A 3x9 Gb/s Shared, All-Digital CDR for High-Speed, High-Density I/O

Matthew Loh, *Student Member, IEEE*, and Azita Emami-Neyestanak, *Member, IEEE*

***Abstract*—This paper presents a novel all-digital CDR scheme in 90 nm CMOS. Two independently adjustable clock phases are generated from a delay line calibrated to 2 UI. One clock phase is placed in the middle of the eye to recover the data ("data clock") and the other is swept across the delay line ("search clock"). As the search clock is swept, its samples are compared against the data samples to generate eye information. This information is used to determine the best phase for data recovery. After placing the search clock at this phase, search and data functions are traded between clocks and eye monitoring repeats. By trading functions, infinite delay range is realized using only a calibrated delay line, instead of a PLL or DLL. Since each clock generates its own alignment information, mismatches in clock distribution can be tolerated. The scheme's generalized sampling and retiming architecture is used in an efficient sharing technique that reduces the number of clocks required, saving power and area in high-density interconnect. The shared CDR is implemented using static CMOS logic in a 90 nm bulk process, occupying 0.15 mm^2. It operates from 6 to 9 Gb/s, and consumes 2.5 mW/Gb/s of power at 6 Gb/s and 3.8 mW/Gb/s at 9 Gb/s.**

***Index Terms*—All-digital CDR, calibrated delay line, clock and data recovery (CDR), eye-monitor, parallel link, per-pin synchronization, shared CDR, static CMOS logic.**

I. Introduction

CURRENT high-speed interconnects need to support very large chip-to-chip data rates, in the range of hundreds of Gb/s [1], [2]. As the number and computational power of the cores integrated on a single CPU die or package continues to increase, the demands placed on these interconnects will get more severe. However, cost and compatibility considerations limit the bandwidth of the physical channel. This suggests that a combination of bandwidth- and density-enhancing techniques is vital to implement future interconnects, which will rely on both a large number of pins as well as high data rates per pin to provide high-speed communication.

As data rates scale, process, voltage and temperature variations cause sufficient mismatch in delay between pins to require per-pin phase alignment [2]. Per-pin phase alignment also relaxes path-length matching requirements between traces, allowing greater signal densities over limited routing resources. The power and area overheads of adding dedicated CDR circuitry to each data pin in order to achieve per-pin phase alignment are prohibitive. This is particularly so for traditional analog-based techniques, which rely on VCOs, analog loop filters and charge pumps. The design space of these components is quite different from that of the digital logic comprising the rest of the system; considerable area, power, design and manufacturing overheads are necessary to accommodate these differences. Recent work on CDR has focused on more-digital techniques [3]–[6]. However, they still rely on analog components such as a voltage regulator, resistors or varactors to control the core VCO. This work proposes a true all-digital CDR scheme [7]; except for the input slicers (StrongARM latches [8]), the system is implemented entirely in static CMOS logic, and heavy use is made of standard cell blocks, automatic synthesis and place-and-route.

Traditional CDR techniques, such as Alexander's [9], rely on edge/data clock pairs to sample the incoming bit stream and extract phase information. Eye-monitor-based CDRs provide an alternative approach where the data-sampling hardware is replicated for comparison purposes, to determine BER. To date, eye-monitors have focused on two-dimensional (voltage and time) approaches for adaptive equalizers [10], [11]. Others still rely on Alexander-type CDRs, but use the eye-monitor data to fine-tune the data clock for improved error tolerance [12], [13]. Although [14] does incorporate timing recovery using only eye-monitor data, it requires external PC-based adaptation. Other work has investigated the use of eye-monitors for off-line channel characterization [15].

This work focuses specifically on timing recovery, and conducts one-dimensional eye-monitoring (time only), avoiding the overhead of a variable-threshold sampler. An important innovation is the use of "ping-pong" clocks; the data and eye-monitor functions are swapped between clocks during updates of the data phase. This confers important advantages: it insulates against mismatch between the phases of the clocks and allows the realization of an infinite delay range using only a loosely-calibrated delay line, instead of a PLL or DLL. Additionally, the search technique used is designed specifically for efficient on-chip implementation, in contrast to the PC-based control of [14].

Since this CDR uses a statistical approach to finding the eye before updating the data sampling position, it has relatively low bandwidth and is not appropriate for use in links requiring large (hundreds of ppm) frequency offset tolerance. However, it can be applied in mesochronous or source-synchronous links. Although these links share a common system clock and have a

Manuscript received June 02, 2011; revised October 17, 2011; accepted November 10, 2011. Date of publication January 16, 2012; date of current version February 23, 2012. This paper was approved by Associate Editor Jared Zerbe. This work was supported by the National Science Foundation, Intel, and the C2S2 Focus Center, funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation subsidiary.

The authors are with the California Institute of Technology (Caltech), Pasadena, CA 91125 USA (e-mail: rui@caltech.edu).

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.2011.2178557

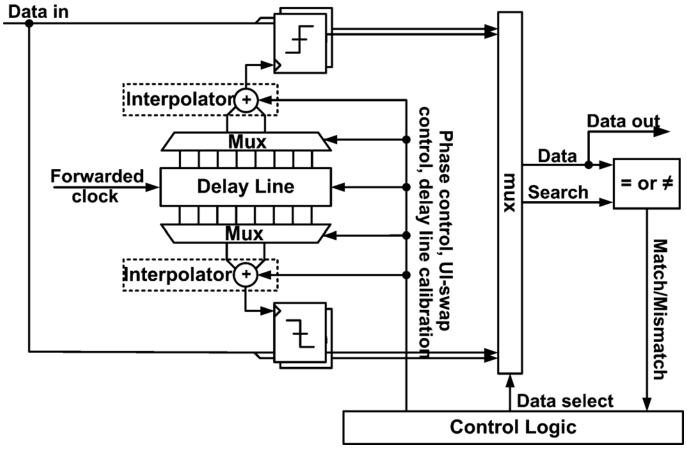


Fig. 1. Single-pin system architecture overview.

degree of correlated jitter and frequency offset tracking [16], transmission path length mismatch, in addition to variability and noise in clock multiplication/distribution, can create the need for full CDR with infinite delay range. The low CDR bandwidth requirements of such links encourage sharing techniques that allow a single set of CDR hardware to calibrate multiple pins. The proposed CDR is particularly suited for sharing, since the sampling and re-timing hardware required for the ping-pong clocks can be reassigned between pins.

II. ALL-DIGITAL CDR

Traditional approaches to CDR [9] use two clocks to sample the incoming signal. The first clock is used to sample the data, while the second clock is used to sample the edge of the eye, and is fixed at 90° phase offset from the first. In eye-monitor-based CDR, the second clock's phase is decoupled from the first and allowed to move independently. This work inherits the decoupled phase inherent to eye-monitoring schemes, with both clocks free to move at discrete intervals ("phase positions") within a 2 UI delay. One clock is placed in the middle of the eye to recover data ("data clock"), the other is swept across the 2 UI delay ("search clock"). Eye information is collected by comparing the samples produced by these clocks. This is used to determine the best phase position for data recovery. The search clock is placed at this phase position, the search and data functions are traded between clocks, and the algorithm repeats.

This implementation targets a source-synchronous link, and a delay line is used to generate these two clocks from the forwarded clock (Fig. 1). This delay line is slowly and digitally calibrated to achieve approximately 2 UI delay (as explained below, exact calibration is unnecessary). Adjacent output phases of the delay line are selected and interpolated independently for each clock. A multiplexer following the input slicers routes their outputs to either "search" or "data". Synthesized digital logic aggregates this data to determine the location of the eye opening and controls the movement and swapping of the clocks.

The samples generated by the search clock are compared with those produced by the data clock. Where these samples match, the eye is open. Conversely, a mismatch between these samples indicates that the eye is closed. As the search clock is swept through the 2 UI delay line, match/mismatch information is collected at each phase position. The collated information can be thought of as a binary reduction of an eye diagram (Fig. 2). Transitions between match and mismatch correspond to the edges of the eye, so the control logic can use these transitions to place the search clock at the mean of the detected eye edges, maximizing timing margin. No assumption is made that the optimum sampling point is 90° away from the edge. The functionality of the search and data clocks is then traded and the algorithm repeats. Trading off between the search and data clocks allows the CDR to realize an infinite delay range. When the eye opening begins to drift off the extent of the delay line, the CDR can place the search clock in the middle of the following or preceding eye opening and invert it before trading the data and search clock functionality. In a half-rate architecture, inverting the search clock before the functionality swap allows the CDR to skip backward (in the inverse scenario, forward) a UI without introducing errors such as added or dropped bits in the data (Fig. 3).

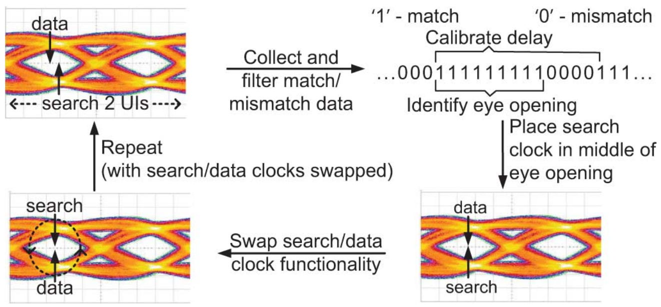

data
search 2 UIs
Collect and filter match/ mismatch data
'1' - match
'0' - mismatch
Calibrate delay
...0001111111110000111...
Identify eye opening
Place search clock in middle of eye opening
data
search
Swap search/data clock functionality
search
data
Repeat (with search/data clocks swapped)

Fig. 2. Eye-monitoring CDR algorithm.

Match/mismatch data is generated by sweeping the search clock's phase, so the eye information is based on the actual phase shift introduced by the search clock's own phase generation and distribution path. Since the search clock uses this information to determine its sampling point when it becomes the data clock, accurate data clock placement is not dependent on matching with any other clock's phase generation/distribution. This is particularly important in a multi-pin environment, where many clocks are required and matching between their paths becomes prohibitively difficult.

A. Search Algorithm

In normal operation, it is unnecessary to search the entire 2 UI delay in order to update the data clock. Instead, operation is hastened by stopping the search once two edges bounding a single eye opening are detected (marked 1 and 2 in Fig. 4). The search clock movement described in Fig. 4 minimizes the time to find these bounding eye edges. A more complete search is only conducted when the eye opening begins to drift off the extent of the delay line, or if a delay line calibration is requested. For instance, the phase offset between clock and data may be gradually increasing (Fig. 5) due to a frequency offset or large jitter transient. As data drifts to the right, the CDR tracks it and updates the data phase accordingly. When the data drifts far enough, one edge of the current eye moves off the end of the delay line and cannot be found. The CDR then searches for eye

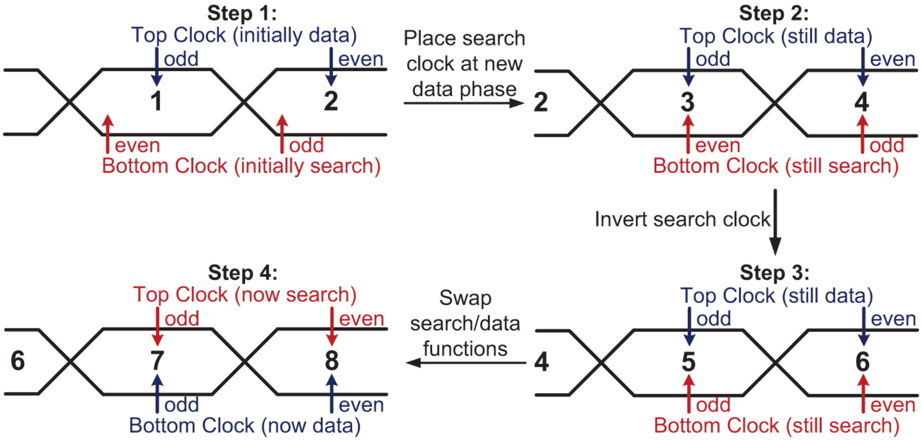


Fig. 3. Steps in a UI swap process. Even (rising) and odd (falling) edges of clocks marked. In Step 1, the search process has just finished and a UI swap is required. In Step 2, the even (rising) edge of the search clock is placed in the previous UI ("3"). To line the even and odd edges up, the search clock is inverted in Step 3. Once this is complete, the clock functions can be interchanged without added/dropped bits (Step 4).

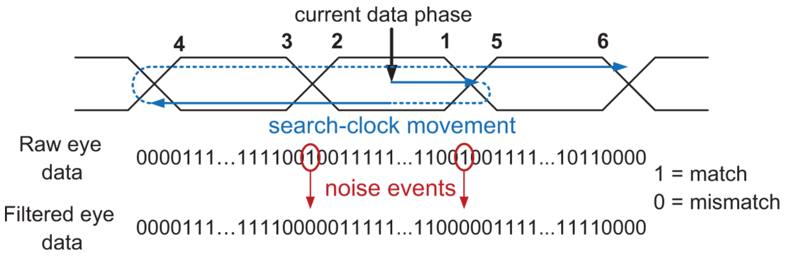


Fig. 4. Search procedure, showing movement of search clock and numbered eye edges. Filtering removes noise events and allows detection of eye opening.

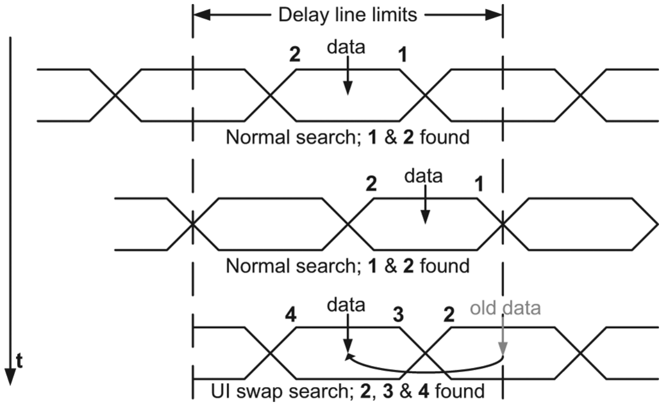

Delay line limits
2
data
1
Normal search; 1 & 2 found
2
data
1
Normal search; 1 & 2 found
4
data
3
2
old data
UI swap search; 2, 3 & 4 found
t

Fig. 5. CDR operation with eye drifting until UI swap is required. Relevant eye edges are marked and numbered.

edges 3 and 4 to acquire the preceding eye opening. It places the search clock in the middle of this eye opening, inverts the search clock, then trades the data and search functionality, completing the data clock update.

A longer search is also required to calibrate the delay line. In this case, the control logic seeks to establish the length of 1 UI in terms of phase positions, then adjust the length of the delay line such that 1 UI of delay occupies half the available phase positions (hence the overall delay will be 2 UI). This information is obtained by extending the search one further eye edge—to 3, or 5 if 3 is not found. The distance between eye edges 3 and 1 or 2 and 5 corresponds to the length of 1 UI. Exact calibration of the delay line length is unnecessary, since the algorithm will never need to search through a full 2 UI; in the worst case it only needs sufficient range to find the preceding or following eye opening when the present eye opening has begun to drift off the delay line. This calibration process can therefore proceed loosely and slowly, so it has limited impact on the overall bandwidth of the CDR.

B. *Search Filtering*

An important consideration is the collection of consistent and reliable match and mismatch information. Data-dependent phenomena such as ISI and transient events such as noise spikes may cause spurious match/mismatch decisions, corrupting the detection of the eye opening. These effects are suppressed through the use of a mismatch counter, which acts as a pre-filtering averager, and an AND/OR filter.

The mismatch counter observes the incoming data stream; when a transition occurs, it makes a comparison between the corresponding search and data samples. Conducting averaging over n such transitions, the probability that at least one of the collected comparisons is conducted on dissimilar search and data samples, therefore that phase position i will be declared a mismatch, is given by a geometric distribution:

$$P(\text{mismatch}_i) \approx 1-(1-\text{BER}_i)^{n/\rho} \tag{1}$$

where BER_i is the bit-error rate at the ith phase position and ρ is the average transition density. Ideally, the search should transition from generating matches to mismatches at a consistent phase position, tracking the eye opening if it moves. Consider the case where the search is moving downwards from the data phase (from a higher to lower phase position) and from generating matches to generating mismatches. In this case, the probability that phase position i will be the first mismatch generated can be defined recursively:

$$P(\mathrm{firstMismatch}_i) \approx \left(1 - \sum_{j=i+1}^{d} P(\mathrm{firstMismatch}_j)\right) P(\mathrm{mismatch}_i). \quad (2)$$

For $i < d$, where d is the current data phase position, and it is assumed that $P(\mathrm{firstMismatch}_d) = 0$. The most consistent results are produced when the distribution generated by (2) is tightest—ideally, concentrated on a single phase position. This suggests that slope of the bathtub produced from (1) should be as steep as possible, which can be achieved by making n large. However, large values of n result in long averaging times at each phase position, reducing the bandwidth of the CDR.

A more reasonable approach is to semi-dynamically size n, collecting more samples only in ambiguous cases. To this end, a new averaging period, n_{base}, can be defined. If search and data clocks produce two or more dissimilar samples in the first n_{base} transitions, the mismatch counter will immediately declare the phase position a mismatch. An ambiguity occurs when there is only one discrepancy between search and data samples in the first n_{base} transitions, in which case the mismatch counter will collect samples over a further n_{base} transitions. If another discrepancy occurs, it will then declare a mismatch. With this ability to repeat the search, the probability of a mismatch declaration at phase position i becomes

$$P(\mathrm{mismatchWithRepeat}s_i) = P(\mathrm{outrightMismatch}_i) + P(\mathrm{repeatMismatch}_i) \quad (3)$$

$$P(\mathrm{outrightMismatch}_i) = \sum_{j=0}^{n_{\mathrm{base}}/\rho} \binom{1+j}{j} \mathrm{BER}_i{}^2 \times (1 - \mathrm{BER}_i)^j \quad (4)$$

$$P(\mathrm{repeatMismatch}_i) = P(\mathrm{mismatch}_i) \times \big(P(\mathrm{mismatch}_i) - P(\mathrm{outrightMismatch}_i)\big) \quad (5)$$

$P(\mathrm{mismatchWithRepeats}_i)$ can be substituted in place of $P(\mathrm{mismatch}_i)$ in (2) to obtain the distribution of the first mismatch using the modified method. Fig. 6 presents the probabilities of mismatch declaration for $n = n_{\mathrm{base}} = 32$. A plot of the peak probability values in the first mismatch distribution for reasonable values of n_{base} and n (Fig. 7) suggests an optimum of $n_{\mathrm{base}} \approx 32$, the value selected for this implementation. The search can be further hastened by declaring a mismatch immediately once the required number of discrepancies between search and data have occurred, instead of waiting for all n_{base} transitions.

Subsequent to the averaging, an AND/OR filter is used to suppress the presence of subsidiary "false eyes" that might result from reflections in the channel, crosstalk or large transient noise events. The previous k match/mismatch declarations are ANDed to eliminate spurious matches, and the output of the AND is then ORed to restore the original eye opening size (Fig. 8). k defines the minimum eye-opening size the CDR is expected to track, and should be set high enough to reject false eyes, but small enough to maintain sensitivity. Exact selection of k is not performance-critical, and a value of 4 is chosen for this implementation. The k-decision latency introduced by the AND/OR filter can be accounted for in the control logic.

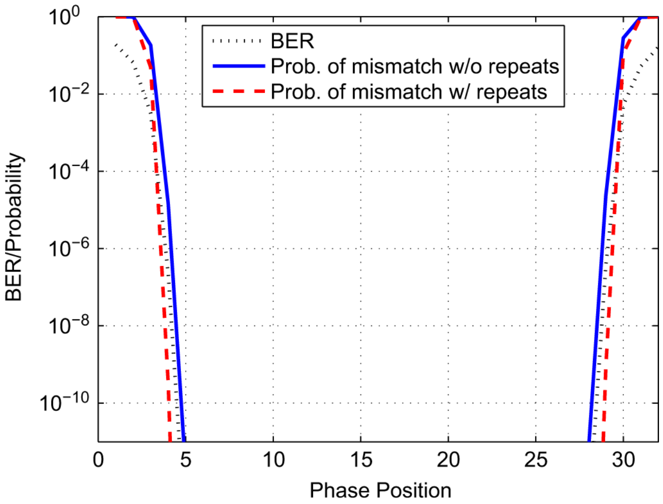


Fig. 6. Typical BER bathtub, and the probability of mismatch declaration at each phase position with repeated averaging ($n_{\mathrm{base}} = 32$) and without repeated averaging ($n = 32$).

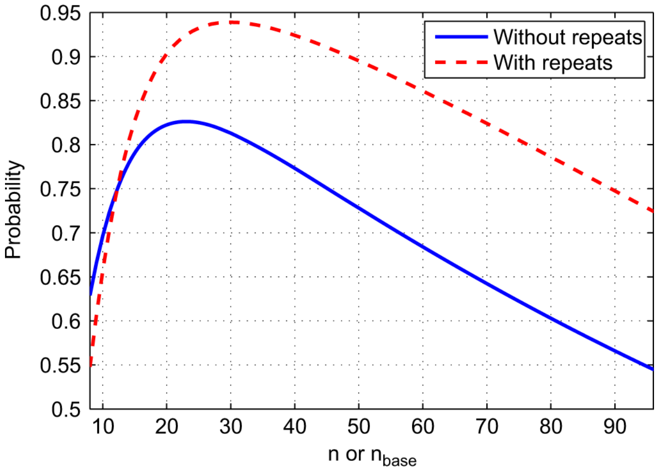


Fig. 7. Peak probability in distribution of first declared mismatch, with and without repeated averaging.

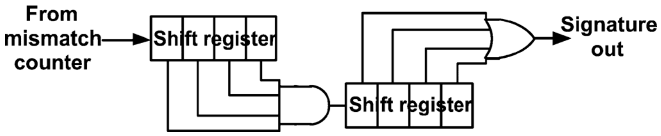
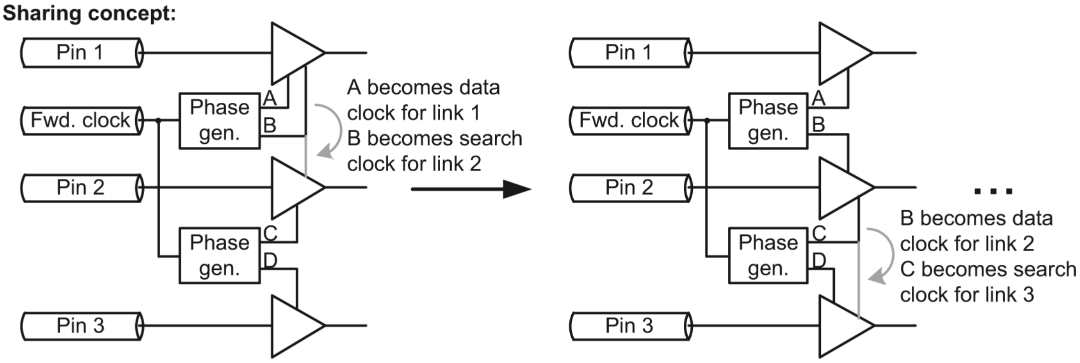


Fig. 8. AND/OR filter with $k = 4$.

III. Shared CDR

The independent adjustability of each clock and the generalized sampling and re-timing paths of the proposed CDR allow it to be easily adapted to a shared multi-pin system. Instead of trading search/data clock functionality on a single pin, the search clock can "bubble" through multiple pins (Fig. 9). Only


Sharing concept:
Pin 1
Fwd. clock
Phase gen.
A
B
Pin 2
Phase gen.
C
D
Pin 3
A becomes data clock for link 1
B becomes search clock for link 2
Pin 1
Fwd. clock
Phase gen.
A
B
Pin 2
Phase gen.
C
D
Pin 3
...
B becomes data clock for link 2
C becomes search clock for link 3

Modified Draft:

Repeats

Clock	Clock assignments at each step					
	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
A	**Search, pin 1**	Data, pin 1	Data, pin 1	Data, pin 1	Data, pin 1	Data, pin 1
B	Data, pin 1	**Search, pin 2**	Data, pin 2	Data, pin 2	Data, pin 2	**Search, pin 1**
C	Data, pin 2	Data, pin 2	**Search, pin 3**	Data, pin 3	**Search, pin 2**	Data, pin 2
D	Data, pin 3	Data, pin 3	Data, pin 3	**Search, pin 3**	Data, pin 3	Data, pin 3

Fig. 9. Sharing concept and algorithm.

one pin is calibrated at a time, so only one extra clock is necessary. Thus, for N pins only $N + 1$ clocks are required, instead of the $2N$ required without sharing.

The pins are calibrated in sequence, with the calibrated pin's data clock swapping with the search clock at each step. The most equitable algorithm would be analogous to a standard draft; each pin would be calibrated in sequence, from first to last, and the calibration would then return to the first pin. The time between calibrations of a particular pin would thus be N complete search and data phase update periods. However, the standard draft requires each clock to successively sample all of the pins, making its hardware cost prohibitive—it multiplies the number of input slicers required, complicates the input routing and calls for larger high-speed multiplexers to route the search and data samples. While this hardware overhead may be reasonable for small N, it does not scale well to large numbers of pins.

To avoid this constraint, a modified draft algorithm is used. Instead of returning to the first pin after the last pin has been calibrated, the modified calibration sweeps back-and-forth through the pins (Fig. 9). It is less equitable and results in a $2N-1$ period gap between calibrations of any one pin, but requires each clock to sample no more than two adjacent pins and is therefore more hardware efficient, scaling well to large N. This draft scheme is well-suited for dense source-synchronous environments, where CDR bandwidth requirements are low and reducing hardware overhead is paramount.

IV. Implementation

A block-diagram overview of the implemented system is presented in Fig. 10. To save power and ease design constraints, the mismatch counter operates on quarter- and eighth-rate clocks, while the AND/OR filter, eye detection, clock phase placement and multiplexer control logic operates on a distinct low-frequency clock. All are synthesized from standard cells. The high-speed phase generation, slicing and multiplexing circuitry operates on a half-rate clock and is custom digital. As much as possible was implemented using static CMOS logic.

As the phase generator architecture used (described in Section IV-A) naturally generates clocks in pairs, and $N + 1$ clock phases are needed (one for each link's data clock, plus a bubbling search clock), an odd number of pins is called for. In this case, a three-link system is implemented to allow the performance of the shared CDR to be fully evaluated and extrapolated to wider links.

A. Phase Generator

The core of the phase generator (Fig. 11) is a direct digitally-modulated, differential delay line (Fig. 12), with nine evenly-spaced output phases. Each cell of the delay line [Fig. 12(a)] is composed of tri-state buffers, which can be turned on or off to adjust the drive strength of each stage, thus the overall delay of the line [17]. Weak cross-coupled inverters are placed at the output of each cell to maintain phase alignment between the differential paths and duty cycle. This scheme has the advantage of allowing an adjustable delay line implementation in pure static CMOS. However, the array of tri-state buffers and the wiring necessary to connect them imposes significant extra loading on the output of each delay cell, thus limiting the practical resolution of the delay adjustment. To overcome this drawback, the output of each delay cell is fed-forward to the calibration input ($\mathrm{cal}/\mathrm{cal}_\mathrm{b}$ in Fig. 12) two cells away [18], thus reducing the size of the tri-state buffers necessary to achieve a large delay range.

An important consideration is the consistency of the phases of the output clocks when the delay line length is changed. These clocks are generated by interpolation of the outputs of a delay line; if the delay line length is changed abruptly, the phase of its outputs will likewise jump, thus causing a deviation in the phase of the generated clocks which could result in errors in CDR tracking. This effect is particularly severe when the outputs near the end of the delay line are being used to generate the clock,

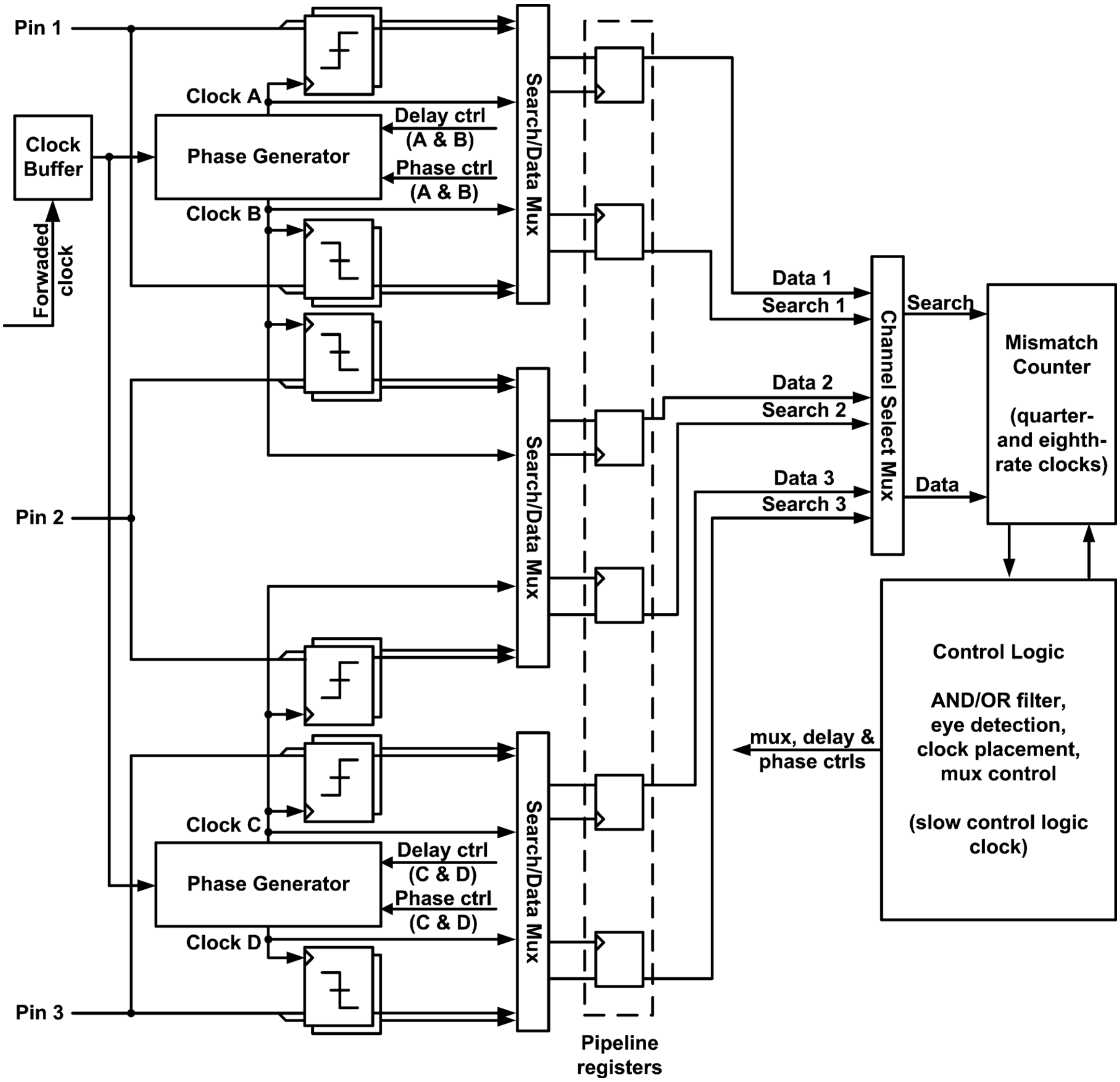


Fig. 10. Three-pin system architecture.

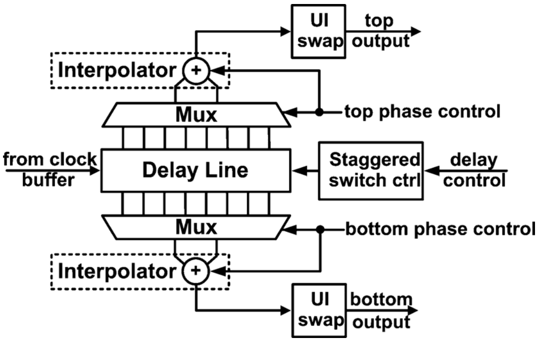


Fig. 11. Phase generator architecture.

since the accumulated change in delay is largest at this point. To minimize this effect, the delay control code is updated in a stepwise manner, with hysteresis added to ensure that small changes in the number of phase positions per UI do not result in control code dithering. Additionally, the delay line is split into four sections of two delay cells each, with the delay updated section by section. The slow calibration of the delay line allows the updates to each section to be staggered across several data phase updates; the phase effect of the delay update is therefore spread out, and the CDR only has to deal with it incrementally. In simulation, staggering reduces the phase discontinuity per data phase update from 6 ps to 2.5 ps, or 0.7 phase positions at 9 Gb/s.

Two adjacent output phases of the delay line are selected via a multiplexer and interpolated to generate finer granularity. The phase interpolator itself (Fig. 12(b)) is composed of a pair of tri-state buffer arrays with shorted outputs; the interpolation ratio is controlled by turning portions of these arrays on or off, while maintaining a constant total number of active tri-states, thus ensuring a consistent output drive. Since there are 8 possible pairs of adjacent output phases from the delay line, and 8 settings of the phase interpolator, the complete phase generator has 64 total output phases, for an overall phase adjustment resolution of 6 bits.

The linearity and resolution of the phase generator affects the final accuracy of the data clock placement by the CDR algorithm. Let Φ_1 and Φ_2 be the detected locations of eye edges 1

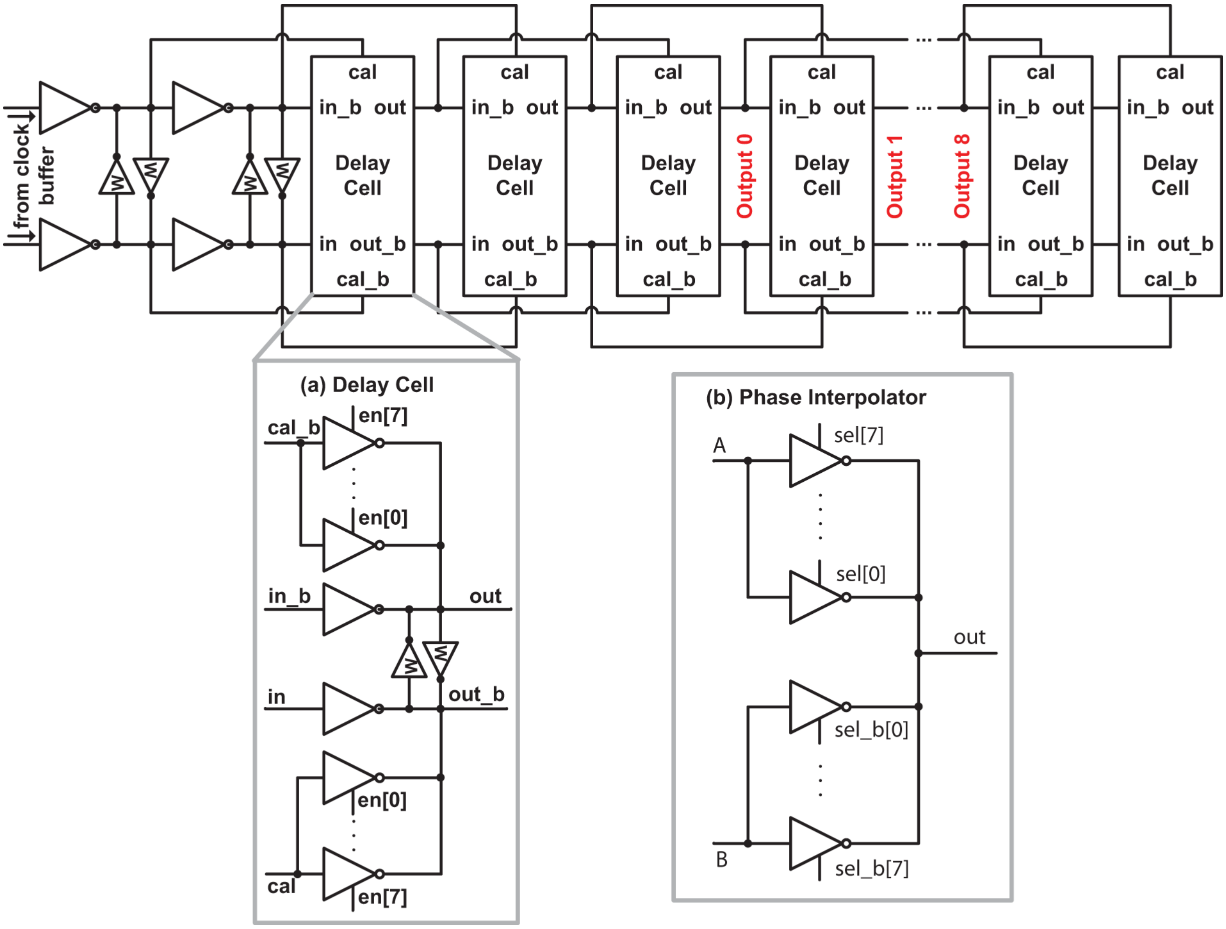


Fig. 12. Delay line with (a) delay cell and (b) phase interpolator. Weak cross-coupled inverters are marked with a W.

and 2, respectively (Fig. 4). Φ_1 and Φ_2 are the phase positions bounding the detected eye opening, and are used to determine the next data phase. The phase generator has limited resolution and could introduce nonlinearity, so there is some error in Φ_1 and Φ_2 relative to the actual positions of eye edges 1 and 2 (Φ_1' and Φ_2'):

$$\Phi_1 = \Phi_1' + \epsilon_1, \quad \Phi_2 = \Phi_2' + \epsilon_2. \tag{6}$$

where the worst-case error in terms of phase position can be written by observing that it is affected by the resolution of the phase generator and its worst-case DNL:

$$\epsilon_1 = \epsilon_2 = \frac{(1 + \mathrm{DNL}_{\mathrm{worst}})}{2}. \tag{7}$$

The algorithm will place the next data phase at the average of the two phase positions:

$$\Phi_{\mathrm{data}} = \frac{(\Phi_1 + \Phi_2)}{2} = \frac{(\Phi_1' + \Phi_2' + \epsilon_1 + \epsilon_2)}{2}. \tag{8}$$

Finally, the placement of the data phase itself will be affected by the INL between Φ_1 and Φ_2, thus yielding an overall worst-case error, in terms of phase positions, of

$$\begin{aligned}\epsilon_{\mathrm{data}} &= \frac{(\epsilon_1 + \epsilon_2)}{2} + \mathrm{INL}_{\mathrm{worst}} \\ &= \frac{(1 + \mathrm{DNL}_{\mathrm{worst}})}{2} + \mathrm{INL}_{\mathrm{worst}}. \end{aligned} \tag{9}$$

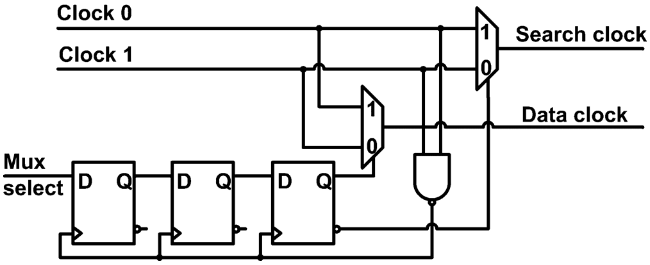


Fig. 13. Search/Data multiplexer (only clock routing shown; sample routing is similar).

B. Multiplexers

As the control logic is shared between multiple links, a key implementation challenge is the design of a high-speed multiplexer tree to route the search, data and clock signals of the pin-under-calibration to the control logic.

The search/data multiplexers must be able to change the data clock of each pin without introducing errors. This is accomplished by delaying changes in the select signals until the input clocks to each search/data multiplexer are both high (Fig. 13). This ensures that the swap is made when no transition is occurring in either the clock or data inputs. However, the flip-flop storing the multiplexer state is clocked asynchronously with its input. Since the flip-flop clock is generated from the overlap of two half-rate clocks, the timing margin is small. Therefore,

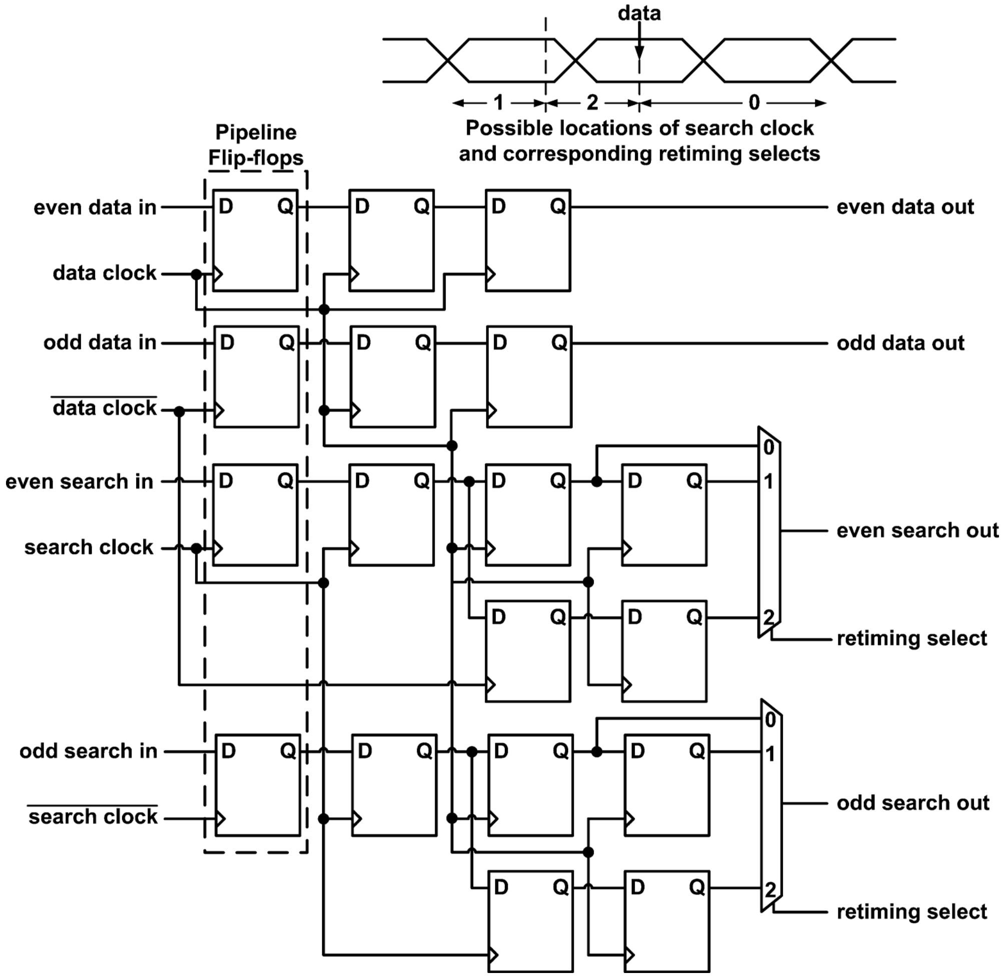


Fig. 14. Retiming logic.

a cascade of two synchronizers is used to account for metastability and ensure correct operation. The synchronizers introduce latency in the switch between clocks, but the control logic for the clock update runs at a much lower frequency, so this latency is inconsequential.

A further challenge is the long and asymmetric wiring run necessary to connect the search/data multiplexer of each pin and the pin-select multiplexer, which routes the search and data samples of the pin-under-calibration to the control logic. To ensure that proper timing is maintained between the recovered clock and data signals from each channel, they are co-routed and pipeline registers are inserted at the output of the search/data multiplexers.

C. Retiming Logic

The mismatch counter needs to compare samples arriving from both the data and search clock. Since the search clock is at a varying (but known) phase offset from the data clock, the incoming samples need to be retimed before this comparison can be made. This is accomplished through chains of flip-flops (Fig. 14); as the phase offset varies from small (1/32 UI, a single phase step) to large (as much as 1.5 UI, depending on the search type and the location of the data phase), each path in the retiming block is dedicated to a range of phase offsets. The appropriate path is selected based on the known location of the clocks.

The retiming logic takes samples from the odd phase of the clock to the even phase, and the timing for this transition is tight—it needs to complete in a full-rate instead of half-rate period. To maximize the timing margin available, pipeline flip-flops (outlined in Fig. 14) are added to the odd inputs, with the equivalent added to the even inputs for delay-matching purposes.

V. Hardware Measurements

The CDR was fabricated in a 90 nm bulk CMOS process. The die micrograph and core detail are presented in Fig. 15. Core area is 460 μm × 330 μm, in a 2.35 mm × 1.45 mm die. Correct operation over an infinite delay range was verified by sweeping the input phase of each channel independently at data rates from 6 to 9 Gb/s. A PRBS-31 input achieved BER $< 10^{-13}$.

Delay line response to calibration code was measured with a 4.5 GHz clock (i.e., data rate of 9 Gb/s), yielding a range of 183–278 ps. This corresponds to data rates between 7.2 Gb/s to 10.9 Gb/s (Fig. 16), if the delay line is required to match 2 UI exactly. The CDR operated correctly (with $>$ 1 UI of jitter) as low as 6 Gb/s, well below this range. This confirms that exact delay line calibration is unnecessary for the eye-monitoring algorithm to function.

Phase generator linearity was measured, with a worst-case DNL of 0.44 LSB (where 1 LSB = 1 phase position), and a

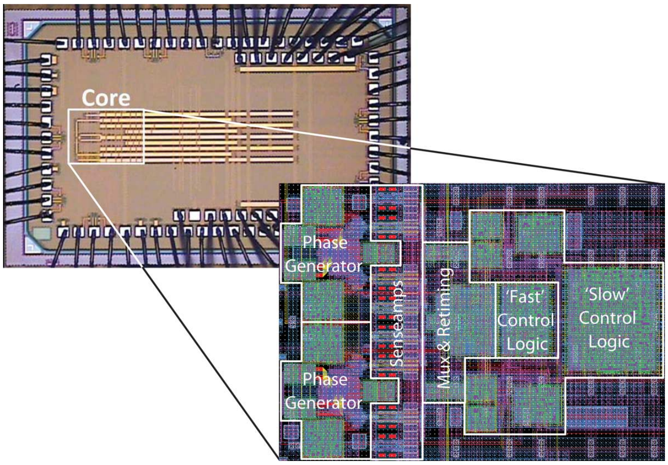


Fig. 15. Die micrograph and core detail.

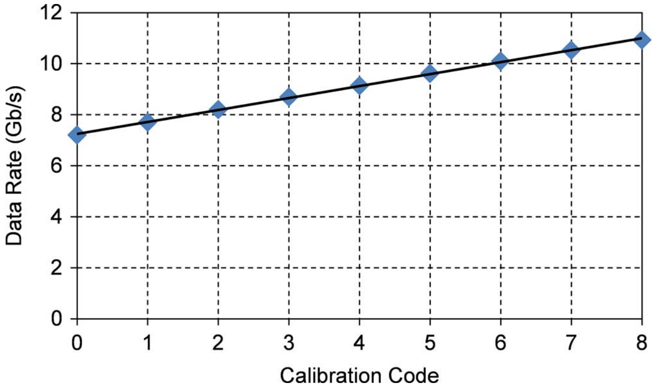


Fig. 16. Data rate for 2 UI of delay, over the control code range.

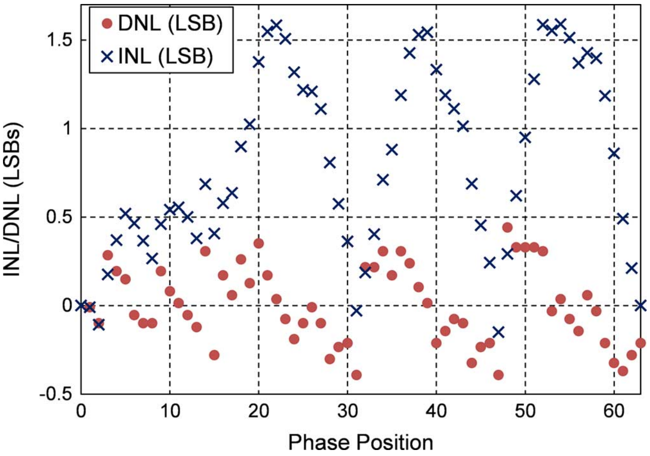


Fig. 17. Phase generator nonlinearity.

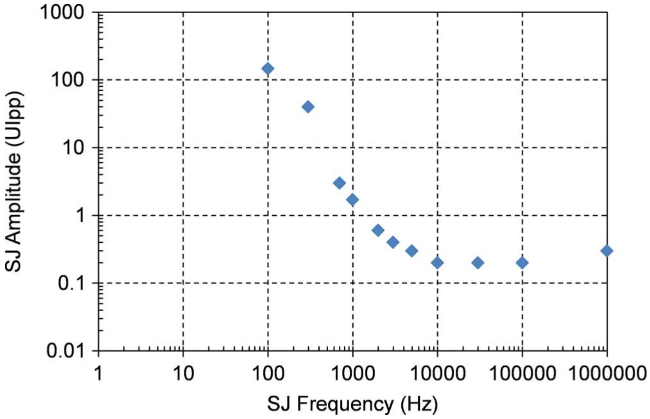


Fig. 18. SJ tolerance with control logic clock at 40 MHz, for $3 \times 9\ \mathrm{Gb/s}$ PRBS-7 input & $\mathrm{BER} < 10^{-12}$.

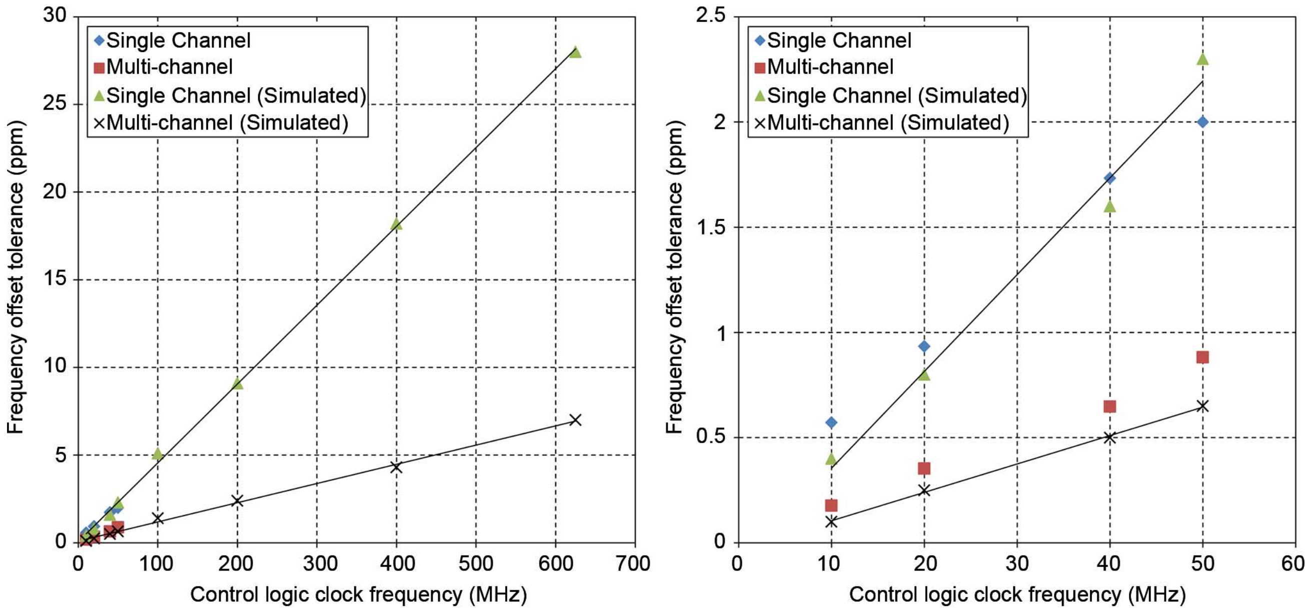
worst-case INL of 1.6 LSB. Using (9), this yields an overall phase-placement error of 2.32 LSB, or about 0.07 UI. A plot of the phase generator linearity (Fig. 17) shows that INL rises and then falls as the interpolator moves from one pair of delay line outputs to the next. This is a result of the way the delay line output-select multiplexer is designed; it selects one of the odd-numbered delay line outputs to feed into the first input of the interpolator, and one of the even-numbered delay line outputs to feed into the second input of the interpolator. As a result, the direction of the INL of the phase generator flips between pairs of delay line outputs. INL is the primary contributor to data clock placement error, so this error is limited by mismatch between the two variable-strength inverters in the interpolator.

Sinusoidal jitter (SJ) tolerance was measured with a control logic clock of 40 MHz (Fig. 18[1]). The period between data clock phase updates is limited primarily by the speed of the control logic, so an almost directly proportional relationship exists between the frequency offset tolerance (equivalently, the SJ tolerance bandwidth) and the control logic clock frequency. This is confirmed by measured results up to 50 MHz (limited by the design of the control logic that emphasized low-power operation at the expense of speed), which match simulated results closely (Fig. 19). Simulation at faster clocks shows that a linear relationship is maintained up to 625 MHz. This suggests a direct tradeoff between system performance and control logic power consumption; since the presented implementation is source-synchronous, low CDR bandwidths are tolerable and control logic power consumption is prioritized by targeting a lower clock frequency. Higher performance can be achieved by targeting a faster control logic clock, allowing the CDR to calibrate plesiochronous links with small frequency offsets.

The filtering parameter n_{base} also has a significant effect on CDR bandwidth, as described in Section II. Frequency offset tolerance was simulated at different values of n_{base}, using the highest logic frequency (625 MHz) to minimize the effect of logic delays (Fig. 20). $n_{\mathrm{base}} < 32$ results in faster searches and more frequent data phase updates, but CDR bandwidth is not improved since gains in speed are offset by a decrease in eye detection accuracy (Fig. 7). $n_{\mathrm{base}} > 32$ slows the search process and also degrades eye detection accuracy, so bandwidth decreases. These results validate the choice of $n_{\mathrm{base}} = 32$ indicated by the theoretical analysis.

Overall power consumption of the 3-pin system, operating at 9 Gb/s, is 103.3 mW, or 3.8 mW/Gb/s. Operation at 6 Gb/s, with a slight reduction in the supply voltage, yields an overall power consumption of 45.6 mW, or 2.5 mW/Gb/s. A module-by-module breakdown of power consumption was inferred by scaling measured results using simulation data (Fig. 21). By reducing the number of clocks required, the shared CDR brings the phase generation power consumption

[1] Previous simulated results [7] did not account for control logic overhead, which is the bandwidth-limiting factor in the implemented system.



(a) (b)

Fig. 19. (a) Frequency offset tolerance scaling for $3 \times 9\ \mathrm{Gb/s}$ PRBS-7 input (measured $\mathrm{BER} < 10^{-12}$ and simulated $\mathrm{BER} < 10^{-6}$), with (b) low frequency detail.

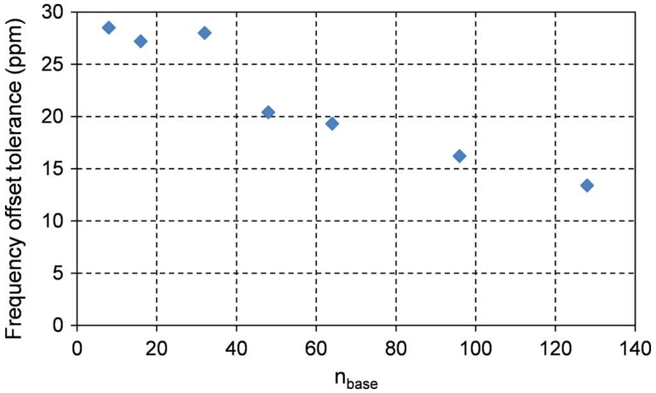
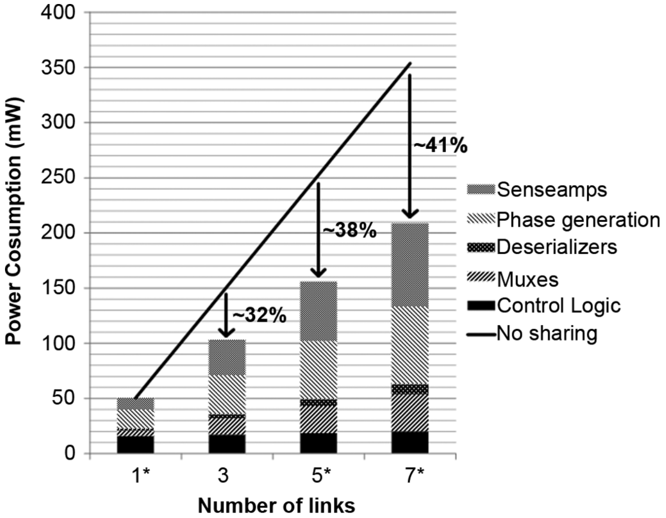


Fig. 20. Effect of nbase on frequency offset tolerance, simulated on single channel at 625 MHz with PRBS-7 input and $\mathrm{BER} < 10^{-6}$.



*Estimated by scaling measured power consumption

Fig. 21. Power breakdown and scaling performance.

in-line with that of the slicers, the next most significant component. Sharing the high-speed re-timing and mismatch counter logic, as well as the low-speed control logic, helps minimize control logic overhead at the expense of expanding multiplexer complexity. Further optimization is possible. As a simple example, this implementation keeps all the slicers running all of the time. Since the slicers exist in pairs of sets per pin (one set each for search and data), and only the data set is required unless the pin is being calibrated, it is possible to further reduce power consumption by gating the clocks to the unused slicers.

Even without such optimizations, the three-pin implementations uses about 32% less power than a naïve tripling of the single-pin system. Further scaling benefits can be realized by extending the system to wider links. The amount of sharing would ultimately be limited by the width of the channel-select multiplexer, the routing to this multiplexer and/or the desired jitter and frequency offset tolerance. It is possible to control multiplexer complexity by repeating the re-timing and mismatch counter logic over several subsets of pins in the overall link, and to mitigate performance loss due to sharing by increasing the control logic clock, although both these approaches sacrifice some of the power efficiency of the shared system. Overall performance of the system is summarized in Table I.

TABLE I
PERFORMANCE SUMMARY

Technology	90 nm
Data Rate (Gb/s)	3 x 9
Supply Voltage (V)	0.9 / 1.2 (6 Gb/s) 1.0 / 1.25 (9 Gb/s)
Power (mW)	45.7 (6 Gb/s) 103.34 (9 Gb/s)
Area (mm^2)	0.15
FOM (Power, mW/Gb/s)	2.5 (6 Gb/s) 3.8 (9 Gb/s)
FOM (Area, mm^2/Gb/s)	0.008 (6 Gb/s) 0.006 (9 Gb/s)

VI. Conclusion

The algorithm for, and design of, an all-digital, eye-monitoring CDR has been presented. Since each clock generates its own calibration information, the CDR eliminates the need for precisely-controlled clock generation, and is insensitive to clock-to-clock phase mismatch. The search algorithm and clock-swapping scheme enables the system to tolerate a poorly and slowly calibrated delay line, while still realizing an infinite delay range. In turn, the infinite delay range enables frequency offset and jitter tolerance characteristics that scale with control logic clock speed. A sharing technique that takes advantage of the generalized clock/sampling architecture used by the CDR algorithm has been discussed, which allows a reduction in number of clocks generated and an increase in power and area efficiency, at the cost of reducing CDR bandwidth. Using control logic clock frequency and the amount of sharing to tradeoff between power and CDR bandwidth allows the system to be scaled for power-efficient performance across a range of link types: source-synchronous, mesochronous and weakly plesiochronous. The all-static-CMOS, standard-cell heavy implementation maximizes flexibility and portability. These properties make the system well-suited for implementation in deep submicron and SOI technologies that have been optimized for the fabrication of digital systems, and in which mismatch becomes an increasing concern.

Acknowledgment

The authors acknowledge the contributions of J. Yoo for helpful technical discussions, H. Mani for advice in the fabrication of the test board, and the funding support of NSF, Intel and the C2S2 Focus Center.

References

[1] HyperTransport™ I/O Link Specification, HyperTransport Consortium, document no. HTC20051222–0046-0035 2010.

[2] N. Kurd, J. Douglas, P. Mosalikanti, and R. Kumar, “Next generation Intel® micro-architecture (Nehalem) clocking architecture,” in *2008 IEEE Symp. VLSI Circuits Dig.*, Jun. 2008, pp. 62–63.

[3] J. 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.

[4] M. Perrott, Y. Huang, R. Baird, B. Garlepp, D. Pastorello, E. King, Q. Yu, D. Kasha, P. Steiner, L. Zhang, J. Hein, and B. Del signore, “A 2.5-Gb/s multi-rate 0.25-μm CMOS clock and data recovery circuit utilizing a hybrid analog/digital loop filter and all-digital referenceless frequency acquisition,” *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2930–2944, Dec. 2006.

[5] P. K. Hanumolu, M. G. Kim, G.-Y. Wei, and U. k. Moon, “A 1.6 Gbps digital clock and data recovery circuit,” in *Proc. IEEE Custom Integrated Circuits Conf. (CICC’06)*, Sep. 2006, pp. 603–606.

[6] H. Song, D.-S. Kim, D.-H. Oh, S. Kim, and D.-K. Jeong, “A 1.0–4.0-Gb/s all-digital CDR with 1.0-ps period resolution DCO and adaptive proportional gain control,” *IEEE J. Solid-State Circuits*, vol. 46, no. 2, pp. 424–434, Feb. 2011.

[7] M. Loh and A. Emami-Neyestanak, “All-digital CDR for high-density, high-speed I/O,” in *2010 IEEE Symp. VLSI Circuits Dig.*, Jun. 2010, pp. 147–148.

[8] J. Montanaro, R. Witek, K. Anne, A. Black, E. Cooper, D. Dobberpuhl, P. Donahue, J. Eno, W. Hoeppner, D. Kruckemyer, T. Lee, P. Lin, L. Madden, D. Murray, M. Pearce, S. Santhanam, K. Snyder, R. Stehpany, and S. Thierauf, “A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor,” *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1703–1714, Nov. 1996.

[9] J. Alexander, “Clock recovery from random binary signals,” *Electron. Lett.*, vol. 11, no. 22, pp. 541–542, Dec. 1975.

[10] Y. Tomita, M. Kibune, J. Ogawa, W. Walker, H. Tamura, and T. Kuroda, “A 10-Gb/s receiver with series equalizer and on-chip ISI monitor in 0.11-μm CMOS,” *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 986–993, Apr. 2005.

[11] B. Analui, A. Rylyakov, 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.

[12] E.-H. Chen, J. Ren, B. Leibowitz, H.-C. Lee, Q. Lin, K. Oh, F. Lambrecht, V. Stojanovic, J. Zerbe, and C.-K. Yang, “Near-optimal equalizer and timing adaptation for I/O links using a BER-based metric,” *IEEE J. Solid-State Circuits*, vol. 43, no. 9, pp. 2144–2156, Sep. 2008.

[13] T. Suttorp and U. Langmann, “A 10-Gb/s CMOS serial-link receiver using eye-opening monitoring for adaptive equalization and for clock and data recovery,” in *Proc. IEEE Custom Integrated Circuits Conf. (CICC’07)*, Sep. 2007, pp. 277–280.

[14] 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.

[15] D. Oh, H. Lan, C. Madden, S. Chang, L. Yang, and R. Schmitt, “In-situ characterization of 3D package systems with on-chip measurements,” in *Proc. 60th Electronic Components and Technology Conf. (ECTC)*, Jun. 2010, pp. 1485–1492.

[16] E. Prete, D. Scheideler, and A. Sanders, “A 100 mW 9.6 Gb/s transceiver in 90 nm CMOS for next-generation memory interfaces,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC 2006) Dig. Tech. Papers*, Feb. 2006, pp. 253–262.

[17] J. Tierno, A. Rylyakov, and D. Friedman, “A wide power supply range, wide tuning range, all static CMOS all digital PLL in 65 nm SOI,” *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 42–51, Jan. 2008.

[18] S.-J. Lee, B. Kim, and K. Lee, “A novel high-speed ring oscillator for multiphase clock generation using negative skewed delay scheme,” *IEEE J. Solid-State Circuits*, vol. 32, no. 2, pp. 289–291, Feb. 1997.

Matthew Loh (S’04) received the B.S. degree in electrical and computer engineering from Lafayette College, Easton, PA, in 2004, and the M.S. degree in electrical engineering from the California Institute of Technology (Caltech) in 2009. He is currently working towards the Ph.D. in electrical engineering at Caltech.

Since 2007, he has been a member of the Mixed Signal Integrated Circuits and Systems (MICS) group at Caltech, designing efficient chip-to-chip and on-chip communication systems, and low-power interfaces for biomedical devices. In the summer of 2011, he interned at IBM Research, Yorktown Heights, NY, where he worked on low-power equalization for high-rate communication over server backplanes.

Azita Emami-Neyestanak (S’97–M’04) was born in Naein, Iran. She received the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1999 and 2004, respectively. She received the B.S. degree with honors from Sharif University of Technology, Tehran, Iran, in 1996.

She is currently an Assistant Professor of Electrical Engineering at the California Institute of Technology, Pasadena, CA. From July 2006 to August 2007, she was with Columbia University, New York, NY, as an Assistant Professor in the Department of Electrical Engineering. She also worked as a Research Staff Member at IBM T. J. Watson Research Center, Yorktown Heights, NY, from 2004 to 2006. Her current research areas are high-performance mixed-signal integrated circuits and VLSI systems, with the focus on high-speed and low-power optical and electrical interconnects, clocking, biomedical implant and compressed sensing.