

On-Chip Jitter Measurement Using Jitter Injection in a 28 Gb/s PI-Based CDR

Joshua Liang^{ID}, Ali Sheikholeslami, *Senior Member, IEEE*,
Hirotaka Tamura, *Fellow, IEEE*, and Hisakatsu Yamaguchi

Abstract—We present a technique to measure random jitter in a phase interpolator (PI)-based clock and data recovery (CDR) circuit by injecting a controlled amount of square-wave jitter into its edge clock and monitoring its effect on the autocorrelation function of the CDR’s bang-bang phase detector output. Jitter is injected by adjusting the code of the edge PI while the autocorrelation function is measured by on-chip counters. Since the injected jitter only affects the edge clock, the CDR remains operational during jitter injection. Using this technique, the rms relative jitter between the clock and data at the CDR input can be estimated with sub-picosecond accuracy as demonstrated in measurements of a 28 Gb/s half-rate digital CDR fabricated in 28 nm CMOS.

Index Terms—Bang-bang phase detector (BB-PD), built-in self-test (BIST), clock and data recovery (CDR), jitter measurement, phase interpolator (PI).

I. INTRODUCTION

AS DATA rates increase and the unit interval (UI) shrinks, it becomes increasingly important to minimize the jitter of clock and data recovery (CDR) circuits. This can be difficult, however, as jitter in a CDR is contributed by numerous sources such as the transmitter, intersymbol interference (ISI), as well as various receiver components such as the phase detector (PD), phase interpolator (PI), and phase-locked loop (PLL). Jitter is also contributed by secondary sources such as power supply noise. As a result, it is difficult to accurately predict and budget for each jitter contribution when designing a CDR. Furthermore, even if a design is optimized, process voltage and temperature variation and ageing can still degrade performance by pushing the design away from its optimal operating point.

On-chip jitter measurement can help address these challenges, allowing a CDR to be adaptively tuned based on the observed jitter. For example, a jitter measurement circuit

Manuscript received July 19, 2017; revised October 12, 2017; accepted November 8, 2017. This paper was approved by Guest Editor Samuel Palermo. This work was supported in part by NSERC. (*Corresponding author: Joshua Liang.*)

J. Liang was with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada. He is now with Huawei Canada, Markham, ON L3R 5B4, Canada (e-mail: joshua.liang@mail.utoronto.ca).

A. Sheikholeslami is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada.

H. Tamura and H. Yamaguchi are with the Fujitsu Laboratories Limited, Kawasaki 211-8588, Japan.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSSC.2017.2776307

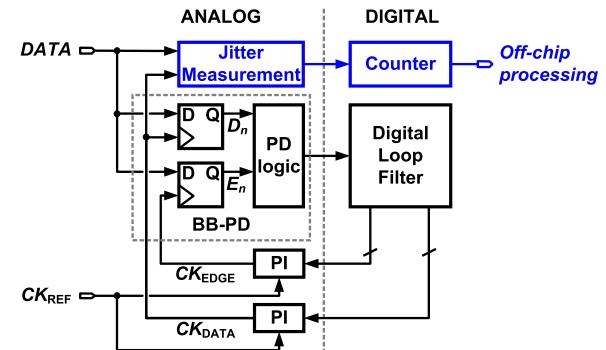


Fig. 1. On-chip jitter measurement using a dedicated jitter measurement circuit in a CDR.

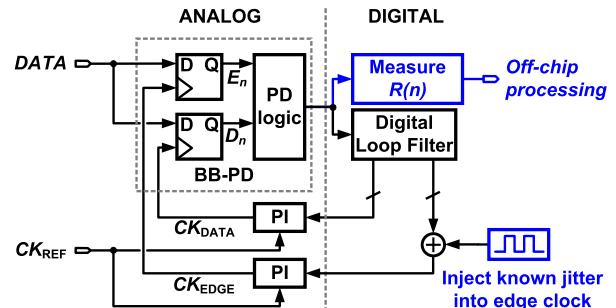


Fig. 2. Proposed technique for on-chip jitter measurement.

can be added to the CDR as shown in Fig. 1, and used to measure the relative jitter between the CDR’s input data and recovered clock. Gradient descent [1] or other optimization techniques can then be used to minimize the relative jitter, and help achieve the lowest bit error rate (BER). Unfortunately, adding jitter measurement circuits is generally costly in terms of power, area, and the extra circuits that need to be designed.

In this paper, we present a technique to measure the rms value of the relative jitter in PI-based CDRs with reduced circuit overhead. By injecting a known amount of square-wave jitter using the CDR’s PI and monitoring the autocorrelation function of the CDR’s bang-bang PD (BB-PD) output [2], the rms relative jitter of the CDR is estimated with sub-picosecond accuracy without requiring any dedicated analog circuits such as eye monitors. The scheme, shown in Fig. 2, is demonstrated in a 28 Gb/s half-rate CDR implemented in 28 nm CMOS. Measurements results also show how this measurement scheme can be used to select the optimal loop filter and equalizer settings of the CDR.

The remainder of this paper is organized as follows. First, Section II provides background on the most common on-chip

jitter measurement techniques. Section III then describes the proposed technique, before presenting the implemented test chip and measurement results in Sections IV and V.

II. BACKGROUND

To be of practical use in high data rate applications, an on-chip jitter measurement circuit must have sufficiently fine measurement resolution. At 25 Gb/s and beyond, the random jitter (RJ) of reference clocks, usually generated by high-performance PLLs, is tightly constrained to be hundreds of femtoseconds or less [3], [4]. This is because such RJ is unbounded and scales with the target BER. Optimizing these PLLs using on-chip jitter measurement would require femtosecond resolution, which may not be practical with existing techniques. PLLs can instead be optimized using other methods [5]. On the other hand, deterministic jitter sources in the CDR such as dithering, PI nonlinearity, ISI, and power supply-induced jitter are bounded, and can have much higher rms values, pushing the total rms jitter of CDRs beyond 1 ps [6], [7]. A measurement technique with sub-picosecond resolution can therefore be valuable in optimizing a CDR's performance.

Jitter can be measured on-chip in a CDR by adding a dedicated analog jitter measurement circuit or eye monitor. Examples of jitter measurement circuits include PD-based circuits [8], which use a linear PD to convert jitter into analog signals, or time-to-digital converters (TDCs), which convert timing information to digital codes using some form of delay line [9], [10]. Achieving sub-picosecond resolution from either of these circuits is challenging and may require the use of calibration [11]. The sampling rates of published TDCs such as [9]-[11] are also limited to several gigahertz. Another class of measurement circuits estimate jitter statistically by sampling a signal asynchronously [12]. In [13], a dedicated jittery voltage-controlled oscillator generates an asynchronous clock, which is then used to estimate the rms jitter of another clock by sampling it along with a reference signal. These approaches all require the design of custom analog circuits and often also require additional supporting circuitry such as dedicated reference clocks that are sometimes supplied from off-chip.

In CDRs, jitter can also be measured using an eye monitor [14]-[18]. As shown in Fig. 3, an eye monitor consists of a dedicated sampler with an adjustable clock phase that can be generated by a PI. The cumulative distribution function of the jitter can be measured by comparing the eye sampler output S_n to the adjacent data samples D_n and D_{n+1} and counting the number of transitions occurring before and after S_n , as the phase of the eye monitor clock CK_{EYE} is swept across the UI. If the eye sampler has an adjustable voltage threshold, it can also be used to measure the receiver's vertical eye opening, which can help drive adaptation [14].

As with other measurement circuits, eye monitors require extra analog circuits, which consume additional power, area, and design effort. More importantly, these circuits also increase loading on the critical clock and high-speed data paths of the analog front end, potentially degrading CDR

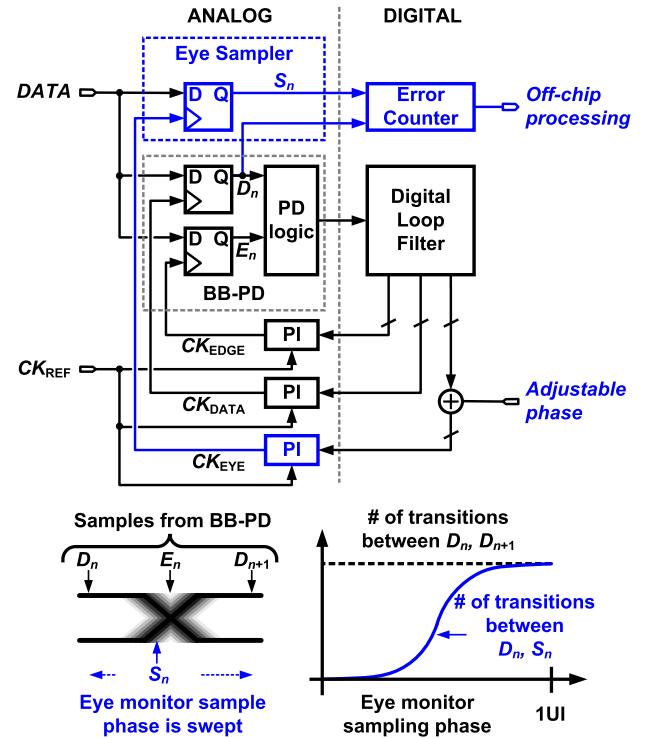


Fig. 3. On-chip jitter measurement using an eye monitor in a CDR.

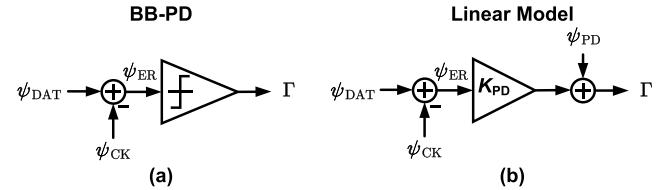


Fig. 4. (a) Model and (b) linearized model of BB-PD operation.

performance. In this paper, we avoid this additional analog overhead by using the CDR's existing BB-PD to measure jitter.

III. PROPOSED TECHNIQUE

Although a BB-PD only outputs the sign of the relative jitter between the clock and data (ψ_{ER}), it can also be modeled as a linear gain K_{PD} with quantization noise ψ_{PD} added at its output [19] as shown in Fig. 4. As such, the BB-PD output Γ can be used to measure ψ_{ER} . However, K_{PD} itself also depends on the probability density function (PDF) and standard deviation of ψ_{ER} (σ_{ER}). An eye monitor can be used to estimate σ_{ER} and, therefore, K_{PD} [18], but that would require additional analog hardware. Instead in this paper, a controlled amount of square-wave jitter (ψ_{INJ}) is injected into the CDR by adding a signal to the PI code of the CDR's edge clock CK_{EDGE} , as shown in Fig. 2. Since the injected jitter amplitude is known, it can be used as a reference to extract K_{PD} and σ_{ER} . When injecting jitter, K_{PD} can be estimated by either correlating the PD output Γ with the injected signal, or by estimating the autocorrelation function $R(n)$ of the PD output as shown in Fig. 5(a). The second method has the additional benefit of allowing the power spectral density (PSD) of the PD output and, therefore, jitter, to be estimated [20] and is, therefore, used in this paper.

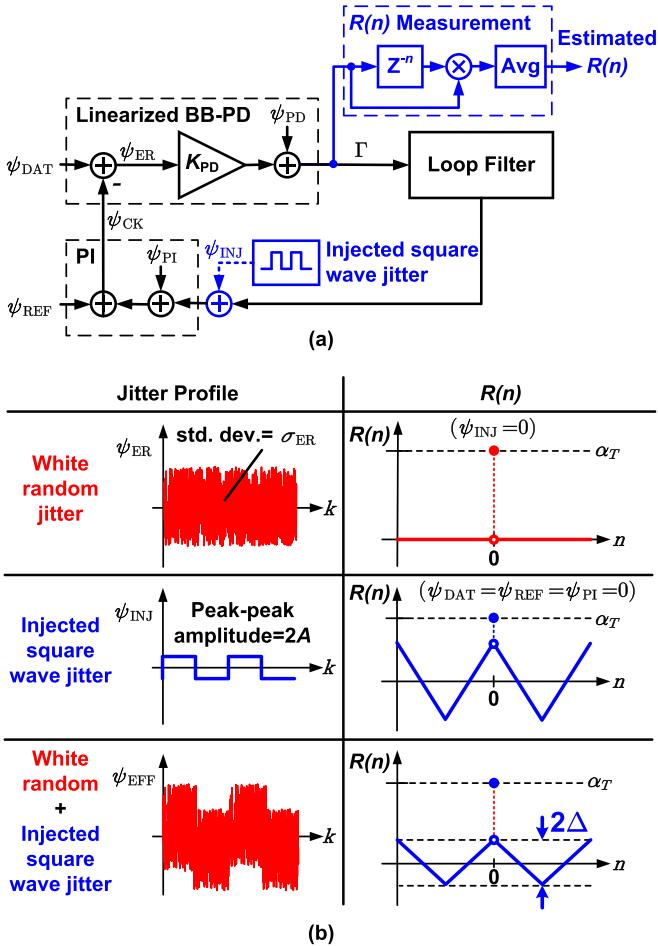


Fig. 5. (a) CDR model showing estimation of $R(n)$ with injected jitter. (b) $R(n)$ for white jitter, injected square-wave jitter, and combination of both.

A. Analysis Using Linear Model

Fig. 5(a) shows the linear model of a PI-based CDR with BB-PD. Here ψ_{ER} has contributions from the data (ψ_{DATA}), reference clock (ψ_{REF}), PI (ψ_{PI}), and BB-PD (ψ_{PD}). $R(n)$ is defined as

$$R(n) = E[\Gamma(k)\Gamma(k-n)] \quad (1)$$

$$= E[K_{PD}^2\psi_{ER}(k)\psi_{ER}(k-n) + \psi_{PD}(k)\psi_{PD}(k-n)] \quad (2)$$

where we have assumed that ψ_{ER} and ψ_{PD} are uncorrelated and that all of the jitter is stationary so that $R(n)$ is not a function of time k . The variance of the PD output (σ_{PD}^2) is found by evaluating $R(n)$ at $n = 0$, which is proportional to the variances of ψ_{ER} and ψ_{PD}

$$R(0) = E[\Gamma^2] = \sigma_{\Gamma}^2 = K_{PD}^2\sigma_{ER}^2 + \sigma_{PD}^2. \quad (3)$$

Since a BB-PD only outputs $+1$, -1 , or 0 when there is no transition, the variance of the PD output is also equal to the transition density of the input data α_T . We can, therefore, solve for σ_{ER} by setting (3) equal to α_T . This gives

$$\sigma_{ER} = \sqrt{\frac{\alpha_T - \sigma_{PD}^2}{K_{PD}^2}}. \quad (4)$$

The value of σ_{PD} depends on the distribution of ψ_{ER} . For example, if ψ_{ER} is Gaussian, σ_{PD} can be derived [19] as

$$\sigma_{PD} = \sqrt{\alpha_T - \frac{2}{\pi}\alpha_T^2}. \quad (5)$$

To extract K_{PD} , we inject square-wave jitter with amplitude A into the recovered clock. Then to estimate $R(n)$, we correlate the PD output with itself delayed by n , and take the average as shown in Fig. 5(a). (Note that since ψ_{INJ} is periodic, ψ_{ER} is actually cyclostationary, making $R(n)$ a periodic function of k . Here for simplicity, $R(n)$ refers to the time-averaged autocorrelation function, which is averaged over k [18].)

Fig. 5(b) shows how $R(n)$ is affected by the injected jitter. When $\psi_{INJ} = 0$, and if ψ_{ER} is largely white, $R(n)$ is dominated by a delta function. When ψ_{INJ} is a square wave and the other jitter sources are zero, $R(n)$ becomes a triangular wave, with a delta function caused by ψ_{PD} , which we also assume to be white. When ψ_{INJ} is added to the existing jitter, $R(n)$ is again a delta function with a triangular wave superimposed on top of it. By measuring the amplitude of the triangular wave (Δ) compared to the rest of $R(n)$, we can extract σ_{ER} , the rms value of the CDR's relative jitter. With jitter injection, the total CDR input jitter (ψ_{EFF}) is equal to $\psi_{ER} + \psi_{INJ}$ and has a variance equal to $\sigma_{EFF}^2 = \sigma_{ER}^2 + \sigma_{INJ}^2$. This gives

$$R(0) = \alpha_T = K_{PD,EFF}^2[\sigma_{ER}^2 + \sigma_{INJ}^2] + \sigma_{PD}^2 \quad (6)$$

where $K_{PD,EFF}$ is the PD gain in the presence of jitter injection. Since ψ_{INJ} is uncorrelated with ψ_{ER} , Δ corresponds to the power of ψ_{INJ} at the PD output

$$\Delta \approx K_{PD,EFF}^2\sigma_{INJ}^2. \quad (7)$$

Combining (7) and (6) and setting $\sigma_{INJ} = A$ gives

$$\sigma_{ER} \approx A \sqrt{\frac{\alpha_T - \sigma_{PD}^2}{\Delta} - 1}. \quad (8)$$

Assuming that ψ_{ER} is Gaussian and A is small compared to ψ_{ER} , ψ_{EFF} can still be approximated as Gaussian. σ_{PD} can then be substituted with (5), giving σ_{ER} as a function of Δ

$$\sigma_{ER} \approx A \sqrt{\frac{2\alpha_T^2}{\pi\Delta} - 1}. \quad (9)$$

Since the BB-PD output is binary, $R(n)$ can be measured with simple logic and digital counters that count the number of correlated outputs [18]. Δ can then be measured from the resulting waveform. Because a PD output of zero provides no useful information, we can also calculate a normalized $R(n)$ ($R'(n)$) only using the non-zero PD output samples

$$R'(n) = E[\Gamma(k)\Gamma(k-n)|\Gamma(k) \neq 0, \Gamma(k-n) \neq 0]. \quad (10)$$

Calculating $R(n)$ in this way removes the effect of transition density. α_T then becomes 1 in (9), making σ_{ER} only a function of Δ . Although the relationship is nonlinear, in a digital implementation, it could be stored in a simple lookup table, whose number of addresses would depend on the number of bits with which Δ is estimated, and whose bit width would be determined by the accuracy with which the response is to be fitted. In addition, in some applications, the above calculations may not all be required. For example, to minimize jitter in an adaptive system, Δ itself can be used to drive the adaptive loop, since the largest value of Δ corresponds to the smallest value of σ_{ER} . In this case, a precise estimate of σ_{ER} is not needed and since measuring Δ provides enough information, the implementation can be simplified.

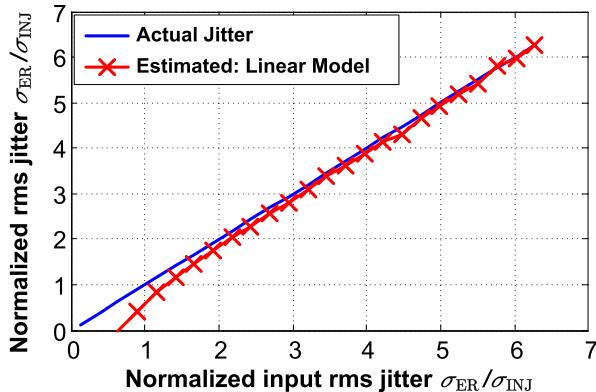


Fig. 6. MATLAB simulation results with Gaussian ψ_{ER} , comparing rms jitter estimated using (9) compared to actual jitter.

B. Simulation Results

The proposed technique was simulated in MATLAB by adding a square wave to Gaussian RJ and modeling the BB-PD as a sign operation. $R(n)$ was estimated and the amplitude of the triangular wave Δ was found by taking the average difference between the peaks and valleys of $R(n)$. Fig. 6 shows the actual and estimated rms jitter (σ_{ER}), found using (9) and normalized to σ_{INJ} . The plot shows good agreement between the two curves when $\sigma_{\text{ER}}/\sigma_{\text{INJ}}$ is greater than approximately 2. However, the curves diverge when σ_{ER} becomes too small, since the derivations of Section III-A assumed that $\sigma_{\text{INJ}} \ll \sigma_{\text{ER}}$, and also used the linearized model of the PD, which is only an approximation. To improve the accuracy of the jitter estimates when σ_{ER} is small, the effect of the injected jitter can be analyzed in terms of the PDF of ψ_{ER} . Next, we provide this analysis for the case when ψ_{ER} is Gaussian.

C. Analysis Using Jitter PDF

Without jitter injection, the BB-PD simply outputs the sign of ψ_{ER} . We can, therefore, write $R(n)$ as

$$R(n) = E[\text{sgn}(\psi_{\text{ER}}(k))\text{sgn}(\psi_{\text{ER}}(n-k))]. \quad (11)$$

Normalizing $R(n)$ as in (10) allows us to ignore the effect of transition density. $R'(n)$ can, then, be written as

$$\begin{aligned} R'(n) &= P[\text{sgn}(\psi_{\text{ER}}(k)) = \text{sgn}(\psi_{\text{ER}}(n-k))] \\ &\quad - P[\text{sgn}(\psi_{\text{ER}}(n)) \neq \text{sgn}(\psi_{\text{ER}}(n-k))]. \end{aligned} \quad (12)$$

Assuming that the PSD of ψ_{ER} does not have large low-frequency content, then samples of ψ_{ER} separated by a long time n should be uncorrelated with each other. This means that the signs of $\psi_{\text{ER}}(k)$ and $\psi_{\text{ER}}(n-k)$, which are also assumed to have zero mean, are also uncorrelated. In other words, without jitter injection $R'(n) \approx 0$ for large n .

When ψ_{INJ} is added, peaks and valleys are introduced into $R'(n)$. $R'(n)$ will reach its peak when $\psi_{\text{INJ}}(k)$ and $\psi_{\text{INJ}}(n-k)$ have the same sign and phase and reach a minimum when they have the opposite sign. Taking the case when $\psi_{\text{INJ}}(k) = \psi_{\text{INJ}}(n-k) = A$, the peak of $R'(n)$ can be written as

$$\begin{aligned} R'_{\max} &= P[\text{sgn}(\psi_{\text{ER}}(k) + A) = \text{sgn}(\psi_{\text{ER}}(n-k) + A)] \\ &\quad - P[\text{sgn}(\psi_{\text{ER}}(k) + A) \neq \text{sgn}(\psi_{\text{ER}}(n-k) + A)]. \end{aligned} \quad (13)$$

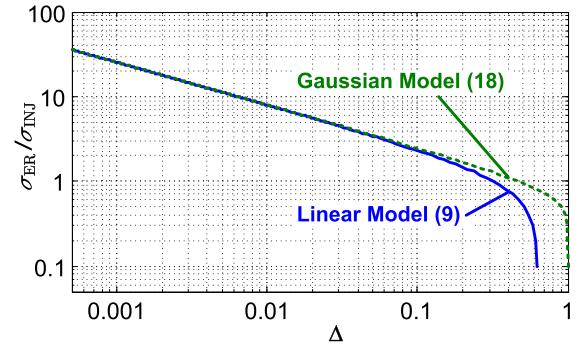


Fig. 7. Comparison between analyses based on linear model (9) and Gaussian model (18).

Assuming that ψ_{ER} is white and stationary, then any samples $\psi_{\text{ER}}(n)$ and $\psi_{\text{ER}}(n-k)$ of ψ_{ER} are also independent and identically distributed (*iid*) random variables. R'_{\max} can then be evaluated in terms of the PDF of ψ_{ER} ($f_{\psi_{\text{ER}}}(\psi_{\text{ER}})$)

$$\begin{aligned} R'_{\max} &= P[\psi_{\text{ER}} > -A]^2 + P[\psi_{\text{ER}} < -A]^2 \\ &\quad - 2P[\psi_{\text{ER}} > -A]P[\psi_{\text{ER}} < -A] \end{aligned} \quad (14)$$

$$= [2P[\psi_{\text{ER}} > -A] - 1]^2 \quad (15)$$

$$= \left[2 \left(\int_{-A}^{\infty} f_{\psi_{\text{ER}}}(\psi_{\text{ER}}) d\psi_{\text{ER}} \right) - 1 \right]^2. \quad (16)$$

If ψ_{ER} has a zero-mean Gaussian distribution, the integral in (16) can be replaced with the Q -function

$$R'_{\max} = \left[2Q\left(\frac{-A}{\sigma_{\text{ER}}}\right) - 1 \right]^2. \quad (17)$$

Since $R'_{\max} \approx \Delta$, σ_{ER} can then be estimated from Δ as

$$\sigma_{\text{ER}} \approx \frac{-A}{Q^{-1}\left(\frac{1+\sqrt{\Delta}}{2}\right)}. \quad (18)$$

Fig. 7 shows σ_{ER} normalized to σ_{INJ} comparing the values as estimated using (9) and (18). As shown in Fig. 7, the two equations diverge when $\sigma_{\text{ER}}/\sigma_{\text{INJ}}$ becomes less than approximately 2, which is also where (9) starts to become less accurate in Fig. 6. Fig. 8 re-plots the results of Fig. 6 comparing jitter as estimated using (9) from the linear analysis of Section III-A, compared to (18). As shown in Fig. 8, as σ_{ER} becomes small compared to σ_{INJ} , the jitter estimate based on the Gaussian PDF becomes more accurate. Therefore in cases where the shape of the jitter PDF is known, or can be assumed to be Gaussian, an accurate jitter estimate can be obtained even when σ_{ER} is much smaller than σ_{INJ} . When a PI is used for jitter injection, the smallest value of σ_{INJ} (which is also equal to the PI's step size) is approximately 0.56 ps, it is therefore theoretically possible to estimate jitter in the hundreds of femtosconds. However, several sources of error will limit the achieved performance.

D. Limitations and Sources of Error

The accuracy of the proposed technique depends on the precision of the injected jitter's amplitude and on the accuracy

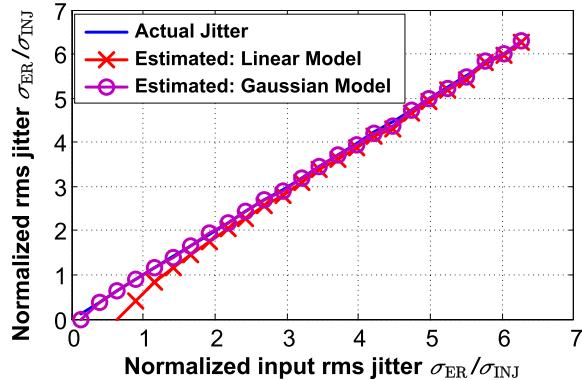


Fig. 8. MATLAB simulation results with Gaussian ψ_{ER} , comparing rms jitter estimated using linear and Gaussian models to actual jitter.

with which we can measure and model the effect of the injected jitter on $R(n)$.

The amplitude accuracy of the jitter injection is primarily limited by nonlinearity in the PI, which can cause the injected jitter amplitude to differ from the expected value A . Such inaccuracy will cause gain error as seen from the expressions in Section III-A. However, this error is mitigated whenever frequency offset is present, as in this paper. With frequency offset, the CDR continuously rotates through all PI codes. Any error in the PI step size will, therefore, be averaged across all PI codes, greatly reducing the impact of any nonlinearity on jitter measurement. However, when no frequency offset is present, the CDR edge PI will only alternate between two codes, making PI nonlinearity more important. To address this, the PI step size could be calibrated using similar techniques to those used in TDCs [11], [21].

Accurate modeling of $R(n)$ also depends on making correct assumptions about the jitter's PDF and spectral content. For example, if the jitter being measured has a uniform, rather than Gaussian distribution, (9) and (18) will become inaccurate, and have a gain error of approximately 38%. This is because uniformly distributed jitter leads to a lower K_{PD} value compared to the Gaussian case [22], changing the expression for σ_{PD} in (8), and the PDF used in (16).

Last, we assumed in Section III-C, that $R(n)$ is zero for large n , which makes Δ easy to extract. This assumption will not be valid if the jitter has large low-frequency, or periodic content. Large low-frequency content will give $R(n)$ a slowly decaying response as a function of n , so that $R(n)$ will not be zero until n is much larger than the time constant of the jitter. In the case of periodic jitter, which could include any excessive CDR dithering, a periodic waveform will also exist in $R(n)$. If the periodic jitter amplitude becomes comparable to that of the injected jitter, this waveform will start to reduce the accuracy with which Δ is estimated. However, both of these cases will be observable from the $R(n)$ waveform and could be addressed by using large values of n and more detailed fitting or analysis of the $R(n)$ waveform.

IV. IMPLEMENTATION

The proposed measurement technique was applied to the 28 Gb/s half-rate PI-based digital CDR shown in Fig. 9.

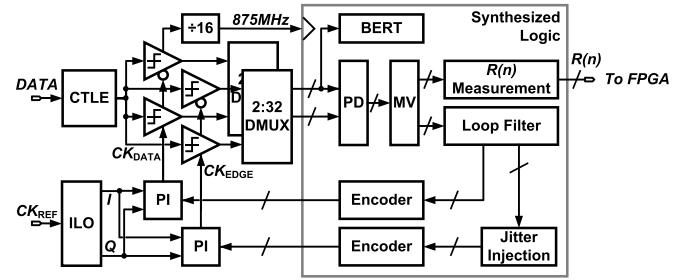


Fig. 9. Overview of implemented half-rate CDR.

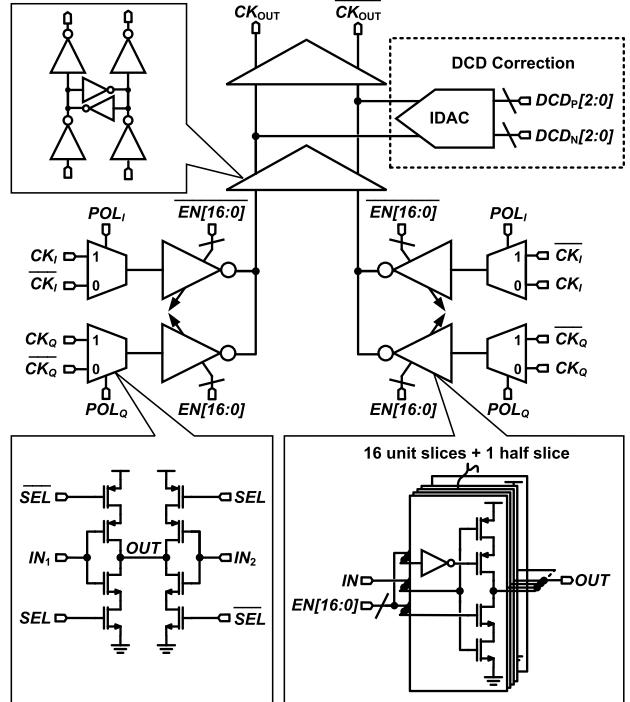
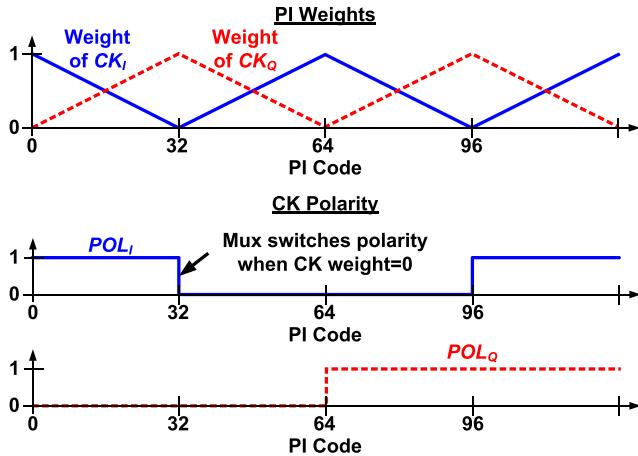


Fig. 10. PI circuit.

The input data are first equalized by a continuous time linear equalizer (CTLE) before being sampled by the half-rate front end. Half-rate clocks are generated by an injection-locked oscillator, which takes an external 14 GHz reference clock and generates quadrature clocks for the 7-bit CMOS inverter-based PIs shown in Fig. 10.

In each PI, the upper 2 bits of the PI code select the polarities of the two input clocks while the lower 5 bits control their respective weights by enabling and disabling inverter slices for each input. The polarity of each clock input is selected using a mux, which passes either the clock or its inverted version. A source-switched mux is used, which achieves higher bandwidth at the cost of increased signal feed-through. The 5-bit inverter weight is decoded to control 16 thermometer-coded slices. To save area, the LSB is implemented by adding a half-size LSB slice. Fig. 11 shows the control signals for CK_{OUT} in Fig. 10. As shown in Fig. 11, at each quadrant boundary, the polarity of either CK_I or CK_Q is switched to achieve full 2π phase rotation. Because the polarity signals POL_I and POL_Q switch when the weights of their corresponding clocks are zero, the effect of the mux switching is reduced.

Fig. 11. Control signals for CK_{OUT} in Fig. 10.

Duty cycle distortion (DCD) in each PI was corrected from off-chip by programming an on-chip current-mode digital-to-analog converter (IDAC) shown in Fig. 10. To calibrate the DCD of CK_{EDGE} , the IDAC code is adjusted to minimize the difference between the average of the even and odd PD outputs when the CDR is locked. This is because when locked, DCD in CK_{EDGE} , which is aligned to the edges of the input data, will cause the average even and odd PD outputs to differ, with one becoming biased early and the other being biased late. To calibrate the DCD of CK_{DATA} , the polarity of the CDR's PD is inverted in a calibration mode, so that the CDR aligns CK_{DATA} to the edges of the input data instead of the center. DCD can, then, be corrected as before by comparing the even and odd PD outputs. This scheme assumes of course that the input data does not have significant DCD and can be used as a reference. To avoid this reliance on the duty cycle of the input data, an analog DCD calibration circuit could also be included on-chip [23].

Data from the analog front end is demuxed by 32 and sent to the digital core clocked at 875 MHz. In the digital core, the edge and data samples pass through PD logic, generating 32 parallel BB-PD outputs. Majority voting (MV) reduces this parallel output to a single binary output which drives the digital loop filter and $R(n)$ measurement blocks.

Each value of $R(n)$ is measured using a FIFO, which first delays the MV output by the desired delay n , and two counters that measure the correlation between the delayed and current MV outputs. One counter counts the total number of samples, while the second counts the number of correlated outputs over the same interval. In this paper, approximately twenty thousand samples of the MV output are used to measure each point of $R(n)$. This corresponds to a measurement time of roughly 23 μ s per point. If incorporated in an adaptive loop, this could lead to adaptation times on the order of milliseconds, depending on the adaptation step size, the number of samples used to estimate each value of $R(n)$ (corresponding to the amount of averaging used), and the number of points of $R(n)$ being measured. While $R(n)$ is measured on-chip, in this paper Δ is extracted off-chip. In this implementation, measuring $R(n)$ from the downsampled

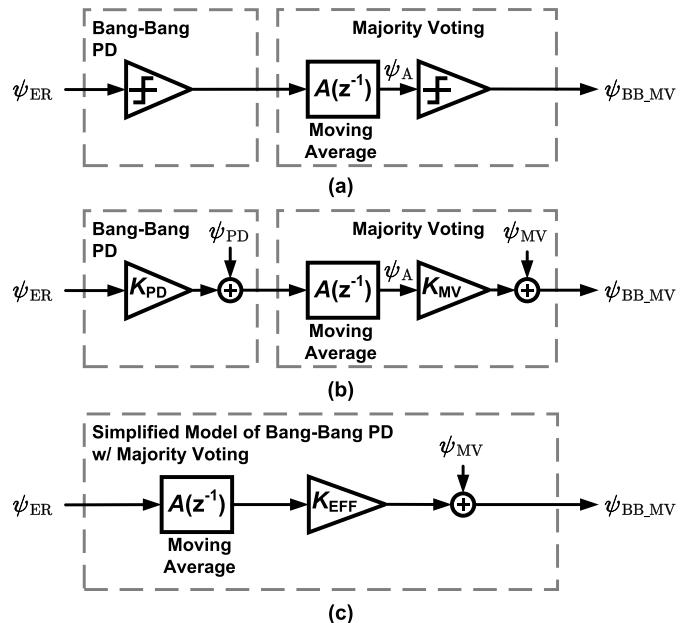


Fig. 12. (a) Functional model, (b) linear model, and (c) simplified linear model of BB-PD followed by MV.

PD output and after MV greatly simplifies the $R(n)$ measurement block. However, MV also introduces a lowpass filtering effect that must be considered.

A. Effect of Majority Voting

MV consists of a moving average lowpass filter (LPF) followed by a slicer as shown in Fig. 12(a). The slicer can be modeled identically to a BB-PD, as a linear gain K_{MV} with a quantization noise source ψ_{MV} added to its output as shown in Fig. 12(b). The values of both K_{MV} and ψ_{MV} depend on the PDF of the output of the moving average filter ψ_A .

If the BB-PD output is a white, stationary random process with iid samples, then if the demux ratio N is large, by the central limit theorem [24], the output of the moving average filter ψ_A , which is the sum of N consecutive PD outputs is a random process that can be approximated as having a zero-mean Gaussian distribution. This gives a lower bound for σ_A

$$\sigma_A \geq \sqrt{N} \sigma_\Gamma. \quad (19)$$

Note that σ_A could be higher if the BB-PD outputs are correlated with each other. Analogous to the analysis in [19] for a BB-PD, and since there is no effect from transition density, the slicer gain K_{MV} and the standard deviation of ψ_{MV} (σ_{MV}) can, then, be approximated by

$$K_{MV} \approx \sqrt{\frac{2}{\pi}} \frac{1}{\sigma_A} \quad (20)$$

$$\sigma_{MV} \approx \sqrt{1 - \frac{2}{\pi}} \approx 0.6. \quad (21)$$

Because of its lowpass filtering effect, MV also heavily suppresses the quantization noise of the BB-PD ψ_{PD} that precedes it, although in doing so, it also increases quantization noise [25]. Ignoring ψ_{PD} for simplicity, K_{PD} and K_{MV} can be combined into an effective gain K_{EFF} as shown in Fig. 12(c). To check the validity of this simplification, a CDR was

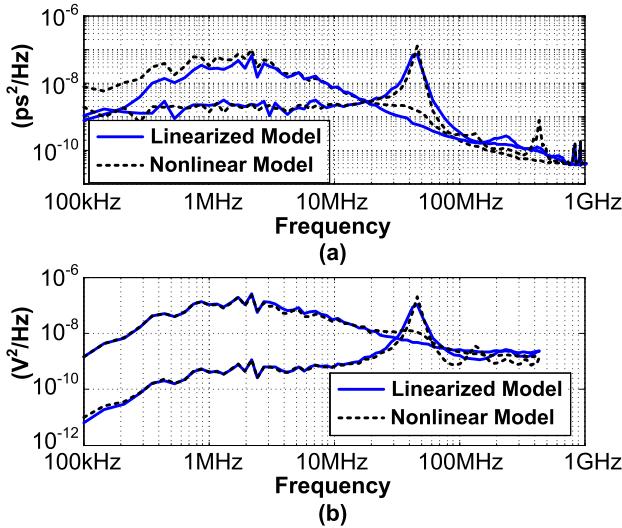


Fig. 13. Simulated spectrum of (a) relative jitter (ψ_{ER}) and of (b) output of MV for CDR using nonlinear and linearized models of BB-PD and MV for two different loop gain settings.

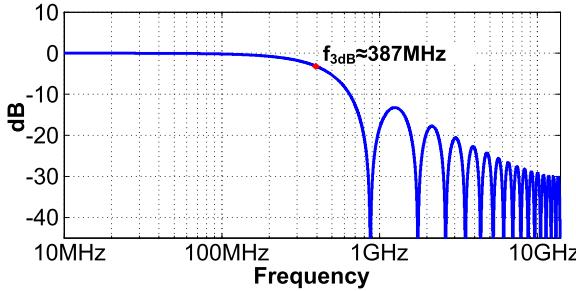


Fig. 14. Normalized frequency response of MV.

simulated in Simulink, replacing the nonlinear BB-PD and MV blocks of Fig. 12(a) with the simplified model in Fig. 12(c). Fig. 13 shows the spectrum of the CDR's relative jitter ψ_{ER} and output from the MV block with the CDR configured with two different loop gain settings. As shown in Fig. 13, the results of the nonlinear and linearized models match quite closely. The linear model does, however, fail to predict some spurs, due to its lack of nonlinearity.

Note that the model in Fig. 12(c) is identical to that of a BB-PD preceded by a LPF. The analysis of Section III therefore still applies, except that the LPF response of MV, which is shown in Fig. 14 for this design, also filters the jitter that can be observed from the PD output. This suppresses wideband jitter such as ISI jitter, and limits the effective measurement bandwidth to roughly 387 MHz in this implementation. Since the measurement bandwidth is inversely proportional to N , if wider measurement bandwidth is required, voting could be performed over fewer samples (by reducing N), or voting could be omitted entirely at the cost of higher complexity in the $R(n)$ measurement block.

B. Jitter Injection

In the half-rate architecture, jitter can be injected into the CDR's edge clock without affecting the data sampling phase, by passing the PI code of the edge clock through a jitter

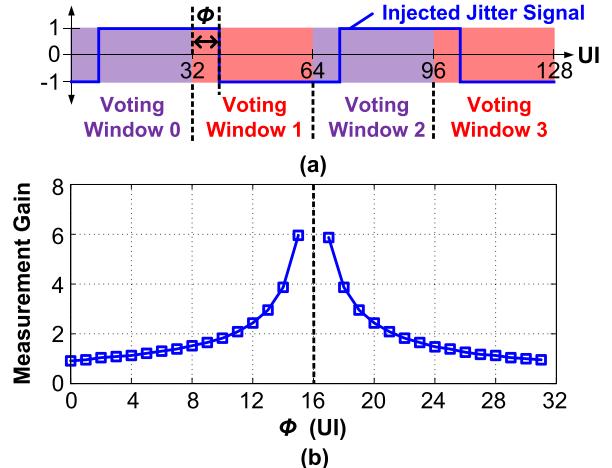


Fig. 15. (a) Illustration of skew between injected jitter and voting window and (b) its simulated effect on the effective gain of jitter measurement.

injection stage which alternately adds or subtracts an offset from the PI code. As a result, square-wave jitter is injected at 437.5 MHz (half the clock rate of the digital core). Since this is well beyond the CDR's loop bandwidth of approximately 10 MHz, it gets heavily suppressed by the CDR's loop filter, minimizing its impact on the CDR's operation.

Although the injected jitter frequency exceeds the 3 dB frequency of the MV filter response, as long as the injected square wave is aligned to the window over which voting is performed, it will not be attenuated. If, however, the injected jitter is misaligned, as can occur due to the delay of the analog front end, the square wave will be attenuated. This means that σ_{INJ} will be smaller than is assumed by (9) and (18) and the estimated jitter will have a gain error. Fig. 15 shows this effect and plots the simulated gain of the estimated jitter as a function of the phase skew Φ of the injected jitter compared to the voting window. In the extreme case, if the injected jitter is skewed by $\pi/2$ compared to the voting window (corresponding to 16UI in this paper), ψ_{INJ} can be completely averaged out, and jitter cannot be measured.

In this paper, we were able to measure the misalignment of the injected jitter by examining the subsampled PD output at the input to the MV stage, using a test mux. By comparing the first PD output within each voting window, with the other bits within the same voting window, we found the bit index where the average PD output changed sign. Because this bit position corresponds to the UI where the injected jitter changes sign, we determined that the injected jitter was misaligned by approximately 11UI. Based on Fig. 15(b), this corresponds to a measurement gain of approximately 2, meaning that the measured results had to be divided by this factor to correct for the gain error.

V. MEASUREMENT RESULTS

The complete CDR was fabricated in 28 nm CMOS and consumes 106.6 mW while operating at 28 Gb/s. Digital logic accounts for 32% of the total power consumption. Fig. 16 shows the die photograph of the fabricated chip. The digital core occupies relatively large area for two reasons. First,

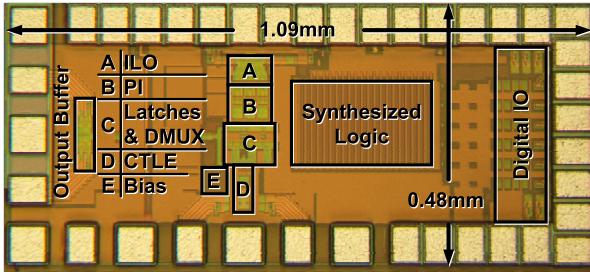


Fig. 16. Die photo of test chip.

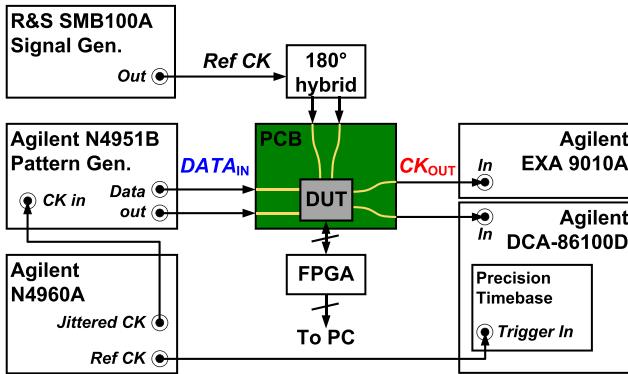


Fig. 17. Test setup.

the density of the synthesized logic was very low. Second, a large amount of test registers were included to monitor internal signals at up to 437.5 MHz. Although a digital power breakdown is not available, $R(n)$ measurement circuits only account for roughly 12.5% of the digital core area.

Fig. 17 shows the test setup. The chip was tested by wire bonding it to a custom printed circuit board (PCB). The CDR's 14 GHz reference clock was supplied by a Rhode and Schwarz SMB100A signal generator. PRBS data were generated by an Agilent N4951B pattern generator, clocked by an N4960A serial BERT controller having an rms jitter of approximately 421 fs as measured using an Agilent EXA 9010A spectrum analyzer and shown in Fig. 18(a). Fig. 18(b) shows the eye diagram of the pattern generator output, which has 1.03 ps rms jitter when observed through short SMA cables using an Agilent DCA-86100D oscilloscope with precision timebase module.

Fig. 18(c) shows the phase noise of the CDR's recovered clock when locked to PRBS31 data with 50 ppm frequency offset. The recovered clock has an rms jitter of approximately 950 fs. At low frequencies, the CDR's phase noise tracks that of the input while mid-band phase noise is determined by the performance of the CDR and includes contributions from PI nonlinearity.

A. PI Linearity

Fig. 19(a) and (b) show the differential nonlinearity (DNL) and integral nonlinearity (INL) of the PI, measured by observing CK_{EDGE} through a test mux, followed by a CMOS buffer chain and output driver. The highest DNL occurs when the PI changes quadrants as the PI's muxes switch the polarity of one of the PI's input clocks. Even though the weight of the clock

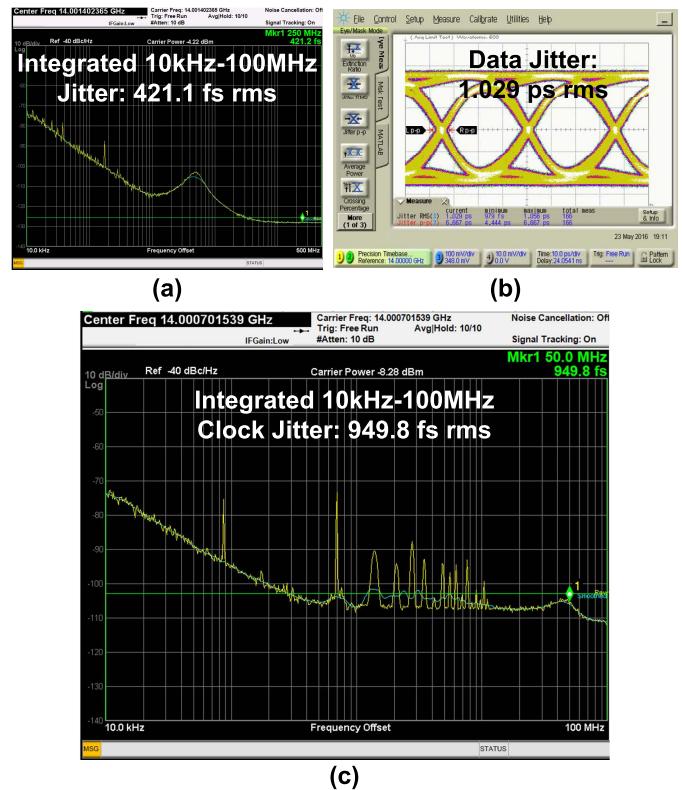


Fig. 18. (a) Phase noise of reference clock for pattern generator, (b) eye diagram of PRBS data input to CDR, and (c) phase noise of CDR's recovered clock with 50 ppm frequency offset.

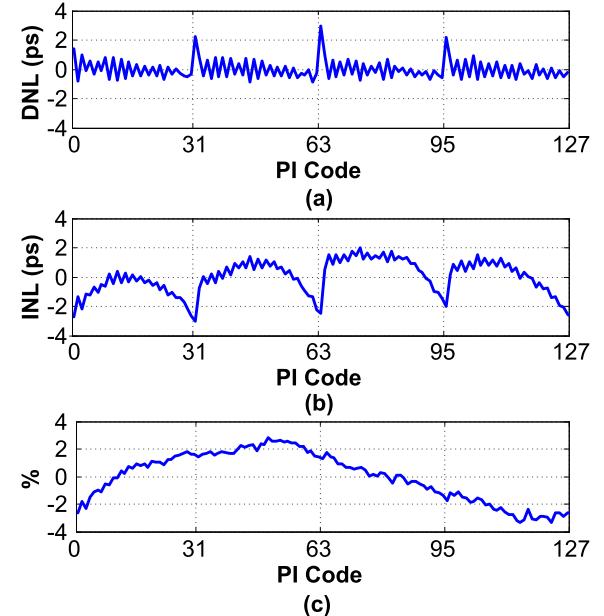


Fig. 19. Measured (a) DNL (b) INL, and (c) % DCD variation of clock versus PI code.

whose polarity is switched is zero at the time the quadrant changes, simulations show that capacitive coupling between the output of the mux and output of the PI's inverter banks contributes to a discontinuity in the PI's response as the mux output changes. Mismatch between the PI inverter slices and

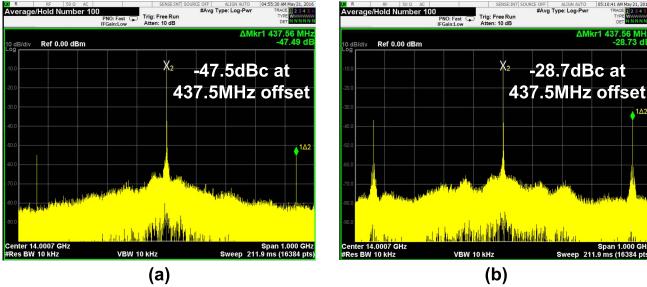
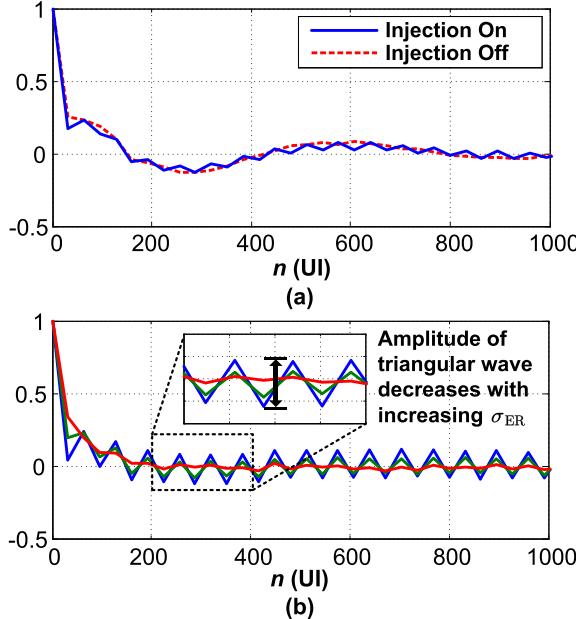


Fig. 20. CDR clock spectrum with jitter injection (a) disabled (b) enabled.

Fig. 21. $R(n)$ measured (a) with and without jitter injected and (b) for three different jitter amplitudes.

the half slice also causes even/odd variation in the PI response. In addition, it was observed that the DCD of $C_{K\text{EDGE}}$ varied versus the PI code as shown in Fig. 19(c), which plots the change in DCD from its nominal value. Because the observed clock passes through multiple mux and buffer stages, it is unclear if the DCD variation occurs within the PI itself, or in the test path.

B. Jitter Injection

Fig. 20. shows the CDR's output clock spectrum before and after jitter injection is enabled. Without any injection, some jitter is already observable at 437.5 MHz in the PI output, but at a very low level, likely caused by coupling. Enabling jitter injection increases jitter substantially. To verify the effect of jitter injection on $R(n)$, Fig. 21(a) shows an example of an $R(n)$ waveform measured using the on-chip counters, with and without jitter injection enabled. As shown in Fig. 21(a), enabling jitter leaves the shape of $R(n)$ relatively unchanged, but causes a triangular wave to appear as expected. Fig. 21(b) compares $R(n)$ while varying amounts of data jitter are applied to the CDR input, thus varying σ_{ER} . As shown in Fig. 21(b), as σ_{ER} increases, the amplitude of the triangular waveform in $R(n)$ decreases as expected.

We found Δ by post-processing the $R(n)$ waveform off-chip and approximated Δ as the average difference between the

even and odd samples of $R(n)$, which are spaced 32UI apart. Since the linear and Gaussian models provide very similar results, σ_{ER} is estimated using the linear model in the remainder of this paper.

C. Jitter Measurement Results

To assess the accuracy of our measurement technique, we measure the rms jitter of the CDR's input data ($\sigma_{\text{DAT,Meas}}$) and recovered clock ($\sigma_{\text{CK,Meas}}$) at the input and output of the CDR using the Agilent DCA-86100D sampling scope, as shown in Fig. 17. All jitter measurement results were performed with the CDR reference clock having a frequency offset of 50–100 ppm with respect to the received data, while the CDR is locked to 28 Gb/s PRBS31 data. Since our technique estimates the relative rms jitter at the CDR input, we must compare our results to $\sigma_{\text{ER,Meas}}$. Because simulations show that the CDR's output jitter is dominated by PI and reference clock jitter and not data jitter, $\psi_{\text{DAT,Meas}}$ and $\psi_{\text{CK,Meas}}$ are essentially uncorrelated, meaning that $\sigma_{\text{ER,Meas}}$ can be approximated as

$$\sigma_{\text{ER,Meas}} \approx \sqrt{\sigma_{\text{CK,Meas}}^2 + \sigma_{\text{DAT,Meas}}^2}. \quad (22)$$

In the following measurements, the PRBS generator output was applied to the DUT through low loss cables. As such, the CTLE only had to compensate for ISI induced by the short PCB trace and wirebond. Unless otherwise noted, the CTLE was always configured to achieve the lowest jitter, as confirmed by jitter tolerance measurements. Nonetheless, since $\sigma_{\text{DAT,Meas}}$ is measured at the input of the PCB, it does not contain any residual ISI that might still be left at the CTLE output. However, as discussed earlier, MV also suppresses wideband jitter such as ISI in our implemented measurement scheme. As such, our estimated jitter can be fairly compared against $\sigma_{\text{ER,Meas}}$.

The accuracy of the proposed technique was tested in several conditions, as the CDR's input data jitter, loop filter settings and CTLE settings were varied.

D. Measuring the Effect of Data Jitter

First, σ_{ER} was estimated as different amounts of RJ were added to the CDR's input data, by driving the external jitter input of the Agilent N4960A serial BERT controller and clock synthesizer with a NoiseCom NC6110 noise generator. Using this setup, RJ could be injected up to a bandwidth of 350 MHz.

Since this falls within our system's measurement bandwidth, the filtering effect of MV does not have a large impact on these jitter estimates. (If RJ is injected over a much wider bandwidth, it could be filtered by MV, leading to offset and/or gain error in the estimated jitter.)

Fig. 22 compares σ_{ER} as estimated with our technique compared to $\sigma_{\text{ER,Meas}}$, derived from oscilloscope measurements. As shown in Fig. 22, the results show good agreement, with results matching well within 1 ps.

E. Measuring the Effect of Loop Gain Setting

Next, we examine the ability of this technique to measure the effect of different loop gain settings on ψ_{ER} . Fig. 23 shows

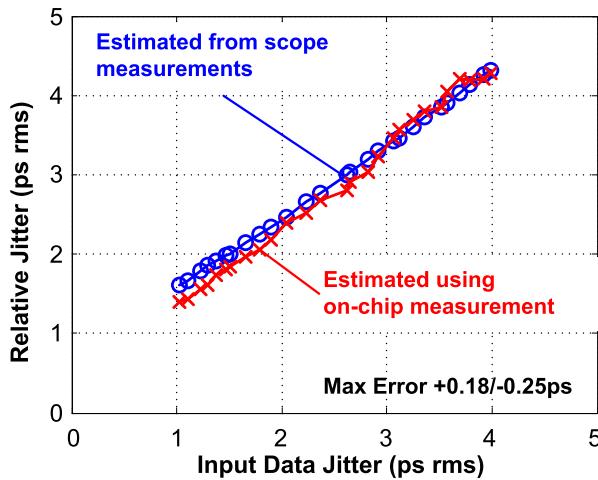


Fig. 22. Estimated relative CK versus data jitter as input data jitter is swept.

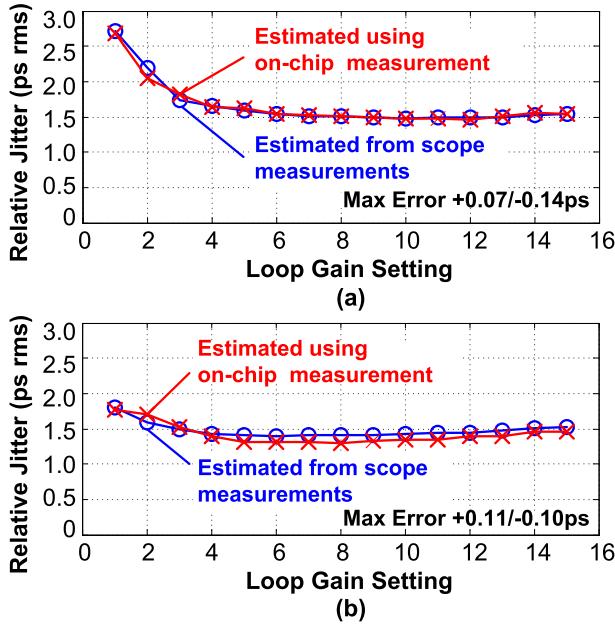


Fig. 23. Estimated relative CK versus data jitter at CDR input as CDR loop gain setting is swept for (a) test case 1 and (b) test case 2.

the estimated and measured jitter as the CDR's loop gain setting is swept for two test cases. In each case, the CDR is driven with reference clocks having different phase noise profiles and frequency offsets.

The results are shown in Fig. 23 and again show good agreement between the estimated and measured jitter. This example demonstrates how this technique can potentially be used in adapting a CDR's loop gain to minimize jitter.

F. Measuring the Effect of CTLE Setting

Jitter was also estimated as the CTLE setting was swept. Since ISI-induced jitter is suppressed in this implementation, the proposed technique can only be used to observe the impact of the CTLE on the CDR's recovered clock jitter. Since the oscilloscope measurements also do not include the effect of CTLE under or over-equalization, the estimated and

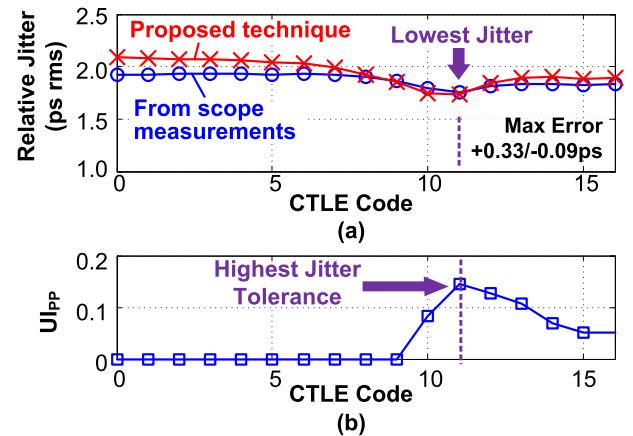
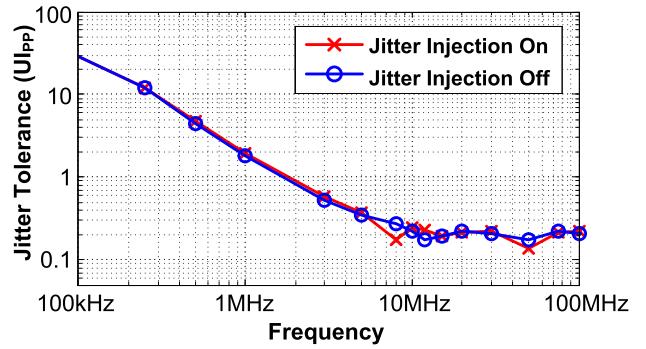


Fig. 24. (a) Estimated relative CK versus data jitter as the CTLE code is swept, with (b) corresponding jitter tolerance at 100 MHz (when jitter injection is disabled).

Fig. 25. Measured jitter tolerance ($\text{BER} < 10^{-12}$, PRBS31) with and without jitter injection enabled.

measured jitter can be fairly compared. Given this, Fig. 24(a) again shows good matching between the jitter as estimated using our technique and based on oscilloscope measurements. Both techniques measure CTLE code 11 as having the lowest jitter.

To test whether this is optimal, the high-frequency jitter tolerance of the CDR was also measured at 100 MHz for the corresponding CTLE codes with jitter injection disabled. The results shown in Fig. 24(b) confirm that the highest jitter tolerance is achieved with code 11 as expected. This demonstrates that even though the proposed technique cannot fully characterize the effect of the CTLE on ISI, the jitter estimates can still be used to help optimize a CDR's equalizer settings.

G. Effect of Jitter Injection on CDR Operation

Last, we also examine the effect of the injected jitter on jitter tolerance. Since only the edge clock is modulated, we expect the data samples to be relatively unaffected. To verify this, we perform jitter tolerance measurements with jitter injection enabled. As shown in the jitter tolerance results plotted in Fig. 25, although some degradation is seen compared to the normal case, the CDR is able to maintain error-free operation with $\text{BER} < 10^{-12}$, making it possible to perform measurements while the CDR remains operational. Note that jitter tolerance was measured with an adaptive loop gain

TABLE I
COMPARISON TO OTHER JITTER MEASUREMENT TECHNIQUES

Reference	Analog Circuit Overhead	Digital Circuit Overhead	Off-Chip Post-Processing	Ext. Ref. Ck?	Freq./Data Rate	Accuracy
[8]	Analog PD	-	N/A	Yes	2.5Gb/s	1.56ps
[9]	TDC	Counters	Yes	No	3.36GHz	<0.7ps
[10]	TDC	-	Yes	Yes	250MHz	<1ps
[11]	TDC	-	Yes	Yes	1GHz	0.1ps
[12]	Sampler, Edge Detect.	FIFO	Yes	Yes	5Gb/s	-
[13]	Sampler, VCO	Counters	No	No	6GHz	<0.3ps
[15]	Eye-Mon.	-	Yes	No	10Gb/s	-
[16]	Eye-Mon.	-	Yes	No	40Gb/s	-
This work	None	Counters	Yes	No	28Gb/s	+0.33ps /-0.25ps

algorithm [20] enabled, which automatically compensates for any changes in the CDR's loop gain caused by variations in jitter. If the injected jitter is too large, it could affect the gain of the BB-PD, and consequently, the loop gain of the CDR. Therefore without adaptation, if large jitter is being injected, the loop filter may have to be re-tuned to maintain optimal performance.

VI. CONCLUSION

A jitter measurement technique has been proposed which makes use of existing circuits in a half-rate PI-based CDR, allowing jitter to be measured without having to add additional analog circuits. Table I compares this work to prior measurement techniques. Similar to most prior techniques, off-chip post-processing is used to recover the estimated jitter, although such processing could easily be incorporated on-chip.

Measurements demonstrate how this technique can be used to characterize the effects of varying data jitter, CDR loop gain, and CTLE settings on CDR jitter performance. When combined with adaptation logic, this technique can be used for on-chip adaptation.

ACKNOWLEDGMENT

The authors would like to thank CMC Microsystems for providing CAD tools and test equipment.

REFERENCES

- M. Mansuri, A. Hadiashar, and C.-K. K. Yang, "Methodology for on-chip adaptive jitter minimization in phase-locked loops," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 11, pp. 870–878, Nov. 2003.
- J. Liang, A. Sheikholeslami, H. Tamura, and H. Yamaguchi, "Jitter injection for on-chip jitter measurement in PI-based CDRs," in *Proc. IEEE Custom Integr. Circuits Conf.*, Apr./May 2017, pp. 1–4.
- R. Navid *et al.*, "A 40 Gb/s serial link transceiver in 28 nm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 814–827, Apr. 2015.
- L. Rodoni, G. von Buren, A. Huber, M. Schmatz, and H. Jackel, "A 5.75 to 44 Gb/s quarter rate CDR with data rate selection in 90 nm bulk CMOS," *IEEE J. Solid-State Circuits*, vol. 44, no. 7, pp. 1927–1941, Jul. 2009.
- T. K. Kuan and S. I. Liu, "A bang bang phase-locked loop using automatic loop gain control and loop latency reduction techniques," *IEEE J. Solid-State Circuits*, vol. 51, no. 4, pp. 821–831, Apr. 2016.
- S.-H. Chu *et al.*, "A 22 to 26.5 Gb/s optical receiver with all-digital clock and data recovery in a 65 nm CMOS process," *IEEE J. Solid-State Circuits*, vol. 50, no. 11, pp. 2603–2612, Nov. 2015.
- Z.-H. Hong, Y.-C. Liu, and W.-Z. Chen, "A 3.12 pJ/bit, 19–27 Gbps receiver with 2-tap DFE embedded clock and data recovery," *IEEE J. Solid-State Circuits*, vol. 50, no. 11, pp. 2625–2634, Nov. 2015.
- M. Ishida *et al.*, "A programmable on-chip picosecond jitter-measurement circuit without a reference-clock input," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 1, Feb. 2005, pp. 512–614.
- K. Niitsu, M. Sakurai, N. Harigai, T. J. Yamaguchi, and H. Kobayashi, "CMOS circuits to measure timing jitter using a self-referenced clock and a cascaded time difference amplifier with duty-cycle compensation," *IEEE J. Solid-State Circuits*, vol. 47, no. 11, pp. 2701–2710, Nov. 2012.
- K. Nose, M. Kajita, and M. Mizuno, "A 1-ps resolution jitter-measurement macro using interpolated jitter oversampling," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2911–2920, Dec. 2006.
- T. Hashimoto, H. Yamazaki, A. Muramatsu, T. Sato, and A. Inoue, "Time-to-digital converter with Vernier delay mismatch compensation for high resolution on-die clock jitter measurement," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2008, pp. 166–167.
- J. D. Schaub, F. H. Gebara, T. Y. Nguyen, I. Vo, J. Pena, and D. J. Acharyya, "On-chip jitter and oscilloscope circuits using an asynchronous sample clock," in *Proc. IEEE Eur. Solid-State Circuits Conf.*, Sep. 2008, pp. 126–129.
- E. Gantsog, D. Liu, and A. B. Apsel, "0.89 mW on-chip jitter-measurement circuit for high speed clock with sub-picosecond resolution," in *Proc. IEEE Eur. Solid-State Circuits Conf.*, Sep. 2016, pp. 457–460.
- K. L. J. Wong, E. H. Chen, and C. K. K. Yang, "Edge and data adaptive equalization of serial-link transceivers," *IEEE J. Solid-State Circuits*, vol. 43, no. 9, pp. 2157–2169, Sep. 2008.
- B. Analui, A. Rilyakov, S. Rylov, M. Meghelli, and A. Hajimiri, "A 10-Gb/s two-dimensional eye-opening monitor in 0.13-μm standard CMOS," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2689–2699, Dec. 2005.
- H. Noguchi, N. Yoshida, H. Uchida, M. Ozaki, S. Kanemitsu, and S. Wada, "A 40-Gb/s CDR circuit with adaptive decision-point control based on eye-opening monitor feedback," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2929–2938, Dec. 2008.
- U. Singh *et al.*, "A 780 mW 4 × 28 Gb/s transceiver for 100 GbE gearbox PHY in 40 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 49, no. 12, pp. 3116–3129, Dec. 2014.
- J. Liang, M. S. Jalali, A. Sheikholeslami, M. Kibune, and H. Tamura, "On-chip measurement of clock and data jitter with sub-picosecond accuracy for 10 Gb/s multilane CDRs," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 845–855, Apr. 2015.
- M.-J. Park and J. Kim, "Pseudo-linear analysis of bang-bang controlled timing circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 6, pp. 1381–1394, Jun. 2013.
- J. Liang, A. Sheikholeslami, H. Tamura, Y. Ogata, and H. Yamaguchi, "A 28 Gb/s digital CDR with adaptive loop gain for optimum jitter tolerance," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 122–123.
- A. Samarah and A. C. Carusone, "A digital phase-locked loop with calibrated coarse and stochastic fine TDC," *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1829–1841, Aug. 2013.
- J. T. Stonick, "DPLL based clock and data recovery," in *Proc. IEEE ISSCC Tuts.*, Feb. 2011.
- S. Joshi *et al.*, "A 12-Gb/s transceiver in 32-nm bulk CMOS," in *Proc. Symp. VLSI Circuits*, Jun. 2009, pp. 52–53.
- A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*, 3rd ed. Upper Saddle River, NJ, USA: Wiley, 2008.
- J. L. Sonntag and J. Stonick, "A digital clock and data recovery architecture for multi-gigabit/s binary links," *IEEE J. Solid-State Circuits*, vol. 41, no. 8, pp. 1867–1875, Aug. 2006.



Joshua Liang received the B.A.Sc. degree in engineering science and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 2007, 2009, and 2017, respectively.

From 2009 to 2011, he was an Analog Designer with Zarlink Semiconductor (now Microsemi), Ottawa, ON, Canada, where he worked on circuits for low-jitter clock synthesis. Since 2012, he has been involved in research on wireline transceivers. In 2017, he joined Huawei, Markham, ON, Canada, where he is focussed on the research and development of high-speed interfaces.

Dr. Liang was a recipient of the Ontario Graduate Scholarship, the Ontario Queen Elizabeth II Graduate Scholarship in Science and Technology, and the NSERC Postgraduate Scholarships. He received the Analog Devices' Outstanding Student Designer Award in 2016.



Hirotaka Tamura (M'02–SM'10–F'13) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Tokyo University, Tokyo, Japan, in 1977, 1979, and 1982, respectively.

In 1982, he joined the Fujitsu Laboratories, Kawasaki, Japan. He was involved in the development of different exploratory devices such as Josephson junction devices and high-temperature superconductor devices. In 1996, he focused on CMOS high-speed signaling and was involved in the development of a multichannel high-speed I/O for server interconnects. Since 1996, he has been involved in the area of architecture- and transistor-level design for CMOS high-speed signaling circuits. Since 2014, he has been expanding his area to cover devices, circuits, and architectures for post-Moore-era computing.



Ali Sheikholeslami (S'98–M'99–SM'02) received the B.Sc. degree in electrical engineering from Shiraz University, Shiraz, Iran, in 1990, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 1994 and 1999, respectively.

In 1999, he joined the Department of Electrical and Computer Engineering, University of Toronto, where he is currently a Professor, an Associate Chair, and a Researcher. He was on research sabbatical with the Fujitsu Laboratories, Kawasaki, Japan, in 2005–2006, and with Analog Devices, Toronto, ON, Canada, in 2012–2013. He has co-authored over 70 journal and conference papers and a graduate-level textbook entitled *Understanding Jitter and Phase Noise*, and holds 10 patents. His current research interests include analog and digital integrated circuits, high-speed signaling, and very large-scale integration memory design.

Dr. Sheikholeslami has received numerous teaching awards including the 2005–2006 Early Career Teaching Award and the 2010 Faculty Teaching Award, both from the Faculty of Applied Science and Engineering at the University of Toronto. He served on the Memory, Technology Directions, and Wireline Subcommittees of the ISSCC in 2001–2004, 2002–2005, and 2007–2013, respectively. He currently serves as the Education Chair for both ISSCC and SSCS, and as a member of the SSCS Administration Committee. Being the SSCS Education Chair, he oversees the SSCS Distinguished Lecturer Program, Webinars, Circuit Contests, and other educational activities. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I during 2010–2012, and the Program Chair of the 2004 IEEE ISMVL. He is an Associate Editor of *IEEE Solid-State Circuits Magazine*, in which he has a regular column entitled *Circuit Intuitions*. He is currently a Registered Professional Engineer at Ontario.



Hisakatsu Yamaguchi received the B.S. degree in electrical engineering from the Tokyo University of Science, Chiba, Japan, in 1994, and the M.S. degree in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1996.

In 1996, he joined the Fujitsu Laboratories Ltd., Kawasaki, Japan, where he was involved in research on DRAMs with high-speed I/Os and was responsible for developing MPEG4 Codec ICs. He is currently developing high-speed I/Os for high-end servers and super-computers.

Mr. Yamaguchi served on the Technical Program Committees of ISSCC from 2012 to 2016.