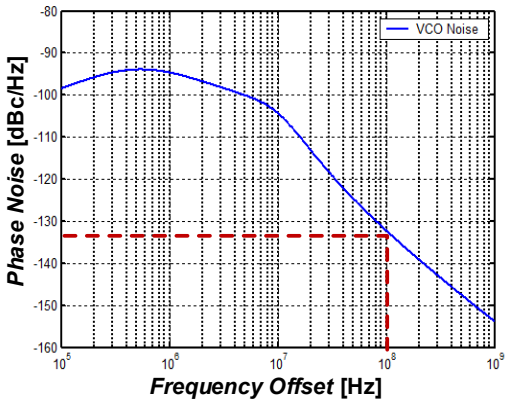
4.6 Summary

We proposed a fully synthesizable and portable architecture of the PI. The proposed circuit does not have either feedback in the signal path or slew rate control. An open loop delay chain generates delayed versions of the input clock and a bank of arbiters quantizes the ratio between the clock period and the delay of a unit delay cell. A glitch-free mux network selects two out of N delayed input clock phases and the selected phases are applied to a CMOS phase blender for finer phase resolution. The proposed phase mixing technique near the phase rotation boundary suppresses worst

case phase step to 1.5 times the unit delay. To guarantee monotonicity, which may be jeopardized by random static delay mismatch, adjustable delay buffers are added in the signal paths and calibrated in the foreground. The entire design is described by SystemVerilog and synthesized/laid out through a standard digital design flow.

Chapter 5

Synthesizable Clock Generator



F_{Target}	10GHz
F_{REF}	100MHz
Bandwidth	10MHz
Target Int. Jitter	0.4ps$_{rms}$
Target PN	-133dBc/Hz @ 100MHz

	Ring	LC
FoM	~ -160dBc/Hz	~ -190dBc/Hz
Required Power	~50mW	~50uW

Figure 5.1: Comparison between Ring and LC oscillator for the given application.

Clock generators of the high-speed serial links are usually implemented by a phase-locked loop (PLL). There are two common choices for the voltage-controlled oscillator (VCO); An LC oscillator and a ring oscillator. Based on the fact that the figure of merit (FOM)[1] of the LC oscillator is greater than that of the ring oscillator by roughly 30dB [90][91][92][93], the LC oscillator is much more power efficient (roughly 1000x)

[1] $FOM_{VCO} = \mathcal{L}(f_{off}) - 20log\left(\frac{f_{out}}{f_{off}}\right) + 10log\left(\frac{Power}{1mW}\right)$, where $\mathcal{L}$ is phase noise, f_{out} is output frequency, and f_{off} is offset frequency.

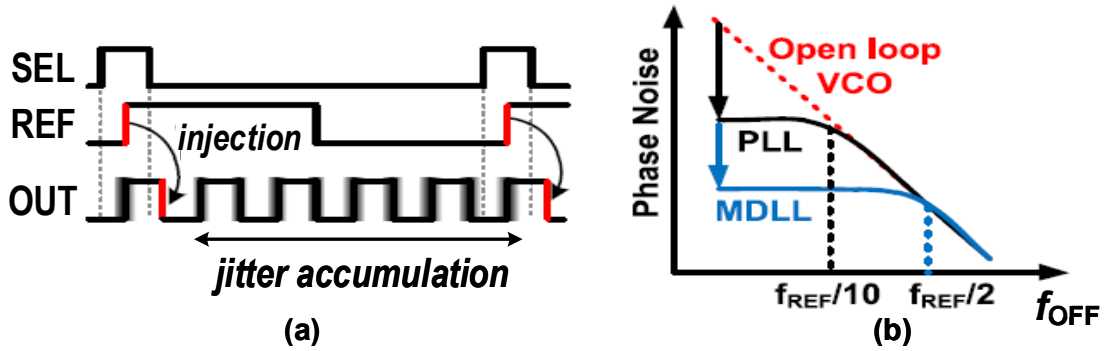


Figure 5.2: (a) Jitter filtering by the clean edge injection of (b) Phase noise comparison between the conventional PLL and the MDLL.

than the ring oscillator for a given target frequency and noise performance, as shown in Figure 5.1.

The tightened jitter requirement, in a given power budget, by the ever-increasing data rate of modern serial link standards left designers one choice: to use an LC tank as the oscillator core of the clock generator [10][11]. However, the LC oscillator is not only hard to make synthesizable, but also hard to make portable since its performance highly depends on process parameters. Moreover, it usually requires multiple tanks to cover a wide range of data rates. A multiplying delay-locked loop (MDLL) is an attractive way to generate a high-frequency clock with low random jitter (RJ) using the ring oscillator, when a reference clock is assumed to be clean enough. One out of N "dirty" output clock edges is periodically replaced by the "clean" reference edge so that the maximum number of cycles that the jitter is accumulated in the ring oscillator is limited to N, as shown in Figure 5.2 (a) (also known as a subharmonically injection-locked oscillator). As a result, the MDLL can achieve better RJ performance by filtering out a wider band of the oscillator's jitter compared to the conventional PLL [94], whose noise bandwidth is usually limited up to 1/10 of the reference frequency [95], as shown in Figure 5.2 (b).

A known issue of the MDLL is that it may suffer from high deterministic jitter (DJ) if the reference edge is not exactly aligned with the output edges. Therefore, most

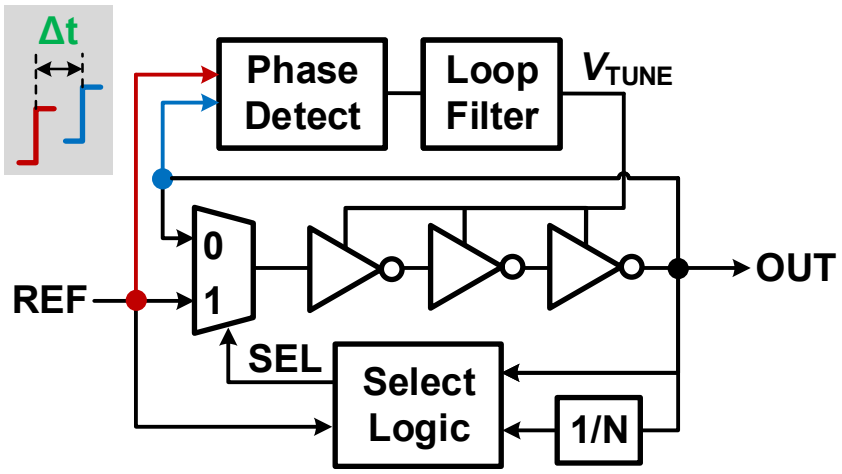


Figure 5.3: Conventional injection locked clock generators (MDLL).

practical designs include a feedback tuning loop to force the free-running frequency of the ring oscillator to be exactly identical to N times the reference frequency, as shown in Figure 5.3. Unfortunately, even though the free-running frequency of the ring oscillator is locked to the reference, with the static timing offset between OUT and REF (Δt), which is common in the automated PnR flow, the clock cycle with the REF injection will be Δt longer than the nominal clock cycles without the injection,

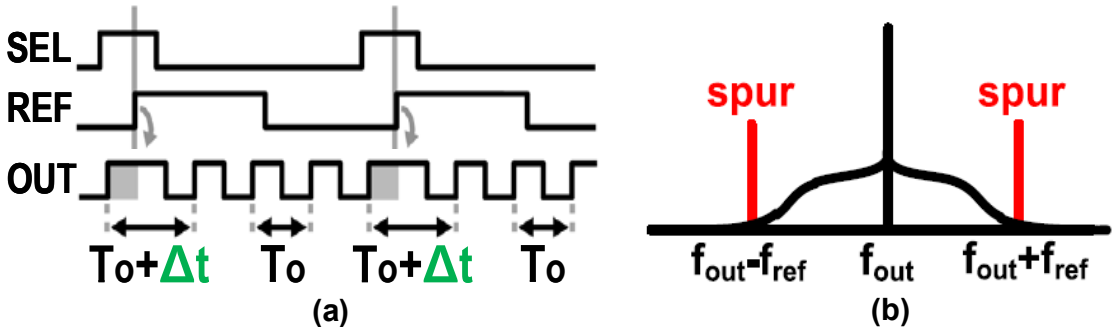


Figure 5.4: Impact of the injection error caused by timing mismatch. (a) Time domain (b) Frequency domain.

as shown in Figure 5.4 (a). This periodic disturbance of the clock cycle causes the DJ or, equivalently, the spurious spectrum at the reference frequency as shown in Figure 5.4 (b).

In this chapter, we first review conventional approaches to suppress the static mismatch-induced DJ of MDLL along with their drawbacks in the context of our synthesizable/portable design goal. Next, we describe the proposed fully synthesizable MDLL.

5.1 Prior Art

A conventional approach to suppress DJ of an MDLL is to actively cancel the timing mismatch between *REF* and *OUT* using a digital-to-time converter (DTC) [96]. To find out the control code of the DTC for optimal performance, Kundu

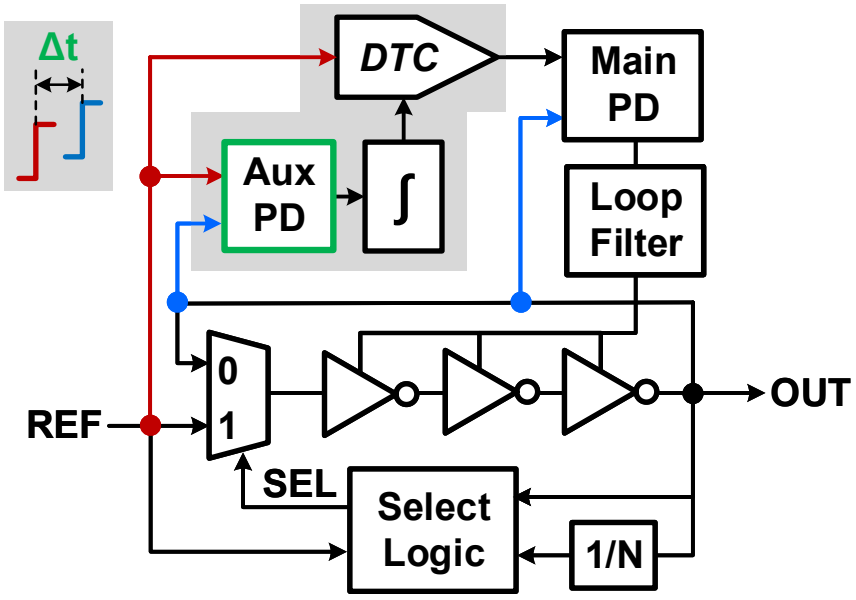


Figure 5.5: Static timing error correction using a digital-to-time converter (DTC) and a zero-offset auxiliary phase detector (PD).

designed a background tracking loop which consists of an auxiliary phase detector (Aux PD) and an integrator as shown in Figure 5.5 [94]. It achieved 2.8ps of RJ and 4ps of DJ with 8mW of power consumption for a 1.4GHz output clock in a 65nm process. Unfortunately, it mandates not only a tight input offset for the Aux PD but also a high resolution for the DTC, which requires high precision analog circuitry with careful custom layout.

Another approach is to employ a correlated double sampling (CDS) technique to estimate the amount of periodic disturbance of the output cycle time, as shown in Figure 5.6. The timing mismatch between REF and OUT, Δt, causes an erroneous cycle time deviated from the nominal cycle time between the ring oscillator, T_o, whenever REF is injected. Both the erroneous cycle time ($T_o + \Delta t$) and the nominal cycle time (T_o) are directly measured by an identical readout circuit and the difference of the results is accumulated and fed back to adjust the ring oscillator as shown in Figure 5.7. Helal employed a high-resolution gated ring oscillator (GRO) based TDC to quantize the cycle times and implemented a digital feedback loop to control the delay [97]. It achieved 0.68ps of RJ and 0.76ps of DJ with 9.2mW of power consumption for a 1.6GHz output clock in a 130nm process. Kim employed

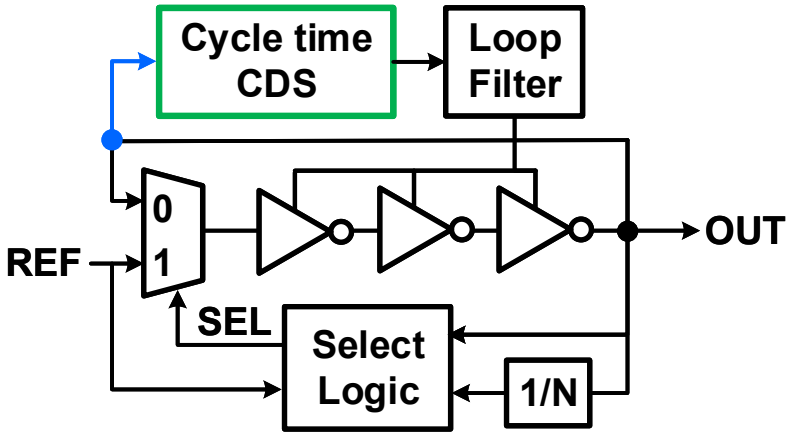


Figure 5.6: Static timing error correction using a cycle time correlated double sampling (CDS).

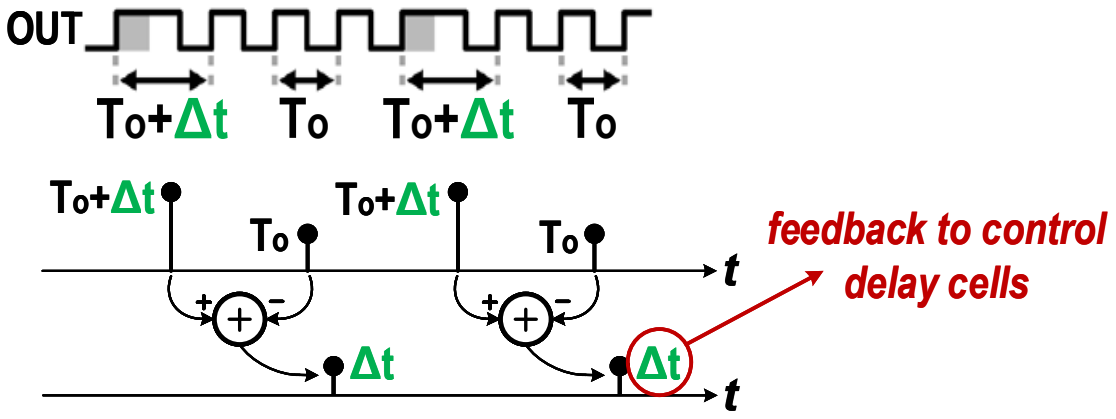


Figure 5.7: Correlated double sampling (CDS) used to readout the injection error (Δt).

a charge pump and a voltage comparator to read out the cycle times and implemented an analog feedback loop to control the delay [98]. It achieved 0.7ps of RJ and 0.53ps of DJ with 1.5mW of power consumption for a 2.4GHz output clock in a 28nm process. Although these works may successfully suppress the DJ they heavily rely on analog precision circuitry and hence, are hard to make synthesizable/portable.

Recently, a fully synthesizable MDLL has been reported by Deng [99]. The free-running frequency of a replica DCO is locked to the reference frequency using a feedback loop. *REF* is injected into the main DCO, which is outside of the loop, and the DCO shares its control code with the replica DCO, as shown in Figure 5.8. It achieved 2.8ps of RJ and 9.9ps of DJ with 0.78mW of power consumption for a 0.9GHz output clock in a 65nm process. The entire design can be implemented by standard logic gates only, and hence, it is fully synthesizable. However, note that an underlying assumption of this work is perfect matching between the two DCOs, which is not practical in the automated PnR design flow in general.

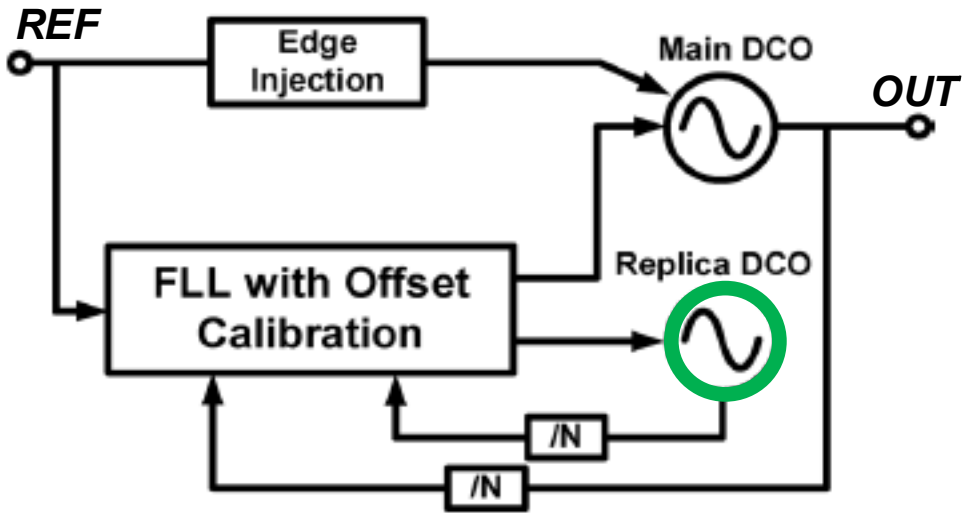


Figure 5.8: Synthesizable implementation of MDLL using a replica DCO.

5.2 Proposed Architecture

Figure 5.9 shows the architecture of the proposed MDLL (differential paths are not shown). It consists of a DCO, a frequency divider, an edge selection logic, a loop filter, a delta-sigma modulator (DSM), a coarse frequency tuning loop, and a fine phase tuning loop. At startup, the initial tuning of the DCO is done by a conventional bang-bang frequency feedback loop to set the coarse delay cells in the DCO. After the bang-bang loop is locked, output edges of the DCO, OUT, are periodically replaced by reference clock edges, REF, through a mux, in every N^{th} cycle. The selection logic along with the frequency divider generates an appropriated pulse, SEL, to control the mux.

The cycle time difference between the injected cycle and the nominal cycle of the DCO, which is mainly due to accumulated random delay mismatch in the signal paths, is measured by a sub-sampling stochastic CDS using an uncorrelated sampling clock and filtered to create a 13-bit delay control word, D_{ctrl}. The upper 7 bits of the

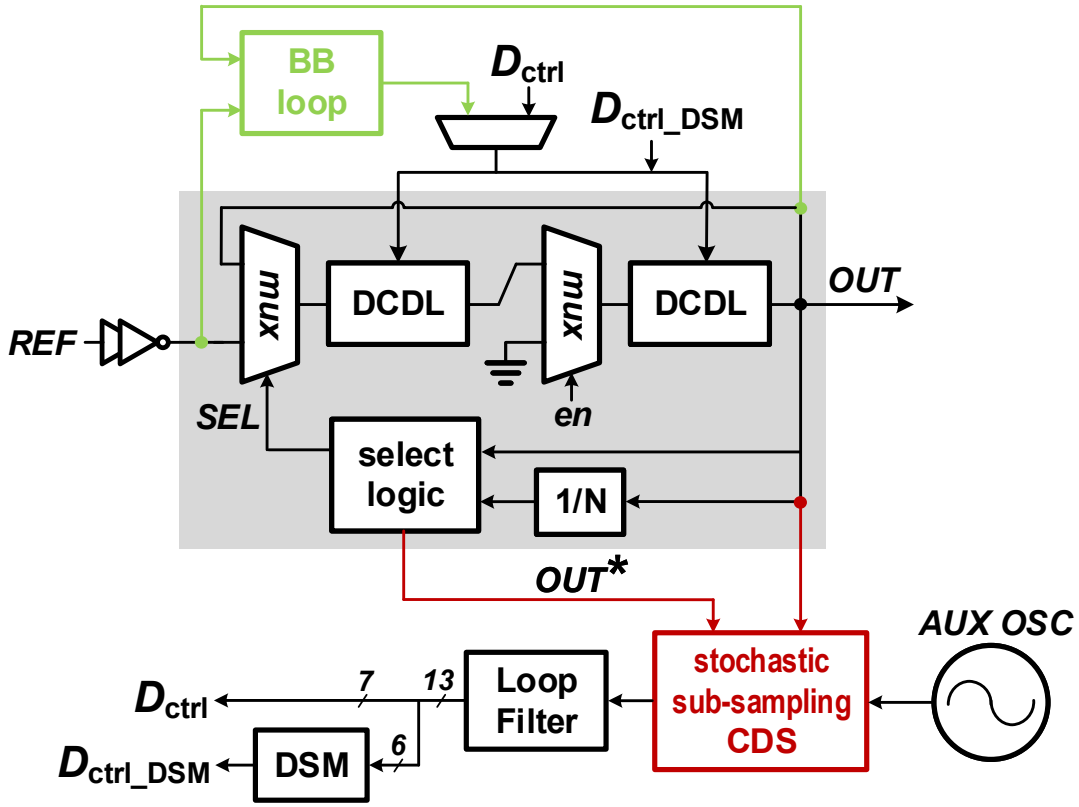


Figure 5.9: Architecture of the proposed MDLL.

control word go to fine tuning cells in the DCO, which have an LSB step size of 0.06ps. This gives a cycle time resolution of 0.24ps. While this is small, with a multiplication factor (N) of 16, this can cause $\sim$ 4ps of DJ. To reduce this DJ accumulation, a DSM with the lower 6 bits of the control word dithers the LSB.

The detailed implementation of the sub-sampling stochastic CDS is shown in Figure 5.10. An auxiliary oscillator (AUX_OSC) is used to create rising edges that are asynchronous with the output of the MDLL, OUT. The rising edges of the AUX_OSC are equally likely to occur anywhere in the OUT waveform. This means that the average number of rising edges counted in any time window is proportional to the width of that window. We use this technique to measure a pulse width in

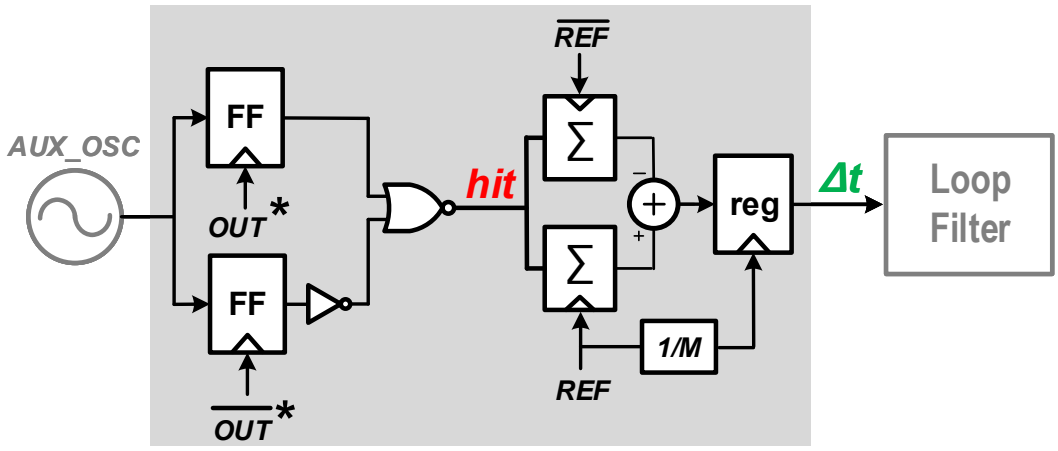


Figure 5.10: Stochastic sub sampling correlated double sampling (CDS).

sub-ps resolution [88].

The selection logic generates OUT^*, which consists of two pulses in a reference cycle. The width of each pulse captures the duration of the injected cycle and the nominal cycle, respectively, as shown in Figure 5.11. The auxiliary clock is sampled at the rising and falling edge of OUT^*, asserting *hit* if a rising edge of the auxiliary clock occurred during the pulse of OUT^*. These *hit*s are accumulated for the two pulses

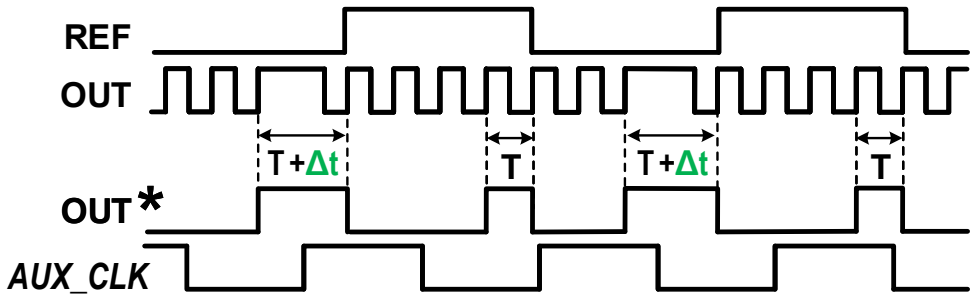


Figure 5.11: Waveforms of the MDLL.

(one for the injected cycle and the other for the nominal cycle), and the difference between them is sent to the loop filter. In this design, the frequency of the AUX_OSC is designed around 1GHz, and the duration of the nominal cycle (T_o) is roughly 200ps, so each pulse will count 0.2 edges per cycle on average. Using 100k cycles (20k counts) gives a resolution of 10fs (200ps/20k counts) for the timing measurement, and standard deviation of the measurement (σ) is about 1.5ps. This error estimate, Δt, is then filtered reducing σ to 200fs, and used to drive the time difference, Δt, to zero. Since the same circuit path is used to generate and measure the pulse for the injected cycle and the nominal clock cycle, delays along the path are common mode, and no matching needs to be considered.

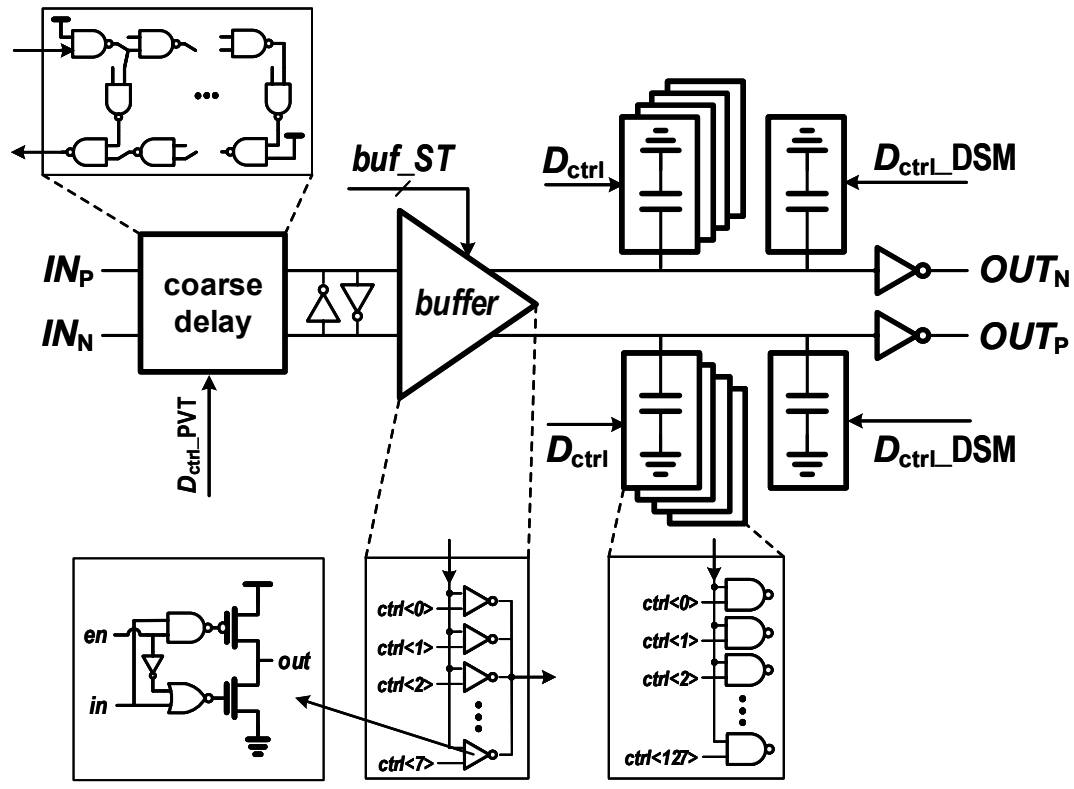


Figure 5.12: Implementation of the DCDL.

The DCO consists of two identical digitally controlled delay lines (DCDLs) and muxes. The clock is inverted in the feedback loop, so the clock cycle time of the MDLL is 4x the DCDL plus mux delay. Detailed implementation of the DCDL is shown in Figure 5.12. The delay of the DCDL is controlled in a two-step process. The length of a chain of series NAND gates sets the coarse delay of the DCDL, allowing it to cover a wide range of frequencies, and to compensate for PVT variation. This chain then drives a bank of parallel NAND gates. The capacitance of the driven input of the NAND gates depends on the logic state of the other input so as to provide fine frequency tuning [99]. A bank of tri-state buffers is inserted between the coarse and fine delay line in order to adjust the step size of the fine tuning by controlling its driving strength. One extra slice of the parallel NAND bank is added to dither LSB for the finer effective resolution of the frequency tuning. Nominal step sizes for the coarse and fine-tuning stage are 8ps and 0.06ps respectively.

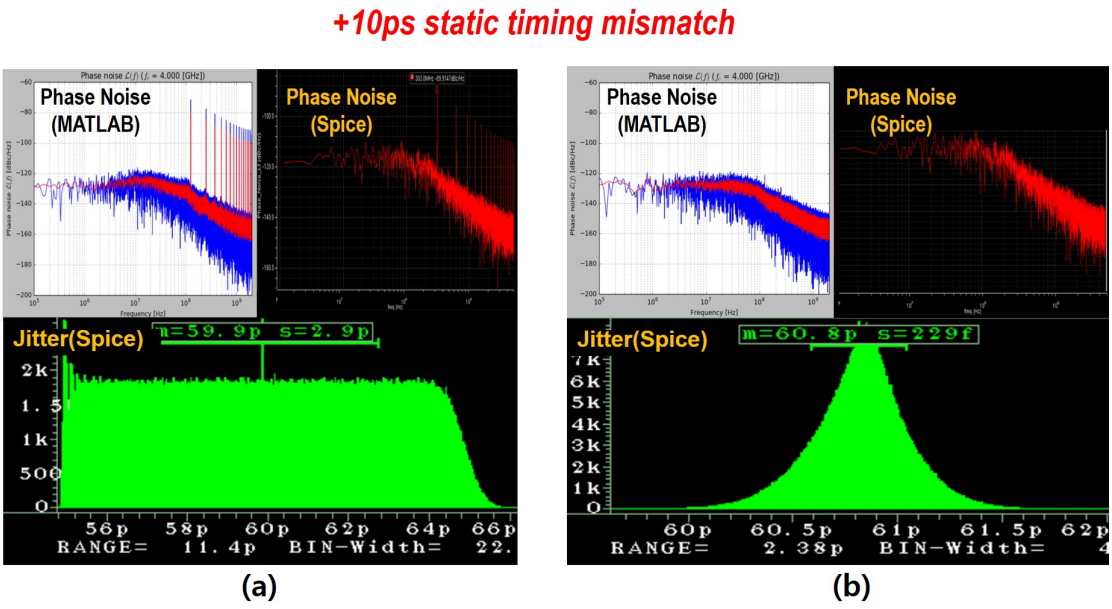


Figure 5.13: Simulation results of the clock generator with an intentional timing mismatch. (a) With conventional bang-bang calibration loop. (b) With the proposed stochastic sub-sampling CDS calibration loop.

Figure 5.13 shows transient and periodic steady state (PSS) simulation results of the proposed MDLL for a frequency multiplication factor of 16 and an ideal reference clock of 312.5MHz. Under an intentional timing mismatch of 10ps, the performance of the proposed MDLL is compared with that of a conventional MDLL based on the bang-bang phase calibration loop. The proposed stochastic sub-sampling CDS loop successfully suppresses the effect of the mismatch and results in low enough DJ with negligible amount of additional hardware and power.

5.3 Summary

This chapter describes a clock source that can be used for the proposed open-source high-speed serial link. Since an LC oscillator is hard to make synthesizable, this design uses a Multiplying DLL (MDLL) to reduce accumulated jitter that would occur in a ring oscillator-based PLL. A major drawback of using an MDLL is high deterministic jitter (DJ) when the free-running frequency of the delay line in the MDLL is not exactly matched with the desired target frequency. A bang-bang phase detector (PD) based offset calibration loop is usually used to deal with this issue, but it requires careful design to minimize the PD offset. To avoid the need for precise design, we use correlated double sampling (CDS) for the phase detection, using a TDC to measure the two clock periods we need to match. The stochastic sub-sampling, which is small, low power, and easily synthesized, is used to create the required high resolution TDC. This approach removes the delay matching requirement, which is critical in highly scaled technologies. While the performance and power of our design are not the state of the art for this technology, it meets the target jitter spec even though the entire design is fully synthesized from SystemVerilog and laid out by an automatic PnR tool.

Chapter 6

Design Flow and Implementation

In the previous chapters, we proposed a synthesizable architecture for the AMS building blocks of an ADC-based high-speed serial link receiver. To make this open-source platform complete, this chapter presents an automated design and physical implementation flow for this design. Section 6.1 shows the open-source design procedure including the verification and synthesis associated with a git-based public repository. Section 6.2 introduces a physical implementation flow of the designs using a commercial digital PnR tool along with its potential challenges and corresponding solutions.

6.1 Open Source Design Flow

Conventional design flow of the mixed-signal systems, including high-speed serial links, is divided into two different paths, as shown in Figure 6.1(a). The digital parts of the system are usually designed and verified by Register Transfer Level (RTL) languages, such as Verilog or VHDL, and synthesized to gate-level netlists associated with a given technology. Physical design of the netlists is generated by automatic Place and Route (PnR) systems. Performance of the designed system can be efficiently examined, or guaranteed, through Static Timing Analysis (STA) with pre-characterized timing information of the gates and interconnects, as long as appropriate constraints are set by designers. This digital flow is so powerful that it has enabled complicated

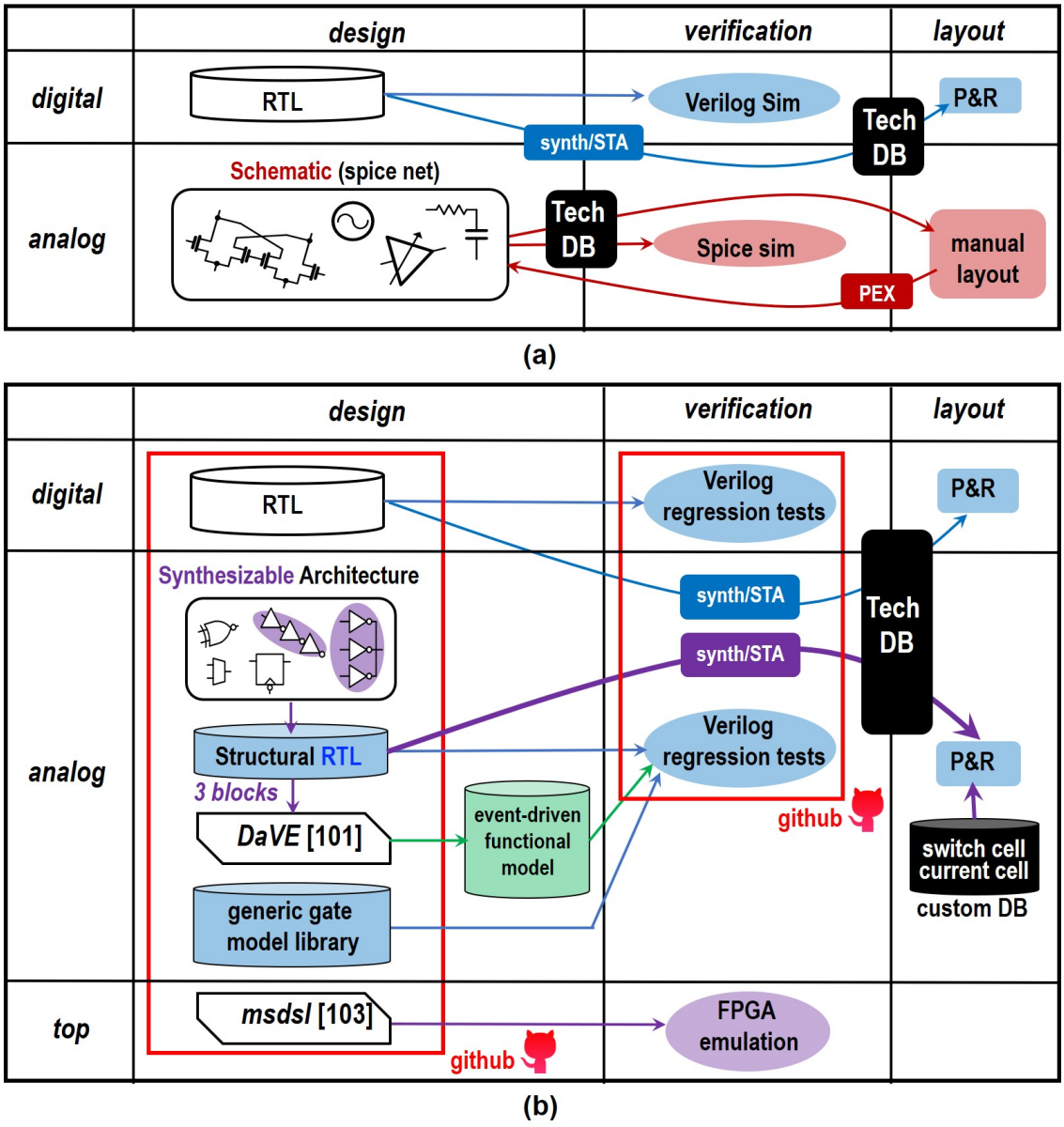


Figure 6.1: (a) The conventional mixed-signal design flow. (b) The proposed automated design flow for the Open-Source PHY.

digital system designs and driven Very Large Scale Integration (VLSI) so far. On the other hand, however, the analog parts of the system are usually designed in transistor level and verified through transient simulation such as HSPICE. Physical design is

laid out manually to meet the required level of precision and accompanied by post-layout iterations. This analog design flow is not only painful and time-consuming but also vulnerable to human errors. Moreover, the burden is getting larger as the technology scales.

Validating modern mixed-signal designs poses additional challenges since one needs to verify a full system including both digital and analog sub-systems. A brute force way is to perform a Verilog-AMS (analog mixed-signal) simulation [100], but it will be incredibly slow when the system is large. A more typical approach employs a lumped functional model of the analog sub-systems. This method works fairly well and is widely adopted in the industry.

Figure 6.1(b) shows the overall design flow of the proposed Open Source PHY. While the design flow of the digital sub-systems is exactly the same as the conventional flow, the flow of the analog sub-systems is significantly different. At first, the design is entirely described in SystemVerilog with generic gate models and is verified through Verilog simulation. The generic gates do not contain any process specific information but are defined by high-level parameters, such as FO4 delay of a given technology node, noise, and jitter. Unfortunately, not all the AMS blocks can be directly verified by the digital flow since the Verilog simulator does not understand the purpose of circuitry containing gates whose outputs are shorted, such as the phase blender and the bias generator. For those few building blocks, piece-wise linear functional Verilog models were generated through DaVE [101], an automated model generation flow. Matching between the actual circuit and the generated functional model can be ensured by mProbo which is a consistency checking engine of DaVE .

Figure 6.2 shows the Verilog simulation results of the proposed ADC as an example of the proposed flow. Note that the proposed flow reduced simulation time by more than 10,000x as compared to conventional SPICE simulation while providing sufficient accuracy. Thanks to the event-driven nature of the functional models, the full-system verification can be performed seamlessly with a minimal amount of overhead. The verified design with the generic gates is mapped into actual gates with specific sizes in the given technology through synthesis and then laid out through

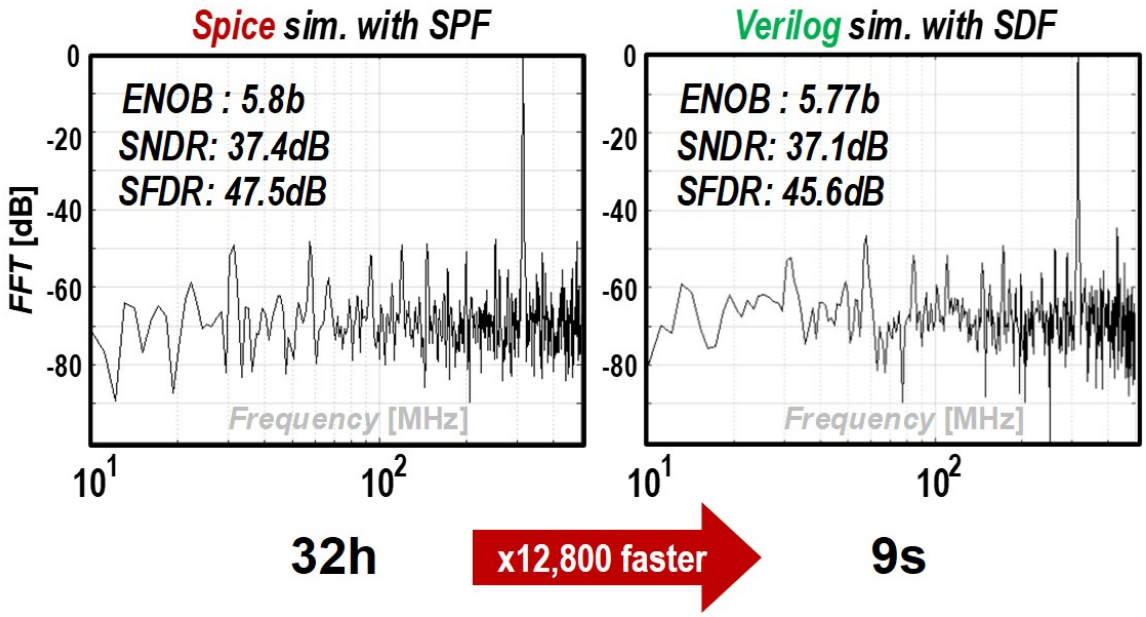


Figure 6.2: AC simulation results of the proposed ADC. (a) Spice simulation with extracted spf. (b) Verilog simulation with extracted sdf.

PnR. For the building blocks containing shorted gates, extra care is required in determining the size of the gates since STA usually cannot map the right size for those gates. Details about the size mapping of the gates with ambiguity will be discussed in the next section. Throughout the entire design, two custom cells, not in the standard cell library, are used; the switch cell and the current unit cells.

We publicly released all required design collateral except technology-related information so that potential users of this open-source platform can generate instances of our design tailored to their unique needs [102]. This includes the RTL, the functional model generator, the generic gate library, self-checking test benches, timing constraints, and a physical design flow. In addition, we also provide an FPGA hardware emulation model generator and corresponding test flow [103]. The users can map the proposed architecture to an FPGA and perform an accelerated emulation for end-to-end tests such as a jitter tolerance test [104].

6.2 Physical Implementation

Physical design of the proposed architecture is done with a standard digital PnR tool (Cadence-Innovus) with no manual layout, aside from the two custom cells mentioned previously (a switch cell and a current unit cell). The entire physical design flow is codified by Tool Command Language (TCL) in a highly parameterized fashion so that the users can customize their own design as necessary. The custom cells are currently designed through the conventional analog design flow (Cadence-Virtuoso) and instantiated in the PnR stage. However, they can also be generated via recent works on automated cell generation as discussed in Section 8.1. Size and port locations of the custom cells are designed to be compatible with standard cell grids so that they can be placed without causing any DRC problems. A Quick Time Model (QTM) supported by Synopsys's Prime Time is used to reflect loading and driving capability of the cells.

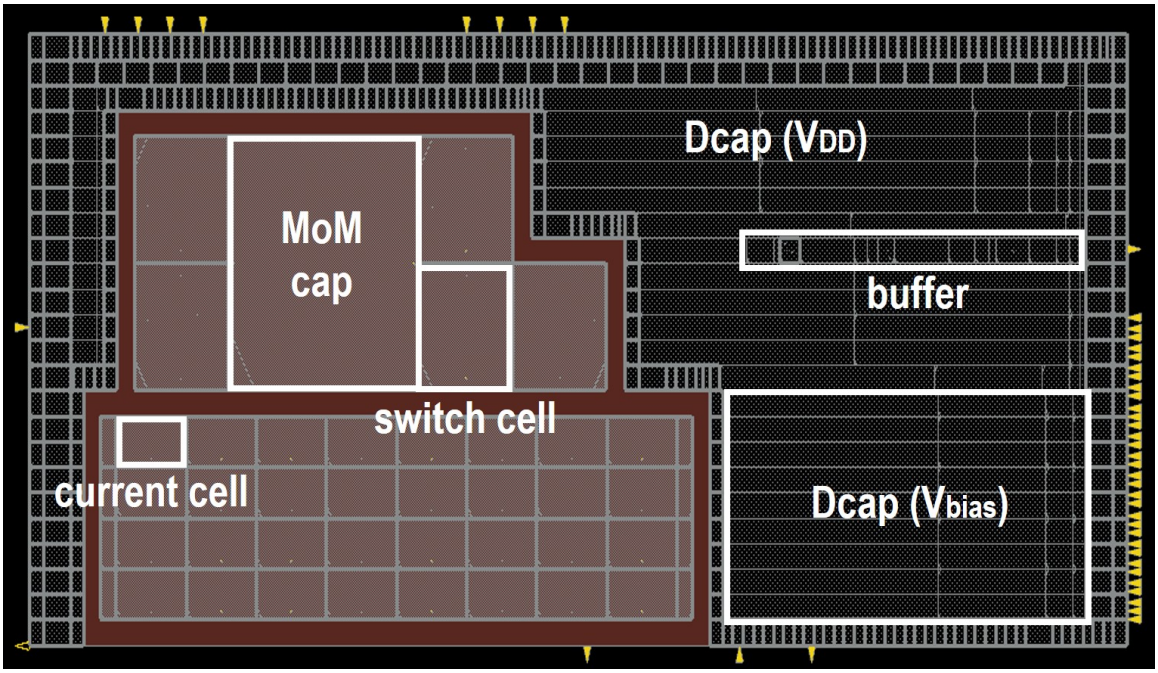


Figure 6.3: Layout of the proposed V2T (single ended).

Figure 6.3 shows the layout of the V2T slice. The sampling capacitor of the V2T can be implemented with either a Metal-oxide-Metal (MoM) pcell given by the process after creating another custom cell dedicated for the cap, or interdigitized metal wires directly drawn by the PnR tool. The value of the capacitance does not need to be accurate because its effect can be calibrated after the quantization (Appendix A).

Layout of the ADC is flat except for the V2Ts to simplify the design as shown in Figure 6.4. For the most part, the physical design of the proposed architecture is compatible with the conventional PnR flow as long as we prevent excessive optimization of the delay lines in the design by applying commands like ***set_dont_touch***. However, there are a couple of exceptions that require extra care. One challenge in the physical design of the ADC is that high frequency clocks connected between the V2T clock generator and the V2Ts require very tight timing. The Clock Tree Synthesis (CTS) with appropriate constraints should be able to put enough buffers so

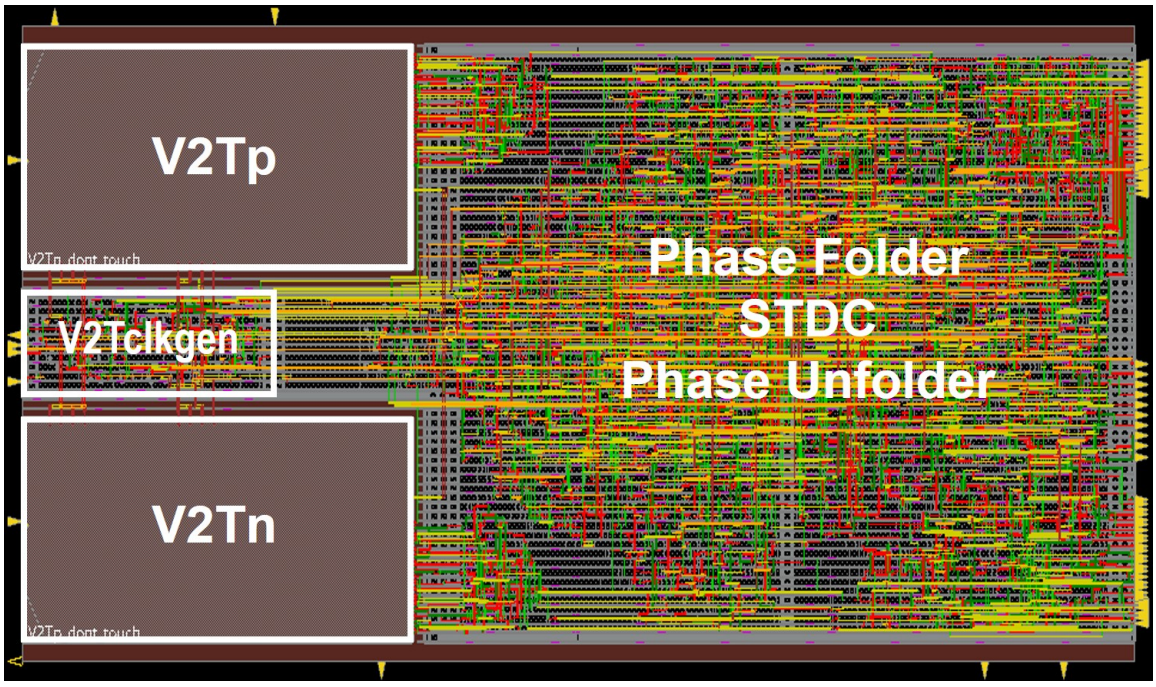


Figure 6.4: Layout of the proposed ADC.

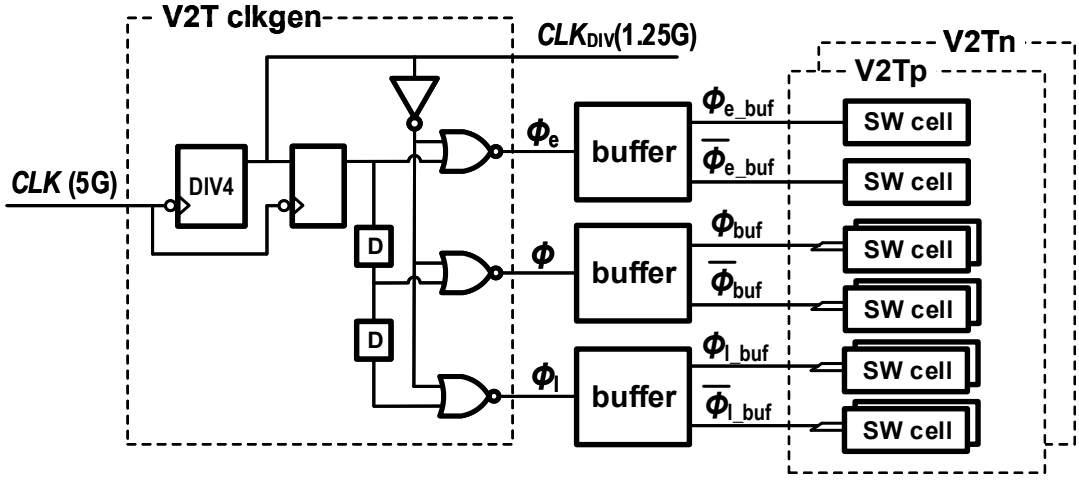


Figure 6.5: High frequency clocks connected between the V2T clock generator, and the V2Ts and the buffer insertion associated with them.

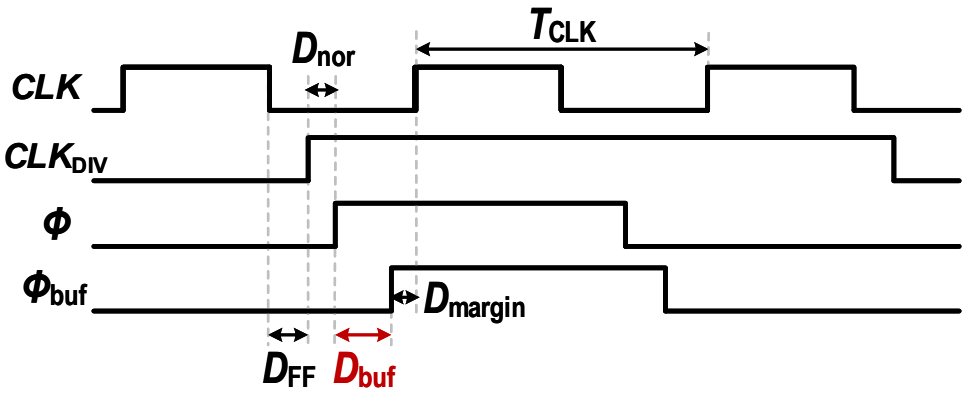


Figure 6.6: Delay requirements of V2T's sampling clocks.

that the differential clocks can successfully drive the CMOS switch cells in the V2Ts and, at the same time, their phase skew should be small enough not to degrade the performance of the V2T, as shown in Figure 6.5. However, it is not an obvious job for

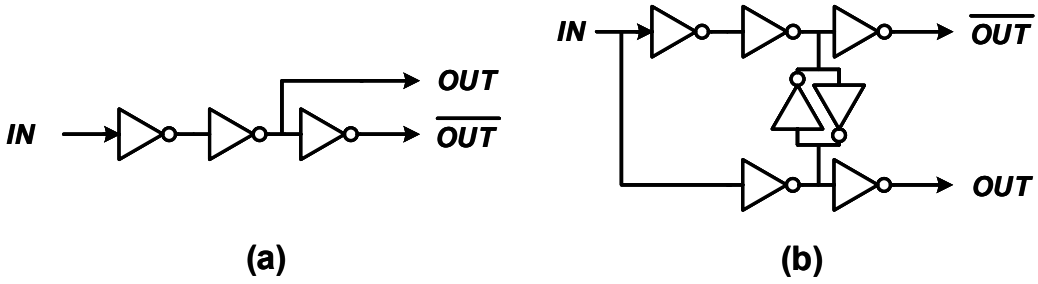


Figure 6.7: Implementation of the V2T clock buffer. (a) A typical example implemented through the conventional PnR flow. (b) A balanced clock buffer implemented by the proposed PnR flow.

the CTS when we think about the delay requirements of these signals in the proposed ADC as shown in Figure 6.6, where T_{CLK} is period of the main clock, CLK, generated from the PI (1/4 of baud rate). The time interleaving works properly only when the buffered V2T clock, Φ_{buf}, toggles from low to high after the falling edge of CLK and before the rising edge of CLK. This means that the overall propagation delay from CLK to Φ_{buf} should be less than $T_{CLK}/2$. Given the fact that clock-to-q delay of a flip-flop is around 50ps and FO4 delay of logic gates is around 10ps in the target technology (16nm FinFET), the maximum number of stages allowed for the clock buffer is only 3. Under these constraints, Figure 6.7(a) shows a typical example when the clock buffer is implemented through the conventional PnR flow. Unfortunately, it can significantly degrade the performance of the V2T due to the mismatches in delay and slew rate between differential output clocks. Figure 6.7(b) shows a balanced clock buffer which is widely used in custom circuit designs but hard to realize in the conventional PnR due to the logical ambiguity (output-shorted gates) in the circuit. It is obvious that the balanced clock buffer is a better solution, in terms of the V2T's sampling performance, as long as it can be implemented by the PnR tools.

Figure 6.8 illustrates the proposed PnR flow dedicated to the balanced clock buffer as a virtual procedure done before the actual PnR flow. First, an auxiliary structural

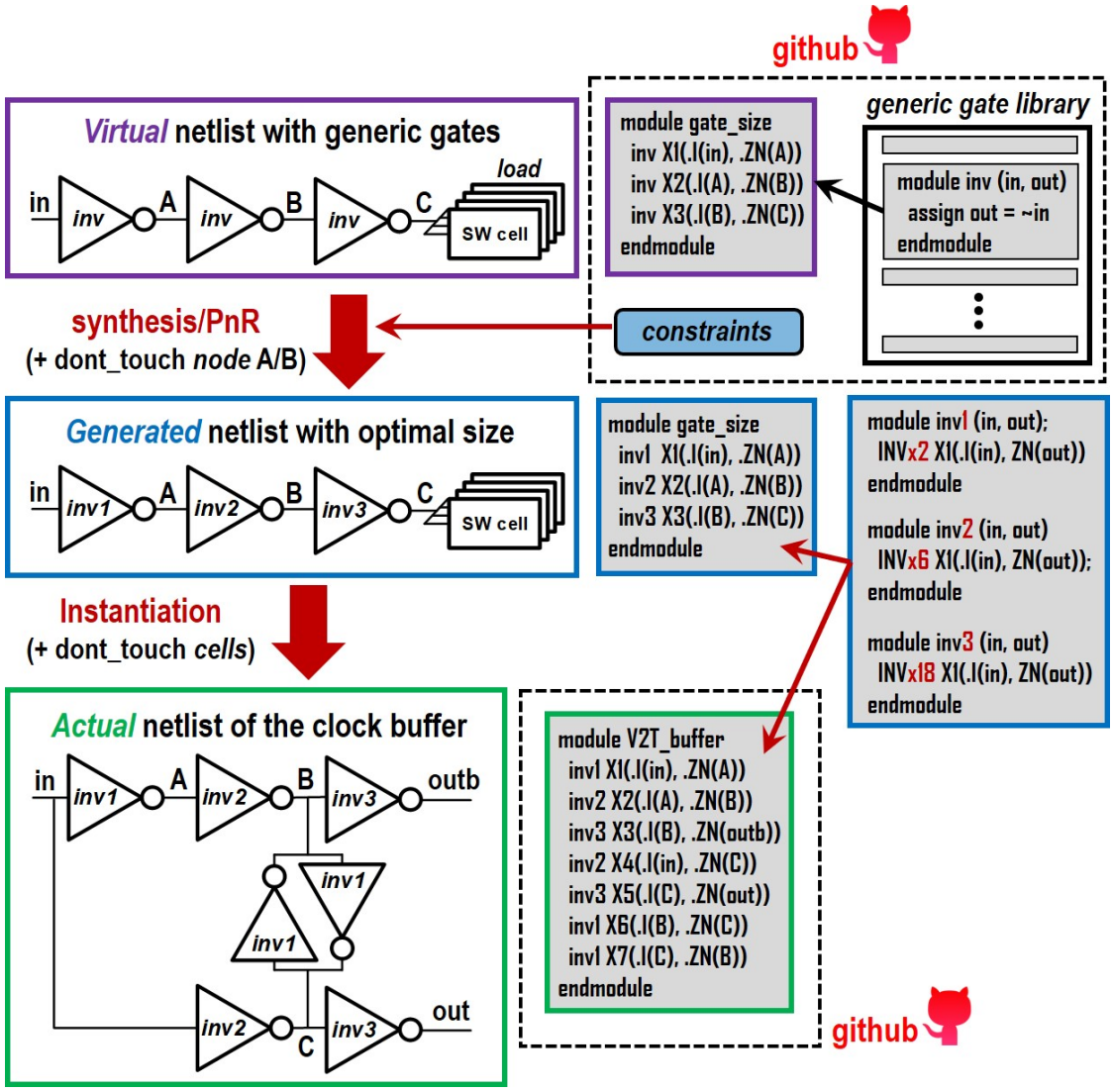


Figure 6.8: Implementation flow of the balanced V2T clock buffer.

netlist (**gate_size.sv**) that contains a 3-stage generic inverter buffer along with its load (4 switch cells in this case) is taken from the repository. This netlist is not an actual design but a virtual design used only to guide the actual implementation of the buffer. Each inverter cell of the buffer has an identical generic definition but unique instance names (inv1, inv2, and inv3). The virtual netlist is synthesized and laid out under appropriate constraints without touching the intermediate nodes of the buffer

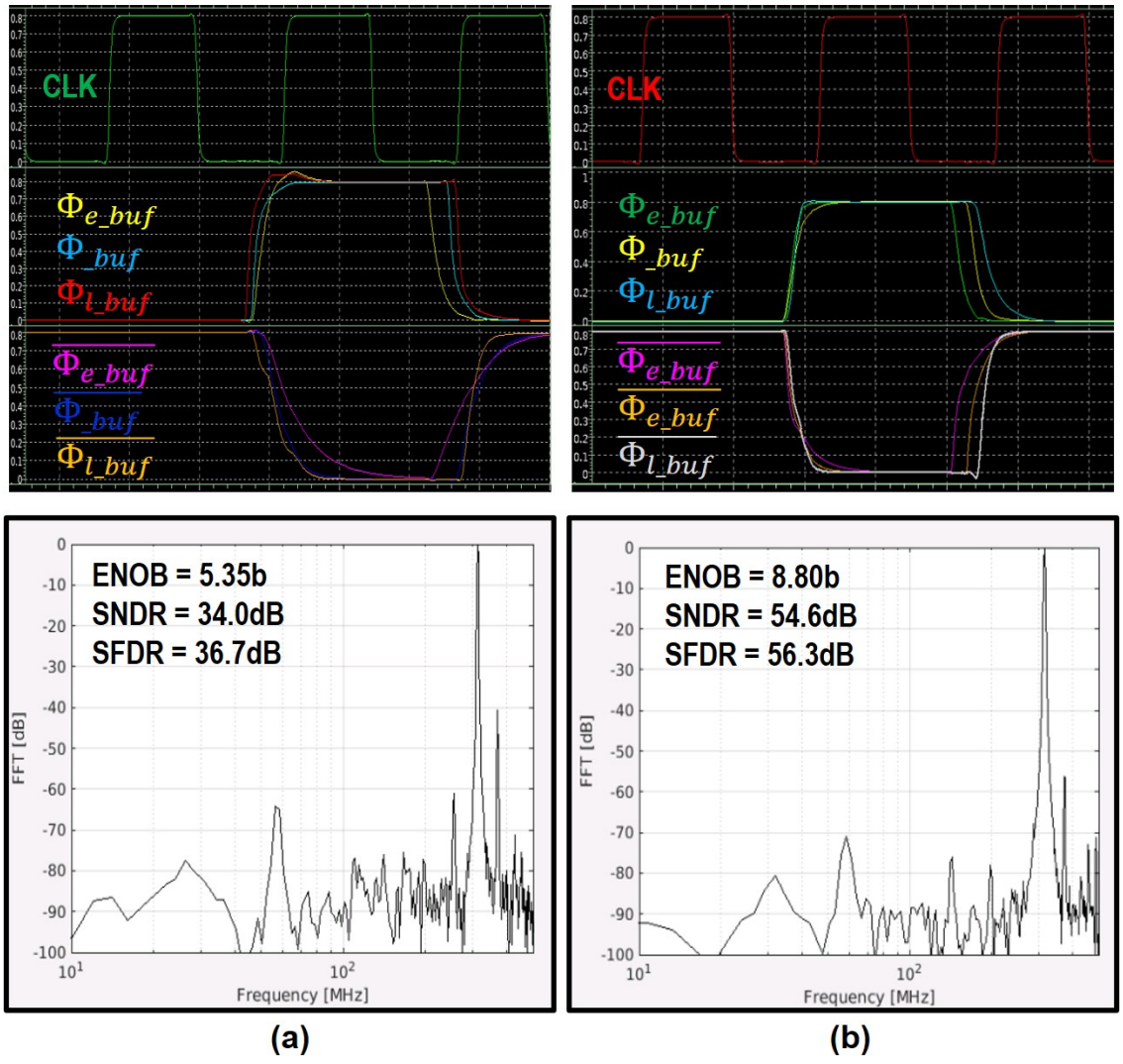


Figure 6.9: SPICE simulation (extracted) results of the V2T clock buffer and corresponding V2T performance (F_{sample}=1GHz, F_{in}=314MHz). (a) Clock buffer designed by the conventional PnR flow. (b) Clock buffer designed with the proposed virtual flow.

(node A and node B). Since there is no logical ambiguity in the netlist, each inverter can be successfully mapped into a specific definition with different sizes according to the constraints. On the other hand, in the actual design, a structural netlist of the buffer (**V2T_buffer.sv**) instantiates the size-mapped inverter cells without touching

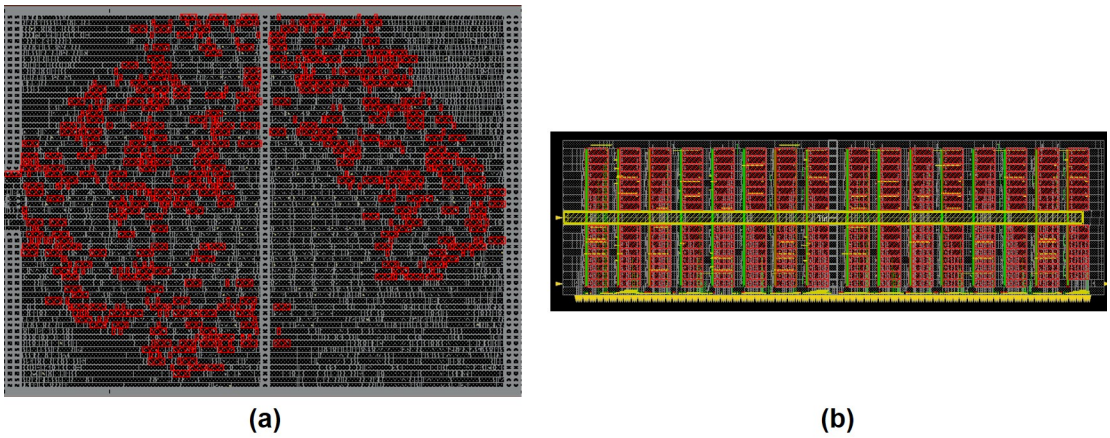


Figure 6.10: Placement of the TDC delay chain where the red boxes denote flip-flops (a) Random placement (conventional). (b) Manual placement with non-default routing (NDR) width.

them. Figure 6.9 shows SPICE simulation results of the clock buffers and corresponding AC characteristic of the V2T (differential). While the conventional clock buffer severely degrades integrity of the clocks, and in turn, the performance of the V2T, the proposed balanced buffer performs well enough so that the V2T does not hurt the performance of the ADC.

Another challenge in the physical design of the ADC is handling the high fan-out node in the stochastic TDC. The output of the phase folder, PF_{OUT}, is a pulse whose width can be as narrow as D_{OFFSET} (∼100ps), and it fans out to the data input of 255 flip-flips. Moreover, since the clock inputs of each flip-flop generated from the delay chain are asynchronous with one another it is hard for the tool to perform a timing analysis with the conventional constraints such as ***set_max_fanout***. The simplest approach for this issue is to disable the timing analysis of all the flip-flops and drive them using a huge-sized buffer. However, it won't help much since parasitic resistance of the interconnects among the randomly placed flip-flops eventually will limit the bandwidth of the node as shown in Figure 6.10(a) and Figure 6.11(a). To

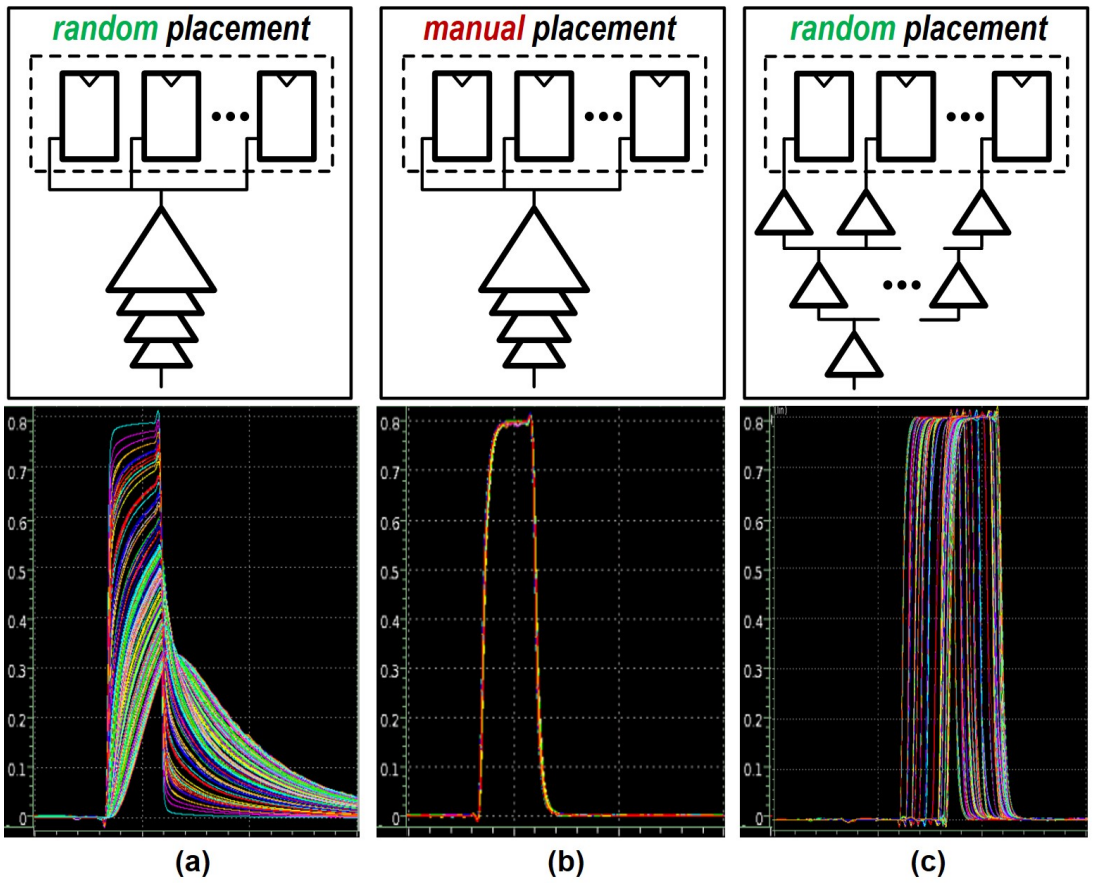


Figure 6.11: Simulated waveforms at data input of the flip-flops with respected to different buffering topology and placement policy. (a) A huge buffer and randomly placed cells. (b) A huge buffer and manually placed cells with wide metals. (c) A balanced buffer tree and randomly placed cells.

minimize the interconnect resistance the flip-flops may be manually placed to the optimal coordinates and routed with wider metals as shown in Figure 6.10(b). Although this may achieve very good results with the minimal amount of power consumption, as shown in Figure 6.11(b), it is not aligned with the direction this dissertation is pursuing. It not only tends to require the users of this open-source design platform to care about low-level design parameters (such as metal resistance and placement coordinates), but also is not flexible for process porting. Figure 6.11(c) shows the

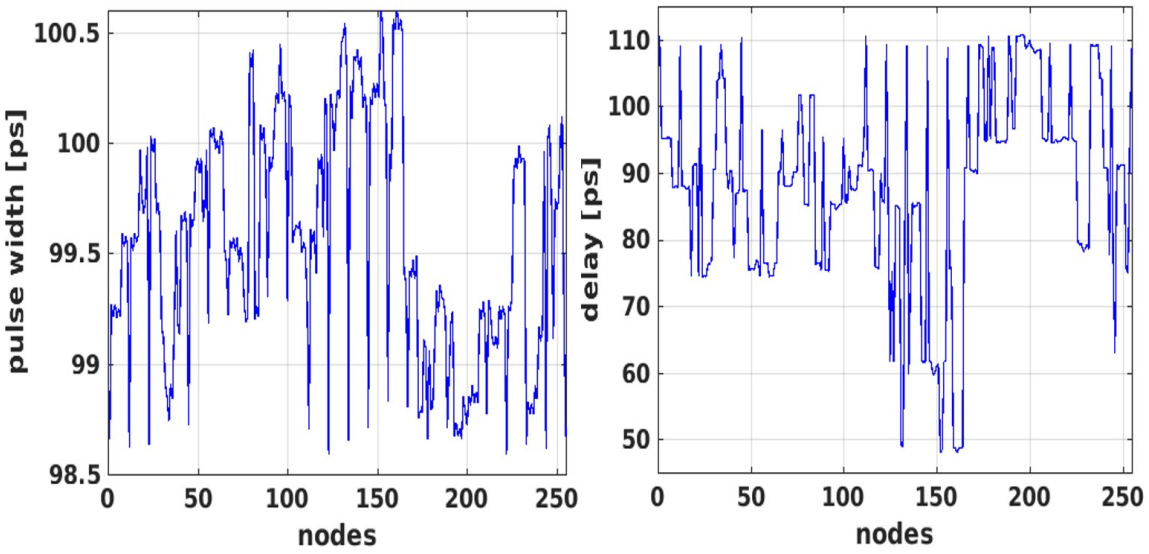


Figure 6.12: Width and delay spread of the pulses at data input of the randomly placed flip-flops when CTS is applied (Case (c) in Figure 6.11).

proposed approach. We allowed the tool to place the flip-flops and the data input of the flip-flops is buffered through a clock tree. If the output of the phase folder is defined as a virtual clock source, and data input of the flip-flips is defined as sinks of the clock within an identical skew group, the CTS flow can generate the balanced buffer tree according to the constraints, ***max_tran*** and ***max_delay***. Figure 6.12 shows the spread of width and delay of each pulse at the data input of the flip-flops. The delay of each pulse ranges from 50ps $\sim$ 110ps, however, thanks to the stochastic nature of the proposed TDC, it does not affect the results as long as each width stays the same. The variation of the pulse width over all the flip-flops is less than 1ps which is small enough compared to the target LSB of the proposed TDC.

The PI is flat except for the two cells, the delay unit cell and the 4-to-1 mux cell, as shown in Figure 6.13. The delay unit consists of a delay cell (two inverters), an arbiter, a phase mixer, and an adjustable delay buffer. Unlike the stochastic TDC, the unit cells of the PI are placed manually since the delay of the unit cells and mismatch among them affect the step size of the PI.

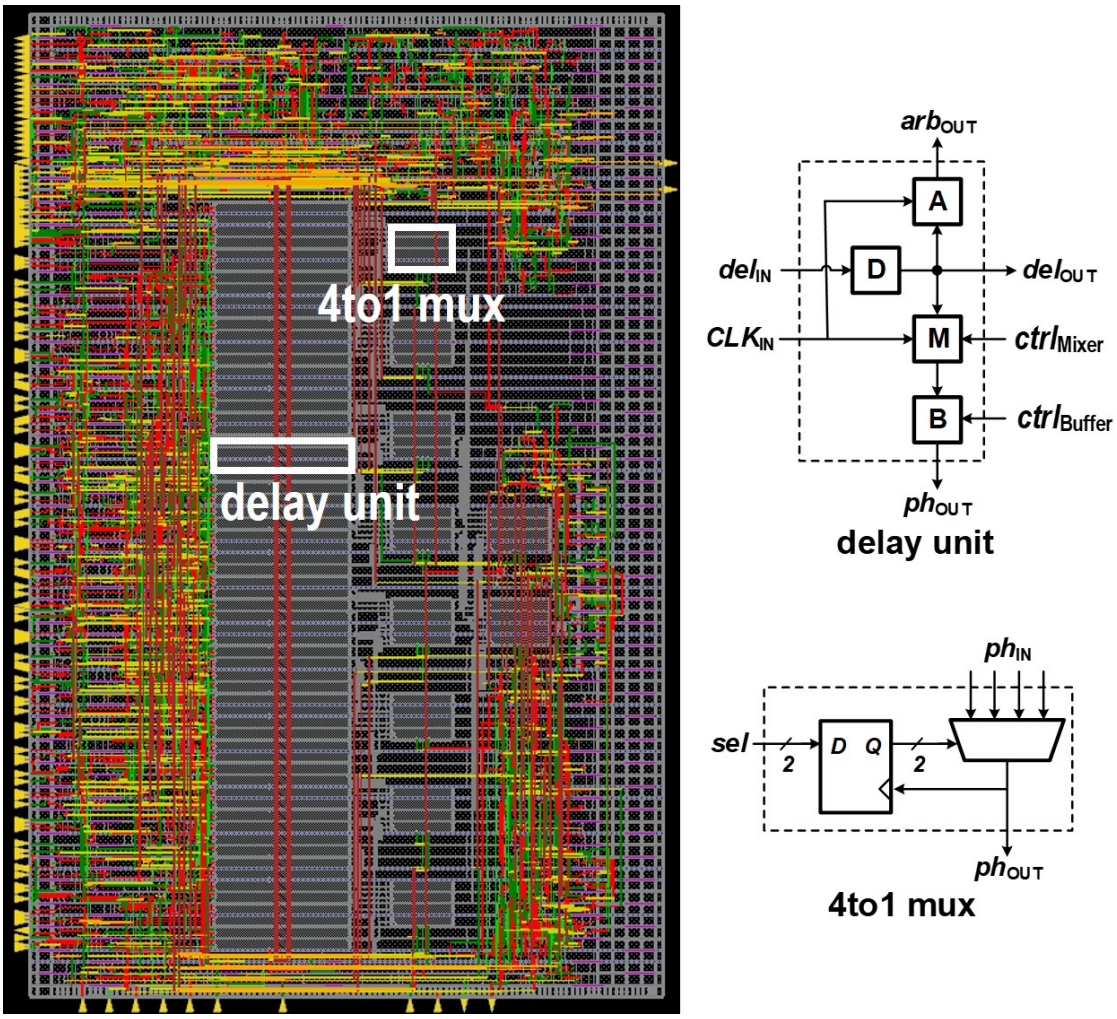


Figure 6.13: Layout of the proposed PI.

6.3 Summary

The proposed designs described entirely in SystemVerilog were verified through Verilog simulation. A structural netlist with process-independent generic logic gate models was used for timing-critical simulations. This reduced simulation time by over 10,000x as compared to conventional SPICE simulation while providing sufficiently accurate results. For the few building blocks that could not be directly simulated by the Verilog simulator, event-driven functional models were generated through DaVE,

an automated model generation flow from Stanford [101][105][106]. Physical design of the proposed architecture was done with a standard digital PnR tool with no manual layout, aside from the two custom cells mentioned previously (a switch and a current source). Physical design of timing critical circuits, such as balanced differential clock buffers, was implemented by the proposed virtual PnR flow. We ran PnR on an individual ADC slice and an individual PI slice, and then instantiated the resulting layouts multiple times in the top-level design which will be discussed in Chapter 7.1.

Chapter 7

Receiver Front-End and Measured Results

In this chapter, we describe the proposed receiver front-end. Section 7.1 shows the proposed architecture of the receiver including ADCs, PIs, bias generators, and macros for testing purposes. A prototype test chip of the receiver along with measurement results is described in Section 7.2.

7.1 Proposed Architecture

The overall architecture of the proposed receiver front-end is shown in Figure 7.1. Four PIs take a quarter-rate input clock and generate 5 GHz quadrature sampling clock phases for the first stage T/H switches. Each PI has an independent control dedicated to cancel clock skew among sampling phases and drives one 1st stage T/H switch and four ADC slices. The quantized data from each ADC slice are synchronized by a double flop aligner to become the final output. An off-chip static non-linearity calibration is supported by an on-chip SRAM. INL of the ADC is calibrated by a post mapping using lookup tables that are generated based on DC transfer function of each ADC slice. The differential input voltage is terminated by a 100 Ω resistor with a middle-tapped common mode voltage and the differential input clock is driven by a custom designed, AC coupled, inverter-based clock buffer. Power domain of the

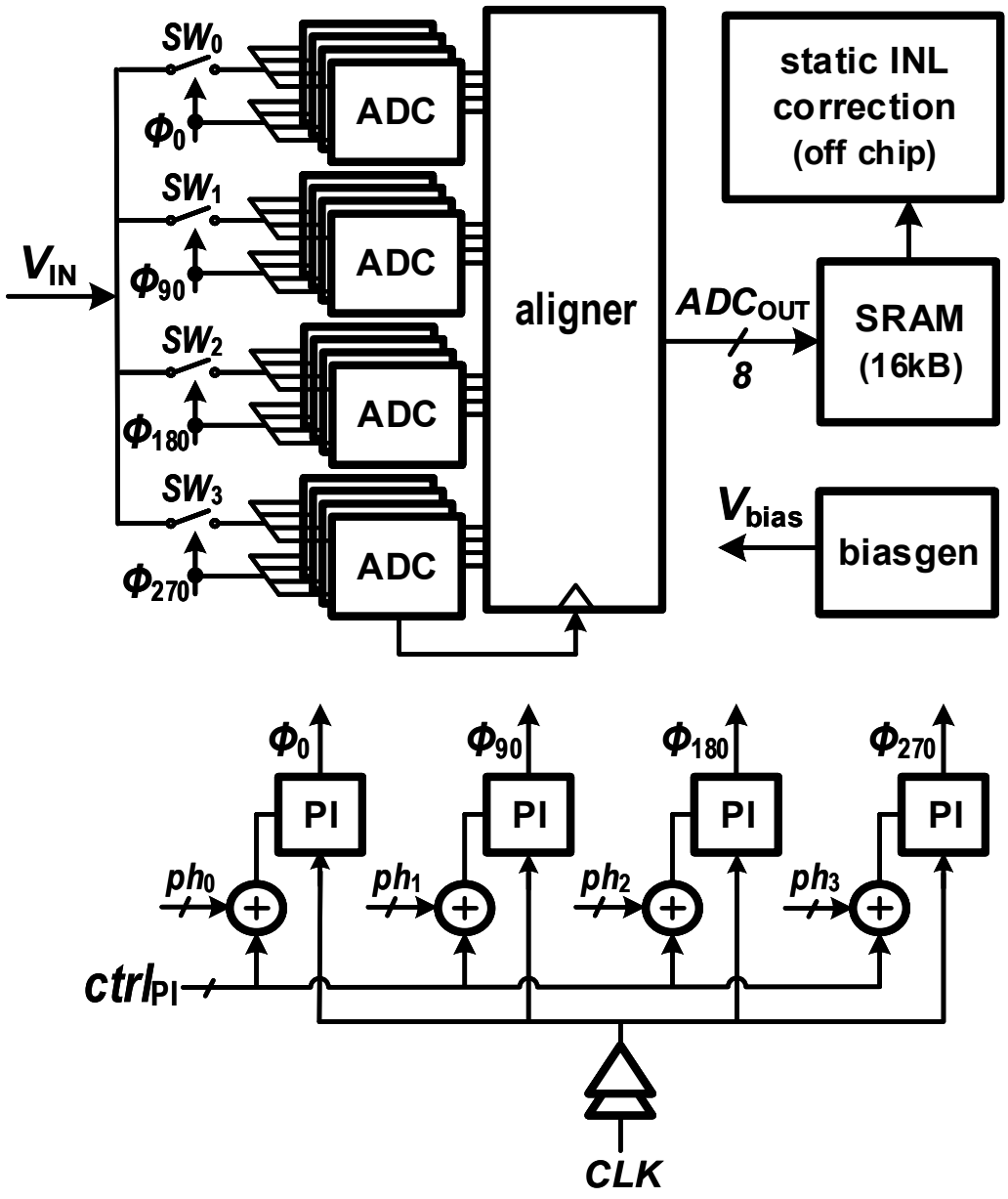


Figure 7.1: Proposed synthesizable receiver front-end.

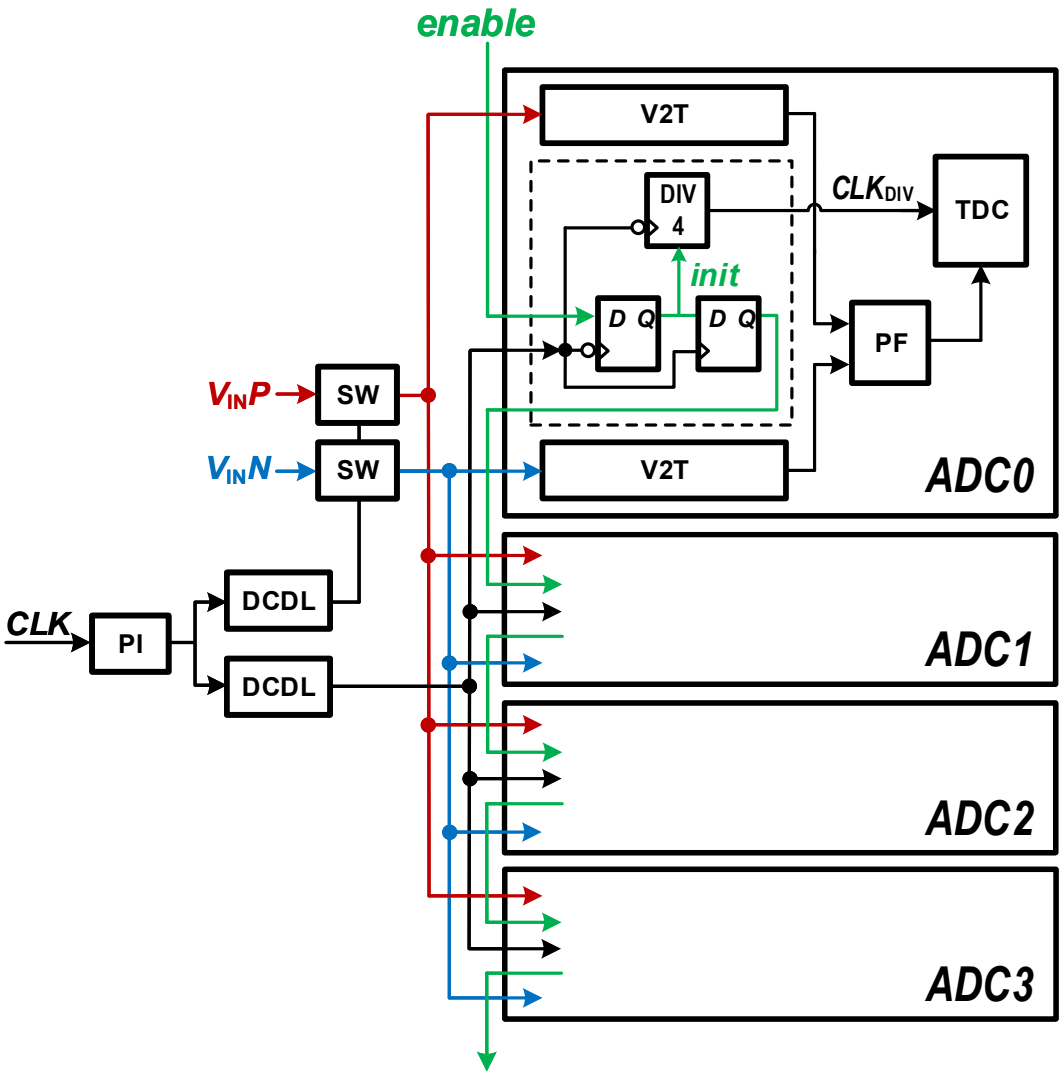


Figure 7.2: Sampling phase aligning based on sequential initialization.

receiver is divided into three sections: V2Ts in each ADC slice are on $AVDD$, circuits for clock generation and distribution are on $CVDD$, and the rest of the circuits are on $DVDD$.

The output clock of each PI is divided into two paths, one dedicating for the T/H switch and the other driving the ADC slices. A 2-bit DCDL is inserted into each path to calibrate the skew between the paths. The frequency divider of the V2T clock generator in each slice of ADC is initialized sequentially by the previous slice to guarantee proper phase alignment as depicted in Figure 7.2.

7.2 Measured Results

A prototype chip was fabricated in the TSMC 16nm FinFET technology, and then packaged in a 672 pin standard FBGA (Flip-chip ball grid array) as shown in Figure 7.3. The areas of an ADC slice and PI are $80\mu m \times 40\mu m$ and $25\mu m \times 80\mu m$ and they consume 8.6mW and 9.6mW, respectively. The area of the 16-channel ADC, including the four PIs, four bias voltage generators, and an input clock divider is $300\mu m \times 340\mu m$ and the total power consumption is 175mW under a 0.9V supply.

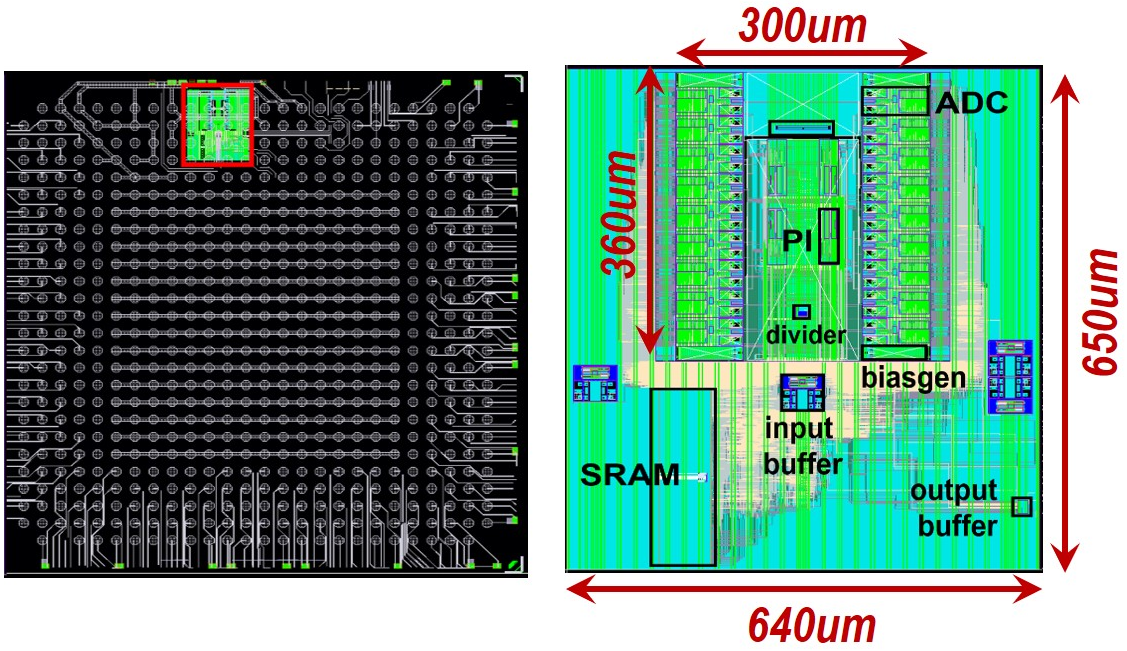


Figure 7.3: Top layout view of the test chip.

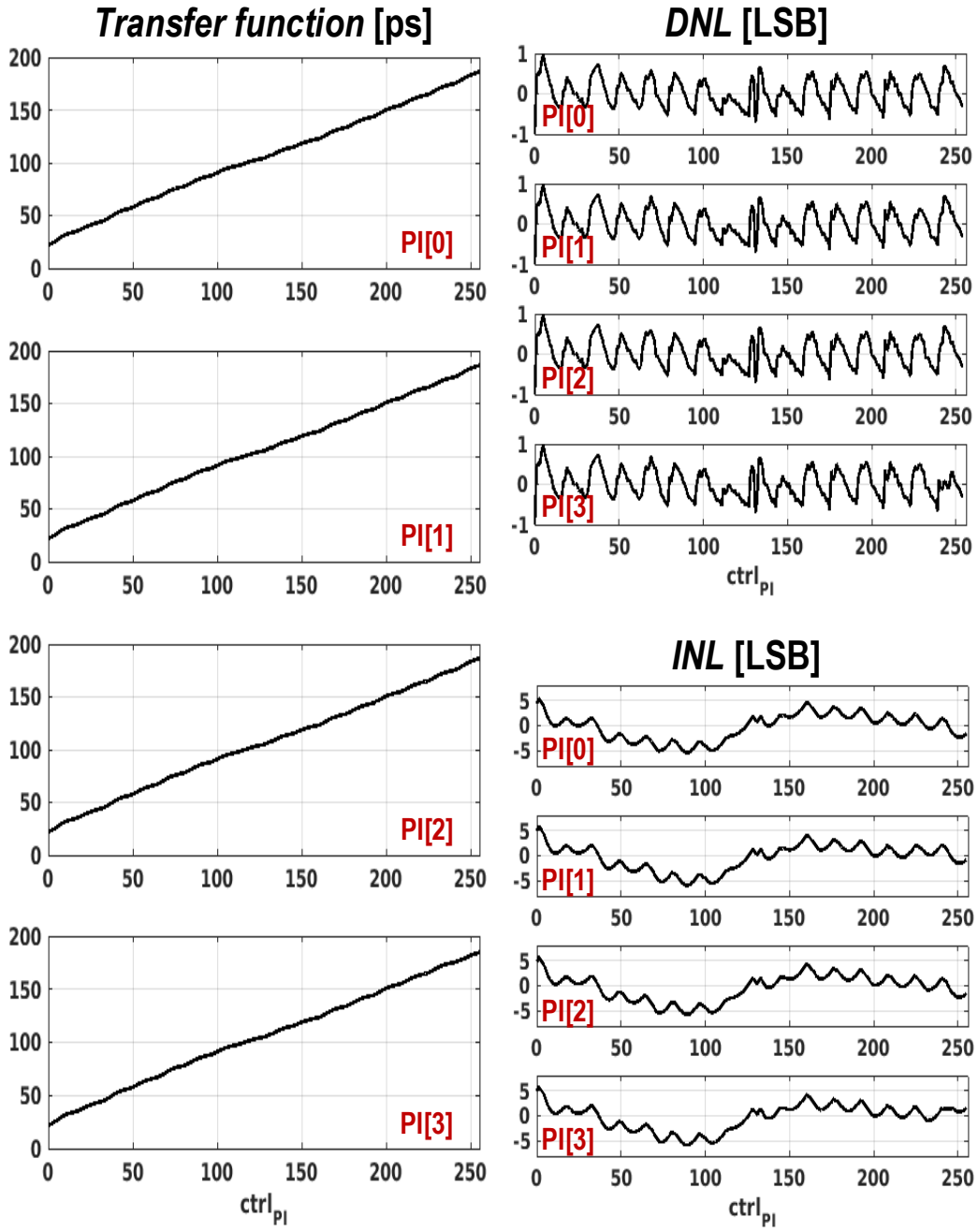


Figure 7.4: Measured results of the proposed PI.

This includes a substantial amount of area dedicated to control signals used for testing, which would not necessarily be included in a production design.

Figure 7.4 shows the measured performance of the proposed PI for a 5GHz input clock. The measured static transfer function of the PI is monotonic with the worst case phase step of 0.7 ps.

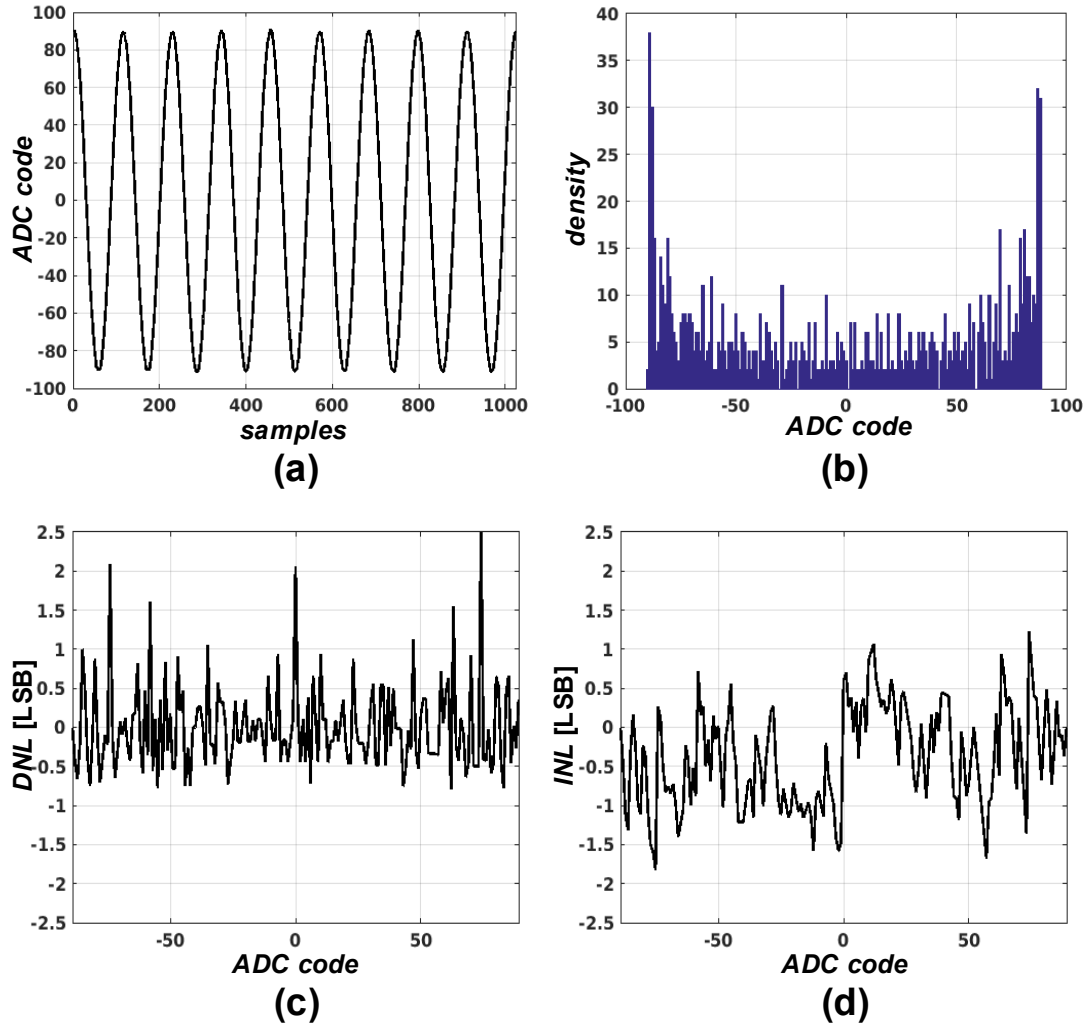


Figure 7.5: Measured result of the single channal ADC. (a) Transient output, (b) histogram, (c) DNL, and (d) INL.

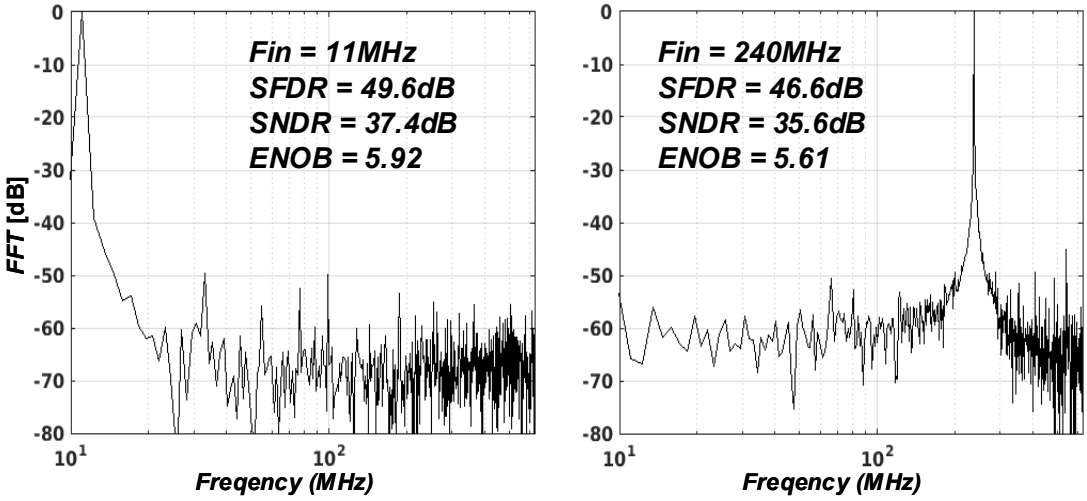


Figure 7.6: Measured results of the single channel ADC. 1,024 points FFT of the output for (a) F_{in}=11MHz and (b) F_{in}=240M.

The measured performance of a single-channel ADC for a full scale sine input is shown in Figure 7.5, demonstrating a DNL of 2.5 LSB and an INL of 1.8 LSB without the linearity correction. The DNL is caused by delay mismatch among the delay cells in the stochastic TDC, mostly due to the interconnects routed by the automated PnR. The abrupt jump in INL near zero is due to the finite resolution of the PF offset calibration loop. Figure 7.6 shows 1,024 points FFT results of the single channel ADC for different input frequencies. The ADC achieves an ENOB of 5.92 for 11MHz and 5.61 for 240MHz of input frequency after the static INL correction in MATLAB.

After characterizing each ADC slice, we measured the performance of the overall time-interleaved ADC. The quantized data from each ADC slice is mapped into an ideal sine wave through a linear regression to find out the sampling clock phase of the slices. Based on this phase estimation, we adjusted the control code of the individual PI ($ph_1 \sim ph_4$) to cancel the sampling phase skew among the slices as shown in Figure 7.7. Figure 7.8 shows measured non-linearity of the 16-channel ADC before and after the INL calibration. DNL of the interleaving ADC is 0.58 LSB, which is much less than that of single channel ADC, since the effect of mismatch among the delay cells

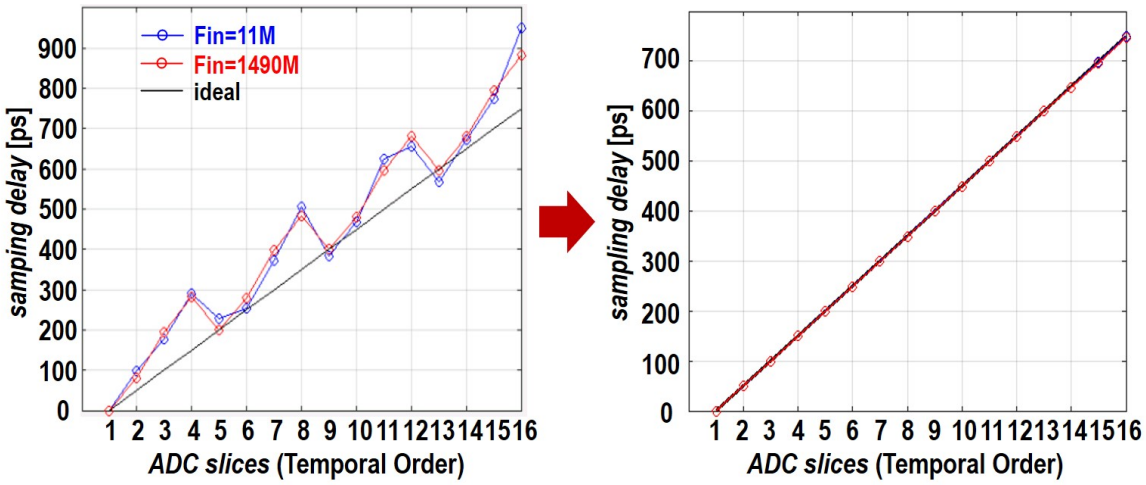


Figure 7.7: Sampling clock phase of each ADC slice before and after the skew calibration.

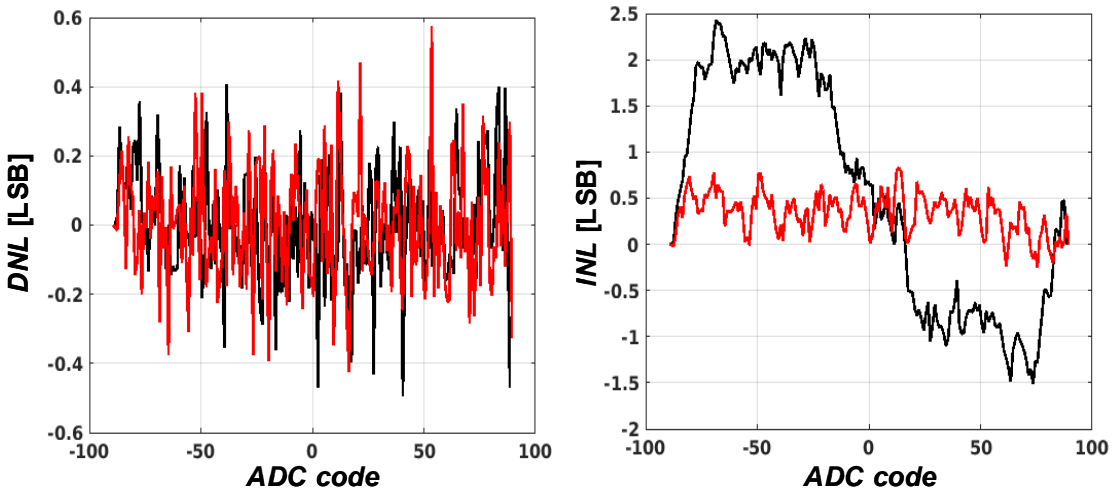


Figure 7.8: Measured results of 16-channel ADC. Static non-linearity of 16-channel ADC before (black) and after (red) the INL calibration. (a) DNL. (b) INL.

in each ADC is averaged in the aggregated output of the interleaved ADC. INL of the interleaved ADC is suppressed to 0.82 LSB by the INL calibration. Figure 7.9 and Figure 7.10 shows 16,384-point FFT results of the 16-channel ADC and its input

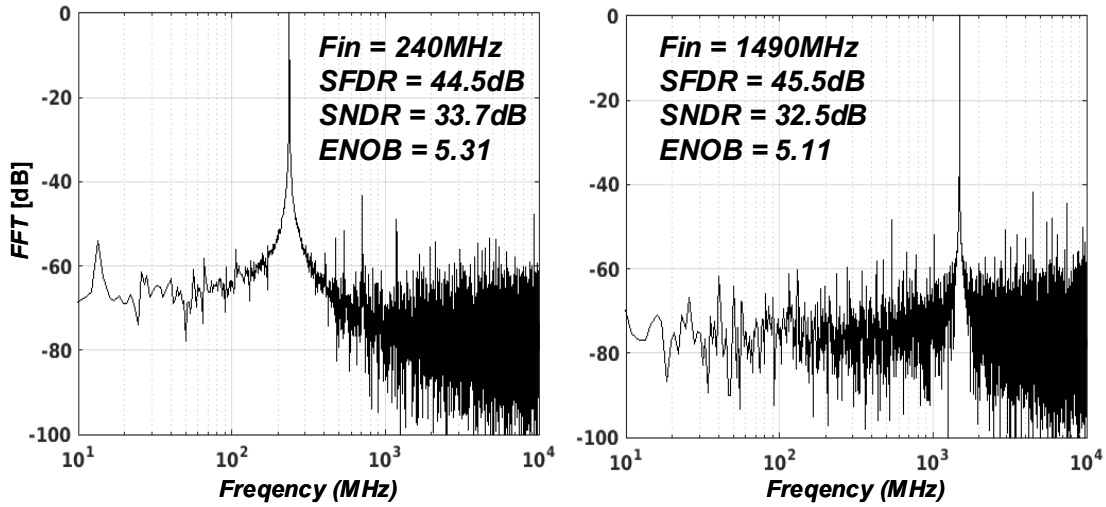


Figure 7.9: Measured results of 16-channel ADC. 16,384 points FFT of the output for (a) F_{in}=240MHz and (b) F_{in}=1490MHz.

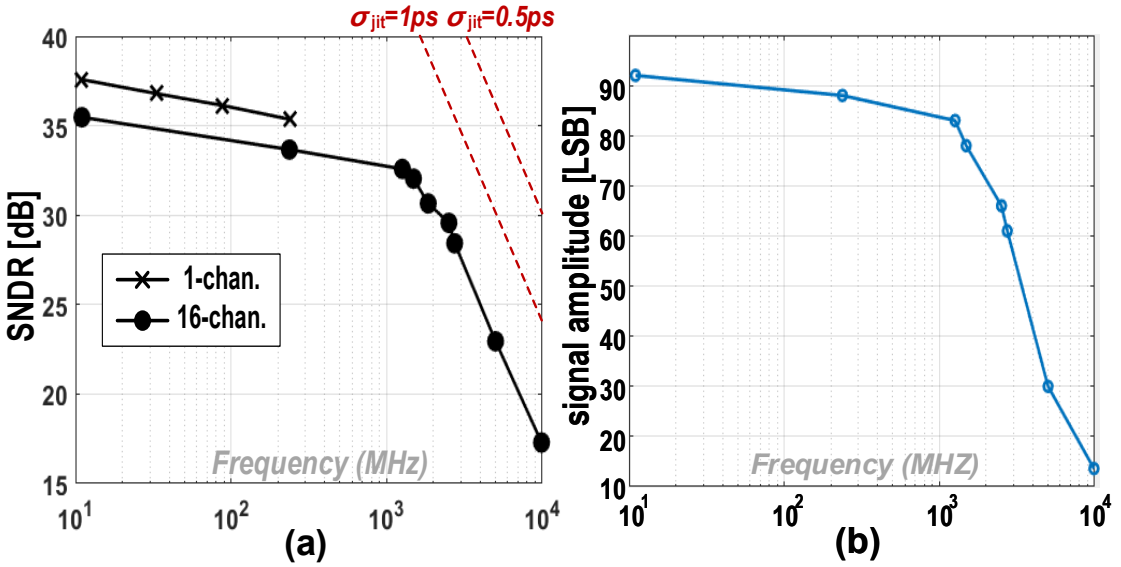


Figure 7.10: (a) ENOB and (b) quantized signal amplitude with respected to input frequency.

frequency dependency of SNDR, respectively. The ENOB is 5.6 at low frequencies, falling to 2.7 at Nyquist. The strong ENOB roll-off at high frequencies is mainly due to input power loss resulting from the poor input network of the package, which was not optimized for high-frequency testing (Figure 7.10(b)).

Figure 7.11 shows power breakdown of the prototype design. The power consumption of an ADC slice is dominated by that of the digital adder (Wallace tree). Fortunately, this scales along with technology scaling. The delay chain dominates the power consumption of the PI. While the unit delay of the delay chain in TDC does not affect the performance of the TDC, the unit delay in PI determines the phase step size of the PI. Thus, the delay chain of PI is implemented by a low threshold voltage (LVT) FO2 delay line and consumes much more power than that of TDC, although its length is shorter.

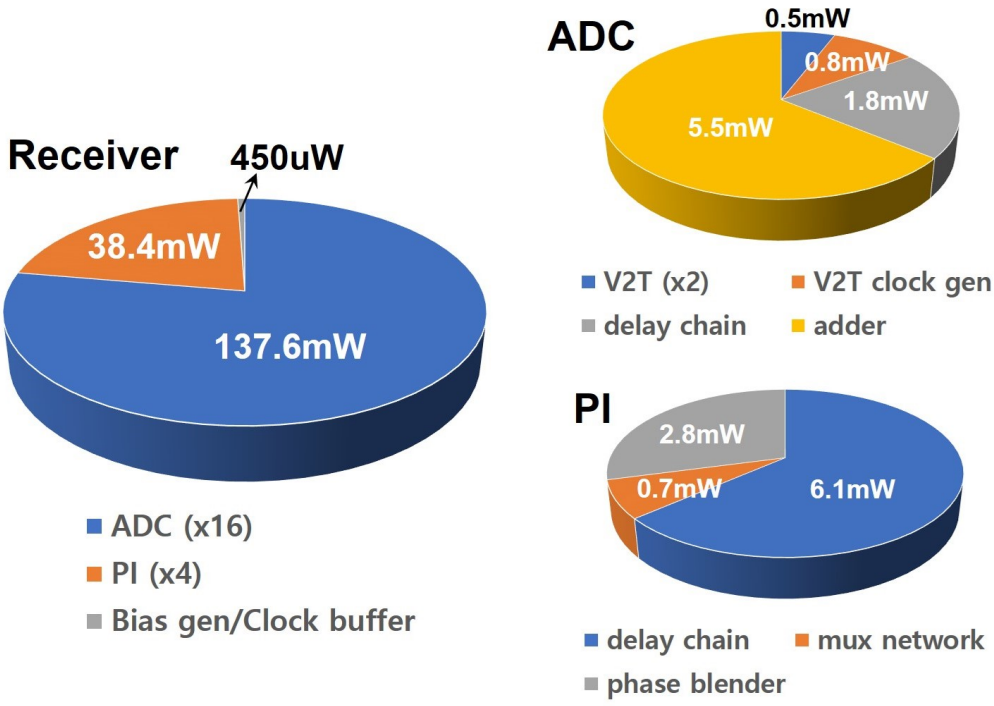


Figure 7.11: Power breakdown of the prototype design.

7.3 Summary

We created a 20 GS/s, 8-bit, time-interleaved ADC, for the use in high-speed serial links, with an ENOB of 5.6, a DNL of 0.58 LSB, and an INL of 0.82 LSB, which dissipated 175mW in 0.102 mm^2 in a 16nm technology. The design is entirely described in an HDL so that it can be ported to other processes with minimal effort and shared as open source.

The performance of the proposed ADC-based receiver front-end is summarized and compared with other papers in Table 7.1. Among previously published synthesizable ADCs, which are based on stochastic flash ADC, the proposed work achieves the smallest size and the best performance in terms of pico-joule-per-bit efficiency. Compared to the state-of-the-art, custom designed high-speed ADC [107], the proposed work shows poorer power efficiency by a factor of $3 \sim 4$. This trade-off between design productivity and power efficiency in the proposed work, along with the necessity of two custom cells (switch cell and current unit cell), will be one of the most interesting topics for the future research (Section 8.1).

	[54] TCAS-II,09 Sundstrom	[62] TCAS-I Weaver	[57] TCAS-II,15 Fahmy	[58] VLSI,04 Paulus	[46] INCSC,14 Chen	[21] JSSC,13 Varzaghani	[107] ISSCC,19 Pisati	This Work	
Type	Stochastic Flash ADC	Stochastic Flash ADC	Stochastic Flash ADC	Stochastic Flash ADC	Stochastic Flash ADC	Flash ADC	SAR	Stochastic Flash TDC	
Process(nm)	90	90	130	130	32	40	7	16	
Supply(V)	1.2	1.2	1.0	1.5	1.1	0.9	0.9	0.9	
Speed(GHz)	1.5	0.21	0.32	4	20	10.3	30	1.25	20
ENOB	3.7	5.69	5.17	5.7	4.82	5.38	5.04	5.9	5.6
Power(mW)	23	34.8	87	990	69.5	242	53.5	8.6	175
FoM(pJ/b)	1.18	3.2	7.55	4.76	0.12	0.56	0.05	0.12	0.18
Area(mm^2)	0.04	0.18	0.51	0.5	0.25	0.27	-	0.003	0.1
Synthesizable	No	Yes	Yes	No	No	No	No	Yes	

Table 7.1: Performance summary of the proposed ADC. Left column is for a single channel ADC, and the right column is for 16x interleaved ADC.

Chapter 8

Conclusions and Future Work

In recent years, many open-source digital circuit generators have emerged to make it easier to create ASICs for demanding applications such as machine learning. Unfortunately, high-speed serial transceivers, which traditionally require precise custom analog circuits, are harder to open-source. As a result, designers still have to obtain proprietary transceiver IPs, which are expensive, hard to verify, and inflexible. To break this bottleneck, this dissertation describes the design and measurement of our open-source, portable analog block generator for a high-speed link receiver. We can open-source this design because it uses digital standard cells and digital Place and Route (PnR) tools and does not rely on precision analog designs. In fact, it takes advantage of inherent circuit variations in its novel stochastic time-to-digital converter (STDC) that, in conjunction with a synthesizable phase interpolator (PI) and synthesizable clock generator, are used to build a time-interleaved analog-to-digital converter (ADC). While the idea of synthesizable analog circuits or taking advantage of circuit variation to enhance performance is not new, our approach in designing high-speed ADCs is much more power-efficient than past efforts. Since the entire design is fully described in SystemVerilog, commercial digital simulators can be used extensively from the initial architecture exploration to the final performance sign-off. Moreover, this makes it easy to incorporate FPGA-based hardware emulation into design flow, especially for time-consuming full-link verification. We validated these techniques and design flow by fabricating a prototype chip.

We created a 20 GS/s, 8-bit ADC with an ENOB of 5.6, a DNL of 0.58 LSB, and an INL of 0.85 LSB, which dissipated 175 mW in 0.102 mm^2 in a 16nm technology. All required design collateral, except technology-related information, is publicly released so that users can generate instances of our design that are tailored to their unique needs. This includes RTL, a functional model generator, a generic gate library, self-checking test benches, timing constraints, a physical design flow, and a system for FPGA emulation.

8.1 Future Work

We think this work can be improved in three different directions.

First, the power consumption of the proposed architecture needs to be reduced. One possible solution is making the delay chain in TDC and PI more power efficient. In the prototype design of the TDC, we only used rising edges of the delayed input clock. Because of the offset delay of the PF, D_{OFF}, and latency of the digital buffer (comparator) in the V2T, the effective signal range of the TDC is only about 35% of the clock period as shown in Figure 8.1 (a). In other words, more than half of the interpolated clock edges do not contribute to the quantization. Once we know the maximum input of the TDC, we can double the effective density of the interpolated edges within the signal range by shifting the phase of the edges outside the range by 180 degrees using additional XOR gates as shown in Figure 8.1 (b). As a result, the hardware resources of the delay chain, sampler, and the following digital adder can be reduced by half for given target performance, saving a considerable amount of power. The number of delay units in the proposed PI was designed to be large enough to cover a wide range of input clock frequencies, and each delay cell consisted of two cascaded inverters in the prototype design. However, not all the delay units should be active for the proposed PI. We can simply change the two cascaded inverters in the delay cell to AND gates so that we are able to adjust the number of the active delay units according to the input clock frequency (data rate) by gating a part of the delay chain.

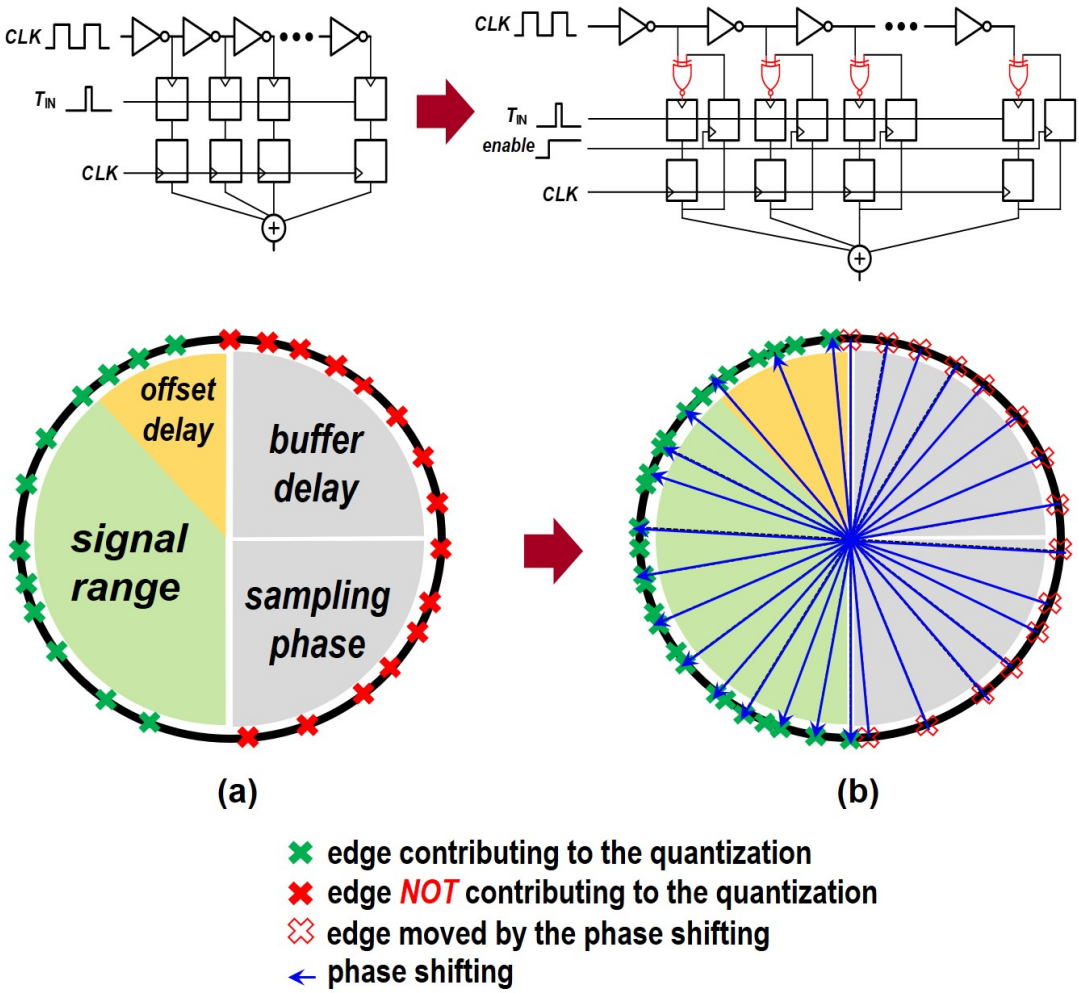


Figure 8.1: TDC delay chain and distribution of interpolated edges. (a) Without phase reversal. (b) With phase reversal.

Second, we can further improve the design productivity. One missing piece for this work to realize complete design automation is the use of custom unit cells, such as a switch cell, and a current unit cell, which required conventional analog design/verification flow. Recently, open-source analog custom cell generators, such as **ALIGN** [108][109], have been reported and we will explore how to integrate these tools into our flow and our designs.

Finally, we can help the effort to establish the open-source ecosystem of the mixed-signal design. We have used many proprietary tools and PDK in this work. However, by shifting our scope beyond the circuit design itself, we can try to migrate our flow to employ an open-source tool chain, such as **Yosys** [110] [111] for synthesis and **Qflow** [112], **OpenROAD** [113][114] for PnR, and an open-source PDK, such as **Skywater130**[115][116], in the future design. If this is successful, both to design and tool flow would be open to everyone.

Appendix A

Per-Channel FFE

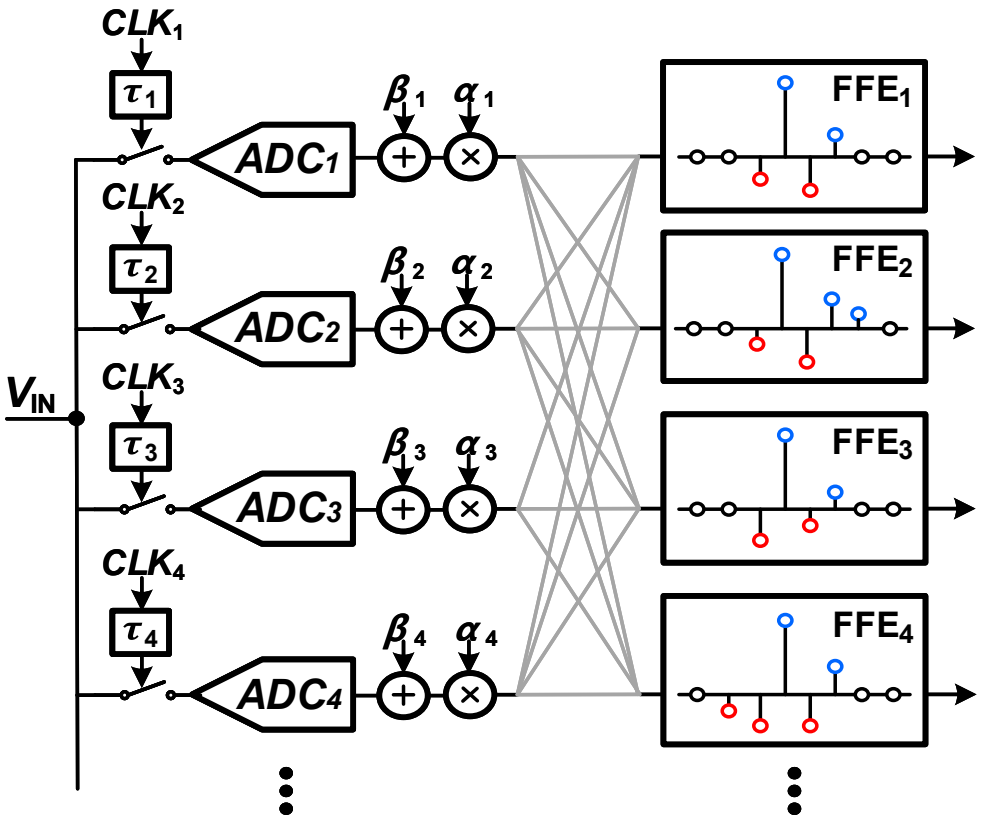


Figure A.1: Per-channel FFE coefficient adaptation for inter-channel mismatch compensation.

The effect of mismatches among the interleaved channels can be considered as a part of the physical channel. Because of the high data rate, linear FIR filters or feed-forward equalizers (FFEs) after the ADC have to be parallelized, and that allows us to customize those filters for each of the ADC slices with minimal overhead. Figure A.1 shows the calibration process for correcting the mismatches among the interleaved ADC slices [20][21]. The design of the FFE is split into sixteen filters, with the adaptation of each FFE coefficient based on samples received from a particular interleave. In the absence of inter-channel mismatches, the corresponding FFE coefficients are identical. In the presence of mismatches, including offset mismatch, gain mismatch, and clock skew, the FFE coefficients deviate from each other to compensate for the mismatches. An on-chip adaptation logic computes the difference between the coefficients and adjusts the inter-channel mismatches to minimize the error utilizing an LMS method. The algorithm assumes a statistically balanced input

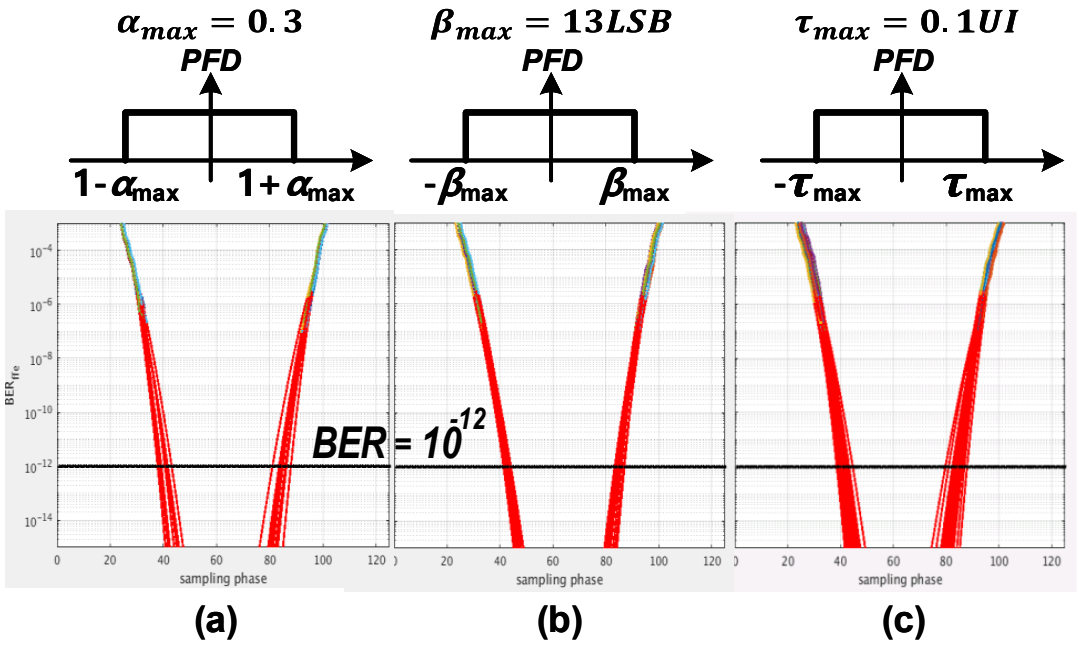


Figure A.2: BER bathtub curve estimation results of the per-channel FFE for (a) gain mismatch, (b) offset mismatch, and (c) sampling clock phase mismatch.

signal. The BER simulation framework discussed in Section 2.3 is used to estimate the performance of the link with the per-channel FFE. Gain mismatch (α), offset mismatch (β), and sampling clock skew (τ) are modeled as a uniform random variable bounded by $\pm 30\%$, $\pm 13LSB$, and $\pm 0.1UI$, respectively as shown in Figure A.2, where the red lines represent the extrapolated portion of the bathtub curves. 100 Monte Carlo iterations show that the per-channel FFE can successfully compensate for the inter-channel mismatches and maintain good performance.

Appendix B

Intermittent MLSD-Based Error Checker

Equalizers are commonly used to compensate channel attenuation by removing inter-symbol interference (ISI). However, the benefits of conventional feed-forward equalizers (FFE) and continuous-time linear equalizers (CTLE) are limited as these amplify noise and degrade the SNR as we discuss in Section 2.2.5. Decision feedback equalizers (DFE) do not amplify noise but discard the information stored in pre-cursors. DFE also results in error propagation and data dependency, which makes the design complex[1].

Maximum likelihood sequence detector (MLSD) is known as the optimal detector for an ISI channel that is subject to Gaussian noise [117]. MLSD makes decisions based on a sequence of received symbols and their ISI-induced correlations, rather than symbol-by-symbol decisions. Therefore, it suppresses failures due to error accumulation and propagation, which hurt conventional DFEs. In various applications requiring detection of digital sequences distorted in a band-limited communication channel, MLSD has been applied to provide a low error rate while meeting constrained latency and complexity requirements [118][119]. Although MLSD and its variants have been widely present in recent solutions for wireless communications

[1]Most of DFE in modern high-speed serial links usually employs speculative decision with loop unrolling to close timing.

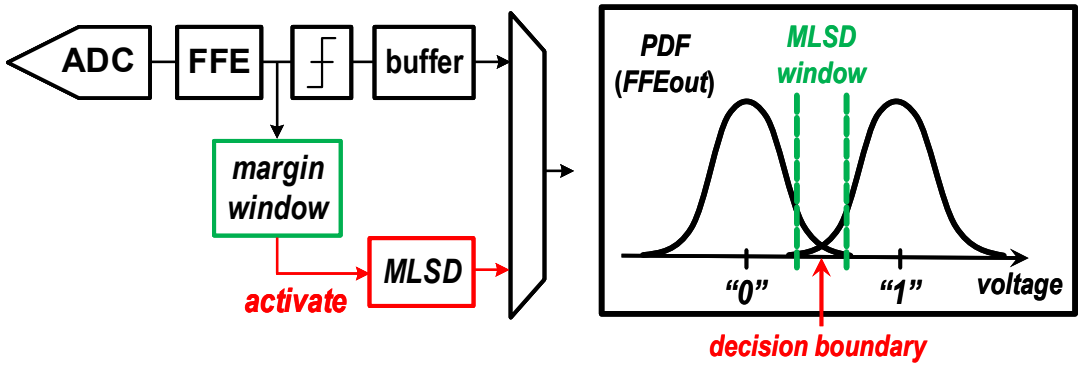


Figure B.1: Proposed intermittent MSLD based error checker.

[120][121][122], they have not commonly been reported for use in wireline applications since conventional multi-Gb/s MLSDs consume too much power (on the order of 100pJ/b [123][124][125]), and have very tight data dependencies for modern high-speed serial link standards.

We can employ both FFE and MLSD to achieve better power efficiency for a given target BER performance of a link. An FFE conjunction with a following slicer can be viewed as a maximum likelihood symbol estimator. The MLSD is activated only when the received symbol has a high chance to become an error. In other words, for a given decision margin widow, the output of the FFE is checked if it is close enough to the decision boundary and then, the MLSD makes decisions only for the symbols which are within the margin window as shown in Figure B.1. The final output of the receiver is selected by a mux with a digital buffer added after the slicer to count for the logic depth of the MDLL. For a target BER of the link (for example, 10^{-12}), employing the proposed intermittent MLSD based error checker can reduce the length of the FFE which is one of the key contributors of power consumption of the ADC-based serial links. Given an assumption that the error is still sparse enough with the reduced length FFE (for example, 10^{-6}), average activation rate of the MLSD is low enough so that power consumption of the MLSD is negligible.

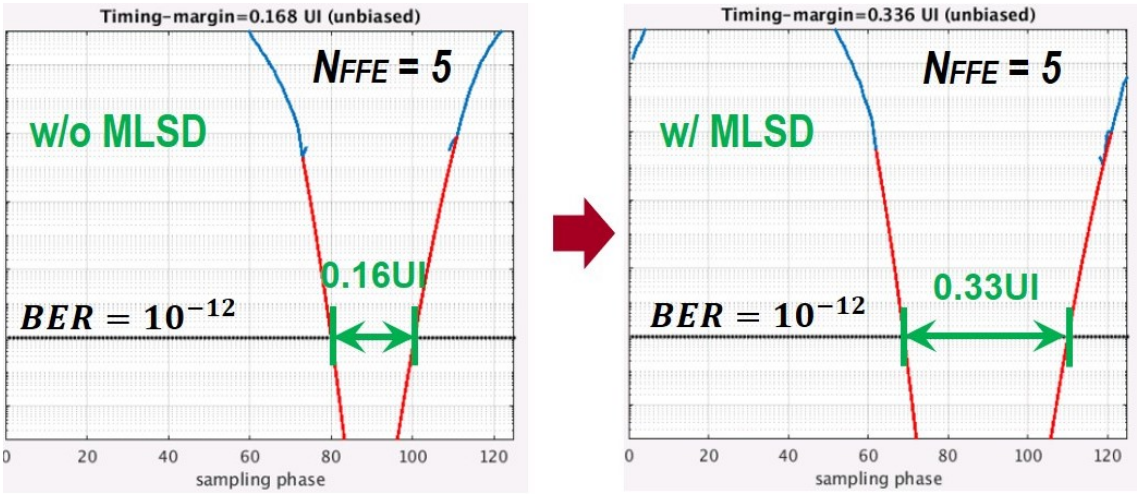


Figure B.2: BER bathtub curve estimation results with and without the proposed error checker when the number of FFE taps is reduced to 5.

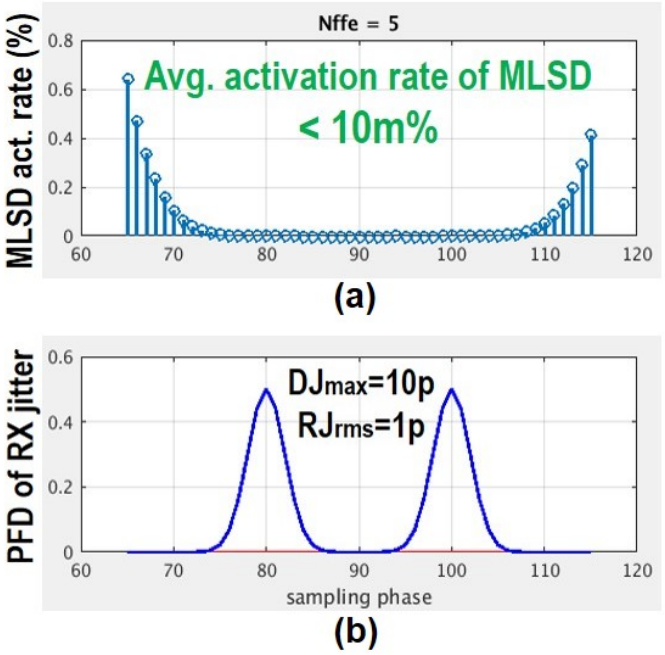


Figure B.3: (a) Activation rate of the MDLL with respect to the sampling phase and (b) RX jitter distribution.

Figure B.2 shows estimated BER where the red lines represent extrapolated portion of the bathtub curves. The horizontal margin of the link is compared with and without the proposed error checker when the length of FFE is reduced from 11 to 5. Figure B.3 shows the activation rate of the MLSD with respect to the RX sampling clock phase when the clock is assumed to have a dual-Dirac jitter distribution with 1ps of RJ and 10ps of DJ. While the average activation rate of the proposed intermittent MLSD is less than $10m\%$, it can successfully compensate for the performance degradation due to the reduction of FFE. For the case when the error sparsity condition does not hold, such as consecutive errors, more sophisticated algorithms may be required on top of this idea. We are working on upgrading the design to cover the corner cases.

Bibliography

[1] CISCO. VNI Global IP Traffic Forecast 2017-2022. [Online] https://davidellis.ca/wp-content/uploads/2019/05/cisco-vni-feb2019.pdf. (Accessed : 2020-12-01).

[2] Ethernet Alliance. Ethernet Speed Roadmap. [Online] https://ethernetalliance.org. (Accessed : 2020-12-01).

[3] PCI-SIG. PCIe Roadmap. [Online] https://pcisig.com. (Accessed : 2020-12-01).

[4] IPnest. IP Category Evolution 2017-2019. [Online] https://semiwiki.com/ip/arm/288187-interface-ip-category-to-overtake-cpu-ip-by-2025. (Accessed : 2020-12-01).

[5] OSI model. [Online] https://en.wikipedia.org/wiki/osi_model. (Accessed : 2020-12-01).

[6] Jafar Savoj, Kenny Hsieh, Parag Upadhyaya, Fu-Tai An, Ade Bekele, Stanley Chen, Xuewen Jiang, Kang Wei Lai, Chi Fung Poon, Aman Sewani, Didem Turker, Karthik Venna, Daniel Wu, Bruce Xu, Elad Alon, and Ken Chang. A Wide Common-Mode Fully-Adaptive Multi-Standard 12.5Gb/s Backplane Transceiver in 28nm CMOS. *IEEE Symposium on VLSI Circuits*, 2012.

[7] Brian S. Leibowitz, Jade Kizer, Haechang Lee, Fred Chen, Andrew Ho, Metha Jeeradit, Akash Bansall, Trey Greer, Simon Li, Ramin Farjad-Rad, William Stonecypher, Yohan Frans, Barry Daly, Fred Heaton, Bruno W., Garleppl,

Carl W. Werner, Nhat Nguyen, Vladimir Stojanovic, and Jared L. Zerbel. A 7.5Gb/s 10-Tap DFE Receiver with First Tap Partial Response, Spectrally Gated Adaptation, and 2nd-Order Data-Filtered CDR. *IEEE International Solid-State Circuits Conference*, 2007.

[8] Hyosup Won, Taehun Yoon, Jinho Han, Joon-Yeong Lee, Jong-Hyeok Yoon, Taeho Kim, Jeong-Sup Lee, Sangeun Lee, Kwangseok Han, Jinhee Lee, Jinho Park, and Hyeon-Min Bae. A 0.87W Transceiver IC for 100 Gigabit Ethernet in 40 nm CMOS. *IEEE Journal of Solid-State Circuits*, 50(2):399–413, 2015.

[9] E-Hung Chen, Ramy Yousry, and Chih-Kong Ken Yang. Power Optimized ADC-Based Serial Link Receiver. *IEEE Journal of Solid-State Circuits*, 47(4):938–951, 2012.

[10] Yohan Frans, Jaewook Shin, Lei Zhou, Parag Upadhyaya, Jay Im, Vassili Kireev, Mohamed Elzeftawi, Hiva Hedayati, Toan Pham, Santiago Asuncion, Chris Borrelli, Geoff Zhang, Hongtao Zhang, and Ken Chang. A 56-Gb/s PAM4 Wireline Transceiver Using a 32-Way Time-Interleaved SAR ADC in 16-nm FinFET. *IEEE Journal of Solid-State Circuits*, 52(4):1101–1110, 2017.

[11] Yoel Krupnik, Yevgeny Perelman, Itamar Levin, Yosi Sanhedrai, Roee Eitan, Ahmad Khairi, Yizhak Shifman, Yoni Landau, Udi Virobnik, Noam Dolev, Alon Meisler, and Ariel Cohen. 112-Gb/s PAM4 ADC-Based SERDES Receiver With Resonant AFE for Long-Reach Channels. *IEEE Journal of Solid-State Circuits*, 55(4):1077–1085, 2020.

[12] Rajan Lakshmi Narasimh and Naresh Shanbhag. Energy-efficient performance budgeting in FEC-based high-speed I/O links. *Conference on Electrical Performance of Electronic Packaging and Systems*, 2009.

[13] Hong Ahn, Amanda Dong, Alan Wong, Sai Lalith Chaitanya Ambatipudi, Xi Luo, and Geoff Zhang. 56 Gbps PAM4 SerDes Link Parameter Optimization for Improved Post-FEC BER. *Conference on Electrical Performance of Electronic Packaging and Systems*, 2019.

[14] H. Johnson and M. Graham. *High-Speed Digital Design: A Handbook of Black Magic*. Prentice Hall, 1993.

[15] Sam Palermo. ECEN 720: High-Speed Links Circuits and Systems. [Online] `https://people.engr.tamu.edu/spalermo/ecen720.html`. (Accessed : 2020-12-01).

[16] B. Widrow. A Study of Rough Amplitude Quantization by Means of Nyquist Sampling Theory. *IRE Transactions on Circuit Theory*, 3(4):266–276, 1956.

[17] B. Widrow. Statistical analysis of amplitude-quantized sampled-data systems. *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, 79(6):555–568, 1961.

[18] Naoki Kurosawa, Haruo Kobayashi, Kaoru Maruyama, Hidetake Sugawara, and Kensuke Kobayashi. Explicit Analysis of Channel Mismatch Effects in Time-Interleaved ADC Systems. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I*, 48(3):261–271, 2001.

[19] Behzad Razavi. Design Considerations for Interleaved ADCs. *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, 48(8):1806–1817, 2013.

[20] Tsung-Heng Tsai, Paul J. Hurst, and Stephen H. Lewis. Correction of Mismatches in a Time-Interleaved Analog-to-Digital Converter in an Adaptively Equalized Digital Communication Receiver. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(2):307–319, 2009.

[21] Aida Varzaghani, Athos Kasapi, Dimitri N. Loizos, Song-Hee Paik, Shwetabh Verma, Sotirios Zogopoulos, and Stefanos Sidiropoulos. A 10.3-GS/s, 6-Bit Flash ADC for 10G Ethernet Applications. *IEEE Journal of Solid-State Circuits*, 48(12):3038–3048, 2013.

[22] N. Da Dalt and A. Sheikholeslami. *Understanding Jitter and Phase Noise*. Cambridge University Press, 2018.

[23] David R. Stauffer, Kent Dramstad, Amanullah Mohammad, Michael A. Sorna, Jeanne T. Mechler, Clarence R. Ogilvie, and James D. Rockrohr. *High Speed Serdes Devicesand Applications*. Spinger, 2008.

[24] Agilent Technology. Jitter analysis: The dualDirac model, RJ/DJ, and Q-scale. [Online] `https://www.keysight.com/upload/cmc_upload/All/dualdirac1.pdf`. (Accessed : 2020-12-01).

[25] Nicola Da Dalt, Moritz Harteneck, Christoph Sandner, and Andreas Wiesbauer. On the Jitter Requirements of the Sampling Clock for Analog-to-Digital Converters. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I*, 49(9):1354–1360, 2002.

[26] V. Stojanovic, A. Amirkhany, and M.A. Horowitz. Optimal linear precoding with theoretical and practical data rates in high-speed serial-link backplane communication. *IEEE International Conference on Communications*, 2004.

[27] Dan Oh, Jihong Ren, and Sam Chang. Hybrid Statistical Link Simulation Technique. *IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY*, 1(5):772–783, 2011.

[28] A. Klein, G.K. Kaleh, and P.W. Baier. Zero forcing and minimum mean-square-error equalization for multi-user detection in code-division multiple-access channels. *IEEE Transactions on Vehicular Technology*, 45(2):276–287, 1996.

[29] B. Widrow, J.M. McCool, M.G. Larimore, and C.R. Johnson. Stationary and nonstationary learning characteristics of the LMS adaptive filter. *Proceedings of the IEEE*, 64(8):1151–1162, 1976.

[30] J. Poulton, R. Palmer, A. M. Fuller, T. Greer, J. Eyles, W. J. Dally, and M. Horowitz. A 14-mV 6.25-Gb/s transceiver in 90-nm CMOS. *IEEE Journal of Solid-State Circuits*, 42(12):2745–2757, 2007.

[31] D. Oh, W. Kim, J.-H. Kim, J. Wilson, R. Schmitt, C. Yuan, L. Luo, J. Kizer, J. Eble, and F. Ware. Study of signal and power integrity challenges in high-speed memory I/O designs using single-ended signaling schemes. *DesignCon*, 2008.

[32] B. K. Casper, M. Haycock, and R. Mooney. An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes. *IEEE Symposium on VLSI Circuits*, 2002.

[33] Pavan Kumar Hanumolu, Bryan Casper, Randy Mooney, Gu-Yeon Wei, and Un-Ku Moon. Analysis of PLL Clock Jitter in High-Speed Serial Links. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II*, 50(11):879–886, 2003.

[34] Shiva Kiran, Ayman Shafik, Ehsan Zhian Tabasy, Shengchang Cai, Keytaek Lee, Sebastian Hoyos, and Samuel Palermo. Modeling of ADC-Based Serial Link Receivers With Embedded and Digital Equalization. *IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY*, 9(3):536–548, 2019.

[35] Anthony Sanders, Mike Resso, and John Ambrosia. Channel Compliance Testing Utilizing Novel Statistical Eye Methodology. *DesignCon*, 2004.

[36] Kyung Suk Oh, Frank Lambrecht, Sam Chang, Qi Lin, Jihong Ren, Chuck Yuan, Jared Zerbe, and Vladimir Stojanovic. Accurate System Voltage and Timing Margin Simulation in High-Speed I/O System Designs. *IEEE TRANSACTIONS ON ADVANCED PACKAGING*, 31(4):722–730, 2008.

[37] Douglas Burns, John Madsen, Todd Westerhoff, and Walter Katz. UNDERSTANDING IBIS-AMI SIMULATIONS,. *DesignCon*, 2015.

[38] K. Inagaki, Y. Hirata, F. Takahata, A. Ogawa, and K. Niwa. International Connection of Plesiochronous Networks via TDMA Satellite Link. *IEEE Journal on Selected Areas in Communications*, 1983.

[39] Jingwei Wei, Xuan Li, Lei Sun, and Dongmei Li. A 63.2μW 11-Bit Column Parallel Single-Slope ADC with Power Supply Noise Suppression for CMOS Image Sensors. *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020.

[40] Hyeon-June Kim. 11-bit Column-Parallel Single-Slope ADC With First-Step Half-Reference Ramping Scheme for High-Speed CMOS Image Sensors. *IEEE JOURNAL OF SOLID-STATE CIRCUITS (Early Access)*, 2021.

[41] Ji-Yong Jeong, Junbo Shim, Seong-Kwan Hong, and Oh-Kyong Kwon. A High Speed and Energy Efficient Multi Bit Cyclic ADC Using Single Slope Quantizer for CMOS Image Sensors. *IEEE Transactions on Circuits and Systems II (Early Access)*, 2021.

[42] Seyed Danesh, Jed Hurwitz, Keith Findlater, David Renshaw, and Robert Henderson. A reconfigurable 1GSps to 250MSps, 7-bit to 9-bit highly time-interleaved counter ADC in 0.13µm CMOS. *Symposium on VLSI Circuits*, 2011.

[43] Valentin Abramzon. *Analog-to-Digital Converters for High-Speed Links*. PhD thesis, Stanford University, 2008.

[44] Deyan Levski, Martin Wany, and Bhaskar Choubey. A 1μs Ramp Time 12-bit Column-Parallel Flash TDC-Interpolated Single-Slope ADC With Digital Delay-Element Calibration. *TRANSACTIONS ON CIRCUITS AND SYSTEMS–I*, 66(1):54–67, 2019.

[45] Sung-Jin Kim, Wooseok Kim, Minyoung Song, Jihyun Kim, Taeik Kim, and Hojin Park. A 0.6V 1.17ps PVT-tolerant and synthesizable time-to-digital converter using stochastic phase interpolation with 16$\times$ spatial redundancy in 14nm FinFET technology. *IEEE International Solid-State Circuits Conference*, 2015.

[46] Vanessa Hung-Chu Chen and Lawrence Pileggi. A 69.5mW 20GS/s 6b Time-Interleaved ADC with Embedded Time-to-Digital Calibration in 32nm CMOS SOI. *IEEE International New Circuits and Systems Conference*, 2014.

[47] Yun-Shiang Shu. A 6b 3GS/s 11mW Fully Dynamic Flash ADC in 40nm CMOS with Reduced Number of Comparators. *IEEE Symposium on VLSI Circuits*, 2012.

[48] Rajan Narasimha, Minwei Lu, Naresh R. Shanbhag, and Andrew C. Singer. BER-Optimal Analog-to-Digital Converters for Communication Links. *IEEE Transactions on Signal Processing*, 60(7):3683–3691, 2012.

[49] Hung-Yen Tai, Yao-Sheng Hu, Hung-Wei Chen, and Hsin-Shu Chen. A 0.85fJ/conversion-step 10b 200kS/s Subranging SAR ADC in 40nm CMOS. *IEEE International Solid-State Circuits Conference*, 2014.

[50] Kostas Doris, Erwin Janssen, Claudio Nani, Athon Zanikopoulos, and Gerard van der Weide. A 480 mW 2.6 GS/s 10b Time-Interleaved ADC With 48.5 dB SNDR up to Nyquist in 65 nm CMOS. *IEEE Journal of Solid-State Circuits*, 46(12):2821–2833, 2011.

[51] Lukas Kull, Thomas Toifl, Martin Schmatz, Pier Andrea Francese, Christian Menolfi, Matthias Braendli, Marcel Kossel, Thomas Morf, Toke Meyer Andersen1, and Yusuf Leblebici. A 90GS/s 8b 667mW 64× Interleaved SAR ADC in 32nm Digital SOI CMOS. *IEEE International Solid-State Circuits Conference*, 2014.

[52] Ahmed M. A. Ali, Huseyin Dinc, Paritosh Bhoraskar, Chris Dillon, Scott Puckett, Bryce Gray, Carroll Speir, Jonathan Lanford, Janet Brunsilius, Peter R. Derounian, Brad Jeffries, Ushma Mehta, Matthew McShea, and Russell Stop. A 14 Bit 1 GS/s RF Sampling Pipelined ADC With Background Calibration. *IEEE Journal of Solid-State Circuits*, 49(12):2857–2867, 2014.

[53] Peter R. Kinget. Device Mismatch and Tradeoffs in the Design of Analog Circuits. *IEEE Journal of Solid-State Circuits*, 40(6):1212–1224, 2005.

[54] Timmy Sundstrom and Atila Alvandpour. Utilizing Process Variations for Reference Generation in a Flash ADC. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 56(5):364–368, 2009.

[55] Skyler Weaver, Benjamin Hershberg, Peter Kurahashi, Daniel Knierim, and Un-Ku Moon. Stochastic Flash Analog-to-Digital Conversion. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(11):2825–2833, 2010.

[56] Min-Ki Jeon, Won-Jun Yoo, Chan-Gyu Kim, and Changsik Yoo. A Stochastic Flash Analog-to-Digital Converter Linearized by Reference Swapping. *IEEE Access*, 5:23046–23051, 2017.

[57] Ahmed Fahmy, Jun Liu, Taewook Kim, and Nima Maghari. An All-Digital Scalable and Reconfigurable Wide-Input Range Stochastic ADC Using Only Standard Cells. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(8):731–735, 2015.

[58] C. Paulus, H.-M. Bluthgen, M. Low, E. Sicheneder, N. Bruls, A. Courtois, M. Tiebout, and R. Thewes. A 4GS/s 6b flash ADC in 0.13um CMOS. *Symposium on VLSI Circuits*, 2004.

[59] Parit Kanjanavirojkul, Nguyen Ngoc Mai-Khanh, Ikeno Rimon, Takahiro J. Yamaguchi, Tetsuya Iizuka, and Kunihiro Asada. A Consideration on LUT Linearization of Stochastic ADC in Sub-Ranging Architecture. *IEEE Custom Integrated Circuits Conference*, 2018.

[60] Chi-Wei Fan and Jieh-Tsorng Wu. ADC clock jitter measurement and correction using a stochastic TDC. *IEEE Asia Pacific Conference on Circuits and Systems*, 2010.

[61] Satoshi Ito, Shigeyuki Nishimura, Haruo Kobayashi, Satoshi Uemori, Yohei Tan, Nobukazu Takai, Takahiro J. Yamaguchi, and Kiichi Niitsu. Stochastic TDC architecture with self-calibration. *IEEE Asia Pacific Conference on Circuits and Systems*, 2010.

[62] Skyler Weaver, Benjamin Hershberg, and Un-Ku Moon. Digitally Synthesized Stochastic Flash ADC Using Only Standard Digital Cells. *IEEE Transactions on Circuits and Systems I: Regulay Papers*, 61(1):84–91, 2014.

[63] Jun-Eun Park, Young-Ha Hwang, and Deog-Kyoon Jeong. A 0.5-V Fully Synthesizable SAR ADC for On-Chip Distributed Waveform Monitors. *IEEE Access*, 7:63686–63697, 2019.

[64] Naoki Ojima, Zule Xu, and Tetsuya Iizuka. A 0.0053-mm^2 6-bit Fully-Standard-Cell-Based Synthesizable SAR ADC in 65 nm CMOS. *IEEE International New Circuits and Systems Conference*, 2019.

[65] Jeahan Park, Cheolmin Ahn, Jeahyeong Hong, and Jae-Yoon Sim. A Picosecond-Resolution Digitally-Controlled Timing Generator With One-Clock-Latency at Arbitrary Instantaneous Input. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(9):1544–1548, 2020.

[66] Yoontaek Lee, Taewook Kang, and Jaeha Kim. A 9–11 bits phase-interpolating digital pulse-width modulator with 1000X frequency range. *IEEE Energy Conversion Congress and Exposition*, 2014.

[67] Behrooz Abiri, Ravi Shivnaraine, Ali Sheikholeslami, Hirotaka Tamura, and Masaya Kibune. A 1-to-6Gb/s Phase-Interpolator-Based Burst-Mode CDR in 65nm CMOS. *IEEE International Solid-State Circuits Conference*, 2011.

[68] Dong-Woo Jee, Yunjae Suh, Byungsub Kim, Hone-June Park, and Jae-Yoon Sim. A FIR-Embedded Phase Interpolator Based Noise Filtering for Wide-Bandwidth Fractional-N PLL. *IEEE Journal of Solid-State Circuits*, 48(11):2795–2804, 2013.

[69] M. Talegaonkar, T. Anand, A. Elkholy, A. Elshazly, R. K. Nandwana, S. Saxena, B. Young, W. Choi, and P. K. Hanumolu. A 4.4-5.4GHz Digital Fractional-N PLL Using $\Delta\Sigma$ Frequency-to-Digital Converter. *IEEE Symposium on VLSI Circuits*, 2014.

[70] Saman Saeedi and Azita Emami. An 8GHz First-order Frequency Synthesizer based on Phase Interpolation and Quadrature Frequency Detection in 65nm CMOS. *IEEE Custom Integrated Circuits Conference*, 2014.

[71] Sanquan Song, John Poulton†, Xi Chen, Brian Zimmer, Stephen G. Tell†, Walker J. Turner, Sudhir S. Kudva, Nikola Nedovic, John Wilson†, C. Thomas Gray, and William J. Dally. A 2-to-20 GHz Multi-Phase Clock Generator with Phase Interpolators Using Injection-Locked Oscillation Buffers for High-Speed IOs in 16nm FinFET. *IEEE Custom Integrated Circuits Conference*, 2019.

[72] Masanobu Tsuji. A 762-μW 16.3-ps Resolution Digital Pulse Width Modulator Using Zooming Phase-Interpolator. *IEEE Asian Solid-State Circuits Conference*, 2017.

[73] Aravind Tharayil Narayanan, Makihiko Katsuragi, Kento Kimura, Satoshi Kondo, Korkut Kaan Tokgoz, Kengo Nakata, Wei Deng, Kenichi Okada, and Akira Matsuzawa. A Fractional-N Sub-Sampling PLL using a Pipelined Phase-Interpolator With an FoM of –250 dB. *IEEE Journal of Solid-State Circuits*, 51(7):1630–1640, 2016.

[74] Anders Jakobsson, Adriana Serban, and Shaofang Gong. A Low-Noise RC-Based Phase Interpolator in 16-nm CMOS. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(1):1–5, 2019.

[75] Guoying Wu, Deping Huang, Jingxiao Li, Ping Gui, Tianwei Liu, Shita Guo, Rui Wang, Yanli Fan, Sudipto Chakraborty, and Mark Morgan. A 1–16 Gb/s All-Digital Clock and Data Recovery With a Wideband High-Linearity Phase Interpolator. *IEEE Transactions on Very Large Scale Integration Systems*, 24(7):2511–2520, 2016.

[76] Parag Upadhyaya, Ade Bekele, Didem Turkur Melek, Haibing Zhao, Jay Im, Junho Cho, Kee Hian Tan, Scott McLeod, Stanley Chen, Wenfeng Zhang, Yohan Frans, and Ken Chang. A Fully-Adaptive Wideband 0.5-32.75Gb/s FPGA Transceiver in 16nm FinFET CMOS Technology. *IEEE Symposium on VLSI Circuits*, 2016.

[77] Shun-Chi Chang and Shen-Iuan Liu. A 5-Gb/s Adaptive Digital CDR Circuit with SSC Capability and Enhanced High-Frequency Jitter Tolerance. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(1):161–165, 2021.

[78] Young-Ho Choi, Byungsub Kim, Jae-Yoon Sim, and Hong-June Park. A Phase-Interpolator-Based Fractional Counter for All-Digital Fractional-N Phase-Locked Loop. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 64(3):249–253, 2017.

[79] Enrico Monaco, Gabriele Anzalone, Guido Albasini, Simone Erba, Matteo Bassi, and Andrea Mazzanti. A 2–11 GHz 7-Bit High-Linearity Phase Rotator Based on Wideband Injection-Locking Multi-Phase Generation for High-Speed Serial Links in 28-nm CMOS FDSOI. *IEEE Journal of Solid-State Circuits*, 52(7):1739–1752, 2017.

[80] Joshua Liang, Ali Sheikholeslami, Hirotaka Tamura, and Hisakatsu Yamaguchi. On-Chip Jitter Measurement Using Jitter Injection in a 28 Gb/s PI-Based CDR. *IEEE Journal of Solid-State Circuits*, 53(3):750–761, 2018.

[81] Stanley Chen, Lei Zhou, Ian Zhuang, Jay Im, Didem Melek, Jinyung Namkoong, Mayank Raj, Jaewook Shin, Yohan Frans, and Ken Chang. A 4-to-16GHz Inverter-Based Injection-Locked Quadrature Clock Generator with Phase Interpolators for Multi-Standard I/Os in 7nm FinFET. *IEEE International Solid-State Circuits Conference*, 2018.

[82] Stefanos Sidiropoulos. High Performance Inter-chip Signaling. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(1):1–5, 2019.

[83] Daniel Weinlader. *Precision CMOS Receivers for VLSI Testing Applications*. PhD thesis, Stanford University, 11 2001.

[84] Archit Joshi and Mukul Sarkar. Nonlinearity Estimation for Compensation of Phase Interpolator in Bang–Bang CDRs. *IEEE Transactions on Very Large Scale Integration Systems*, 25(1):388–392, 2017.

[85] Archit Joshi and Mukul Sarkar. An Odd Phase CDR With Phase Interpolator Trimming. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(1):31–35, 2019.

[86] Rainer Kreienkamp, Ulrich Langmann, Christoph Zimmermann, Takuma Aoyama, and Hubert Siedhoff. A 10-Gb/s CMOS Clock and Data Recovery Circuit With an Analog Phase Interpolator. *IEEE Journal of Solid-State Circuits*, 40(3):736–743, 2005.

[87] Yang-Hang Fan, Ankur Kumar, Takayuki Iwai, Ashkan Roshan-Zamir, Shengchang Cai, Bo Sun, and Samuel Palermo. A 32-Gb/s Simultaneous Bidirectional Source-Synchronous Transceiver With Adaptive Echo Cancellation Techniques. *IEEE Journal of Solid-State Circuits*, 55(2):439–451, 2020.

[88] Mozhgan Mansuri, Bryan Casper, and Frank O'Mahony. An On-Die All-Digital Delay Measur ement Circuit with 250fs Accuracy. *IEEE Symposium on VLSI Circuits*, 2012.

[89] Giuseppe Li Puma and Christophe Carbonne. Mitigation of Oscillator Pulling in SoCs. *IEEE Journal of Solid-State Circuits*, 51(2):348–356, 2016.

[90] Behzad Razavi. The Ring Oscillator. *IEEE Solid-State Circuits Magazine*, 11(4):10–13, 2019.

[91] Jun Yin, Pui-In Mak, Franco Maloberti, and Rui P. Martins. A Time-Interleaved Ring-VCO with Reduced $1/f^3$ Phase Noise Corner, Extended Tuning Range and Inherent Divided Output. *IEEE Journal of Solid-State Circuits*, 49(2):384–391, 2014.

[92] P. Kinget. Integrated GHz voltage controlled oscillators. *Analog circuit Design*, (Kluwer Academic):353–381, 2014.

[93] Shanliang Gan, Yuan Wang, Jian Cao, Song Jia, and Xing Zhang. A 4.9–6.9 GHz LC VCO with low-phase-noise. *IEEE International Conference of Electron Devices and Solid-state Circuits*, 2013.

[94] Somnath Kundu, Bongjin Kim, and Chris H. Kim. A 0.2–1.45-GHz Subsampling Fractional- N Digital MDLL With Zero-Offset Aperture PD-Based Spur Cancellation and In Situ Static Phase Offset Detection. *IEEE Journal of Solid-State Circuits*, 52(3):799–811, 2017.

[95] F. Gardner. Charge-Pump Phase-Lock Loops. *IEEE Transactions on Communications*, 28(11):1849–1858, 1980.

[96] Giovanni Marzin, Andrea Fenaroli, Giovanni Marucci, Salvatore Levantino, Carlo Samori, and Andrea L. Lacaita. A spur cancellation technique for MDLL-based frequency synthesizers. *IEEE International Symposium on Circuits and Systems*, 2013.

[97] Belal M. Helal, Matthew Z. Straayer, Gu-Yeon Wei, and Michael H. Perrott. A Highly Digital MDLL-Based Clock Multiplier That Leverages a Self-Scrambling Time-to-Digital Converter to Achieve Sub-picosecond Jitter Performance. *IEEE Journal of Solid-State Circuits*, 43(4):855–863, 2008.

[98] Hyunik Kim, Yongjo Kim, Taeik Kim, Hojin Park, and SeongHwan Cho. A 2.4GHz 1.5mW digital MDLL using pulse-width comparator and double injection technique in 28nm CMOS. *IEEE International Solid-State Circuits Conference*, 2016.

[99] Wei Deng, Dongsheng Yang, Tomohiro Ueno, Teerachot Siriburanon, Satoshi Kondo, Kenichi Okada, and Akira Matsuzawa. A Fully Synthesizable All-Digital PLL With Interpolative Phase Coupled Oscillator, Current-Output DAC, and Fine-Resolution Digital Varactor Using Gated Edge Injection Technique. *IEEE Journal of Solid-State Circuits*, 50(1):68–80, 2015.

[100] Synopsys. AMS verification. [Online] `https://www.synopsys.com/verification/ams-verification.html`. (Accessed : 2020-12-01).

[101] Stanford University. DaVE project. [Online] `https://github.com/StanfordVLSI/DaVE`. (Accessed : 2020-12-01).

[102] Stanford University. DragonPHY project. [Online] `https://github.com/StanfordVLSI/dragonphy`. (Accessed : 2020-12-01).

[103] Steven Herbst, Byong Chan Lim, and Mark Horowitz. Fast FPGA Emulation of Analog Dynamics in Digitally-Driven Systems. *IEEE/ACM International Conference on Computer-Aided Design*, 2018.

[104] Keysight. Jitter Fundamentals:Jitter Tolerance Testing with Agilent 81250 ParBERT. [Online] `http://literature.cdn.keysight.com/litweb/pdf/5989-0223EN.pdf`. (Accessed : 2020-12-01).

[105] ByongChan Lim and Mark Horowitz. Error Control and Limit Cycle Elimination in Event-Driven Piecewise Linear Analog Functional Models. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(1):23–32, 2016.

[106] ByongChan Lim, James Mao, Ji-Eun Jang, and Jaeha Kim Mark Horowitz. Digital Analog Design: Enabling Mixed-Signal System Validation. *IEEE Magazine*, pages 44–52, 2015.

[107] Matteo Pisati, Fernando De Bernardinis, Paolo Pascale, Claudio Nani, Marco Sosio, Enrico Pozzati, Nicola Ghittori, Federico Magni, Marco Garampazzi, Giacomino Bollati, Antonio Milani, Alberto Minuti, Fabio Giunco, Paola Uggetti, Ivan Fabiano, Nicola Codega, Alessandro Bosi, Nicola Carta, Demetrio Pellicone, Giorgio Spelgatti, Massimo Cutrupi, Andrea Rossini, Roberto Massolini, Giovanni Cesura, and Ivan Bietti. A Sub-250mW 1-to-56Gb/s Continuous-Range PAM-4 42.5dB IL ADC/DAC-Based Transceiver in 7nm FinFET. *IEEE International Solid-State Circuits Conference*, 2019.

[108] Tonmoy Dhar, Kishor Kunal, Yaguang Li, Meghna Madhusudan, Jitesh Poojary, Arvind K. Sharma, Wenbin Xu, Steven M. Burns, Ramesh Harjani, Jiang Hu, Desmond A. Kirkpatrick, Parijat Mukherjee, Sachin S. Sapatnekar, and Soner Yaldiz. ALIGN: A System for Automating Analog Layout. *IEEE Design Test*, 2020.

[109] ALIGN project. [Online] `https://github.com/ALIGN-analoglayout/ALIGN-public`. (Accessed : 2020-12-01).

[110] Yosys Open SYnthesis Suite. [Online] `http://www.clifford.at/yosys`. (Accessed : 2020-12-01).

[111] YosysHQ. [Online] `https://github.com/YosysHQ/yosys`. (Accessed : 2020-12-01).

[112] Qflow 1.3: An Open-Source Digital Synthesis Flow. [Online] `http://opencircuitdesign.com/qflow`. (Accessed : 2020-12-01).

[113] OpenROAD. [Online] `https://theopenroadproject.org`. (Accessed : 2020-12-01).

[114] The OpenROAD Project. [Online] `https://github.com/The-OpenROAD-Project`. (Accessed : 2020-12-01).

[115] Google. skywater open source PDK. [Online] `https://www.skywatertechnology.com/design-enablement`. (Accessed : 2020-12-01).

[116] Google. skywater. [Online] `https://github.com/google/skywater-pdk`. (Accessed : 2020-12-01).

[117] G.D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[118] Hazar Yueksel, Matthias Braendli, Andreas Burg, Giovanni Cherubini, Roy D. Cideciyan, Pier Andrea Francese, Simeon Furrer, Marcel Kossel, Lukas Kull, Danny Luu, Christian Menolfi, Thomas Morf, and Thomas Toifl. A 4.1 pJ/b 25.6 Gb/s 4-PAM reduced-state sliding-block Viterbi detector in 14 nm CMOS. *European Solid-State Circuits Conference*, 2016.

[119] Hazar Yueksel, Giovanni Cherubini, Roy D. Cideciyan, Andreas Burg, and Thomas Toifl. Design considerations on sliding-block viterbi detectors for high-speed data transmission. *International Conference on Signal Processing and Communication Systems*, 2016.

[120] Sha Hu, Harald Kröll, Qiuting Huang, and Fredrik Rusek. Optimal Channel Shortener Design for Reduced- State Soft-Output Viterbi Equalizer in Single-Carrier Systems. *IEEE Transactions on Communications*, 65(6):2568–2582, 2017.

[121] Hao Peng, Rongke Liu, Yi Hou, and Ling Zhao. A Gb/s parallel block-based Viterbi decoder for convolutional codes on GPU. *International Conference on Wireless Communications Signal Processing*, 2016.

[122] Yao Wang and Vijaya Kumar. Improved Multitrack Detection With Hybrid 2-D Equalizer and Modified Viterbi Detector. *IEEE Transactions on Magnetics*, 53(10), 2017.

[123] Peter J. Black and Teresa H.-Y. Meng. A 1Gb/s, 4-State, Sliding Block Viterbi Decoder. *Symposium on VLSI Circuits*, 1993.

[124] H.-M. Bae, J.B. Ashbrook, J. Park, N.R. Shanbhag, A.C. Singer, and S. Chopra. An MLSE Receiver for Electronic Dispersion Compensation of OC-192 Fiber Links. *IEEE Journal of Solid-State Circuits*, 41(11):2541–2553, 2006.

[125] Thomas Veigel, Thomas Alpert, Felix Lang, Markus Grözing, and Manfred Berroth. A Viterbi Equalizer Chip for 40 Gb/s optical communication links. *European Microwave Integrated Circuit Conference*, 2013.