

# A Reference-Less Single-Loop Half-Rate Binary CDR

Mohammad Sadegh Jalali, *Student Member, IEEE*, Ali Sheikholeslami, *Senior Member, IEEE*, Masaya Kibune, and Hirotaka Tamura, *Fellow, IEEE*

**Abstract**—This paper proposes a half-rate single-loop reference-less binary CDR that operates from 8.5 Gb/s to 12.1 Gb/s (36% capture range). The high capture range is made possible by adding a novel frequency detection mechanism which limits the magnitude of the phase error between the input data and the VCO clock. The proposed frequency detector produces three phases of the data, and feeds into the phase detector the data phase that minimizes the CDR phase error. This frequency detector, implemented within a 10 Gb/s CDR in Fujitsu's 65 nm CMOS, consumes 11 mW and improves the capture range by up to  $6\times$  when it is activated.

**Index Terms**—Burst-mode CDR, clock and data recovery, cycle-slipping, frequency detection, gated VCO.

## I. INTRODUCTION

CLOCK and data recovery circuits typically have two locking mechanisms: one for frequency and one for phase. Frequency locking is achieved either by locking the CDR frequency to an external oscillator whose frequency is close to a fraction of the data rate (in referenced CDRs) [1]–[6], or by using a frequency detector (in reference-less CDRs) [7]–[14]. In both architectures, at the beginning of the operation, the frequency correction mechanism brings the clock frequency to within a few hundred ppm of the lock frequency. At this point, the phase detector takes over to correct the remaining frequency offset and also to lock the phase.

Reference-less CDRs mainly target applications in which the use of an external crystal is not feasible [7]. One example is a repeater for either optical or copper media in which the space and number of pins are severely limited to include an external crystal oscillator. Also, adding a low-noise, rate-adjustable crystal could increase the overall cost and complexity of these receivers [7]. In these applications, frequency and phase detection can be done using the incoming data with the help of a frequency detector. Conventional frequency detectors monitor the change in data phase with respect to clock phase over many unit intervals (UI), and accordingly adjust the VCO frequency

so as to reduce the frequency offset. This adjustment, however, may go against the adjustment required by the phase detector, causing delays in phase locking. These frequency detectors are known as rotational frequency detectors and they date back to 1954 [10].

Single-loop CDRs in which both phase and frequency detection is performed in the same loop is becoming popular in recent years [11]–[14]. Refs. [11] and [12] propose single-loop architectures in which the CDR starts the VCO control voltage from zero and slowly sweeps the control voltage until the CDR acquires lock. Although this solution leads to a wide capture range, it is slow. [13] and [14] periodically reset the the phase of the clock in order to limit the phase error. This enables the phase detector to deal with frequency offset, but it comes at the cost of limited improvement in capture range [13] and a complicated delay calibration circuitry [14]. We will discuss these two FDs in more detail in Section II.

In this paper, we propose a frequency detection scheme that aligns the data phase to the clock phase, and thereby improves the FD gain, capture range, and power consumption compared to previous work [13]. Since the delay in resetting data phase is fixed (and small), no delay calibration circuit is needed. We will show that while the power consumption of this FD is similar to that of [13], its performance is improved by about  $2\times$ .

The remainder of this paper is organized as follows. Section II reviews the underlying issues with the existing solutions. Section III presents the proposed work and Section IV compares the proposed scheme against previous works. Section V describes the circuit implementation of the proposed scheme. Section VI shows the measurement results of this work and Section VII concludes the paper.

## II. BACKGROUND

### A. Rotational Frequency Detectors

As shown in Fig. 1, a single-loop phase tracking CDR is composed of a (linear or binary) phase detector (PD), charge-pump (CP), loop filter (LF) and VCO. When the CDR is locked,  $CK_I$  is aligned to the center of the UI and the phase error is zero, while, at the beginning of the operation, there is a random phase difference between clock and data.

The relation between the data phase and the clock phase can be represented either in time domain or in phase domain. Since we explain the operation of the PD and FD using phase diagrams, we review the conversion between these two domains in Fig. 2(a). For representation purposes, and in order to uniquely identify the data phase (i.e., the phase of the data edge), assume

Manuscript received December 26, 2014; revised March 22, 2015; accepted May 02, 2015. This paper was approved by Associate Editor Pavan Kumar Hanumolu. This work was supported by NSERC.

M. S. Jalali and A. Sheikholeslami are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario M5S 3G4, Canada.

M. Kibune and H. Tamura are with Fujitsu Laboratories Limited, Kawasaki-shi, Kanagawa-ken, 2118588, 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.2015.2429714

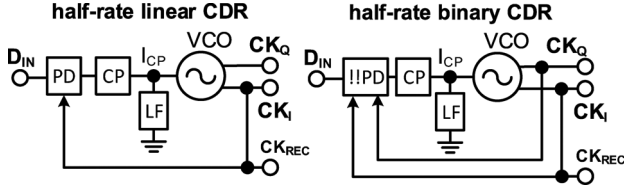
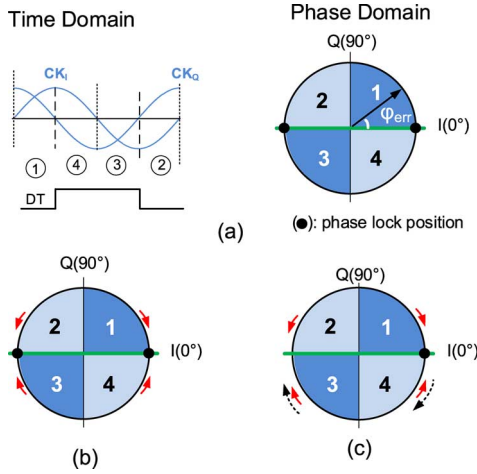


Fig. 1. Conventional half-rate linear and binary CDRs.


 Fig. 2. (a) Time domain and phase domain representation of data and clock (b) response of CDR to phase offset (c) response of CDR to positive frequency offset ( $f_{DATA} > 2f_{CK}$ ).

data samples both clocks,  $CK_I$  and  $CK_Q$ . When the CDR is locked, the data edge is aligned to the clock peaks (shown by the black dots in the right figure), where  $CK_Q$  is zero and  $CK_I$  is at its peak. When the CDR is not locked, data edge can arrive anywhere in the time domain. Therefore, the pair of  $CK_I$  and  $CK_Q$  samples map directly to a circle. In the phase diagram, the phase of the data edge at some given time with respect to the clock phase is shown by the arrow. The difference between data phase and phase of  $CK_I$  is defined as phase error,  $\phi_{err}$ . The role of the CDR loop is to zero the phase error. Therefore, as shown in Fig. 2(b), if the phase of data falls in regions 1 and 3, it will be rotated clockwise (negative PD output), while if the phase falls in regions 2 and 4, it will be forced to rotate counter clockwise (positive PD output) with respect to clock phase.

While in a half-rate design and at the beginning of the operation, it is desirable for the average VCO frequency to be twice the data rate, process, temperature and voltage variations shift the initial VCO frequency, causing frequency offset between clock and data. To justify the need for frequency detection, Fig. 2(c) shows the behavior of the CDR in the presence of a positive frequency offset. Here, two effects work simultaneously on the data phase: one is the effect of the loop (solid red arrow) and the other is the effect of frequency offset (black dotted arrow). In region 4, these two effects oppose each other, while in region 3, they both move the data phase in a clockwise direction. Therefore, the data phase spends more time in region 4 than region 3, and on average, the VCO frequency will increase. This CDR will eventually lock after a few periods of cycle-slipping. However, if the frequency offset is large, the effect of the loop becomes negligible as the change in phase due to

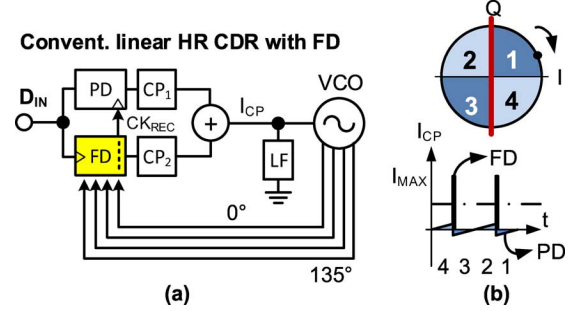


Fig. 3. (a) Conventional Dual-loop CDR (b) operation of a rotational frequency detector.

frequency offset dominates the total change in phase error (see (5) in the Appendix). Data phase goes through region 4 with the same speed that it goes through region 3, and on average, the VCO frequency remains unchanged.

A rotational frequency detector detects the direction of this data phase rotation and opposes it. Fig. 3(a) shows a dual loop CDR in which the FD is placed in parallel to the PD. Fig. 3(b) shows the operation of a half-rate FD [8] which detects the data phase crossing between the boundary of regions 1 and 2, and also 3 and 4. It also detects the direction of this boundary crossing, which is used to find if data frequency is higher or lower than the VCO frequency. However, guaranteeing a smooth transition between the FD loop and PD loop, as well as the area and power overhead of the second FD loop, limits the use of dual-loop CDRs [11].

### B. Frequency Detection Based on Clock Phase Adjustment

In a half-rate CDR, phase error with respect to either  $CK_I$  or  $CK_Q$  (which are  $90^\circ$  apart) will move the VCO in the correct direction. A clock-phase-selection (CPS) FD [13], takes advantage of this observation and by feeding the “correct” clock phase into the PD, eliminates the FD loop. This can be done by selecting the clock phase with a lower value of phase error on the edge of the data. Fig. 4(a) shows a half-rate linear CDR with a CPS FD, where the “correct” clock phase is found by selecting the higher of  $CK_I$  and  $CK_Q$  if  $CK_{REC}$  is positive and the lower of the two if  $CK_{REC}$  is negative [13]. As shown in Fig. 4(b), by switching the PD clock, the CPS FD makes the average CP current negative if  $f_{DATA} > 2f_{CK}$  and positive if  $f_{DATA} < 2f_{CK}$ .

Although [13] eliminates the need for a secondary FD loop, the CPS FD still allows the CDR to cycle slip in regions 1 and 3, as shown in Fig. 4(b), reducing the efficiency of the PD in the presence of frequency offset. To eliminate cycle-slipping, [14] and [17] periodically re-align the clock phase with the data phase to reset the phase error of the CDR. This is done, as shown in Fig. 5, by replacing the VCO with a gated-VCO (GVCO) [19]. By resetting the GVCO, a clock phase reset (CPR) FD aligns the clock phase to the data phase every  $N$  data cycles. This also reduces the errors that the CDR make before lock [14]. However, the delay required to reset the GVCO could be as large as 200–300 ps [14]. To solve this problem, [14] delays the data that gets to the phase detector by up to 300 ps. This ensures that after the VCO is reset, its phase is aligned with that of the

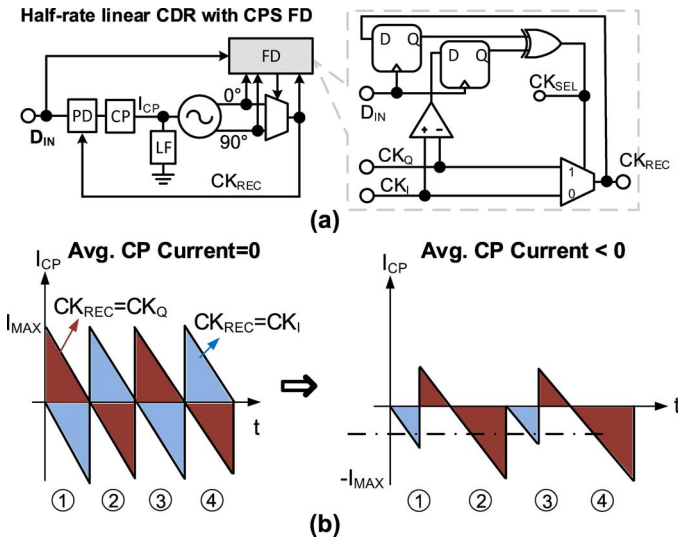


Fig. 4. (a) CDR with CPS FD (b) phase error with and without CPS FD when  $f_{DATA} < 2f_{CK}$ .

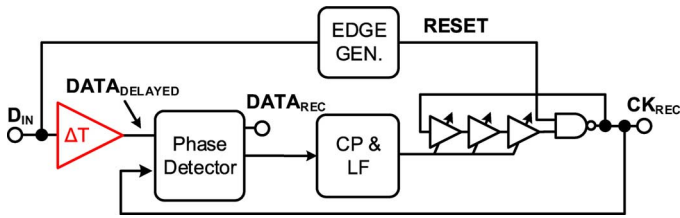


Fig. 5. Implementation of the CDR with “phase reset” [14].

delayed data. The delay line consumes 13 mW of power, and calibrating it is power and area hungry. Also, perfect calibration is not possible, as the delay to reset the GVCO depends on the VCO phase at the arrival time of the data edge. It can be shown that a residual delay mismatch of  $\tau$  causes the CDR to lock with a forced frequency offset of  $\tau / (2N \times T^2)$ , where  $T$  is the data period and  $N$  is the number of data edges between subsequent phase reset events. To reduce this frequency offset, [14] increases  $N$  to 8, which comes at the cost of capture range while [17] increases  $T$ , running the CDR at 2.2 Gb/s.

### III. PROPOSED HALF-RATE FREQUENCY DETECTOR

To enable the phase detector to deal with frequency offset, the frequency detection mechanism should adjust the phase error of the PD prior to frequency locking. In order to achieve this, we propose to occasionally shorten or elongate the data UI by inserting a unit interval adjuster (UIA) block prior to the PD. Fig. 6 shows the phase error of a half-rate binary CDR with the proposed UIA block along with the block diagram of the UIA. For the case of a positive frequency offset ( $f_{DATA} > 2f_{CK}$ ), the data UI is smaller than half the clock period, and hence the data edge moves to the left with respect to clock, increasing the phase error. When the phase error becomes large enough (greater than  $UI/4$ ), we replace a UI with a “long UI”, so as to reduce the phase error by  $UI/4$ . The  $UI/4$  threshold was chosen as it can easily be detected by comparing the signs of

$CK_I$  and  $CK_Q$  at the data edge (and without knowing their amplitude). In case of a negative frequency offset ( $f_{DATA} < 2f_{CK}$ ), the data UI is larger than half the clock period, and hence the data edge moves to the right with respect to clock, moving the phase error in the negative direction. When it becomes smaller than  $-UI/4$ , we replace the UI with a “short UI”, again limiting the phase error. By not letting the phase error exceed  $UI/4$ , cycle slipping is totally eliminated. A “long UI” is approximately 1.25 UI long while a “short UI” is approximately 0.75 UI long. The long and the short UIs are created respectively by either delaying or advancing the data by 0.25 UI.

As shown in Fig. 6, the UIA block is comprised of a phase error monitor block, a CMOS delay line and a transmission gate based multiplexer. The long and the short UIs are produced by passing data through a 4-tap delay line with approximately 0.25 UI delay between adjacent taps. A half-rate CDR has two locking points, shown in Fig. 2(a). As far as the PD is concerned, the phase error due to the  $0^\circ$  and the  $180^\circ$  phases of the data is the same. Therefore, only three phases of the data needs to be generated. The data phase goes back to the  $0^\circ$  phase after it arrives at the  $135^\circ$  phase. The CDR makes an error during this switch, however this error occurs before lock acquisition and can therefore be tolerated. Also, this introduces a glitch into the data stream which can be tolerated by the loop, as the frequency of the occurrence of these glitches are low. Furthermore, these data glitches are acceptable as they occur prior to CDR lock. The phase error monitor block is responsible for detecting when the phase error is greater or smaller than  $UI/4$  or  $-UI/4$ , respectively. The four-bit, one-hot, output of this block controls the multiplexer.

Fig. 7 shows the block diagram of the binary half-rate CDR with the proposed UIA block, where we pass the data through a unit interval adjuster (UIA) block before it gets to the phase detector. Since the magnitude of the phase error is kept small, a binary PD [16] was used in the design.

Fig. 8 shows the phase diagram and the implementation of the phase error monitor block. One full circle in the phase diagram represents one full period of the clock and two data UIs under frequency lock condition. The phase diagram on the left identifies the regions where the phase error is less than  $UI/4$ . In this region, UIA is inactive. The phase diagram in the middle corresponds to a data with a negative frequency offset where the data phase has moved counter-clockwise to arrive in the blue region. In this region, the UIA shortens one UI so as to align the next data edge to the clock. A positive frequency offset, shown in the right phase diagram, rotates the data phase clockwise to arrive in the red region. In this region, the UIA elongates one UI. The operation described above is implemented as shown in Fig. 8. The phase error monitor block samples  $CK_I$ ,  $CK_I + CK_Q$  and  $CK_I - CK_Q$  to produce the  $RL$  (replace with long UI) and  $RS$  (replace with short UI) signals. The data is delayed if  $RL$  is asserted and advanced if  $RS$  is asserted; otherwise, the data phase remains unchanged. A simplified phase interpolator [18] is used to realize addition and subtraction of the two clock phases. After the CDR acquires lock, the  $RL$  and  $RS$  signals stop toggling (this is because the data phase is no longer rotating with respect to the clock phase). This is detected by a toggle detector, which

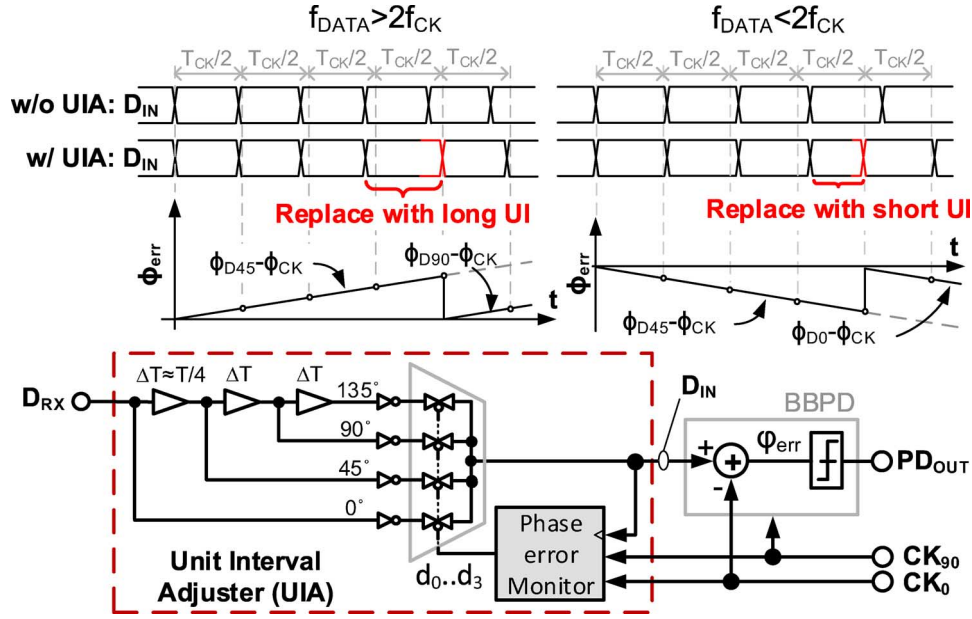


Fig. 6. The UIA block measures the accumulated phase error between data and clock and occasionally elongates or shortens the UI to limit the magnitude of the phase error, hence eliminating cycle slipping.

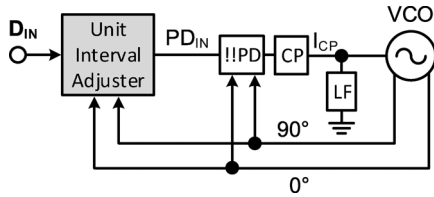


Fig. 7. Half-rate CDR with data phase selection FD.

disables the UIA. At this point, the UIA-enabled CDR behaves like a conventional binary CDR.

Finally, to mitigate the effects of jitter and ISI on the data phase switching, we filter the  $RL$  and  $RS$  signals so as to switch the data phase only if the  $RL$  or the  $RS$  signals repeat themselves on two consecutive data transitions. Our simulations show (this was verified during measurements) that filtering  $RL$  and  $RS$  on two consecutive data transitions provides a good trade-off between tolerance to ISI and capture range (the peak FD gain goes down by a factor of 3.5 when the filter depth is increased to 3). Note that under extreme jitter and ISI conditions, it is possible for an immature phase switch to take place, however, the CDR will acquire lock as long as the magnitude of the phase error is kept small most of the time. The filtering of the  $RL$  and  $RS$  signals reduce the occurrence of false phase switches and are therefore critical in achieving desired operation in the presence of jitter (and ISI).

As previously mentioned, the value of the delay need not to be accurate. In fact, the delay of the delay line changes by  $\pm 18\%$  over PVT. To see how the system behaves in the presence of a delay mismatch, Fig. 9(a) shows the locking transient of the CDR assuming three different corners for the delay line. To isolate the effect of the change in the delay (and not the change in the performance of the other blocks across process corners), the delay was calculated from our RC-extracted simulations in each

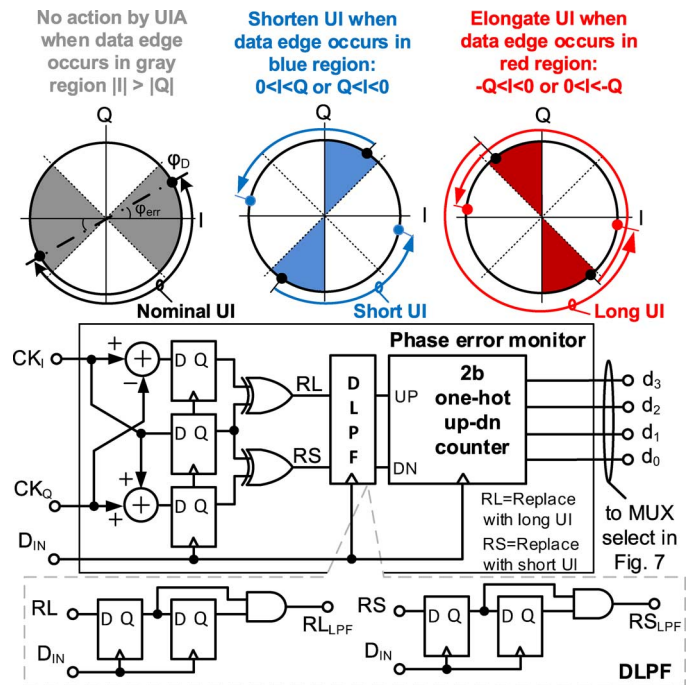


Fig. 8. Phase error monitor operation and implementation.

corner and imported into MATLAB for locking behavior simulations. The transient behavior of the CDR remains the same across all corners. Fig. 9(b) shows the case where we are creating a  $\pm 20\%$  of random mismatch between delay elements in the delay line. Again, the general behavior of the VCO control voltage remains the same across corners. This is because to eliminate cycle slipping, the value of the phase error need not to be zeroed after the insertion of a long or a short UI; keeping its magnitude small enough would serve the same purpose.

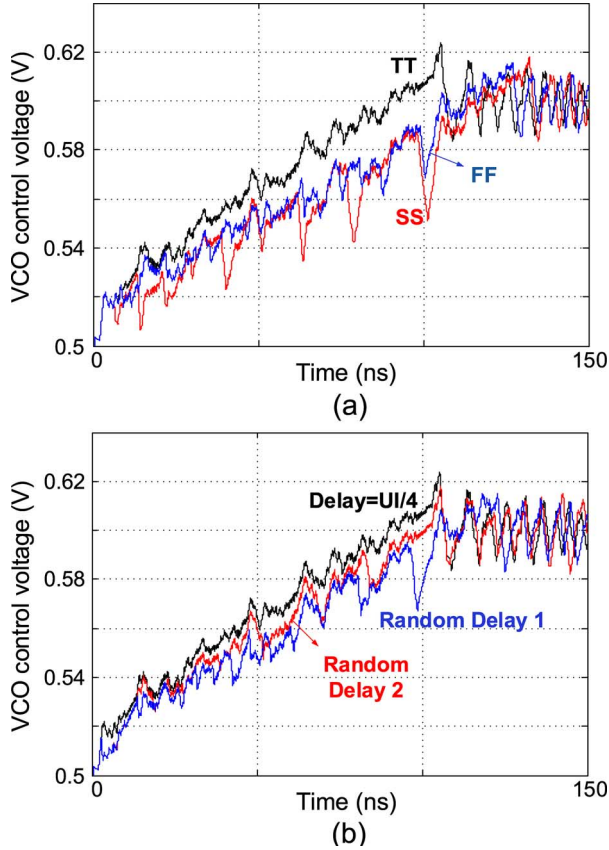


Fig. 9. Locking behavior of the CDR in the presence of an inaccurate delay line and with a PRBS7 pattern (a) across corners (b)  $\pm 20\%$  of delay variation.

To show the details of the lock transient of the proposed CDR, we plot the phase error and the VCO control voltage for a 3.3% and  $-3.3\%$  of frequency offset in Fig. 10(a) and (b), respectively. Prior to activating the UIA ( $t < 20$  ns), cycle slipping prevents the CDR to acquire lock and the control voltage remains unchanged. After the UIA is activated ( $t > 20$  ns), the CDR moves towards lock without cycle slipping. The high ripple on the VCO control voltage after lock is due to the activity of the binary phase detector and is not caused by the operation of the UIA block. Also, note that in this region, the phase error is always positive in Fig. 10(a) and always negative in Fig. 10(b).

The Appendix shows that the lock time of an analog CDR with a first-order RC loop filter (a resistance of  $R$  and a capacitance of  $C$ ) for a clock pattern input with the UIA operation is

$$t_{lock} = \frac{I_{CP}}{C \times k_{VCO}} [f_0 - f_{avg}] \quad (1)$$

where the charge-pump current is  $I_{CP}$  and the VCO gain is  $k_{VCO}$ .  $f_0$  is the initial frequency offset,  $f_{avg} = \Delta\Phi_{loop}/2\pi$ , and  $\Delta\Phi_{loop}$  is:

$$\Delta\Phi_{loop} = 2\pi k_{VCO} T \left[ RI_{CP} + \frac{TI_{CP}}{2C} \right] \quad (2)$$

where  $T$  is the period of data.

#### IV. COMPARISON TO PREVIOUS WORK

In this section, we compare the capture range and the lock time of the proposed frequency detection scheme to the capture

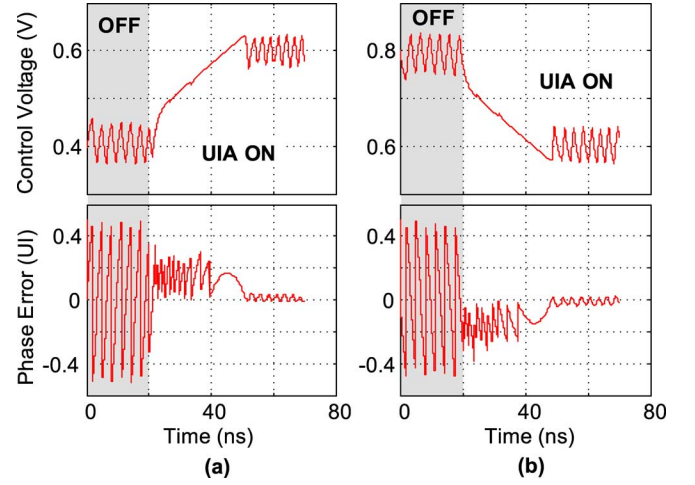


Fig. 10. Behavioral simulation results of the CDR with UIA with clock pattern and with (a) 3.3% frequency offset (b)  $-3.3\%$  frequency offset.

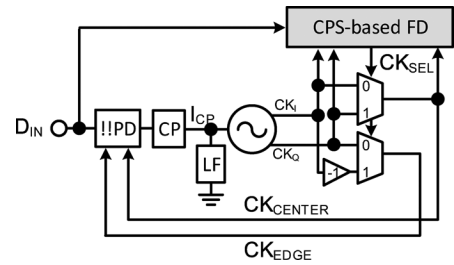


Fig. 11. Clock phase selection FD with a binary PD.

range and the lock time of the CPS FD and the conventional FD in [8]. To be able to compare these frequency detection schemes, we slightly modify the design of the CPS FD in [13] to enable it to operate with the same half-rate binary PD. This is shown in Fig. 11.

#### A. Capture Range

Fig. 12 compares the normalized average charge-pump current (obtained by integrating the CP current) versus frequency offset for a PRBS7 input pattern for a half-rate binary CDR with a conventional FD [8], a CPS FD [13] and a CDR with UIA. The CDR with the proposed UIA was also simulated with a PRBS31 pattern. The CDR with UIA provides a gain that is  $6.5 \times$  and  $2.5 \times$  higher than those of a CDR with a conventional FD and a CPS FD, respectively. The gain of the UIA FD goes down by a factor of 2.2 when the data pattern is switched from PRBS7 to PRBS31. Our RC-extracted simulation results show that for a 10% frequency offset, the CDR with UIA achieves a lock time of  $1 \mu\text{s}$ , which is  $2.1 \times$  smaller than that of the CDR with CPS FD, while the CDR with the FD in [8] fails to acquire lock. For a 36% frequency offset (corresponding to the measured tuning range), the proposed CDR achieves a lock time of  $4 \mu\text{s}$ .

To study the performance of the proposed FD under extreme frequency offset, Fig. 13 re-simulates the CDR with data phase selection (DPS) FD using a PRBS7 pattern and a clock pattern for a  $\pm 60\%$  of frequency offset. It can be observed that the DPS FD operates reliably if the initial VCO frequency is between 3.5 GHz and 6.5 GHz (corresponding to a capture range of about

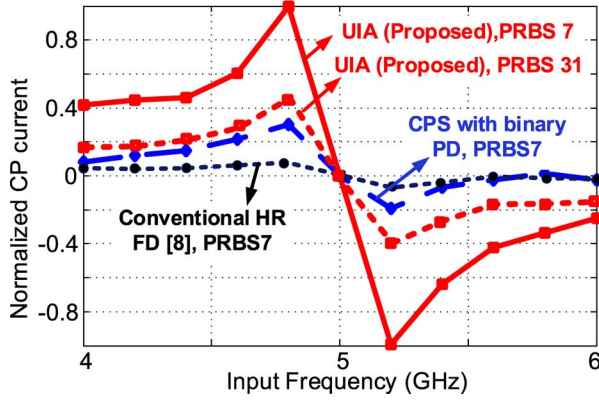


Fig. 12. Normalized charge-pump current versus frequency error with a PRBS7 pattern (and PRBS31 for UIA-enabled CDR) for a binary CDR with conventional FD, CPS FD, and the proposed UIA block.

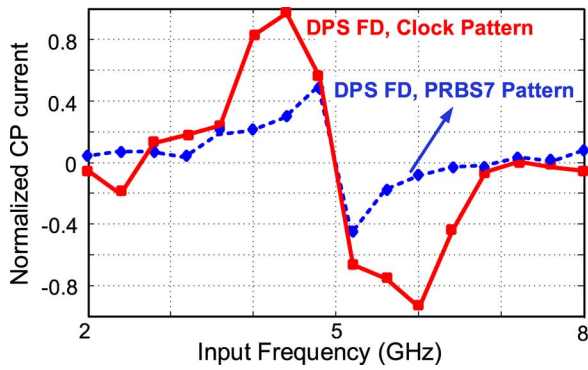


Fig. 13. Simulink results of the normalized charge-pump current versus frequency error for a PRBS7 and clock pattern; The VCO tuning range is increased beyond the original design to study the performance of the system in the case of an extreme frequency offset.

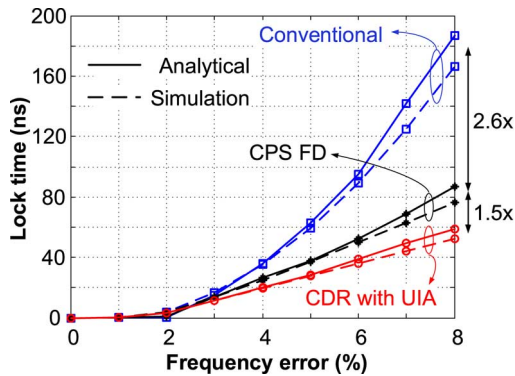


Fig. 14. Comparison between the lock time of a conventional CDR and a CDR with a CPS FD and a DPS FD.

70%), when PRBS7 pattern is used. As the frequency offset increases beyond this point, it becomes likely for data edges to completely miss the blue or the red regions of Fig. 8, and as a result, the FD fails to reset the phase error. Changing the pattern to clock, extends this capture range by about 20%, as the clock pattern contains more edges than the PRBS7 pattern.

It should be mentioned that minimizing the phase error to eliminate cycle slipping can also be achieved by choosing the correct clock phases into the PD (similar to the CPS FD). To

extend the idea of the CPS FD to achieve a phase error less than  $UI/4$  at all times, one of the four clock phase should be selected as  $CK_I$  for the PD, requiring a 4:1 multiplexer to be inserted at the VCO output (consequently, another 4:1 MUX is required for  $CK_Q$ ). The latency of the additional multiplexer, as well as the additional power consumption required to route all the four clock phases out of the VCO, makes this idea less attractive. Alternatively, the clocks can be delayed. However, this requires two delay lines (since both  $CK_I$  and  $CK_Q$  need to be delayed), which doubles the power consumption of the delay line.

### B. Lock Time Analysis

It can be shown that the lock time of an analog CDR with a first-order RC loop filter (a resistance of  $R$  and a capacitance of  $C$ ) for a clock pattern input is [21]:

$$t_{lock} = \frac{C}{2\Delta\phi_{loop}I_{CP}k_{VCO}} \left[ 2\pi(f_{avg} - f_0)T + \frac{\pi^2 - \Delta\phi_{loop}^2}{\pi} \times \ln \left( \frac{1 - 2\pi T f_{avg}}{1 - 2\pi T f_0} \right) \right]. \quad (3)$$

In the Appendix, we show that the lock time of a CDR with a CPS FD is:

$$t_{lock} = \frac{C}{I_{CP}k_{VCO}} \left[ \frac{2\pi(f_0 - f_{avg})}{\pi - \Delta\Phi_{loop}} + \frac{\Delta\Phi_{loop}(\pi + \Delta\Phi_{loop})}{\pi T(\pi - \Delta\Phi_{loop})} \times \ln \left( \frac{\Delta\Phi_{loop} + 2\pi T f_{avg}}{\Delta\Phi_{loop} + 2\pi T f_0} \right) \right]. \quad (4)$$

Fig. 14 compares the lock time of a conventional CDR and a CDR with a CPS and a CDR with the proposed UIA. The dotted lines represent the analytical results while the solid lines are obtained from simulations. Also, due to symmetry, only the lock time with a positive frequency offset is shown. The CDR with UIA achieves a lock time  $1.5 \times$  smaller than the CPS FD and  $3.9 \times$  smaller than the conventional CDR. We will show in the next section that the CDR with UIA also has a wider capture range than the CPS FD.

## V. CIRCUIT IMPLEMENTATION

An active and or a passive delay lines can be used to implement the required delay on the data path. Passive delay lines are based on some form of a transmission line, while active delay elements take advantage of the inherent delay of a buffer. Although a transmission line consumes no power and has very wide bandwidth (thus creating negligible amount of ISI), it occupies a large area. On the contrary, active delay lines occupy a very small area, but consume power.

To implement the buffer in the delay line, several options were considered. Although the delay of a CML buffer can easily be made adjustable, it has a high power consumption. However, as previously discussed, since the exact value of the delay is unimportant, this option was not used. We found that the required delay can easily be implemented using CMOS buffers. To this end, and as shown in Fig. 15, the differential input data is first fed into a CML-to-CMOS converter. The CMOS data is then fed into a CMOS delay line, in which the data is delayed three time to construct the four phases of data. Note that each

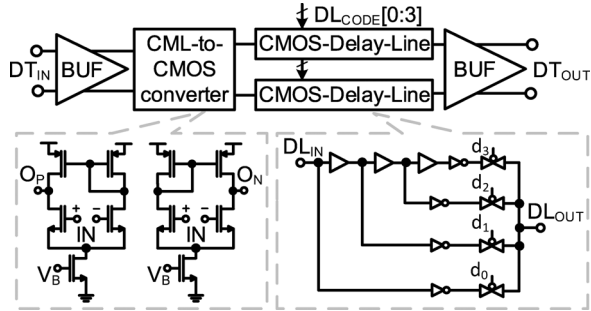


Fig. 15. Delay line implementation.

buffer consists of two back-to-back CMOS inverters. One of the four data phases is chosen by the UIA. A CML buffer, with a high common-mode rejection, converts the CMOS data back to CML, which is then fed to the phase detector. The CML-to-CMOS converter is built using two differential to single-ended converter circuits with a gain of around 2.

To study the impact of the delay line on the received data, Fig. 16 shows the simulated large signal transfer function of the delay line for the four cases of  $d_3d_2d_1d_0 = 0001$  (corresponding to the shortest delay),  $d_3d_2d_1d_0 = 0010$ ,  $d_3d_2d_1d_0 = 0100$ , and  $d_3d_2d_1d_0 = 1000$  (corresponding to the longest delay), where  $d_3$ ,  $d_2$ ,  $d_1$ , and  $d_0$  are the one-hot output signals of the phase error monitor block, shown in Fig. 8. The figure shows that the limited bandwidth of the delay line introduces a worst case simulated loss of approximately 0.6 dB at 6 GHz, which is small compared to the loss introduced by the channel (6.4 dB at 6 GHz which will be shown later in the measurement section). Furthermore, this delay line can be bypassed and powered down after locking is acquired to reduce the power consumption of the CDR. This introduces a momentary phase offset, which is corrected by the loop.

Fig. 17 shows the implementation of a binary half-rate phase detector [16]. Here,  $CK_I$  and  $CK_Q$  sample the data on the center and edge of the UI, respectively. The recovered half-rate data,  $D_{OUT}[even]$  and  $D_{OUT}[odd]$ , is also shown in the figure.

Fig. 18 shows the circuit implementation of the VCO. The VCO delay cell is based on a differential pair with a cross-coupled stage. The delay of each stage is controlled by  $V_{TUNE}$ , which adjusts the trans-conductance of the negative-gm stage, varying delay.  $V_{TUNE}$  is used in a differential fashion to maintain a constant common-mode at the VCO output (leading to constant current drawn from the supply). The single-ended to differential converter circuit converts the single-ended  $V_{CNT}$  to the differential  $V_{TUNE}$ . A CML buffer is connected to the VCO core not to load to VCO with the PD. To maintain symmetry among clock phases, the internal VCO nodes are connected to dummy buffers.

Fig. 19 shows the circuit implementation of the charge-pump. An on-chip DAC is used to adjust the CP current during capture range measurements. The nominal CP current is 100  $\mu$ A.

## VI. MEASUREMENT RESULTS

The chip, shown in Fig. 20, is fabricated in Fujitsu's 65 nm CMOS process. The area of each block is shown in the figure.

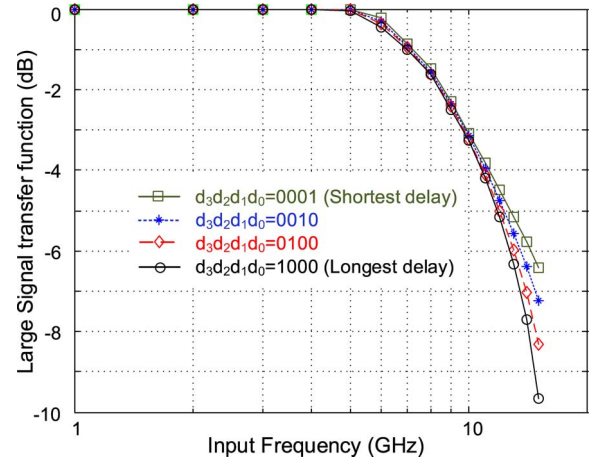
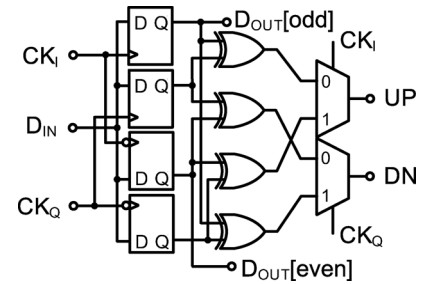

 Fig. 16. Simulated large signal transfer function of the delay line for four cases of  $d_3d_2d_1d_0 = 0001$ ,  $d_3d_2d_1d_0 = 0010$ ,  $d_3d_2d_1d_0 = 0100$ , and  $d_3d_2d_1d_0 = 1000$ .


Fig. 17. Implementation of the phase detector.

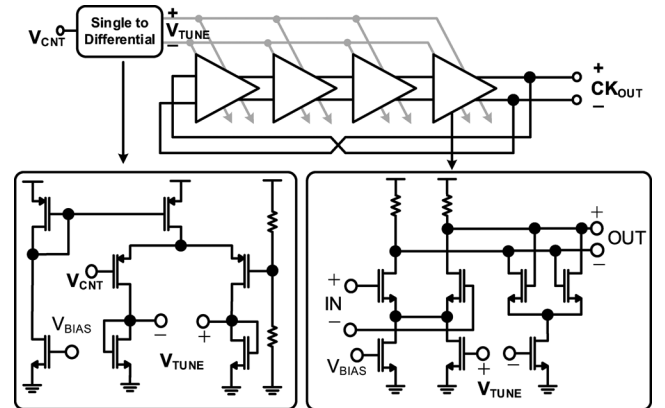


Fig. 18. Half-rate linear CDR with the proposed half-rate FD.

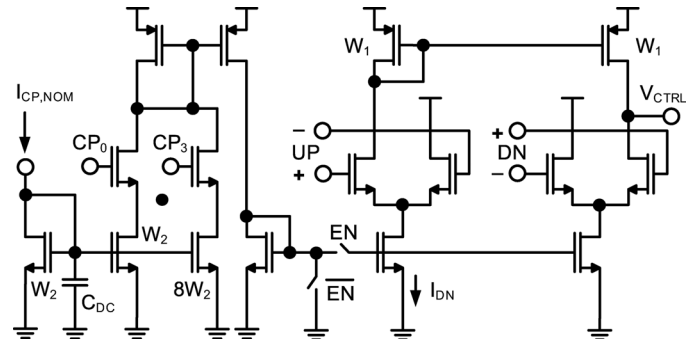


Fig. 19. Circuit diagram of the charge-pump.

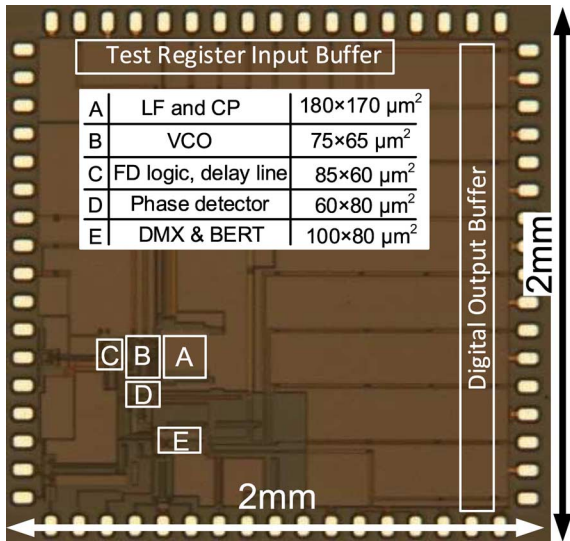


Fig. 20. Chip photo.

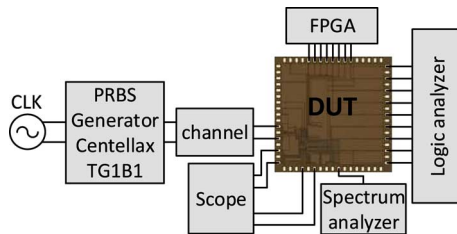


Fig. 21. Measurement setup.

The VCO and the FD logic consume 21 mW, the delay line (including the differential to single-ended converter and the CML buffers) consume 6.6 mW, and the CDR core consumes 15.4 mW. The receiver consumes 3.5 mW/Gb/s, when the data rate is set to 12.1 Gb/s.

The measurement setup is shown in Fig. 21. The PRBS generator (Centellax TG1B1-A) is clocked with an 8–12 GHz clock source (Centellax TG1C1-A), and is connected to the chip through a 48” SMA cable or a 7” FR4 channel. For jitter tolerance measurements, sinusoidal jitter (SJ) is inserted on the clock of the PRBS generator. The recovered clock is fed into a spectrum analyzer (N9010A), while the half-rate recovered data output is fed into an oscilloscope (DSA-X 91604A). An FPGA programs the chip and a logic analyzer monitors the state of the chip.

Fig. 22 shows the tuning range of the VCO. In this measurement, the CDR loop is opened and the VCO control voltage is manually swept. The VCO frequency is from 4.2 GHz–6.35 GHz. Note that the CDR cannot achieve lock in the end points of this curve.

Fig. 23 shows the recovered half-rate data eye diagram for the 10 Gb/s PRBS7 input. The recovered eye is fully open and the measured BER is less than  $10^{-12}$ . The loss of the probe card and a 1 mm on-chip trace has degraded the quality of the measured eye. Fig. 24 shows the spectrum of the recovered clock of the CDR locked to both PRBS7 and PRBS31 data. As expected, the recovered clock spectrum has a dominant term at 5 GHz.

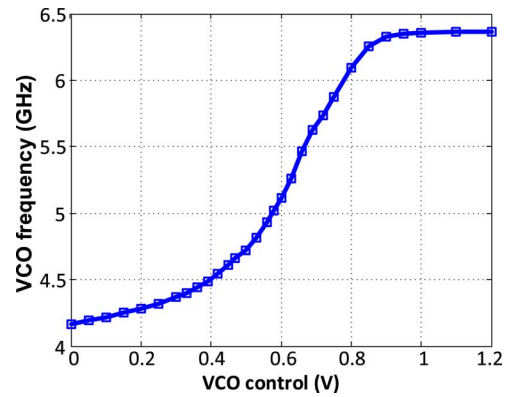


Fig. 22. Measured VCO tuning range.

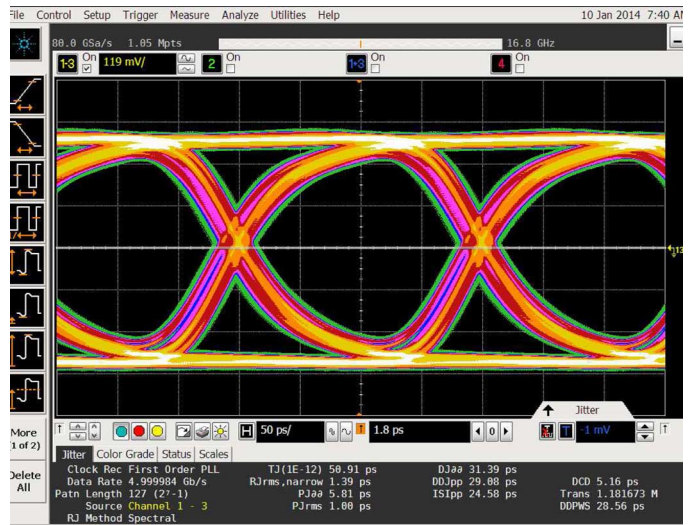


Fig. 23. 10 Gb/s half-rate eye.

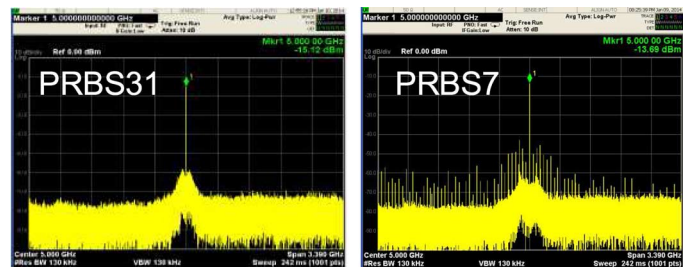


Fig. 24. 5 GHz locked PRBS7 and PRBS31 clock spectrum.

Fig. 25, shows the measured jitter tolerance of the CDR for a 10 Gb/s PRBS7 input data and a bit error rate less than  $10^{-12}$ . We verified that once the CDR is locked, the UIA is deactivated automatically and as such had no effect on jitter tolerance. To measure the jitter tolerance, we used a full-rate on-chip BERT (instead of using an off-chip BERT on the recovered half-rate data). Furthermore, the CDR is first locked to a 10 Gb/s pattern before the number of errors are counted. Incoming data does not go through the FR4 channel in this measurement.

Fig. 26 shows the measured capture range of the CDR. To measure the capture range, the VCO control voltage is driven to the supply voltage, setting the VCO to its maximum frequency. The frequency of the PRBS7 input is then swept, and the CDR



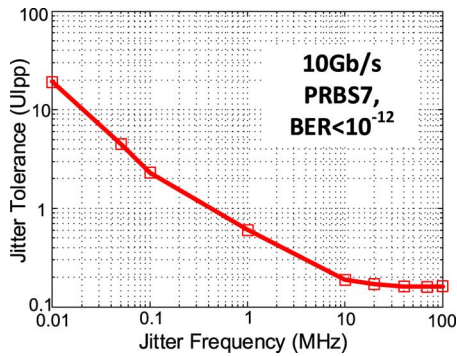


Fig. 25. Measured jitter tolerance at 10 Gb/s.

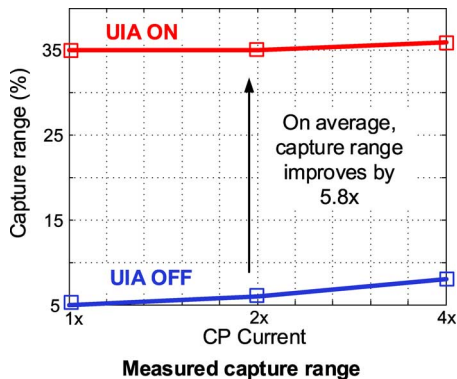


Fig. 26. Measured capture range with UIA on and off.

locking is monitored by verifying that the half-rate recovered data eye is open and the recovered clock frequency is correct. As shown, activating the UIA increases the capture range by  $5.8 \times$ . At  $4 \times$  the CP current ( $I_{CP}$ ), the CDR capture range is from 11.6 Gb/s to 12.1 Gb/s with the UIA off, while it is from 8.5 Gb/s to 12.1 Gb/s with the UIA on. The measured (linear) tuning range of VCO is from 4.2 GHz to 6.1 GHz. Note that while in a conventional CDR, increasing the charge-pump current increases the capture range (this is because increasing  $I_{CP}$  increases  $\Delta\Phi_{loop}$ ), it also increases the generated jitter. This is in contrast with the proposed CDR where the capture range is not affected by the charge-pump current and is mainly limited by the VCO tuning range. This is because the UIA operation keeps the phase error small and thus the PD will always move the VCO in the correct direction. Therefore, minimum  $I_{CP}$  can be used to reduce jitter generation. Also, we verified that changing the supply voltage of the delay line by  $\pm 5\%$  does not affect the capture range (note that while increasing this supply beyond 5% does not affect the capture range, decreasing it decreases the band width of the delay line, which affects the capture range). The measured capture range decreases by around 10% when a PRBS31 data is used. This is because our FD, relies on data edges to correct the PD phase error.

To verify that the UIA can work in the presence of ISI, the data is fed to the chip through a 3'' and a 7'' FR4 channels. The response of these two channels is shown in Fig. 27. The probe card has a measured loss of 1.8 dB at 6 GHz and 1.5 dB at 5 GHz. No equalization is used in this chip. The 3'' FR4 channel does not decrease the capture range of the CDR, while with the 7'' FR4 channel, the phase detector only acquires lock if the

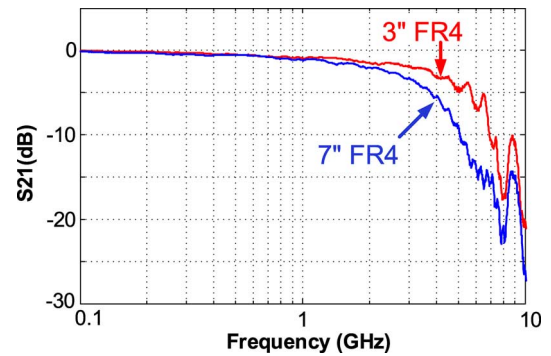


Fig. 27. Transfer characteristic of the 3'' and the 7'' FR4 channels.

TABLE I  
SUMMARY AND COMPARISON WITH PREVIOUS WORK

FD	Type	Tech. (nm)	Lock Rate (Gb/s)	Lock range (%)	FD power (mW)	Does data sample clock in FD?
[7]	Full-rate	65	10	30	NA	yes
[8]	Half-rate	180	3.125	11.52	30.6	yes
[9]	Half-rate	180	10	14.3	42.2	no
[12]	Half-rate	65	10	65	NA	no
[11]	Full-rate	180	10	21	NA	no
[13]	Half-rate	65	10	23.25	8.4	yes
[14]	Full-rate	65	10	20	37	yes
[20]	Full-rate	180	3.125	16	15.5	yes
<b>This work</b>	<b>Half-rate</b>	<b>65</b>	<b>10</b>	<b>36</b>	<b>11</b>	<b>yes</b>

data rate is below 10.35 Gb/s. The capture range with the 7'' FR4 channel is from 8.5 Gb/s to 10.35 Gb/s. This measurement shows that the proposed frequency detection mechanism can operate without the need for equalization for channels with less than 7 dB of loss.

Finally, Table I summarizes the results and compares this paper against previous work. Also, note that simulating all frequency detectors in 65 nm CMOS at 10 Gb/s (with the same gates) result in a power consumption of 6 mW for the proposed FD and 29.5 mW and 28.6 mW for the FDs in [8] and [9], respectively. Since the details of the design in [7] and [20] are not available, they cannot be simulated. Also exact transistor count cannot be obtained.

## VII. CONCLUSION

This paper presents a novel frequency detector based on data phase selection. By feeding the correct data phase into the phase detector, the effect of cycle slipping is mitigated and the CDR capture range is increased to the VCO tuning range. Also, by filtering the pattern out of the FD, this frequency detector tolerates up to 7 dB of loss without compromising the capture range.

## APPENDIX

The lock time of a conventional binary CDR with a first-order RC loop filter is derived in [21]. The analysis is then slightly modified in [14] to obtain the lock time of a binary CDR with a CPR FD. In this Appendix, we derive the lock time of a binary CDR with a CPS FD. For convenience, some of the intermediate steps of the derivation is repeated here. This section assumes that the input is a clock pattern, the loop has a resistance of  $R$  in

series with a capacitance of  $C$ , the charge-pump current is  $I_{CP}$  and the VCO gain is  $k_{VCO}$ .

It was shown in [14] that the change in phase and frequency during a period of  $UP$  or  $DN$  pulse is:

$$\begin{aligned}\Phi_{err}(T) - \Phi_{err}(0) &= \text{sgn}[\Phi_{err}(0)] \times (\Delta\Phi_{loop} + \Delta\Phi_{fos}) \\ f_{err}(T) - f_{err}(0) &= \text{sgn}[\Phi_{err}(0)] \frac{I_{CP}T}{C} k_{VCO}\end{aligned}\quad (5)$$

where  $f_{err} = k_{VCO}V_1 - f_{in}$  and  $\Delta\Phi_{loop}$  and  $\Delta\Phi_{fos}$  are defined as:

$$\begin{aligned}\Delta\Phi_{loop} &= 2\pi k_{VCO}T \left( RI_{CP} + \frac{TI_{CP}}{2C} \right) \\ \Delta\Phi_{fos} &= 2\pi T f_{err}.\end{aligned}\quad (6)$$

It shows in [21] that for a conventional CDR the number of  $UP$  and  $DN$  pulses during each cycle-slip period is:

$$\begin{aligned}N_{DN} &= \frac{\pi - \Delta\Phi_{loop}}{2\pi f_{ave}T - \Delta\Phi_{loop}} \\ N_{UP} &= \frac{\pi + \Delta\Phi_{loop}}{2\pi f_{ave}T + \Delta\Phi_{loop}}\end{aligned}\quad (7)$$

where  $f_{ave}$  is the average value of frequency error during a cycle-slipping period.

Now, let us revisit Fig. 4(b). While a conventional CDR increases the VCO frequency for  $TN_{UP}$  and decreases it for  $TN_{DN}$ , the CPS FD increases the VCO frequency for  $TN_{UP}/2$  and decreases it for  $T(N_{DN} + N_{UP}/2)$ . This, combined with (6) yield the total change in the CDR frequency error during a cycle-slipping period:

$$\begin{aligned}\Delta f_{ave} &= \frac{I_{CP}T}{C} k_{VCO} \left( \frac{N_{UP}}{2} - N_{DN} + \frac{N_{UP}}{2} \right) \\ &= -\frac{I_{CP}T}{C} k_{VCO} N_{DN}.\end{aligned}\quad (8)$$

Since each cycle-slip period takes  $(N_{DN} + N_{UP})T$ , the change in frequency error versus time can be expressed as:

$$\begin{aligned}\frac{df_{ave}}{dt} &= -\frac{I_{CP}k_{VCO}}{C} \\ &\times \frac{(\pi\Delta\Phi_{loop} - \Delta\Phi_{loop}^2) + (2\pi^2T - 2\pi\Delta\Phi_{loop}T)f_{ave}}{4\pi^2Tf_{ave} - 2\Delta\Phi_{loop}^2}.\end{aligned}\quad (9)$$

Solving the above differential equation yields the lock time of the CDR

$$\begin{aligned}t_{lock} &= \frac{C}{I_{CP}k_{VCO}} \left[ \frac{2\pi(f_0 - f_{avg})}{\pi - \Delta\Phi_{loop}} \right. \\ &\left. + \frac{\Delta\Phi_{loop}(\pi + \Delta\Phi_{loop})}{\pi T(\pi - \Delta\Phi_{loop})} \times \ln \left( \frac{\Delta\Phi_{loop} + 2\pi T f_{avg}}{\Delta\Phi_{loop} + 2\pi T f_0} \right) \right].\end{aligned}\quad (10)$$

For the CDR with UIA, the situation is simpler, as cycle-slipping is completely eliminated. Since  $\text{sgn}[\Phi_{err}(t)]$  is now independent of time, (6) simplifies to:

$$f_{err}(t+T) = f_{err}(0) \pm \frac{I_{CP}T}{C} k_{VCO}\quad (11)$$

$t_{(f_0 - f_{avg})}$  can easily be found to be:

$$t_{(f_0 - f_{avg})} = \frac{C}{I_{CP}k_{VCO}} |f_0 - f_{avg}|.\quad (12)$$

#### ACKNOWLEDGMENT

The authors thank CMC Microsystems for providing measurement equipment and CAD tools, and J. Liang and W. Rahman for technical discussions.

#### REFERENCES

- [1] M. Hsieh and G. Sobelman, "Architectures for multi-gigabit wire-linked clock and data recovery," *IEEE Circuits Syst. Mag.*, pp. 45–57, Sep. 2008.
- [2] L. Henrickson *et al.*, "Low power fully integrated 10-Gb/s SONET/SDH transceiver in 0.13- $\mu\text{m}$  CMOS," *IEEE J. Solid-State Circuits*, vol. 38, no. 10, pp. 1595–1601, Oct. 2003.
- [3] H. Muthali, T. Thomas, and I. Young, "A CMOS 10-Gb/s SONET transceiver," *IEEE J. Solid-State Circuits*, vol. 39, no. 7, pp. 1026–1033, Jul. 2004.
- [4] N. Kalantari and J. Buckwalter, "A multichannel serial link receiver with dual-loop clock-and-Data recovery and channel equalization," *IEEE Trans. Circuits Syst. I*, pp. 2920–2931, Nov. 2013.
- [5] C. Kromer, G. Sialm, C. Menolfi, M. Schmatz, F. Ellinger, and H. Jackel, "A 25-Gb/s CDR in 90-nm CMOS for high-density interconnects," *IEEE J. Solid-State Circuits*, pp. 2921–2929, Dec. 2006.
- [6] B. Zhang *et al.*, "A 195 mW/55 mW dual-path receiver AFE for multi-standard 8.5-to-11.5 Gb/s serial links in 40 nm CMOS," in *IEEE ISSCC Dig. Tech. Papers*, 2013, pp. 34–35.
- [7] N. Kocaman *et al.*, "An 8.5–11.5-Gb/s SONET transceiver with referenceless frequency acquisition," *IEEE J. Solid-State Circuits*, pp. 1875–1884, Aug. 2013.
- [8] R. Yang, S. Chen, and S. Liu, "A 3.125-Gb/s clock and data recovery circuit for the 10-Gbase-LX4 ethernet," *IEEE J. Solid-State Circuits*, vol. 39, no. 8, pp. 1356–1360, Aug. 2004.
- [9] J. Savoj and B. Razavi, "A 10-Gb/s CMOS clock and data recovery circuit with a half-rate binary phase/frequency detector," *IEEE J. Solid-State Circuits*, vol. 38, no. 1, pp. 13–21, Jan. 2003.
- [10] D. Richman, "Color-carrier reference phase synchronization accuracy in NTSC color television," *Proc. IRE*, vol. 42, pp. 106–133, Jan. 1954.
- [11] S. Huang, J. Cao, and M. Green, "An 8.2-to-10.3 Gb/s full-rate linear reference-less CDR without frequency detector in 0.18  $\mu\text{m}$  CMOS," in *IEEE ISSCC Dig. Tech. Papers*, 2014, pp. 152–153.
- [12] S. Guanghua, W. Choi, S. Saxena, T. Anand, A. Elshazly, and P. Hanumolu, "A 4-to-10.5 Gb/s 2.2 mW/Gb/s continuous-rate digital CDR with automatic frequency acquisition in 65 nm CMOS," in *IEEE ISSCC Dig. Tech. Papers*, 2014, pp. 150–151.
- [13] M. Jalali, R. Shivnaraine, A. Sheikholeslami, M. Kibune, and H. Tamura, "An 8 mW frequency detector for 10 Gb/s half-rate CDR using clock phase selection," in *Proc. CICC*, 2013, pp. 1–4.
- [14] R. Shivnaraine, M. Jalali, A. Sheikholeslami, M. Kibune, and H. Tamura, "An 8–11 Gb/s reference-less bang-bang CDR enabled by "Phase reset"," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 61, no. 6, pp. 2129–2138, Jun. 2013.
- [15] J. Savoj and B. Razavi, "A 10-Gb/s CMOS clock and data recovery circuit with a half-rate linear phase detector," *IEEE J. Solid-State Circuits*, vol. 36, no. 5, pp. 761–768, May 2001.
- [16] A. Rezaee and K. Martin, "A 9–16 Gb/s clock and data recovery circuit with three-state phase detector and dual-path loop architecture," in *Proc. IEEE ESSCIRC*, 2004, pp. 683–686.
- [17] W. Choi, T. Anand, S. Guanghua, and P. Hanumolu, "A fast power-on 2.2 Gb/s burst-mode digital CDR with programmable input jitter filtering," in *IEEE VLSI Symp. Dig.*, 2013, pp. C280–C281.
- [18] J. Liang, M. Jalali, A. Sheikholeslami, M. Kibune, and H. Tamura, "On-chip measurement of clock and data jitter with sub-picosecond accuracy for 10 Gb/s multilane CDRs," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 845–855, Apr. 2015.
- [19] M. Nogawa *et al.*, "A 10 Gb/s burst-mode CDR IC in 0.13  $\mu\text{m}$  CMOS," in *IEEE ISSCC Dig. Tech. Papers*, 2005, pp. 228–235.
- [20] K. Hsiao, M. Lee, and T. Lee, "A clock and data recovery circuit with wide linear range frequency detector," in *Proc. IEEE Int. Symp. VLSI Design, Automat. Test*, 2008, pp. 121–124.

- [21] M. Chan and A. Postula, "Transient analysis of bang-bang phase locked loops," *IET Circuits Devices Syst.*, vol. 3, pp. 76–82, 2009.



**Mohammad Sadegh Jalali** (S'11–M'14) received the Bachelor degree (with honors) in electrical engineering from the University of Tehran, Tehran, Iran, the Master degree from the University of British Columbia, Vancouver, BC, Canada, and the Ph.D. degree from the University of Toronto, Toronto, ON, Canada, in 2008, 2010, and 2014, respectively.

In 2014, he joined Semtech-Snowbush IP, and has been engaged in the development of multistandard SerDes IP.



**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, Canada, in 1994 and 1999, respectively, all in electrical engineering.

In 1999, he joined the Department of Electrical and Computer Engineering at the University of Toronto where he is currently a Professor. He was on research sabbatical with Fujitsu Labs in 2005–2006, and with Analog Devices in 2012–2013. His research interests are in analog and digital integrated circuits, high-speed signaling, and VLSI memory design. He has coauthored over 50 journal and conference articles and 8 patents.

Dr. Sheikholeslami served on the Memory, Technology Directions, and Wireline Subcommittees of the ISSCC in 2001–2004, 2002–2005, and 2007–2013, respectively. He is currently an Associate Editor for the Solid-State Circuits Magazine and the Educational Events Chair for ISSCC. He was an Associate Editor for the IEEE TCAS-I for 2010–2012, and the program chair for the

2004 IEEE ISMVL. He has received numerous teaching awards including the 2005–2006 Early Career Teaching Award and the 2010 Faculty Teaching Award, both from the Faculty of Applied Science and Engineering at the University of Toronto. He is a registered professional engineer in Ontario, 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., Kanagawa, Japan. He has been engaged in research and design of high-speed IO with CMOS.



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

He joined Fujitsu Laboratories in 1982. After being involved in 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 multi-channel 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.