A Blind Baud-Rate ADC-Based CDR

Clifford Ting, Joshua Liang, Ali Sheikholeslami, *Senior Member, IEEE*, Masaya Kibune, and Hirotaka Tamura, *Fellow, IEEE*

***Abstract*—This paper proposes a 10-Gb/s blind baud-rate ADC-based CDR. The blind baud-rate operation is made possible by using a 2UI integrate-and-dump filter, which creates intentional ISI in adjacent bit periods. The blind samples are interpolated to recover center-of-the-eye samples for a speculative Mueller–Muller PD and a 2-tap DFE operation. A test chip, fabricated in 65-nm CMOS, implements a 10-Gb/s CDR with a measured high-frequency jitter tolerance of $0.19\mathrm{UI}_{\mathrm{PP}}$ and ±300 ppm of frequency offset.**

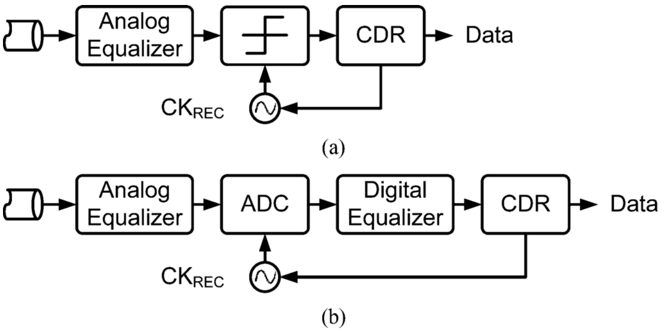
***Index Terms*—ADC-based clock and data recovery (CDR), all-digital CDR, baud-rate CDR, blind-sampling CDR, Mueller–Muller PD (MMPD).**

I. Introduction

As data rates continue to increase, the transmit signals in wireline communications are subjected to higher attenuation by legacy channels. This requires more sophisticated equalization schemes than what analog equalization is able to provide in binary receivers [see Fig. 1(a)]. In contrast, ADC-based receivers have an analog-to-digital converter (ADC) that allows additional equalization to be performed in the digital domain [e.g., Fig. 1(b)]. Digital blocks are advantageous compared with their analog counterparts because they are more robust to PVT variations, can be designed through HDL code, and are more easily ported to newer, more advanced technologies.

As shown in Fig. 1(b), an ADC-based receiver consists of an ADC, one or more equalizers, and a digital clock and data recovery (CDR). This paper focuses on a novel architecture for a digital CDR. Our work does not include channel equalization and, therefore, recovers data from low-loss channels. However, in our simulated results, we show that the digital CDR can recover data from a high-loss channel when combined with appropriate equalization.

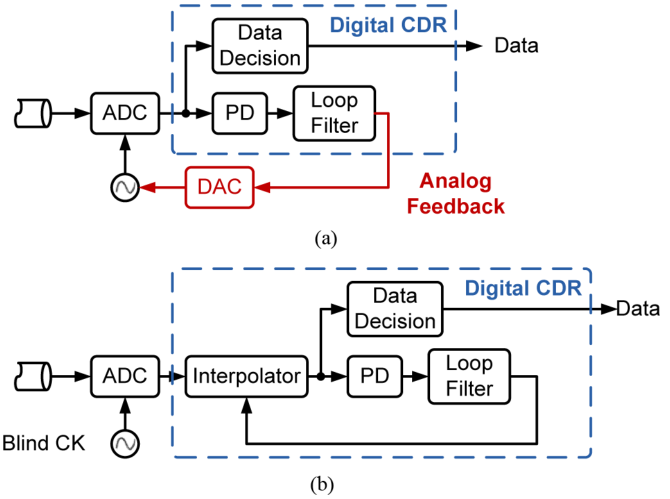
There are two types of ADC-based CDRs: phase-tracking [1]–[5] and blind [6]. In the phase-tracking architecture illustrated in Fig. 2(a), the ADC samples the received signal at the center of the data eye using digital-to-analog feedback. This is time-consuming to design because the analog and digital blocks must be simulated together to ensure the feedback loop works well. In the blind architecture shown in Fig. 2(b), the ADC samples the received signal with a local plesiochronous clock and the digital CDR extracts data from the blind samples. This eliminates the feedback loop between digital and analog domains, and the associated design complexity so that the ADC and the digital CDR can be designed and simulated independently. The digital CDR may have internal feedback, but no feedback goes to the analog blocks.



(a)

(b)

Fig. 1. Comparison of (a) binary versus (b) ADC-based receivers.



(a)

(b)

Fig. 2. Comparison of (a) phase-tracking versus (b) blind ADC-based CDRs.

In this work, we focus on blind ADC-based CDRs. Previous works [6], [7] sampled the incoming data at 2 samples per UI and 1.45 samples per UI to achieve 5 and 6.875 Gb/s, respectively. In an attempt to further increase the data rate to 10 Gb/s, we eliminate oversampling and sample at baud rate (1 sample per UI). Existing baud-rate architectures [1]–[5] rely on a phase-tracking clock to sample at the middle of the data eye. In contrast, this paper presents a blind baud-rate CDR [8] fabricated in 65-nm CMOS.

This paper is organized as follows. Section II provides the background for ADC-based sampling. Section III introduces the

Manuscript received April 12, 2013; revised June 17, 2013; accepted July 19, 2013. Date of publication September 04, 2013; date of current version November 20, 2013. This paper was approved by Guest Editor Azita Emami.

C. Ting, J. Liang, and A. Sheikholeslami are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada M5S 3G4 (e-mail: cliff@eecg.utoronto.ca; liangj@eecg.utoronto.ca; ali@eecg.utoronto.ca).

M. Kibune and H. Tamura are with Fujitsu Laboratories Ltd., 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.2013.2279023

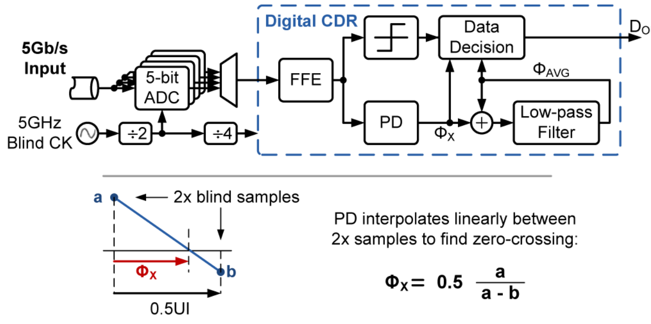


Fig. 3. Blind 2× ADC-based CDR [9].

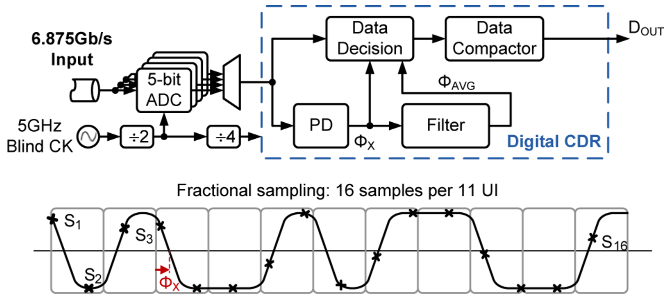


Fig. 4. Blind 1.45× ADC-based CDR [7].

receiver architecture and describes how the CDR handles frequency offset. Section IV discusses the implementation of each block. Section V presents the simulation and measurement results. Section VI summarizes the main concepts and results in the paper.

II. Background

An example of a 2× blind ADC-based CDR [6], [9] is shown in Fig. 3. A 5-Gb/s input is sampled by a 5-bit ADC and is passed to a feed-forward equalizer (FFE) in the digital CDR. After the FFE, the blind samples are processed by the phase detector (PD). If two adjacent blind samples are opposite in sign, a zero-crossing is detected which corresponds to the edge sample in a phase-tracking system. This zero-crossing, denoted by variable ϕ_X, is approximated by the linear interpolation shown in Fig. 3. The instantaneous value of ϕ_X is low-pass filtered into ϕ_{AVG} by the digital filter. The data decision block adds 0.5UI to ϕ_{AVG} to find the center of the eye and compares it to ϕ_X to recover the data. This system uses 2× sampling where the blind samples are 0.5UI apart. However, if oversampling ratio can be decreased, then the data rate can be increased without increasing the frequency of the blind clock.

A subsequent work [7], illustrated in Fig. 4, reduces the oversampling ratio to 1.45×; the receiver takes 16 samples for every 11UI to achieve 6.875 Gb/s. Its architecture is similar to the one presented in [6], but now the samples are farther apart than 0.5UI and the linear interpolation used in the PD to estimate zero-crossings is less accurate. To solve this problem, the PD filters out some of the less accurate results based on sample amplitude. With this architecture, 1.45× seems to provide a good compromise where the oversampling ratio can be reduced without much loss in jitter tolerance. In order to eliminate oversampling altogether, a different CDR architecture is required.

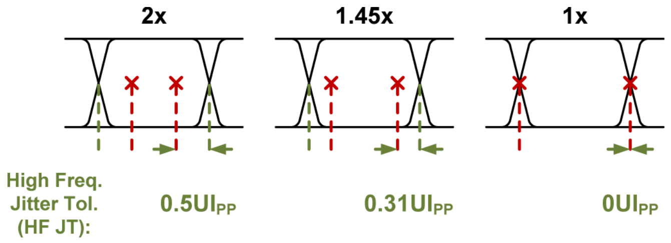


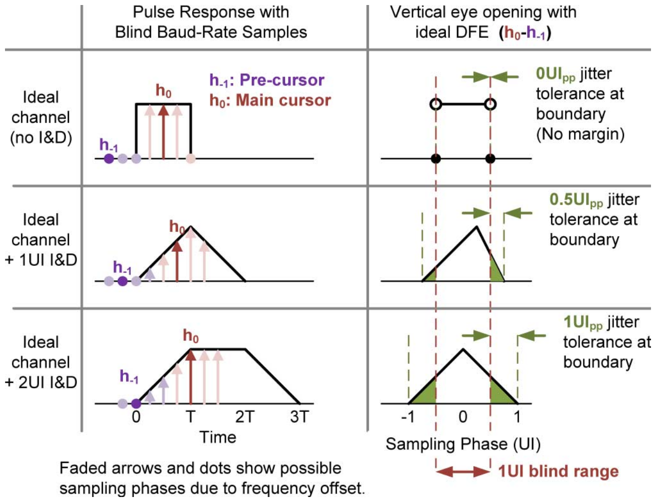
Fig. 5. Worst case for 2×, 1.45×, and 1× sampling on an open eye diagram.

The PDs in the 2× and 1.45× blind CDRs interpolate between the blind samples in order to detect the phase of the zero crossings; they require a finite slope in order to calculate phase. Given a low-loss channel, the data transitions become too sharp and, as a result, the interpolation cannot accurately estimate phase. Unlike phase-tracking CDRs, blind ADC-based CDRs perform poorly with low-loss channels. Since a blind ADC-based CDR should work with a range of channels, we focus most of our analysis on low-loss channels. In Section V, we show how the proposed CDR can recover data from a high-loss channel when combined with additional equalization.

Fig. 5 compares eye diagrams with different sampling rates given a low-loss channel. The worst-case sampling position occurs when adjacent samples are equally far from the center of the eye. For 2× blind sampling, the worst case is where adjacent samples are both 0.25UI from the edge, which leads to a high-frequency jitter tolerance of $0.5\mathrm{UI}_{\mathrm{PP}}$. When the oversampling ratio is decreased to 1.45×, jitter tolerance decreases to $0.31\mathrm{UI}_{\mathrm{PP}}$. At 1×, the samples may occur on the edges. If jitter shifts samples away from each other, then the CDR will not capture the bit at all, which results in zero jitter tolerance. In the following paragraph, we will use the channel's pulse response to elaborate on this issue and to arrive at our proposed solution.

Fig. 6 shows the pulse response of an ideal channel. The best sampling position occurs when the main cursor is at the center of the ideal pulse response. In a clocked phase-tracking system, the sampling would remain at this position. However, with 1× blind sampling, any frequency offset between the data and receiver clock will cause the sampling phase to shift continuously across a 1UI window. When the sampling occurs near the UI boundary, any high-frequency jitter may shift the sampling outside the 1UI phase range, resulting in the loss of data bits (i.e., zero jitter tolerance).

In order to increase the jitter tolerance at baud-rate sampling, we extend the pulse response beyond 1UI by introducing a controlled amount of ISI in the data using a rectangular filter, which we implement via an integrate-and-dump (I&D) circuit [10] in the receiver front end. A rectangular filter is suitable in this case since its response has a finite length of ISI and requires fewer equalization taps compared to the exponentially decaying response of an *RC* filter. A 1UI rectangular filter, convolved with the ideal channel, spreads the pulse response across 2UI. If we have a perfect decision feedback equalizer (DFE) to cancel all post-cursor ISI, then the eye would be open for a range of 1.5UI (this would have been 2UI if we could cancel precursor ISI). If the blind samples shift beyond the 1UI window, there is still a remaining jitter margin of $0.5\mathrm{UI}_{\mathrm{PP}}$. A 2UI rectangular filter


Pulse Response with Blind Baud-Rate Samples
Vertical eye opening with ideal DFE (h0-h-1)
Ideal channel (no I&D)
h-1: Pre-cursor
h0: Main cursor
h0
h-1
0UIpp jitter tolerance at boundary (No margin)
Ideal channel + 1UI I&D
0.5UIpp jitter tolerance at boundary
Ideal channel + 2UI I&D
1UIpp jitter tolerance at boundary
0
T
2T
3T
Time
-1
0
1
Sampling Phase (UI)
1UI blind range

Faded arrows and dots show possible sampling phases due to frequency offset.

Fig. 6. Comparison of theoretical worst case jitter tolerance given the pulse responses of an ideal channel, 1UI I&D, and 2UI I&D. Blind baud-rate samples can shift across a 1UI range due to frequency offset.

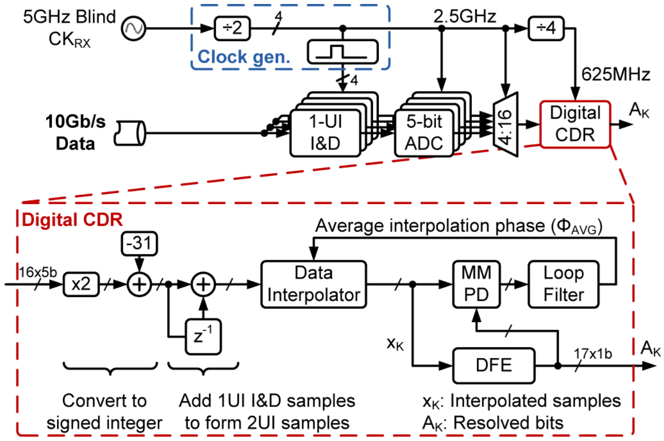


Fig. 7. System block diagram of interleaved analog front-end (1UI I&D and ADC) and digital CDR.

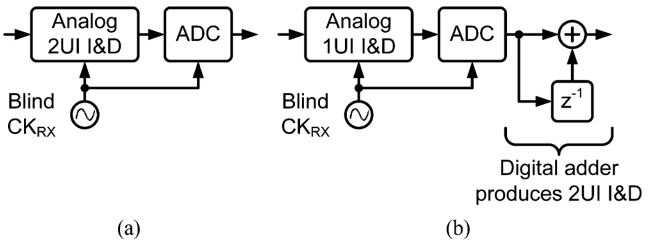


Fig. 8. Comparison of (a) fully analog 2UI I&D and (b) analog and digital 2UI I&D.

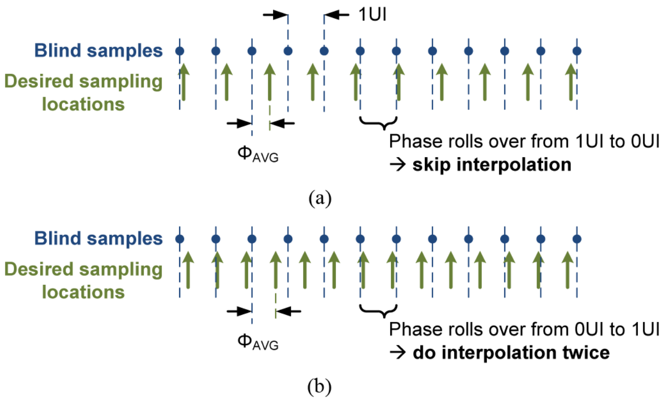

1UI
Blind samples
Desired sampling locations
ΦAVG
Phase rolls over from 1UI to 0UI → skip interpolation
(a)
Blind samples
Desired sampling locations
ΦAVG
Phase rolls over from 0UI to 1UI → do interpolation twice
(b)

Fig. 9. Handling (a) negative frequency offset: data (TX) is slower than blind receiver clock (CK_{RX}) and (b) positive frequency offset: data (TX) is faster than blind receiver clock (CK_{RX}).

increases this margin to $1\mathrm{UI}_{\mathrm{PP}}$ and results in a symmetric eye opening with respect to the blind sampling window. For these reasons, we choose a 2UI I&D circuit in our proposed design.

III. Proposed 1× Blind Receiver Architecture

Fig. 7 shows the system diagram of the receiver including an analog front-end and digital CDR. The analog front-end consists of four interleaved I&D and ADC blocks, each operating at 2.5 GS/s. Fig. 8 shows two possible implementations of a 2UI I&D. The first implementation illustrated in Fig. 8(a) is a fully analog 2UI I&D. We have chosen the second implementation [Fig. 8(b)] where the 2UI I&D consists of two components: one piece is analog and the other digital. The I&D circuit integrates 1UI samples and the ADC converts the samples into 5-bit digital values. An adder in the digital CDR combines adjacent 5-bit 1UI I&D samples to synthesize 6-bit 2UI I&D samples. Since our ADC resolution is limited to 5 bits, if we were to obtain 2UI I&D samples directly in the analog domain and feed them to the ADC, we would have lost the additional 1 bit of resolution.

Simulations showed that the system needed an ADC with a minimum ENOB of 4 bits; hence we chose a 5-bit ADC with a known ENOB of 4.2 bits [9] for our design. The proposed design does not include ADC calibration; the addition of digital calibration for gain, offset, and timing mismatches [11]–[13] would further improve the receiver performance.

The samples in the digital CDR are processed by the data interpolator, which estimates the samples at the center of the eye using the recovered phase, ϕ_{AVG}. The digital data interpolator allows us to use a more sophisticated interpolation algorithm compared to an analog interpolator [14]. A Mueller-Muller PD and loop filter form a feedback loop with the data interpolator. Loop latency is critical in this design because it degrades the stability of the feedback loop. Since the digital CDR operates on a 625-MHz divided clock, each cycle in the loop adds significant delay. Our implementation has a loop latency of seven cycles. A 2-tap DFE recovers the binary data, A_K, from the interpolated samples, x_K.

The data interpolator compensates for frequency offset. As shown in Fig. 9(a), we define negative frequency offset to mean that the transmitter clock is slower than the blind receiver clock. When this occurs, an interpolated sample is skipped each time the phase completes a 1UI rotation. Similarly, Fig. 9(b) shows a positive frequency offset where the transmitter clock is faster than the receiver clock. A positive frequency offset would result in cases where no blind sample exists between two desired samples; the interpolator resolves these cases by interpolating twice between the closest two blind samples when the

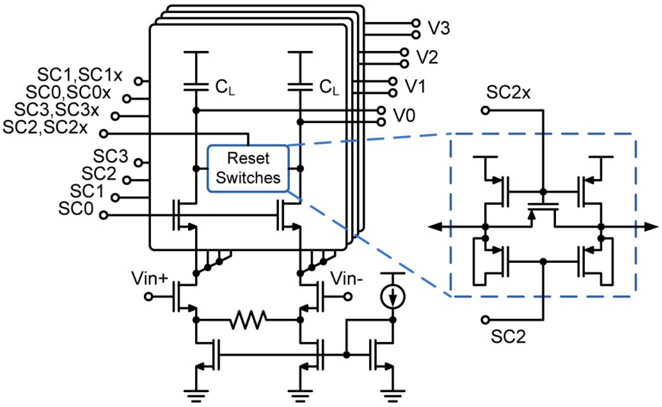


Fig. 10. Implementation of I&D circuit [10].

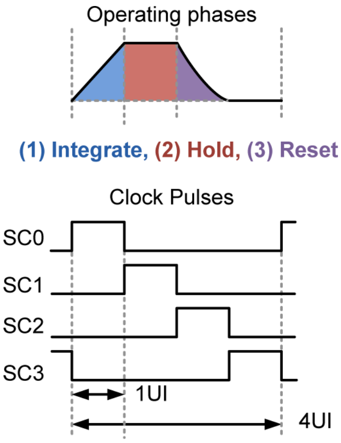


Fig. 11. I&D operating phases synchronized with clock pulses.

decreasing ϕ_{AVG} rolls over from 0UI to 1UI. The range of frequency offset supported by the loop filter is sufficiently low that we can assume the extra interpolated sample is very close to the blind sample at 1UI. Hence, our implementation directly uses the blind sample as the extra interpolated sample.

The data path in the digital CDR is sized for 17 parallel samples. Most of the time, only 16 paths are active. If there is frequency offset and ϕ_{AVG} rolls over, then the number of active paths is temporarily reduced to 15 or increased to 17 for one cycle.

IV. Receiver Implementation

A. I&D Filter

The output from the channel drives the input of the I&D filter. The I&D circuit in Fig. 10 introduces controlled ISI into the ADC input and operates as a frequency-scalable anti-aliasing filter [10]. The circuit consists of a single source-degenerated transconductance stage that converts the input voltage to current and integrates the signal on the input capacitance of the four interleaved ADCs, labeled as C_L in Fig. 10. As shown in Fig. 11, each interleaved I&D block operates in three phases: integrate, hold (during which the ADC samples the value), and reset. The clock pulses (SC0-SC3) and inverted pulses (SC0x-SC3x) reset the outputs (V0-V3) and redirect the current to each of the interleaved ADCs. Each clock pulse is 1UI wide.

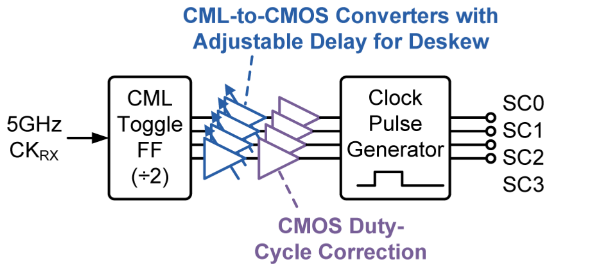

CML-to-CMOS Converters with Adjustable Delay for Deskew
5GHz CK_RX
CML Toggle FF (÷2)
Clock Pulse Generator
SC0
SC1
SC2
SC3
CMOS Duty-Cycle Correction

Fig. 12. Implementation of clock pulse generator with adjustable delay for deskew.

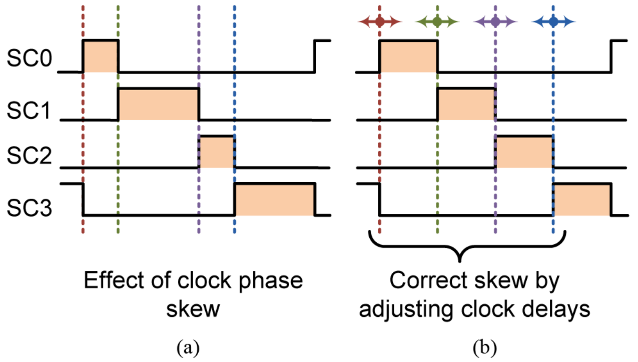

SC0
SC1
SC2
SC3
Effect of clock phase skew
Correct skew by adjusting clock delays
(a)
(b)

Fig. 13. (a) Effect of clock phase skew on the I&D integration period. (b) Equal I&D integration periods after correcting clock skew.

B. Clock Generator

Fig. 12 shows the clock generator which drives the ADC and I&D. A CML toggle flip-flop divides a 5-GHz input clock into four phases, each at 2.5 GHz. The outputs are then converted into single-ended CMOS signals and buffered. The clock pulse generator [10] uses logic gates to generate 1UI wide pulses from the four clock pulses.

Fig. 13(a) shows an example of the clock pulses when skew exists between the 4 phases. First, we note that any skew could change the integration periods when the pulses control the I&D operation. There would be gain mismatch between the four interleaved I&D blocks. Second, when we sample high-speed signals, the clock skew would appear effectively as high-frequency periodic or duty-cycle-dependent (DCD) jitter. Both the gain mismatch and high-frequency jitter will degrade the receiver's jitter tolerance. In simulation, the CDR's high-frequency jitter tolerance is reduced by approximately $0.2\mathrm{UI}_{\mathrm{PP}}$ when the clock pulse widths are 0.95UI, 1.05UI, 0.95UI, and 1.05UI, respectively.

As shown in Fig. 13(b), we compensate for skew by adjusting the clock phase through deskew circuits. In this design, the skews are manually adjusted by observing the ADC outputs (shown in Section V). Fig. 14 shows the deskew circuitry implemented in each of the CML-to-CMOS converters as a 4-bit phase interpolator. The differential clock signal connects to the In+ and In− inputs and a 20-ps delayed clock connects

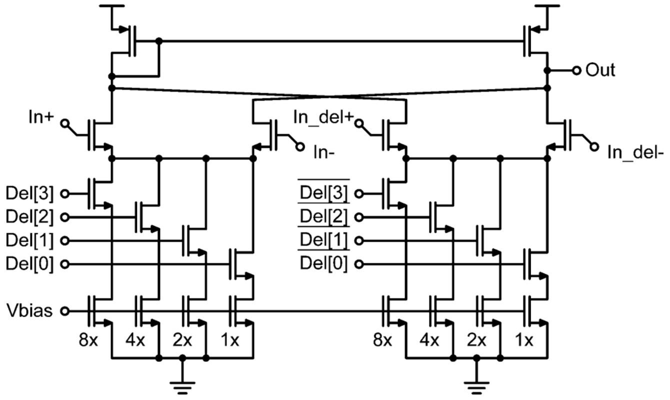


Fig. 14. Adjustable clock delay block.

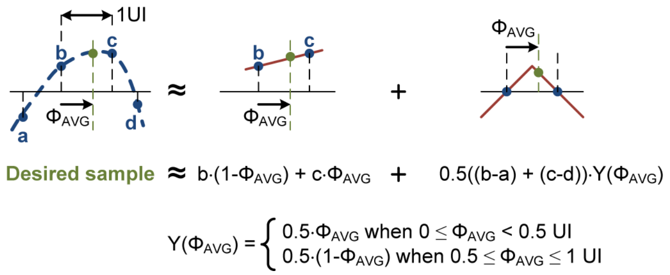

1UI
a
b
c
d
Φ_AVG
≈
+
Desired sample ≈ b·(1-Φ_AVG) + c·Φ_AVG + 0.5((b-a) + (c-d))·Y(Φ_AVG)
Y(Φ_AVG) = { 0.5·Φ_AVG when 0 ≤ Φ_AVG < 0.5 UI
0.5·(1-Φ_AVG) when 0.5 ≤ Φ_AVG ≤ 1 UI

Fig. 15. Piecewise linear interpolation of desired sample from blind samples.

to In_del+ and In_del−. Combining them achieves ±10 ps of deskew range on each of the four clock phases driving the I&D.

C. Data Interpolator

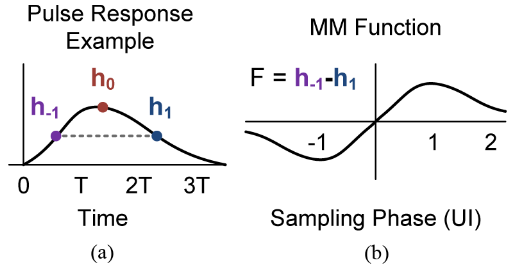
Given the ADC's blind samples and the CDR's recovered phase $\phi_{\rm AVG}$, the data interpolator estimates the value of the data at the center of the eye (i.e., the desired sample). Fig. 15 shows four consecutive blind samples a, b, c, and d that are separated by 1UI. The desired sample is $\phi_{\rm AVG}$ away from sample b. For simplicity, the expression in Fig. 15 assumes that $\phi_{\rm AVG}$ is a floating point value between 0 and 1UI. In our implementation, $\phi_{\rm AVG}$ is represented by a 5-bit value.

The desired sample is estimated first by linearly interpolating between samples b and c. This estimate has a large error because samples b and c are separated by 1UI. To improve accuracy, extrapolation is performed using the slopes $((b-a)/\ \mathrm{UI})$ and $((c-d)/1\mathrm{UI})$. We scale the piecewise linear shape in Fig. 15 by the average of the two slopes and superimpose it on the linear interpolation. Hence, the accuracy of the estimate is improved by using four instead of two blind samples.

D. Mueller–Muller Phase Detector (MMPD)

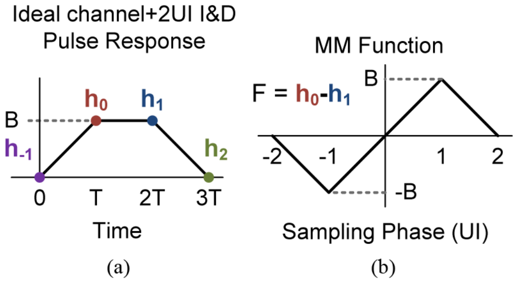
The MMPD is defined by a function we will denote as the MM function F, which should be chosen based on the pulse response of the channel. The MM function is also the transfer characteristic of the MMPD. When placed in a CDR feedback loop, the feedback forces the MM function to zero.

Fig. 16 shows an example that Mueller and Muller presented in their 1976 paper [15]. The MM function demonstrated in [15] was $F = h_{-1} - h_1$ (i.e., the difference between the pre-cursor, h_{-1}, and post-cursor, h_{+1}). Given the example pulse response shape, when the samples h_{-1} and h_1 shift to the left, h_1 becomes greater than h_{-1} and F is negative. Conversely, if the samples shift to the right, F becomes positive. When the CDR locks, the feedback forces F to zero and h_{-1} and h_1 are equal such that the main cursor, h_0, is near the optimal sampling position close to the peak of the pulse response.



(a) (b)

Fig. 16. Example of (a) pulse response and (b) MM function [15].



(a) (b)

Fig. 17. (a) Pulse response of an ideal channel followed by 2UI I&D. (b) Proposed MM function.

In this work, the 2UI I&D provides a wider pulse response such that the MM function in Fig. 16 would not provide the optimal sampling phase. If the receiver includes a DFE to cancel post-cursor ISI, the maximum vertical eye opening occurs when the main cursor, h_0, is at time T in Fig. 17 because h_0 is the maximum value of the pulse response and h_{-1} is zero. Setting the pre-cursor to zero will allow us to fully benefit from the DFE and eliminates the need for FFE. This sampling position occurs when post-cursor ISI h_1 is equal to the main cursor, h_0. To identify this desired phase location, we choose the MM function to be $F = h_0 - h_1$ [16] and force it to zero through the feedback loop. Since our actual sampling phase is blind, we force the desired phase on the interpolating phase, $\phi_{\rm AVG}$.

It can be shown [15] that the pulse response can be estimated using the samples x_K and the recovered data A_K. For convenience, we include the derivation in Appendix A. From (9) and (7), h_0 and h_1 can be estimated by the expected values, $E[x_K A_K]$ and $E[x_K A_{K-1}]$, respectively. We substitute the expected values into the MM function to transform the MM function into the MMPD. The loop filter in the next block performs the expected value operation by averaging the MMPD output.

Note that the above expressions for pulse response are not unique. For example, according to (10), h_0 is also equal to $E[x_{K-1} A_{K-1}]$. In the implementation illustrated in Fig. 18, we can therefore choose $h_0 = E[x_{K-1} A_{K-1}]$ so that A_{K-1} can be factored out of the expressions for h_0 and h_1. The DFE has some latency before it recovers A_{K-1}; factoring out A_{K-1} allows the subtraction to be performed before A_{K-1} becomes

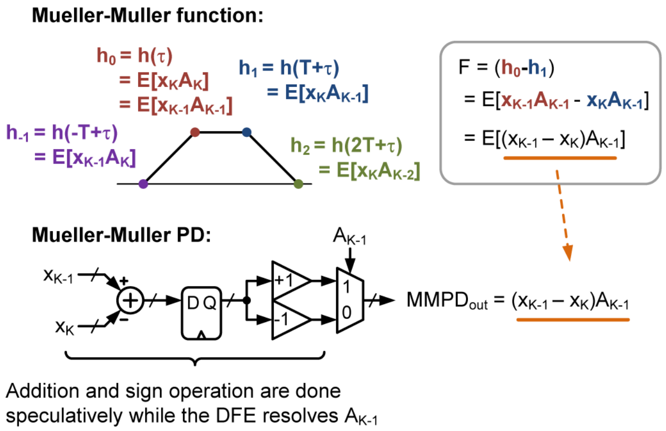
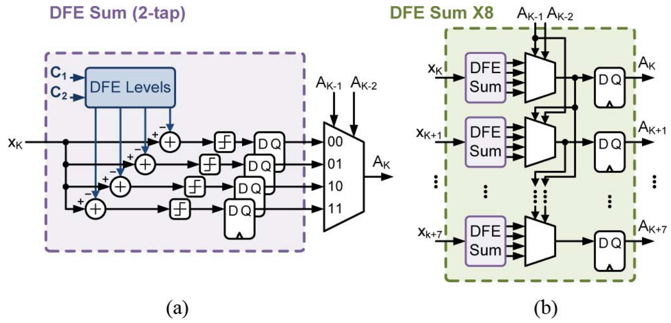

Mueller-Muller function:
h0 = h(τ) = E[xKAK] = E[xK-1AK-1]
h1 = h(T+τ) = E[xKAK-1]
h-1 = h(-T+τ) = E[xK-1AK]
h2 = h(2T+τ) = E[xKAK-2]
F = (h0-h1) = E[xK-1AK-1 - xKAK-1] = E[(xK-1 – xK)AK-1]
Mueller-Muller PD:
xK-1
xK
DQ
+1
-1
AK-1
1
0
MMPDout = (xK-1 – xK)AK-1
Addition and sign operation are done speculatively while the DFE resolves AK-1

Fig. 18. Design and implementation of the speculative MMPD.



(a) (b)

Fig. 19. (a) Speculative 2-tap DFE and (b) the first stage of the parallel speculative DFE that recovers 8 bits per cycle.

available. Since A_{K-1} takes on only two values, +1 and -1, it only affects the sign of the MMPD. In the PD implementation, subtraction is performed first and speculation is used for the sign of A_{K-1}. The DFE's recovered data and the PD output are ready at the same time, thereby reducing latency in the CDR feedback loop and improving loop stability.

E. Decision-Feedback Equalizer

The DFE compensates for post-cursor ISI from the channel and the I&D filter. As can be seen from the pulse response in Fig. 17, recovering data from an ideal channel and 2UI I&D filter would require one DFE tap to equalize post-cursor h_1, while a more attenuative channel may require more taps. Three pipelined stages, operating at 625 MHz, resolve 16 bits in parallel—actually 15 to 17 bits to handle cases of frequency offset as discussed in Section III. DFE adaptation was not included in this design.

To recover 16 bits per clock cycle, 16 parallel DFE sum blocks are required. Speculation is used extensively to reduce latency in the CDR feedback loop. In each DFE summation block shown in Fig. 19(a), the two DFE taps, C_1 and C_2, are manually set and speculation is performed by subtracting the four possible levels from the interpolated sample x_K. When the previous two bits A_{K-1} and A_{K-2} have been recovered, the mux selects the correct A_K.

This speculation removes the adder from the critical path. However, the muxes remain on the critical path since, in order to resolve all 16 bits, data must propagate through 16 muxes. However, at 625 MHz, the data can only propagate through 8 muxes per cycle. Fig. 19(b) shows eight DFE summation blocks that resolve 8 bits in one clock cycle. For this reason, we created another stage of speculation.

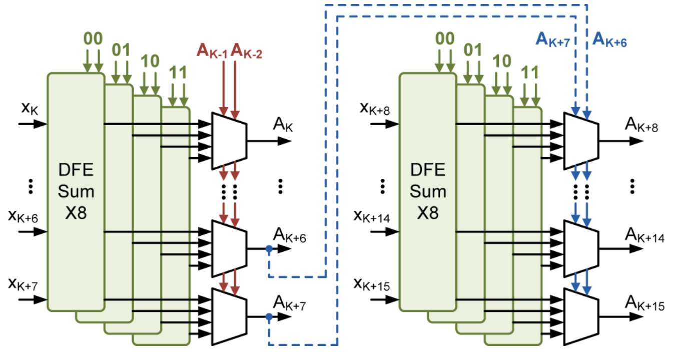


Fig. 20. Second stage of parallel speculative DFE that recovers 16 bits per cycle.

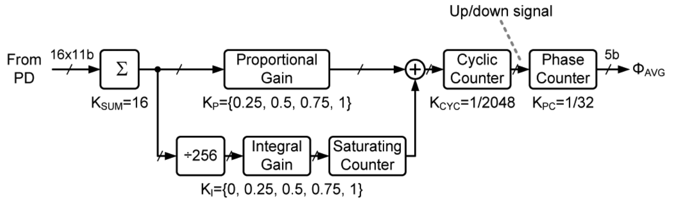


Fig. 21. Loop filter with configurable proportional and integral gains.

In the next stage, we speculate on the A_{K-1} and A_{K-2} inputs to the DFE Sum x8 blocks. As shown in Fig. 20, A_{K-1} and A_{K-2} drive the first four parallel DFE Sum x8 blocks in a speculative structure which resolve bits A_K to A_{K+7}. The last two bits A_{K+6} and A_{K+7} of this first stage then drive a second set of four DFE Sum x8 blocks which resolve bits A_{K+8} to A_{K+15}. In the end, the complete DFE has a latency of three cycles.

F. Loop Filter

The loop filter is a conventional proportional-integral controller as shown in Fig. 21. The parallel PD outputs are summed together and the result is scaled by configurable proportional and integral gains. The saturating counter is sized to handle up to ±1900 ppm of frequency offset. At the output, the 5-bit phase counter produces the recovered CDR phase as discrete ϕ_{AVG} values ranging from 0 to 31 which are fed back to the data interpolator block, closing the CDR feedback loop.

V. Simulation and Measurement Results

Here, we will show, through simulation, that the feedback loop converges correctly, how the system can be modified for a more attenuative channel, and how the system tolerates jitter. Next, we will show the measured eye diagrams and measured jitter tolerance of the proposed CDR.

Fig. 22 illustrates the loop dynamics by showing the transient signals in the loop filter. When the system in Fig. 7 starts up, it appears that the MMPD relies on correctly recovered data to estimate phase and, at the same time, the DFE requires a correct phase to recover the data. To verify that the feedback loop does not enter into a deadlock, we have applied an input

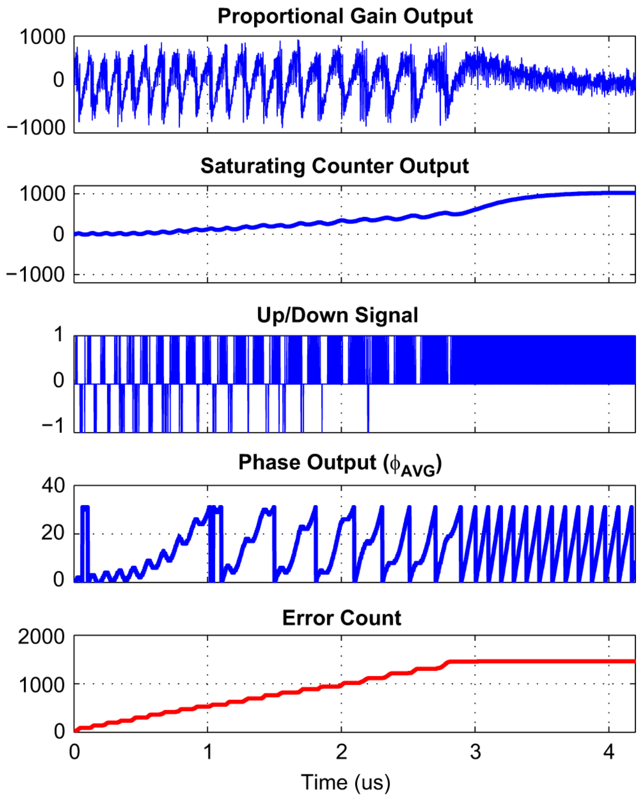


Fig. 22. Simulated loop filter convergence with 1000 ppm of frequency offset for PRBS-7. Signals correspond to nodes on the block diagram of Fig. 21.

with 1000 ppm of frequency offset so as to start the loop with both phase and data errors. The proportional gain and saturating counter outputs are, respectively, the outputs of the proportional and integral paths in the loop filter. The cycle-slipping causes the saturating counter to temporarily decrease at times, but the saturating counter settles to a value corresponding to 1000 ppm within 4 μs. The up/down signal increments or decrements ϕ_{AVG}. In steady state, ϕ_{AVG} ramps from 0 to 31 and wraps around in order to track the frequency offset. After 3 μs, ϕ_{AVG} is sufficiently close to the center of the eye to recover the data correctly (i.e., no more bit errors).

In simulation, the digital CDR has a CID tolerance of approximately 1600UIs when the input has SSC modulation with $\pm$1000 ppm of frequency offset at 32 kHz. The CID tolerance is mainly limited by low-frequency jitter from the SSC modulation and the error at the output of the saturating counter in Fig. 21 (which can be caused, for example, by noise from the MMPD).

As discussed in Section II, the receiver relies on ISI which spreads the pulse response beyond 1UI. We demonstrate through simulation that the $1\times$ blind CDR can work in two cases. In cases where the channel attenuation is low (i.e., there is not enough ISI produced by the channel), we rely on the 2UI I&D to produce the ISI. This situation is demonstrated in Fig. 23 which shows the combined frequency response of a low-attenuation Channel A followed by its associated 2UI I&D filter. In contrast, where the channel is attenuative by itself (i.e., there is enough ISI produced by the channel), we no longer need the 2UI I&D to produce extra ISI; in fact, we require equalization to reduce ISI. This situation is demonstrated by Channel B in Fig. 23. Simulations show that the $1\times$ blind CDR works in both of these cases. If the CDR will be used in applications with a wide variety of channels, then, ideally, the front-end filter should be adaptive such that it decreases the amount of post-cursor ISI generated when the channel has more high-frequency loss and, therefore, reduces the required equalization. However, an adaptive filter is beyond the scope of this work. Our test chip, which we describe later, demonstrates only the first case (i.e., low-attenuation channel with 2UI I&D).

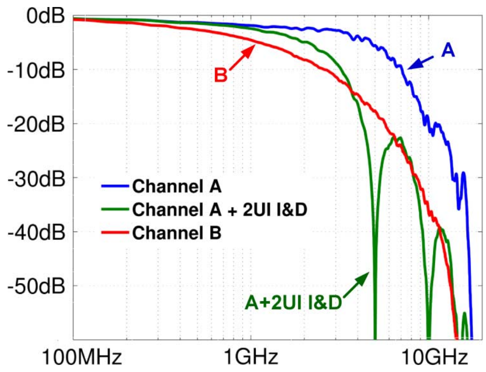


Fig. 23. Frequency response of channel models in simulation.

Figs. 24 and 25 show the eye diagrams from simulations done in Simulink using event-driven models [17]. The data source is 10 Gb/s and has 0.17UI$_{\mathrm{PP}}$ of random jitter. Similarly, the blind receiver clock is simulated with 0.23UI$_{\mathrm{PP}}$ of random jitter. The two leftmost eye diagrams in Fig. 24 show the data eye after Channel A (low attenuation) and I&D. The 5-bit ADC quantizes the samples into discrete values from 0 to 31. The eyes are still open because the analog 1UI I&D does not add much attenuation. The $1+z^{-1}$ filter adds further ISI and closes the eye. In order to obtain the eye diagrams in the digital CDR, we break the feedback loop and set ϕ_{AVG} to 0.5UI. This forces the desired sample halfway between the blind samples and the data interpolator produces the worst-case interpolation error in this condition. The open eye after the DFE adder shows that the data can be successfully recovered.

Fig. 25 demonstrates that the system can recover the data with Channel B without the I&D filter, however, it requires a 20-tap DFE. This large number of taps is necessary for Channel B because it introduces a long tail of ISI. This is not the case for Channel A with the 2UI I&D because it produces far less ISI. Alternatively, a FFE could also be used to suppress the long-tail ISI and reduce the number of DFE taps required for Channel B.

Fig. 26 compares the simulated jitter tolerance for each of the two channels. The simulation assumes a bit error rate (BER) of 10^{-6}. The high-frequency jitter tolerance of the system in Fig. 25 (Channel B) is slightly below that of the system in Fig. 24 (Channel $A+2$ UI I&D). We also note that the former has a lower CDR bandwidth compared to the latter, which is caused by a lower PD gain. Compared to Channel A, Channel B further spreads out the pulse response, which reduces the PD gain (i.e., the slope of the MM function).

We implemented the proposed receiver in Fujitsu's 65-nm CMOS process. Fig. 27 is a photograph of the test chip. The

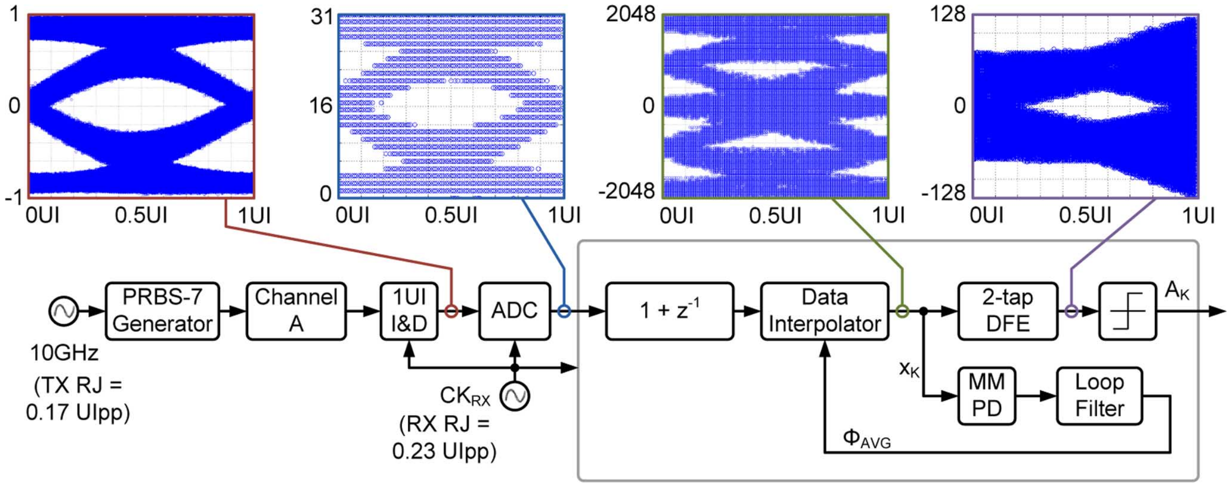


Fig. 24. Simulated eye diagrams using Channel A +2UI I&D.

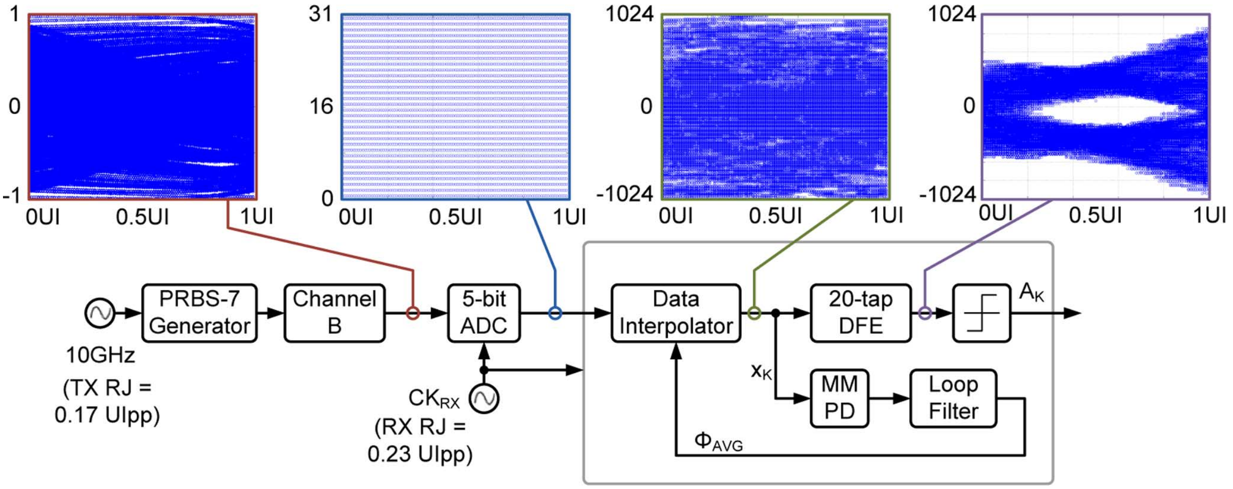


Fig. 25. Simulated eye diagrams using Channel B.

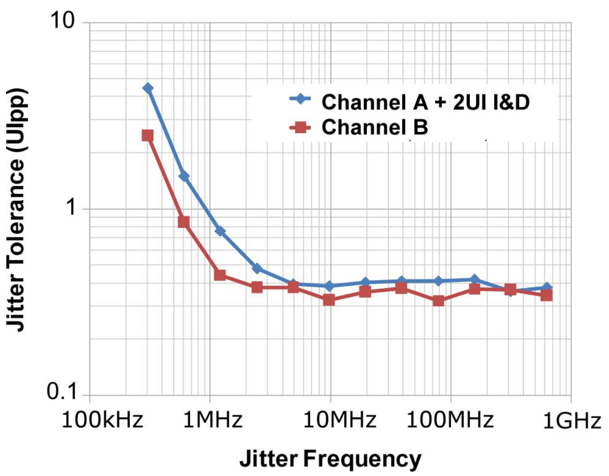
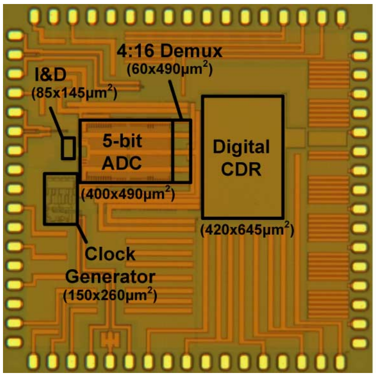


Fig. 26. Simulated jitter tolerance results.

I&D, clock generator, and ADC are custom-designed analog blocks. The digital CDR was designed using Verilog RTL and implemented with standard cell gates.



Process	65nm CMOS
Data Rate	10Gb/s
Supply	1.2V
ADC+Demux Power	109mW
CDR Power	112mW
Clock Gen. Power	83mW
I&D Power	1.7mW
Total Power	306mW

Fig. 27. Chip photograph.

Fig. 28 shows a simplified diagram of our measurement setup. The data source is a PRBS-7 generator. A logic analyzer captures and stores digital waveforms from the test chip [i.e., design-under-test (DUT)]. For jitter tolerance measurements, we apply sinusoidal jitter to the transmitter clock.

Fig. 29 shows the average ADC output when the I&D is given a DC input. On one test chip, we observed that one of the

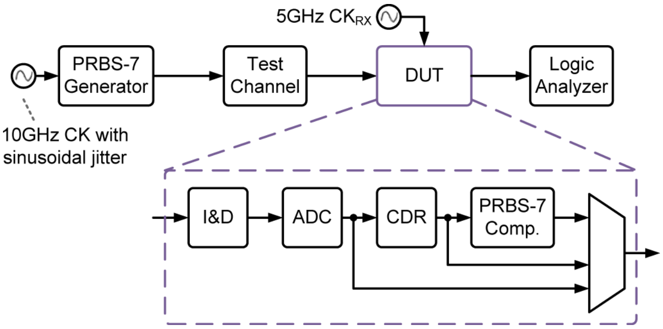


Fig. 28. Measurement setup.

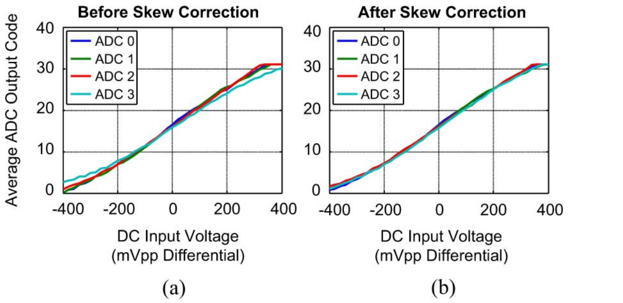


Fig. 29. Average ADC output given DC input (a) before and (b) after skew correction.

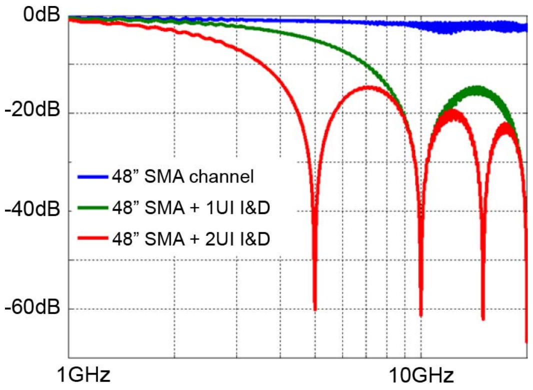


Fig. 30. Measured channel frequency response.

interleaved front end blocks had a lower gain compared with the other blocks as we varied the DC input. As discussed in Section IV, the gain error is mostly caused by systematic clock skew. If left uncompensated, the skew will reduce the CDR's jitter tolerance. Hence, we manually adjusted the delays in the clock generator. Fig. 29(b) shows that the gain at the output of ADC 3 matches more closely with gain of the other interleaved blocks after skew correction.

Our measurements were performed with a 48-in SMA cable as the channel—its frequency response is plotted in Fig. 30. Fig. 31(a) shows the data eye at the output of the channel. Fig. 31(b) shows the eye diagrams taken from the outputs of the interleaved ADCs. It has been partially attenuated by the analog 1UI I&D. There is some mismatch between the four interleaved analog front ends, but the digital CDR is able to tolerate this, as demonstrated in the jitter tolerance measurement.

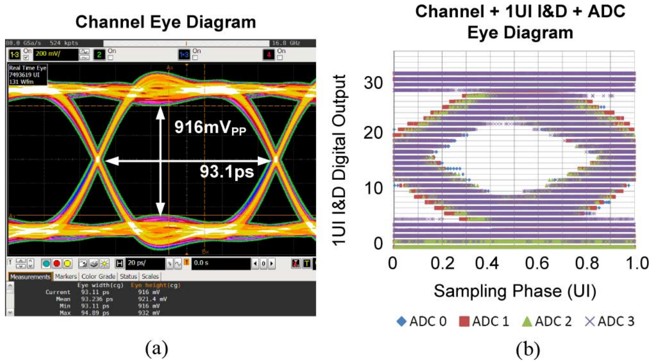


Fig. 31. Measured eye diagrams (a) after the channel and (b) after the ADC.

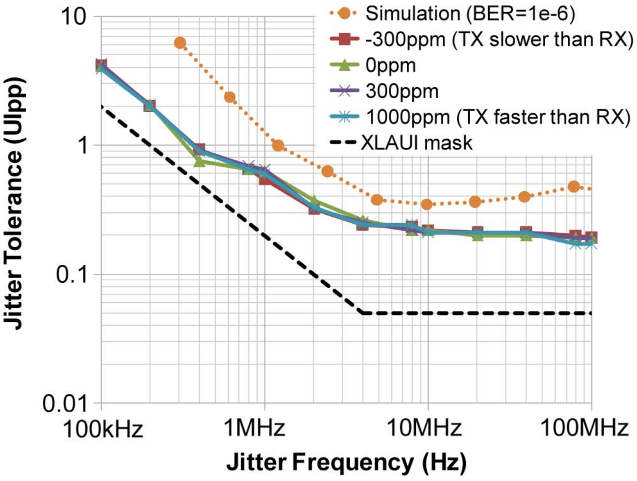


Fig. 32. Measured and simulated jitter tolerance results.

TABLE I
COMPARISON OF ADC-BASED CDRS

	[3]	[4]	[5]	This work
CDR Type	Phase-tracking	Phase-tracking	Phase-tracking	Blind
Data Rate	10Gbps	12.5Gbps	11.5Gbps	10Gbps
Technology	65nm CMOS	65nm CMOS	40nm CMOS	65nm CMOS
Power (mW/Gbps)	50*	26.4*	17**	30.6**
Samples / UI	1	1	1	<1

* Includes TX and RX
* * Includes RX only

We measured jitter tolerance after skew correction and with a maximum BER of 10^{-12} at 10 Gb/s. In Fig. 32, we show the results given -300, 0, 300, and 1000 ppm of frequency offset. A negative frequency offset means that the transmitter is slower than the blind receiver clock (i.e., above baud-rate sampling). A positive frequency offset means that the transmitter is faster than the blind receiver clock—this case is worse for jitter tolerance since we are actually sampling slightly below baud-rate. During measurement, we were able to push the frequency offset to 1000 ppm with only a slight degradation in jitter tolerance.

Fig. 32 also compares the measurement results against a simulation using the measured channel response (Fig. 30) with 300 ppm of frequency offset. Due to simulation time constraints, the simulation assumes a maximum BER of 10^{-6}. For this reason, the simulated jitter tolerance is higher compared with the measured results. We also show the jitter tolerance mask for XL-Attachment-Unit-Interface (XLAUI) in Fig. 32. Although we did not specifically target ethernet applications in the proposed design, we provide the mask as a reference.

Table I compares the proposed CDR with other baud-rate ADC-based CDRs published in [3]–[5].

VI. Conclusion

We have presented a 1× blind ADC-based CDR. In the proposed architecture, we recover data by extending the channel pulse response so that the pulse amplitude is greater than zero, no matter where the blind samples occur within a 1UI window. The receiver adds controlled ISI to the pulse response through the use of an I&D block in the receiver front end. The baud-rate design allows the CDR to operate at 10 Gb/s given a 10-GS/s sampling rate.

We fabricated the proposed design in a 65-nm CMOS process. The test chip successfully recovers 10-Gb/s data with BER below 10^{-12}. Jitter tolerance measurements show that the CDR implementation can recover data with below-baud rate sampling—the CDR operates with ±300 ppm of frequency offset and a high-frequency jitter tolerance of $0.19\mathrm{UI}_{\mathrm{PP}}$.

Appendix
Derivation of Pulse Response Samples

Let $x(t)$ be the received signal, $h(t)$ be the combined pulse response of the transmitter, channel, and receiver, x_k be the sampled signal, and A_k be the resolved bit. The data is assumed to be binary ($A_k = \{-1, 1\}$), independent, and equiprobable ($E[A_k] = 0$):

$$x_k = x(kT + \tau) \tag{1}$$

$$x(t) = \sum_{m=-\infty}^{\infty} A_m h(t - mT). \tag{2}$$

Substitute (2) into (1) yields

$$x_k = \sum_{m=-\infty}^{\infty} A_m h((k-m)T + \tau) \tag{3}$$

$$x_k = \sum_{n=-\infty}^{\infty} A_{k-n} h(nT + \tau) \tag{4}$$

$$E[x_k A_{k-1}] = \sum_{n=-\infty}^{\infty} E[A_{k-n} A_{k-1}] h(nT + \tau). \tag{5}$$

Since the data bits are independent and uncorrelated, we have

$$E[A_{k-n} A_{k-1}] = \begin{cases} 0, & \text{if } n \neq 1 \\ E[A_{k-1}^2] = 1, & \text{if } n = 1 \end{cases}. \tag{6}$$

Now, substitute (6) into (5) to obtain

$$E[x_k A_{k-1}] = h(T + \tau) = h_1. \tag{7}$$

Similarly

$$E[x_{k-1} A_k] = h(-T + \tau) = h_{-1} \tag{8}$$

$$E[x_k A_k] = h(\tau) = h_0 \tag{9}$$

$$E[x_{k-1} A_{k-1}] = h(\tau) = h_0 \tag{10}$$

$$E[x_k A_{k-2}] = h(2T + \tau) = h_2. \tag{11}$$

Acknowledgment

The authors would like to thank CMC Microsystems for CAD tools and measurement equipment and C. Sannomiya for discussions.

References

[1] H.-M. Bae, J. Ashbrook, J. Park, N. Shanbhag, A. Singer, and S. Chopra, "An MLSE receiver for electronic dispersion compensation of OC-192 fiber links," *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2541–2554, Nov. 2006.

[2] O. Agazzi, M. Hueda, D. Crivelli, H. Carrer, A. Nazemi, G. Luna, F. Ramos, R. Lopez, C. Grace, B. Kobeissy, C. Abidin, M. Kazemi, M. Kargar, C. Marquez, S. Ramprasad, F. Bollo, V. Posse, S. Wang, G. Asmanis, G. Eaton, N. Swenson, T. Lindsay, and P. Voois, "A 90 nm CMOS DSP MLSD transceiver with integrated AFE for electronic dispersion compensation of multimode optical fibers at 10 Gb/s," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2939–2957, Dec. 2008.

[3] J. Cao, B. Zhang, U. Singh, D. Cui, A. Vasani, A. Garg, W. Zhang, N. Kocaman, D. Pi, B. Raghavan, H. Pan, I. Fujimori, and A. Momtaz, "A 500 mW ADC-Based CMOS AFE with digital calibration for 10 Gb/s serial links over KR-backplane and multimode fiber," *IEEE J. Solid-State Circuits*, vol. 45, no. 6, pp. 1172–1185, Jun. 2010.

[4] M. Harwood, N. Warke, R. Simpson, T. Leslie, A. Amerasekera, S. Batty, D. Colman, E. Carr, V. Gopinathan, S. Hubbins, P. Hunt, A. Joy, P. Khandelwal, B. Killips, T. Krause, S. Lytollis, A. Pickering, M. Saxton, D. Sebastio, G. Swanson, A. Szczepanek, T. Ward, J. Williams, R. Williams, and T. Willwerth, "A 12.5 Gb/s serdes in 65 nm CMOS using a baud-rate ADC with digital receiver equalization and clock recovery," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers.* , Feb. 2007, pp. 436–591.

[5] B. Zhang, A. Nazemi, A. Garg, N. Kocaman, M. R. Ahmadi, M. Khanpour, H. Zhang, J. Cao, and A. Momtaz, "A 195 mW/55 mW dual-path receiver AFE for multistandard 8.5-to-11.5 Gb/s serial links in 40 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2013, pp. 34–35.

[6] H. Yamaguchi, H. Tamura, Y. Doi, Y. Tomita, T. Hamada, M. Kibune, S. Ohmoto, K. Tateishi, O. Tyshchenko, A. Sheikholeslami, T. Higuchi, J. Ogawa, T. Saito, H. Ishida, and K. Gotoh, "A 5 Gb/s transceiver with an ADC-based feed-forward CDR and CMA adaptive equalizer in 65 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2010, pp. 168–169.

[7] O. Tyshchenko, A. Sheikholeslami, H. Tamura, Y. Tomita, H. Yamaguchi, M. Kibune, and T. Yamamoto, "A fractional-sampling-rate ADC-based CDR with feed-forward architecture in 65 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2010, pp. 166–167.

[8] C. Ting, J. Liang, A. Sheikholeslami, M. Kibune, and H. Tamura, "A blind baud-rate ADC-based CDR," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2013, pp. 122–123.

[9] O. Tyshchenko, A. Sheikholeslami, H. Tamura, M. Kibune, H. Yamaguchi, and J. Ogawa, "A 5-Gb/s ADC-based feed-forward CDR in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 45, no. 6, pp. 1091–1098, Jun. 2010.

[10] T. Tahmoureszadeh, S. Sarvari, A. Sheikholeslami, H. Tamura, Y. Tomita, and M. Kibune, "A combined anti-aliasing filter and 2-tap FFE in 65-nm CMOS for 2x blind 2–10 Gb/s ADC-based receivers," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2010, pp. 1–4.

[11] S. Louwsma, A. J. M. V. Tuijl, M. Vertregt, and B. Nauta, "A 1.35 Gs/s, 10 b, 175 mW time-interleaved AD converter in 0.13 um CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 778–786, Apr. 2008.

[12] P. Schvan, J. Bach, C. Fait, P. Flemke, R. Gibbins, Y. Greshishchev, N. Ben-Hamida, D. Pollex, J. Sitch, S.-C. Wang, and J. Wolczanski, “A 24 GS/s 6b ADC in 90 nm CMOS,” in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2008, pp. 544–634.

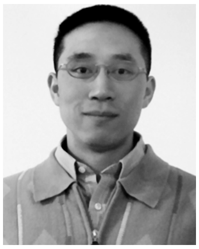
[13] S. Verma, A. Kasapi, L. m. Lee, D. Liu, D. Loizos, S.-H. Paik, A. Varzaghani, S. Zogopoulos, and S. Sidiropoulos, “A 10.3 GS/s 6b flash ADC for 10G ethernet applications,” in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2013, pp. 462–463.

[14] Y. Doi, T. Shibasaki, T. Danjo, W. Chaivipas, T. Hashida, H. Miyaoka, M. Hoshino, Y. Koyanagi, T. Yamamoto, S. Tsukamoto, and H. Tamura, “32 Gb/s data-interpolator receiver with 2-tap DFE in 28 nm CMOS,” in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2013, pp. 36–37.

[15] K. Mueller and M. Muller, “Timing recovery in digital synchronous data receivers,” *IEEE Trans. Commun.*, vol. COM-24, no. 5, pp. 516–531, May May, 1976.

[16] A. Joy, H. Mair, H.-C. Lee, A. Feldman, C. Portmann, N. Bulman, E. Crespo, P. Hearne, P. Huang, B. Kerr, P. Khandelwal, F. Kuhlmann, S. Lytollis, J. Machado, C. Morrison, S. Morrison, S. Rabii, D. Rajapaksha, V. Ravinuthula, and G. Surace, “Analog-DFE-based 16 Gb/s serdes in 40 nm CMOS that operates across 34 dB loss channels at nyquist with a baud rate CDR and 1.2 Vpp voltage-mode driver,” in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2011, pp. 350–351.

[17] M. v. Ierssel, H. Yamaguchi, A. Sheikholeslami, H. Tamura, and W. Walker, “Event-driven modeling of CDR jitter induced by power-supply noise, finite decision-circuit bandwidth, and channel ISI,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 5, pp. 1306–1315, May 2008.



Clifford Ting received the B.A.Sc. degree in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 2007, where he is currently working toward the M.A.Sc. degree in electrical engineering.

He has held the Natural Sciences and Engineering Research Council of Canada (NSERC) postgraduate scholarship and the Ontario Graduate Scholarship (OGS). His research interests are in the design of integrated circuits for high-speed chip-to-chip communications, including clock-and-data recovery blocks and equalizers.

Joshua Liang received the B.A.Sc. degree in engineering science and M.A.Sc. degree in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 2007 and 2009, respectively, where he is currently working toward the Ph.D. degree in electrical engineering.

From 2009 to 2011, he was an Analog Designer with Zarlink Semiconductor (now Microsemi), where he worked on circuits for low-jitter clock synthesis. His research is in the area of circuit design for high-speed wireline and optical communications.

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

In 1999, he joined the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, where he is currently a Professor. He has collaborated with industry on various VLSI design research in the past few years, including work with Nortel and Mosaid, Canada, and with Fujitsu Labs of Japan and America. He was a visiting researcher with Fujitsu Labs in 2005–2006, and with Analog Devices in 2012–2013. His research interests are in the areas of analog and digital integrated circuits, high-speed signaling, and VLSI memory design.

Dr. Sheikholeslami is a registered Professional Engineer in the province of Ontario, Canada. He has received the Best Professor of the Year Award four times (in 2000, 2002, 2005, and 2007) by the popular vote of the undergraduate students in the Department of Electrical and Computer Engineering, University of Toronto. He received 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, in “Recognition of Superb Accomplishment in Teaching.” He served on the Memory, Technology Directions, and Wireline Subcommittees of the IEEE International Solid-State Circuits Conference (ISSCC) in 2001–2004, 2002–2005, and 2007–2013, respectively. He currently serves on the executive committee of the same conference as its Educational Events Chair. He presented a tutorial on ferroelectric memory design at ISSCC 2002 and a tutorial on high-speed signaling at ISSCC 2008. He was an associate editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I—REGULAR PAPERS from 2010 to 2012. He was the program chair for the 34th IEEE International Symposium on Multiple-Valued Logic (ISMVL 2004) held in Toronto, Canada.

Masaya Kibune was born in Kanagawa, Japan, in 1973. He received the B.S. and M.S. degrees in applied physics from Tokyo University, Tokyo, Japan, in 1996 and 1998, respectively.

In 1998, he joined Fujitsu Laboratories, Ltd., Kawasaki, Japan, where he has been engaged in the research and design of high-speed IO with CMOS.

Mr. Kibune has been a TPC member of ASSCC since 2012.

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

He joined Fujitsu Laboratories Ltd., Kawasaki, Japan, in 1982. After being involved with the development of different exploratory devices such as Josephson junction devices and high-temperature superconductor devices, he moved into the field of CMOS high-speed signaling in 1996. His first contribution to this area was in the designing of a receiver front-end for DRAM-to-processor communications. Then, he got involved in the development of a multichannel high-speed I/O for server interconnects. Since then he has been working in the area of architecture- and transistor-level design for CMOS high-speed signaling circuits.