An 8–11 Gb/s Reference-Less Bang-Bang CDR Enabled by “Phase Reset”

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

***Abstract*—This paper embeds a “phase-reset” scheme into a bang-bang clock and data recovery (CDR) to periodically realign the clock phase to the data rising edge using a gated-VCO. This reduces both the CDR lock time and bit errors during pull-in, while increasing the CDR capture range. The CDR is fabricated in 65-nm CMOS, operates at 8–11 Gb/s, and demonstrates a 9× increase in capture range. The CDR consumes 84 mW during lock, and 48 mW in steady state.**

***Index Terms*—Burst-Mode CDR, Clock and data recovery, Cycle-slipping, Gated VCO.**

I. Introduction

The demand for bandwidth in internet applications is increasing in both consumer and back-end communication links. Supporting this demand is often accomplished by the use of multiple channels, and faster individual lanes. However, this has resulted in a rise in power consumption. To curb this increase in power consumption, techniques using a lower supply or current recycling [1] have been used. In situations where much of the traffic is idle, techniques utilizing quick power-down and start-up can be utilized to save power. This method is well suited for server applications where the data traffic is below 100% in more than 85% of the time [2]. Targeting these applications, we propose a technique to improve the lock time of clock and data recovery (CDR) circuits to facilitate quick power up.

CDR circuits are typically built using phase tracking architectures [3], [4] as illustrated in Fig. 1(a). These CDRs, which are typically deployed in applications with continuous data traffic, offer high frequency jitter rejection, and good high frequency jitter tolerance. However, they are not easily adaptable to applications requiring quick lock performance (such as Passive Optical Networks, PON) due to cycle-slipping, where the clock phase drifts relative to the data boundary resulting in a periodic phase error. The bit error rate of phase tracking CDRs before lock is also high. On the other hand, burst-mode CDRs (BM-CDRs), shown in Fig. 1(b), [5], [6], can quickly lock to data and are largely used in applications where data is not sent continuously [7], [8]. Additionally, BM-CDRs can be powered down during periods of idle, which are often present in many signaling applications. However, unlike phase tracking CDRs, BM-CDRs offer poor high-frequency jitter rejection and have reduced high frequency jitter tolerance due to their activity on every data edge. Also, they are sensitive to frequency offset between receiver and transmitter clocks [9], requiring a reference clock to be used for frequency locking.

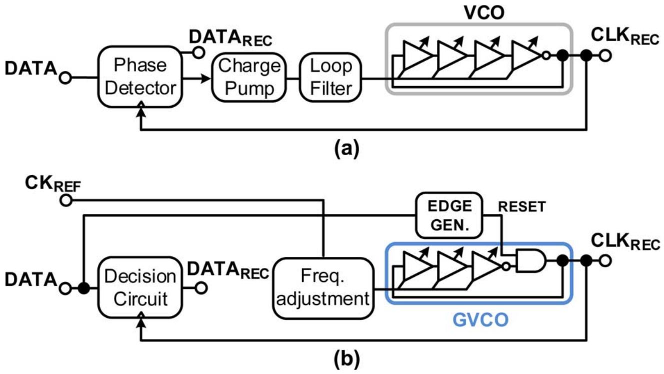


Fig. 1. (a) Conventional phase tracking CDR. (b) Burst-mode CDR.

This paper proposes a scheme in which a gated voltage-controlled oscillator (GVCO) is inserted into a traditional CDR loop to break up cycle-slipping by periodically resetting the phase of the recovered clock. This allows the CDR to quickly settle to the correct control voltage and allows for the correct recovery of bits during pull-in. After the CDR control voltage has settled, the GVCO is no longer reset and the system operates as a conventional CDR. By combining the phase tracking and burst-mode topologies, this work achieves the quick lock time of a BM-CDR and maintains the steady-state jitter performance of a phase tracking CDR. The proposed architecture is both single loop and referenceless.

The remainder of this paper is organized as follows. Section II reviews the underlying issues of the traditional phase tracking CDR and Gated-VCO topologies. Section III presents the proposed work and Section IV describes the circuit implementation of the concept. Section V includes simulation and measurement results of this work and Section VI and VII discuss limitations of this work and conclude the paper.

II. Background

In this section, we review three basic types of CDRs, namely conventional CDR, conventional CDR with frequency detector, and burst-mode CDR.

A. Conventional CDR

A conventional CDR, shown in Fig. 1(a), is composed of a phase detector (PD), charge-pump (CP), loop filter (LF), and

Manuscript received June 29, 2013; revised November 01, 2013; accepted December 29, 2013. Date of publication February 19, 2014; date of current version June 24, 2014. This paper was recommended by Associate Editor F. O’Mahony.

R. Shivnaraine, M. S. Jalali and A. Sheikholeslami are with the Department of Electrical and Computer Engineering, University of Toronto, Canada.

M. Kibune and H. Tamura are with the Fujitsu Laboratories Limited, 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/TCSI.2014.2304668

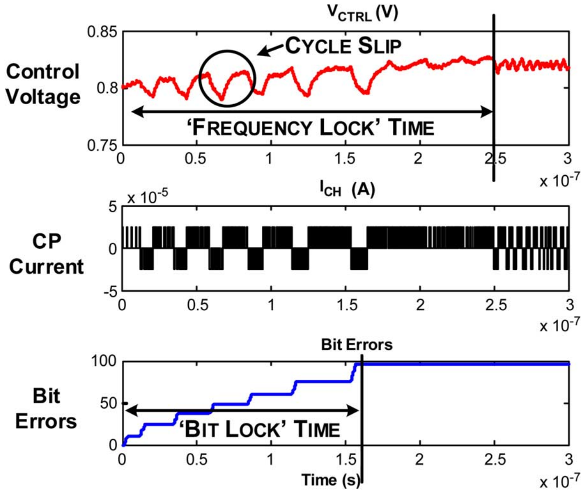


Fig. 2. CDR locking characteristic.

voltage-controlled oscillator (VCO). The phase detector compares the phase of the clock to that of data and thorough the CP, provides a control signal to the loop filter. Since the phase detector is mainly designed to deal with phase offset, its efficiency is limited in the presence of frequency offset. In this case, and during a process which is referred to as "pull-in," the CDR adjusts the control voltage of its VCO so as to bring the VCO frequency close to the data frequency. However, as the CDR control voltage moves toward the direction of reducing the frequency offset, it may momentarily move toward increasing the frequency offset. This process, which is caused by the periodic output of the phase detector, is referred to as cycle-slipping [10].

Shown in Fig. 2 is a behavioral locking characteristic obtained by a Simulink simulation of a phase tracking CDR with binary phase detector. Cycle-slipping is illustrated by the change in the control voltage, V_{CTRL}, which travels repeatedly in the "wrong direction" before settling to its correct value. At lock, the loop transitions from a slow waxing and waning of charge-pump current to a high frequency burst of alternating current which keeps the average of the CP current (proportional to control voltage) almost constant. The CDR does not produce bit errors after lock. Cycle-slipping impacts the system in two different ways. It delays the time it takes for the control voltage to settle which we refer to as "frequency lock time" and delays the time it takes for bit-errors to stop occurring, which we refer to as "bit-lock time." While in a phase tracking CDR these two parameters are coupled, we will show in this paper that the proposed scheme allows for correct phase alignment (bit-lock) even though frequency-lock has not yet been achieved.

An important practical limitation of CDRs with only a phase detector is their limited tolerance to frequency offset caused by cycle-slipping [10]. To extend the CDR's tolerance to frequency offset, a frequency detector is often incorporated into the CDR loop.

B. Conventional CDR With Frequency Detector

A CDR's lock range is typically on the order of its loop bandwidth [10], [11]. To expand the CDR's tolerance to frequency offset, an auxiliary circuit known as a frequency detector (FD) is added to the CDR loop [10], [11] as shown in Fig. 3. The FD compares the frequency of the local clock to that of data and provides a control signal to the loop filter. During lock, both frequency acquisition and phase detection loops are active and the frequency detector is designed to "overpower" the phase detector. After frequency lock, the FD stops producing control signals and the PD takes over and eliminates any residual frequency and phase offset.

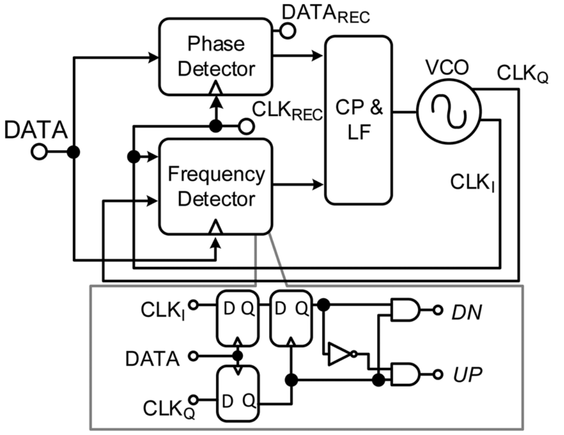


Fig. 3. CDR with frequency detector, rotational FD [12].

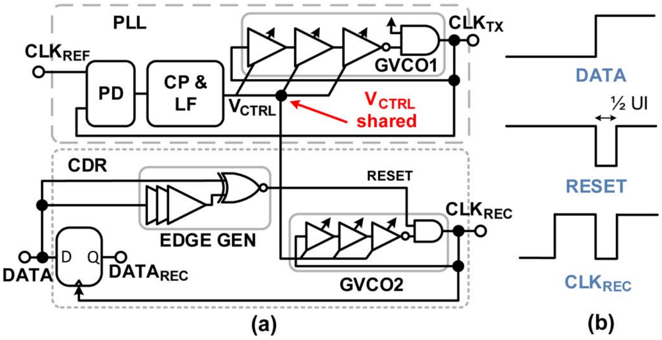


Fig. 4. (a) Block Diagram of BM-CDR with GVCO. (b) Operation waveform.

Rotational frequency detectors [12], [13] compare the movement of two quadrature clocks relative to data. The frequency detector in Fig. 3 [12] operates by sampling two quadrature phases of the local clock, CLK_I and CLK_Q, and looking at their rotation.

C. Burst-Mode CDR

Fig. 4 shows the block diagram of a BM-CDR which consists of a PLL for frequency locking and a CDR for phase locking, along with its timing diagram. The PLL in Fig. 4(a) sets the frequency of the receiver clock (CLK_{REC}). The CDR consists of a GVCO, edge generator, and flip-flop. The edge generator produces a $RESET$ at each data edge which starts and stops oscillation [6]. The delay of the EDGE GEN block can be anywhere from 0UI-1UI, however it is typically designed to be 1/2UI for maximum timing margin. In this case, when oscillation resumes, a rising edge is generated at the center of the data eye to correctly sample the data [see Fig. 4(b)]. However, any mismatch between the two GVCOs results in a static frequency offset between the local clock and data, which limits the CDR's tolerance to continuous identical digits (CID) [8].

In this paper, we propose embedding a GVCO into a conventional CDR loop to reduce cycle-slipping and reduce errors during lock. Reducing cycle-slipping significantly improves the capture range, which eliminates the need for a frequency detector.

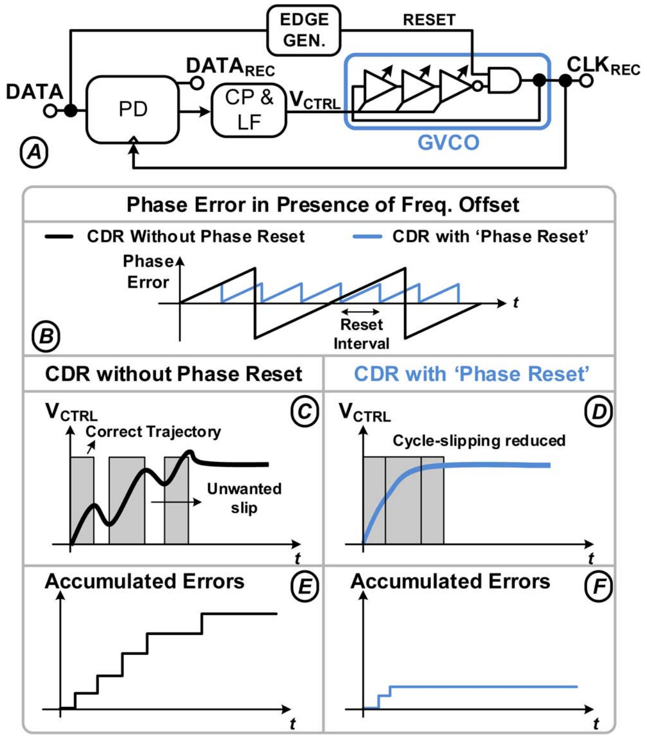


Fig. 5. "Phase reset" scheme. (a) Block diagram of the implementation. (b) Phase error with and without "phase reset." (c) VCO control voltage without "phase reset." (d) VCO control voltage with "phase reset." (e) Errors during lock without "phase reset." (f) Errors during lock with "phase reset."

III. Proposed CDR Architecture

A. Proposed Concept

As mentioned before, a conventional CDR has no frequency offset after lock, but has a limited locking range and a large lock time. In contrast, a BM-CDR has a small lock time, but its performance is deteriorated in the presence of frequency offset, necessitating the use of a reference clock. The proposed concept is illustrated Fig. 5(a), where by combining these two architectures, namely by periodically aligning the phase of the recovered clock with data edge in a conventional CDR loop, we manage to reduce the effect of cycle-slipping. The advantages of the proposed CDR are the following: The VCO control voltage settles faster to the correct value, the CDR has a wider locking range, and bit errors due to cycle-slipping are significantly reduced. The proposed architecture is different from the BM-CDR architecture of Fig. 4 in three ways, 1. The proposed architecture has one loop for both phase and frequency locking while the BM-CDR has two loops, one for frequency (the PLL loop), and one for phase (the CDR loop), 2. The proposed architecture is reference-less while the BM-CDR needs a reference clock 3. The proposed architecture is not sensitive to mismatch as it only uses one GVCO, while the conventional BM-CDR uses two GVCOs, making it sensitive to mismatch.

Shown in Fig. 5(b) is the phase error as a function of time for a CDR without "phase reset" and for one with reset. In the former, the phase error changes sign every time it grows to π, causing a cycle-slip. In the latter, we reset the phase of the clock prior to the phase error reaching π, hence avoiding the sign reversal and cycle-slip. In other words, in the CDR with "phase resets," the phase detector produces an output with a nonzero average, and is biased toward the direction of reducing frequency offset.

Fig. 5(c) and (d) show the control voltage as a function of time for the CDR with and without "phase reset." Cycle-slipping in the former delays the settling time of control voltage, V_{CTRL}, which increases frequency lock time, whereas the latter reduces frequency-lock time. Another important consequence of avoiding cycle-slips is the reduction in the number of errors produced in the CDR with "phase reset." This is illustrated in Fig. 5(e) and (f).

The system uses an Alexander Bang-Bang PD [14] to exploit its full-scale digital output (which is independent of the magnitude of the phase error) toward reducing the frequency offset. This would be in contrast with a linear PD whose output is not at full scale, but is proportional to magnitude of the phase error.

The proposed scheme as presented in Fig. 5(a) requires the edge generator and the NAND gate to respond with zero delay to a data edge in order for the recovered clock to produce an edge in the middle of the data eye. However, in the actual implementation, this delay is nonzero, and hence we must devise a scheme to compensate for this delay. We explain this scheme in Section IV-A.

B. Lock Time of Binary CDR With and Without "Phase Reset"

This section presents an analytical formula for the lock time of an analog CDR with a first-order RC loop filter (a resistance of R and a capacitance of C) with and without "phase reset" (the detailed derivation is relegated to Appendix A). The charge-pump current is I_{CP} and the VCO gain is k_{VCO}. The input to the CDR is a clock pattern. For the conventional case the lock time is found to be (refer to Appendix A)

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

where T is the period of data, f_0 is the initial frequency offset, $f_{avg} = \Delta\Phi_{loop}/2\pi$, and $\Delta\Phi_{loop}$ is

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

The lock time for the CDR with "phase reset" is

$$t_{lock} = \frac{I_{CP}}{C \times k_{VCO}} [f_0 - f_{avg}]. \quad (3)$$

Fig. 6 plots the lock time of the above CDRs versus frequency offset and compares the results of analytical (1) and (3) with behavioral simulations. Since the results are symmetrical, only the part with positive frequency offsets is shown. At 10% frequency offset, a 4× reduction in lock time is achieved when "phase reset" is used.

For a CDR with a second-order loop filter and PRBS7 input pattern, the analysis become complex and we thus resort to simulation results only. Fig. 7 compares the behavioral lock time of a CDR without resets with that of a CDR with resets every four and eight data edges. The model uses a CDR with a tuning range from 9 GHz to 11 GHz centered at 10 GHz and a PRBS7 data input. A lock time of infinity is assumed in this figure when the CDR does not lock. Without "phase resets," the CDR has a

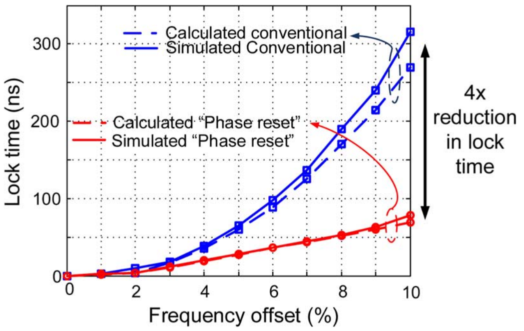


Fig. 6. Analytical and behavioral simulations of CDR lock time with clock pattern and a first-order loop filter. For both systems, $I_{CP} = 100\ \mu$A, $R = 780\ \Omega$ and $C = 16$ pf.

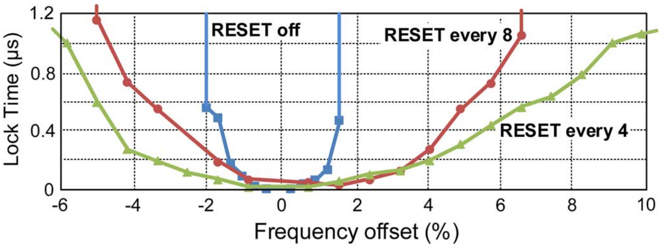


Fig. 7. Behavioral simulations of CDR lock time.

locking range of approximately 9.8 GHz to 10.167 GHz (−2% to 1.67% of the center frequency). With "phase resets" every eight data edges, the CDR lock range is increased to 9.5 GHz to 10.667 GHz (−5% to 6.67%) and resetting every 4 increases it further to a range of 9.417 GHz to 11 GHz (−5.83% to 10%). At −2% frequency offset, the lock time of the CDR with no "phase reset" is 550 ns, while this number is decreased to about 250 ns and 100 ns if we reset the CDR every eight and four data edges, respectively. As expected, the CDR lock time is decreased and the CDR capture range is increased by resetting phase more often.

IV. Circuit Implementation

A. Detailed Implementation

The detailed implementation of the proposed concept is shown in Fig. 8; highlighted blocks are powered down during different phases of operation. Prior to lock, pulses are generated by the edge detector block. These pulses reset the GVCO phase to align the recovered clock to data. After a reset, the phase detector guides the CDR in the direction of reducing frequency offset.

The proposed scheme as presented in Fig. 5(a) requires the edge generator and the NAND gate to respond immediately to a data edge in order for the recovered clock to produce an edge in the middle of the data eye. However, in the actual implementation, this delay is nonzero, and hence we must device a scheme to compensate for this delay. Assume the time it takes to reset the GVCO is, $t_{data-reset}$. Additionally, assume the latency of clock buffers between the GVCO and phase detector also takes, $t_{reset-clock}$. Therefore, we must compensate for a total delay of t_{delay} (equal to $t_{data-reset} + t_{reset-clock}$) in order for the recovered clock to be aligned to data after a reset. To this end, we propose adding a delay block, ΔT in the data path. This delays the data until CLK_{REC} arrives at the phase detector following a "phase reset." To calibrate the delay line, a delay control loop is incorporated into the system. During calibration ($CDR_{ENA} = 0, CAL = 1, R = 0$), "phase resets" are not performed (see Fig. 9 for equivalent system during calibration). In this mode, the CDR loop is opened and a divided-by-8 version of the GVCO's output is used as a "mock data source." The delay of the ΔT block is then compared to the delay of the reset operation. This is achieved through bypassing the GVCO as the recovered clock and using the "mock source" to exercise the edge-detector and GVCO's gating logic delays. Since the GVCO is set to free-run and acts as a "data source," the delays of the edge-detector and GVCO gating logic are accounted for through the use of replica blocks. The phase of the two paths is compared by using the "edge" sample $(U/\overline{D})$ of the phase detector. The PD's edge sample is used as the Up or $Down$ control of a saturating counter to adjust the delay-line.

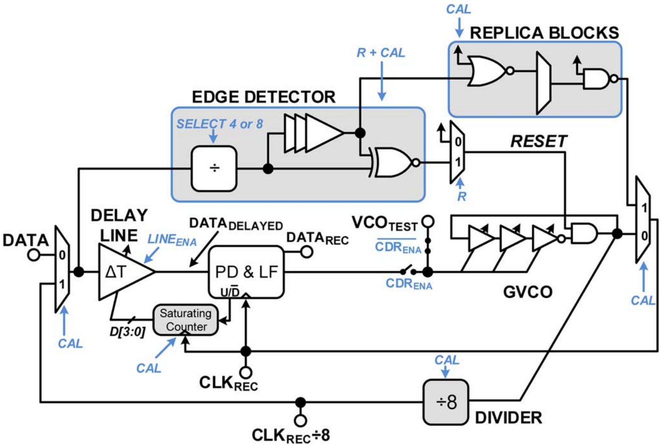


Fig. 8. Detailed implementation of the system with "phase reset."

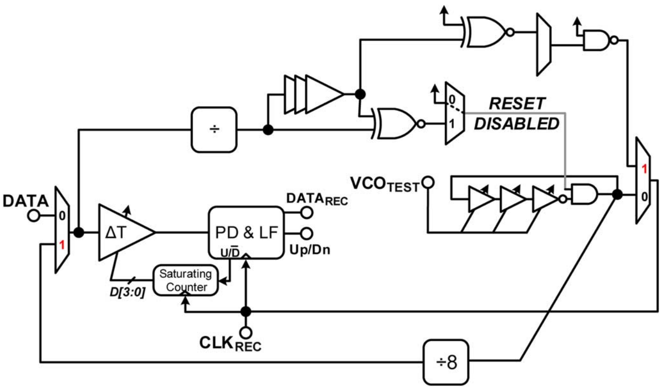


Fig. 9. Equivalent system during calibration phase.

Fig. 10 shows the equivalent system after calibration is complete (during pull-in). The divider, replica blocks, and counter are all powered down and the "phase reset" mode is enabled ($CDR_{ENA} = 1, CAL = 0, R = 1$).

Once the CDR achieves lock, it is switched to "normal operation" ($CDR_{ENA} = 1, CAL = 0, R = 0$) and the edge detector is powered down. In this mode, "phase resets" are not performed and all highlighted blocks in Fig. 8 are powered down, and the control loop is identical to that of a conventional CDR. To characterize the GVCO's frequency and initialize the CDR to a fixed data rate for capture range measurements, the loop filter switch, CDR_{ENA}, is used.

To compensate for frequency offsets below 10%, the GVCO is reset every four or eight data rising edges.

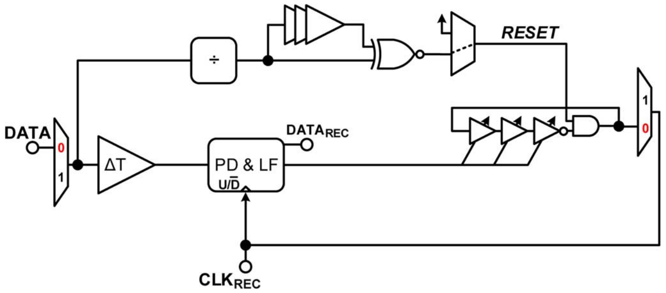


Fig. 10. Equivalent system during pull-in phase.

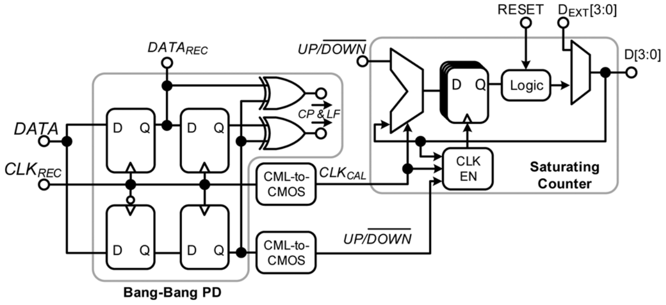


Fig. 11. Saturating counter in delay calibration circuit.

B. Delay Calibration Control Circuit

The delay calibration controller consists of a saturating counter as shown in Fig. 11. This controller compares the phase of the ΔT block and the phase of the reset path by reusing the edge flip-flop of the Alexander phase detector. When $UP/\overline{DOWN}$ is "high" or "low," the delay of the ΔT block, a 4-bit Digitally Controlled Delay Line (DCDL) is increased or decreased respectively. When $UP/\overline{DOWN}$ signal toggles, calibration is completed. The DCDL's code (D[3:0]) is capable of being set externally by D_{EXT}[3:0].

Any residual error that may remain between ΔT and t_{delay} (delay of a reset operation) causes an induced frequency offset, which we explain here. Consider the case of a conventional CDR when locked. The CDR has an equal number of early and late events, and on average maintains a constant control voltage. Adding "phase resets" periodically injects a skew between clock and data, making the number of early and late events unbalanced. To compensate, the CDR changes its frequency such that in the presence of a constantly injected skew the number of early and late events is on average equal. This new locked condition occurs because of an induced frequency offset. This is an unwanted effect because it is desired to disable resetting after both frequency and phase are locked. To mitigate the induced frequency offset, delay line resolution can be increased or resetting phase may be performed less frequently. In this work, we reset phase less frequently by using a divider in the edge generator. By updating phase every nth data edge, the timing skew injected into the loop is averaged over the longer reset period.

Fig. 12 shows the RC-extracted simulation results of delay calibration control loop. After the reset is released, the $DLLcode$ oscillates between 3 and 6. The calibration block picks the code at the falling edge of the $UP/\overline{DOWN}$ signal. Each DLL code corresponds to a delay change of 2–4 ps.

C. Delay Line

The delay-line needs to compensate for roughly 300 ps of delay, and must also have little intersymbol interference (ISI).

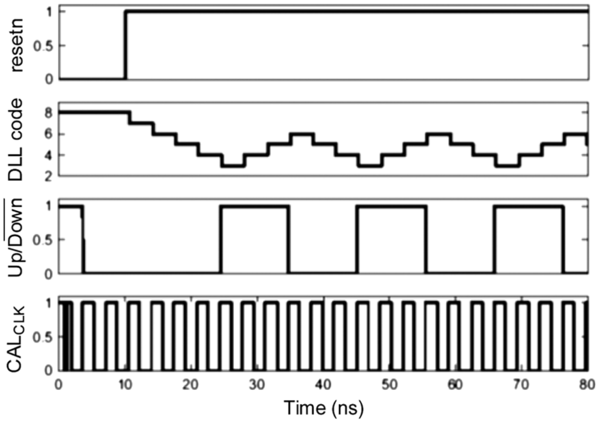


Fig. 12. Simulation (RC-extracted) results of delay calibration control loop.

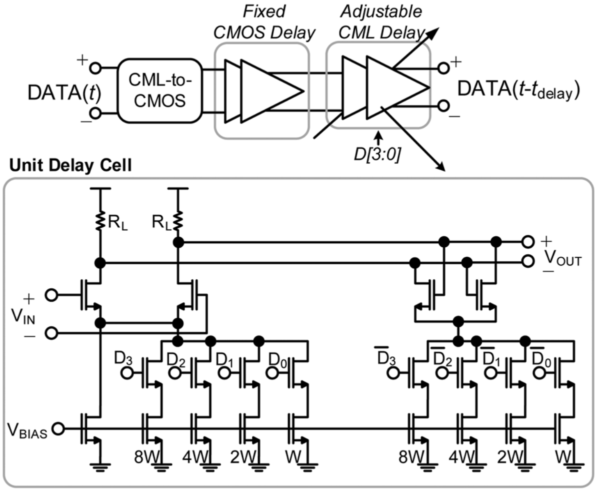


Fig. 13. Delay-line implementation.

One option is to use several CML stages and multiplex between them [5] but this burns a significant amount of power. Phase-mixing delay cells [15], [16] commonly used for clocks were found to introduce a significant amount of jitter for the large delay required. The same result was found for current-starved CMOS inverter delay chains as used in [17]. The solution we use (Fig. 13) relies on CMOS inverters to buffer the signal with low ISI, and provide the bulk of the required delay with low power consumption. The remaining delay was done with an adjustable 4-bit CML delay line. Postlayout simulations showed that the block produced little duty-cycle distortion (DCD), as a result DCD correction was not implemented.

D. Divider

To explore the relationship between capture range and the frequency of "phase resets," a programmable divider chain was included (Fig. 14). The divider was built using a synchronous counter using CML gates. For layout simplicity, a divider chain with more outputs was avoided to keep the number of inputs to each gate to two. One important consideration of the divider chain is that the outputs should have similar delays to limit the required tuning range of the delay line. At the circuit level, the divided outputs have similar delays due to all clocks being generated by a synchronous counter. During postlayout verification, delays were kept close by routing each output to be roughly capacitance matched.

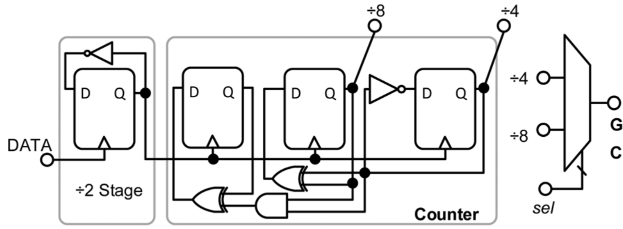


Fig. 14. Divider implementation with synchronous counter.

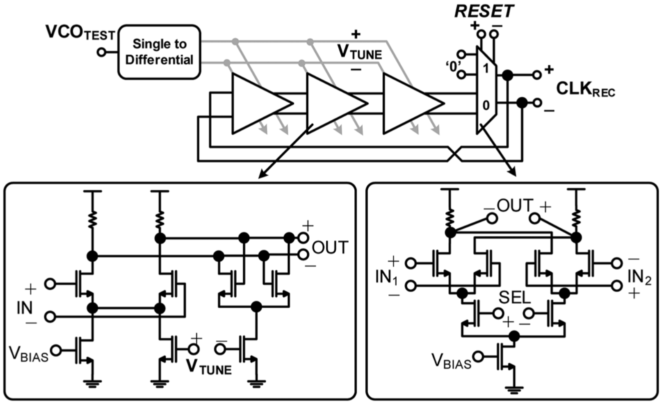


Fig. 15. Gated-VCO implementation.

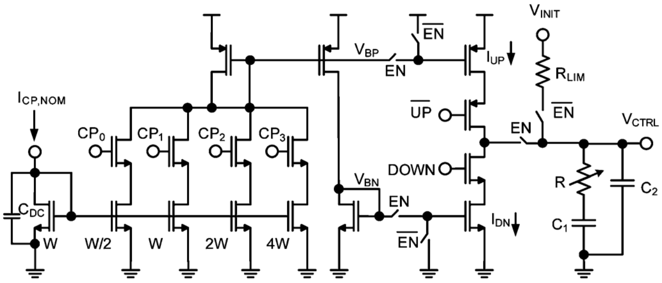


Fig. 16. Programmable charge-pump and loop filter.

E. Gated-VCO

Fig. 15 shows the schematic diagram for the Gated-VCO. The gating block is built using a 2-to-1 multiplexer which either passes the input from the VCO delay cells to the output, starting oscillation, or passes "0" to the output, stopping oscillation. The GVCO 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 its delay. V_{TUNE} is used differentially to maintain a constant common-mode at the VCO output.

F. Programmable Charge-Pump and Loop Filter

Fig. 16 shows the CDR charge-pump and loop filter. The circuit allows for the VCO control voltage, V_{CTRL}, to be driven externally by V_{INIT} to initialize the CDR to a desired data rate. This allows the GVCO tuning range to be measured without the need for a replica GVCO break-out circuit. The loop filter resistance can also be varied by ±10% to shift CDR loop bandwidth if required, and the loop filter current can be digitally controlled by CP_{3-0}.

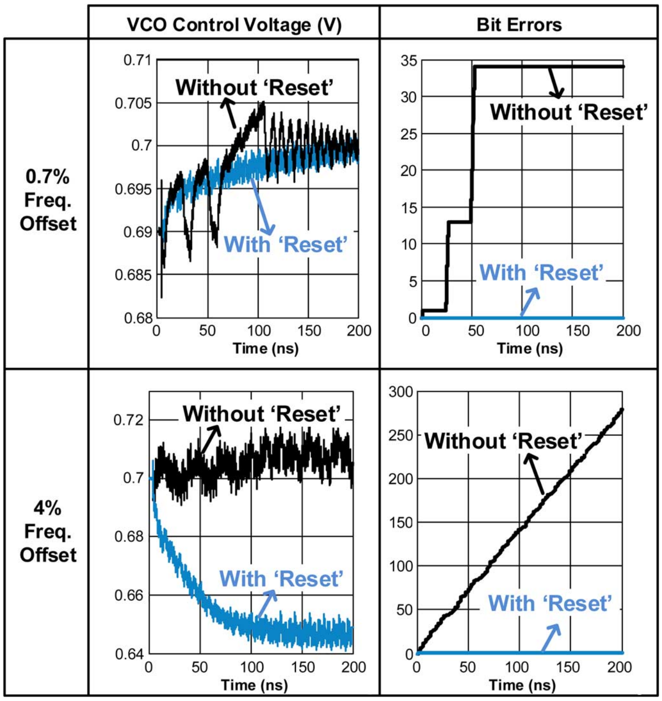


Fig. 17. Simulated (RC-extracted) locking characteristics.

V. Simulation and Measurement Results

A. Simulated Locking Characteristics

Fig. 17 shows simulated (RC-extracted) locking characteristics of the CDR without resets, and with resets performed every four data rising edges for a 10 Gb/s PRBS7 data pattern. During lock, the charge-pump current is increased to eight times its nominal value. In the presence of a 0.7% frequency offset, the CDR without resets cycle-slips while the CDR with resets does not. Unlike the CDR without reset, the proposed scheme produces no errors even though the CDR has not settled to its final value. This is because the proposed CDR is kept at the correct sampling position due to the "phase reset" operations. At a frequency offset of 4%, the CDR without "phase reset" is unable to lock (error count increases without bound) while the proposed scheme achieves lock with no error.

B. Measurement Results

Fig. 18 shows the measurement setup and die photograph of this work. The circuit was fabricated in Fujitsu's 65-nm CMOS process, using a 1.2 V supply. The CDR circuits occupy an area of $600 \times 200\ \mu\text{m}^2$ and the loop filter occupies $200 \times 175\ \mu\text{m}^2$. A signal generator is used as the clock source for the BERT's pattern generator. The CDR recovers the data from the BERT, and outputs the retimed data and clock. The retimed data is viewed on an oscilloscope, and the recovered clock's frequency is verified using a spectrum analyzer. The delay-line calibration codes and bit-error counter value for the on-chip BERT are available on the DIG_{OUT} bus which is monitored using a logic analyzer. Programming of internal registers was performed using an FPGA. DC power supplies were used for supply power rails and bias currents.

To characterize the GVCO frequency within the CDR, the CDR loop is opened and the GVCO control voltage is provided off-chip ($EN = 0$, Fig. 16). Measured results for the GVCO

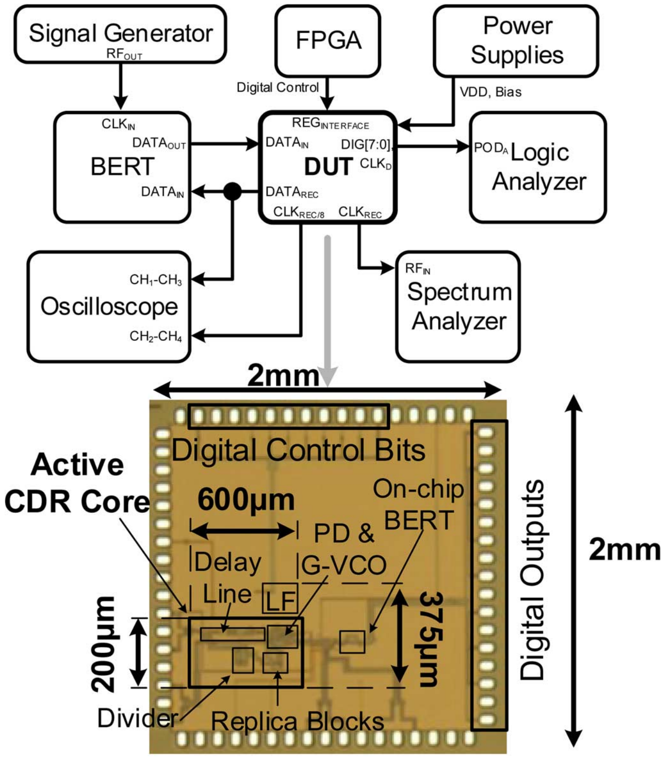


Fig. 18. Test setup and die photograph.

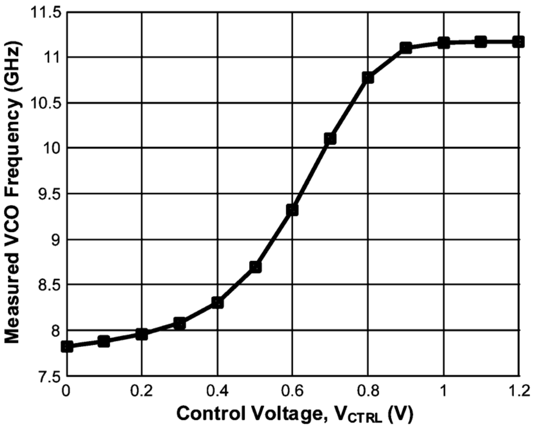


Fig. 19. Measured GVCO tuning range.

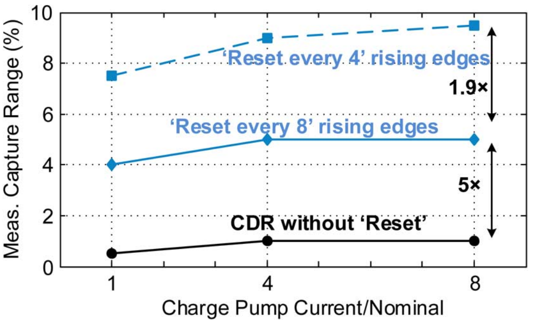


Fig. 20. Measured capture range.

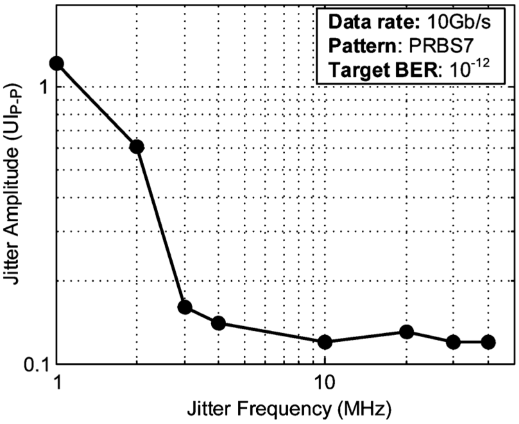


Fig. 21. Measured jitter tolerance.

tuning range are shown in Fig. 19. The measured tuning range of the oscillator is from 7.8 to 11.2 GHz.

Fig. 20 shows the CDR's measured capture range with and without resets for a 10 Gb/s PRBS7 data pattern. Since the lock-range of a CDR is typically boosted by increasing its charge-pump current, the CDR capture range was measured for $1\times (= 125\ \mu\text{A})$, 4× and 8× the nominal current of the charge-pump. For a given mode, increasing current from 4× and 8× did not significantly improve capture range since the charge-pump current is already at a high value. The proposed solution improves capture range by up to five times when resets are performed every eight rising edges and up to nine times when resets are performed every 4 rising edges. The measured CDR jitter tolerance (normal mode, "phase reset" disabled) with a 10 Gb/s PRBS7 pattern at a BER of 10^{-12} is shown in Fig. 21. At high frequencies, the CDR's jitter tolerance is $0.12UI_{P-P}$.

Shown in Fig. 22 are the total errors accumulated during the lock process. Due to measurement limitations, the total number of errors after pull-in were recorded instead of bit errors vs. time. This measurement was performed using an on-chip BERT with an 8-bit error counter. The VCO frequency is initialized to a control voltage corresponding to 8.4 GHz (8.4 Gb/s) via VCO_{INIT} (see Fig. 16) and the incoming data (PRBS7) is set to various frequency offsets as shown. The CDR's loop-filter enable signal, CDR_{ENABLE}, and $BERTRESET$ are activated at the same time and the error count is observed. With a frequency offset of 0.5% both with and without reset have no errors with a charge-pump current of eight times the nominal. The CDR without "phase reset" has errors at the lower charge-pump currents, and "phase resetting" eliminates these errors at all charge-pump currents. At a frequency offset of 0.75%, the "no reset" only locks when the charge-pump current is increased to 4× or 8×. Even for the increased CP current, the error count is larger than 255. On the other hand, "reset every 8" locks even with the 1× CP current. The "reset every 4" shows the best performance as it locks without any errors at all. With a 4% frequency offset, "no reset" does not achieve lock, reset every eight locks after saturating the error counter, and reset every four produces as few as 25 errors with a charge-pump current of 8× the nominal case. The reduction in bit errors during pull-in indirectly demonstrates that the CDR lock time has decreased. Although a bit-lock time measurement could not be performed, the simulation results in Fig. 17 demonstrated an improvement in lock time by over 50 ns for a 0.7% frequency offset.

The half-rate recovered data eye for PRBS7 is shown in Fig. 23. As expected, the eye is fully open after lock. The jitter

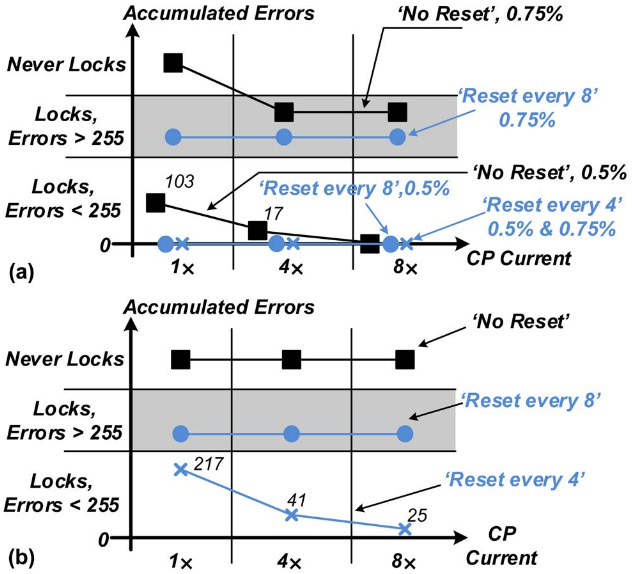


Fig. 22. Measured errors during lock at frequency offsets of (a) 0.5% and 0.75% and (b) 4%.

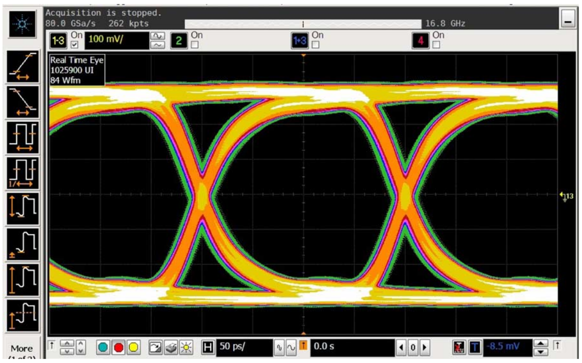
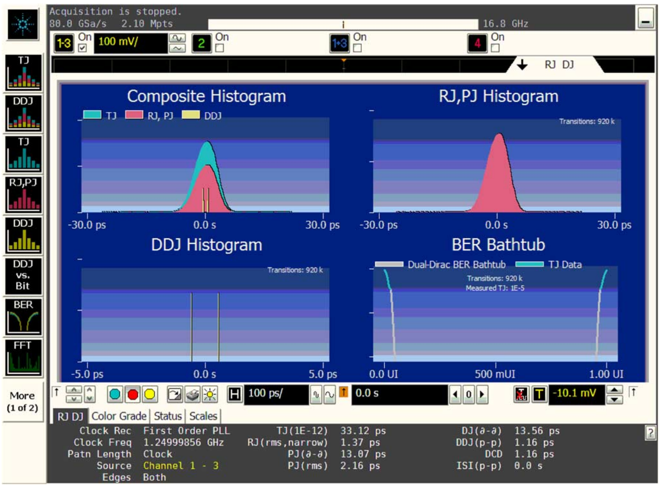

Acquisition is stopped.
80.0 GSa/s 262 kpts 16.8 GHz
1-3 On ☑ 100 mV/ 2 On □ 1+3 On □ 4 On □
Real Time Eye
1025900 UI
84 Wfm
50 ps/ 0.0 s 8.5 mV

Fig. 23. Measured half-rate recovered data eye.


Acquisition is stopped.
80.0 GSa/s 2.10 Mpts 16.8 GHz
1-3 On ☑ 100 mV/ 2 On □ 1+3 On □ 4 On □

100 ps/ 0.0 s -10.1 mV

```
   Clock Rec  First Order PLL          TJ(1E-12)  33.12 ps      DJ(∂-∂)  13.56 ps
  Clock Freq  1.24999856 GHz    RJ(rms,narrow)  1.37 ps      DDJ(p-p)  1.16 ps
 Patn Length  Clock                    PJ(∂-∂)  13.07 ps          DCD  1.16 ps
      Source  Channel 1 - 3            PJ(rms)  2.16 ps      ISI(p-p)  0.0 s
       Edges  Both
```

Fig. 24. Measured spectrum of $CLK_{REC}/8$.

histogram of $CLK_{REC}/8$ after lock is shown in Fig. 24. The total peak to peak jitter on $CLK_{REC}/8$ is 33.12 ps, verifying that the clock is clean after lock.

The CDR operates from a 1.2 V supply and at 8× the nominal CP current, consumes 84 mW, 72 mW, and 48 mW, during ΔT calibration, during "phase reset" being on, and during normal operation, respectively. When the delay line (ΔT) is powered down ($LINE_{ENA} = 0$, Fig. 8), the CDR power without any added enhancement of reset is 35 mW.

TABLE I
COMPARISON OF CDR RESULTS

CDR	Type	Tech. (nm)	Data Rate (Gb/s)	Lock range (%)	Errors during Pull-in	Ref. CK needed
[6]	BM	130	10	NA	few	yes
[9]	BM	65	1-6	NA	few	yes
[20]	BM	55	10	NA	many	yes
[18]	PT	180	3.125	11.52	many	no
[19]	PT	65	10	30	many	no
[22]	PT	65	10	23.25	many	no
This work	**PT**	**65**	**10**	**± 10**	**few**	**no**

Finally, Table I summarizes the results and compares this work against previous work. PT in this table denotes phase tracking.

VI. DISCUSSION

The proposed architecture in this paper assumes a front-end equalizer that provides some equalization in order to clean the ISI-induced jitter in the data and boosts its transitions. Not having an equalizer in the front-end may result in increased BER. Since the system uses the data rising edge to perform a "phase reset," ISI may lead to different delays through the reset path and delay line. This delay mismatch introduces a timing skew, which as discussed earlier, induces a freq. offset, and causes cycle-slipping and bit errors when "phase reset" is turned off. This effect can be mitigated by resetting phase less frequently, but this may add more latency to the reset path and necessitates a large delay line. Other FD circuits such as those in [12] and [22] also use the rising edge of data to sample the clock, and hence suffer similarly from sensitivity to ISI.

It is worth mentioning that prior to lock, the random jitter of data is transferred to the recovered clock. However, this does not prevent CDR from locking as it is evident from our measurement results which includes a 0.1UIpp of random jitter. In addition, this jitter transfer only occurs prior to lock as we turn off the "phase reset" operation after lock.

In this work, "phase reset" is disabled a fixed time after the CDR is enabled. The time to disable resets can be determined by simulating the CDR loop and choosing a time which is larger than the expected settling time. However, this may result in "phase resets" being performed for longer than required, which increases the overall power consumption of the CDR. By incorporating a lock detector into the CDR, the time to disable resets can be determined as the CDR is running.

VII. CONCLUSION

The work presented in this paper places a GVCO within a phase tracking CDR to speed up the bit-lock time and to reduce cycle-slipping. Resetting of the oscillator can be halted after lock to operate the CDR as a conventional CDR. A test-chip was fabricated in Fujitsu's 65-nm process and operates at 8–11 Gb/s. An improvement in CDR capture range of up to 9× was demonstrated. A reduction in the number of errors during the lock process was also shown. The inclusion of "phase resetting" blocks did not hinder the "standalone" CDR performance, and the CDR achieved a BER of better than 10^{-12} at 10 Gb/s.

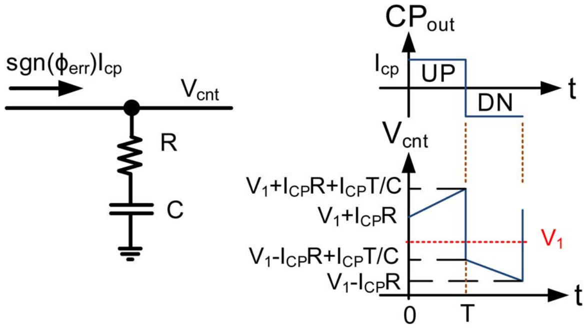


Fig. 25. Loop filter response in presence of an UP, DN command.

Appendix A

To characterize the CDR lock time and capture range [21], assume that a first-order RC loop filter is used in the CDR, and assume that the input is a clock pattern. Assume that the PD outputs an UP signal (at $t = 0$) followed by a DN signal. Both signals have a width of T, which is the clock period. This is shown in Fig. 25.

During the UP pulse, the VCO frequency can be expressed as

$$f_{VCO}(t) = \frac{I_{cp}}{C} k_{VCO} t + I_{cp} k_{VCO} R + k_{VCO} V_1 \qquad (4)$$

where I_{cp} is the charge-pump current, k_{VCO} is the VCO gain, V_1 is the voltage across C just before the UP pulse, and R and C are the values of the loop filter resistance and capacitance, respectively.

Phase error is defined as

$$\Phi_{err}(t) = \Phi_{err}(t_0) + 2\pi \int_{t_0}^{t} \left[f_{VCO}(t) - f_{in} \right] dt. \qquad (5)$$

Therefore, the phase error at the end of the UP pulse can be found by integrating (5)

$$\begin{aligned} \Phi_{err}(T) - \Phi_{err}(0) &= 2\pi k_{VCO} R I_{CP} T \\ &+ \pi k_{VCO} \frac{I_{CP}}{C} T^2 + 2\pi T (k_{VCO} V_1 - f_{in}). \end{aligned} \qquad (6)$$

The above equation can be simplified to

$$\begin{aligned} \Phi_{err}(T) - \Phi_{err}(0) &= 2\pi k_{VCO} R I_{CP} T \\ &+ \pi k_{VCO} \frac{I_{CP}}{C} T^2 + 2\pi T f_{err} \end{aligned} \qquad (7)$$

where $f_{err} = k_{VCO} V_1 - f_{in}$. Repeating the above steps assuming a DN pulse arrives first, and generalizing the results yields

$$\begin{aligned} \Phi_{err}(T) - \Phi_{err}(0) &= sgn\left[\Phi_{err}(0)\right] \\ &\times 2\pi k_{VCO} T \left(R I_{CP} + \frac{T I_{CP}}{2C} \right) + 2\pi T f_{err}. \end{aligned} \qquad (8)$$

The above result can be used to gain insight into CDR design. The first term shows how the phase error is affected by the loop, while the second term shows how it is affected by the existing frequency offset. Let us now define $\Delta\Phi_{loop}$ and $\Delta\Phi_{fos}$ as

$$\begin{aligned} \Delta\Phi_{loop} &= 2\pi k_{VCO} T \left(R I_{CP} + \frac{T I_{CP}}{2C} \right) \\ \Delta\Phi_{fos} &= 2\pi T f_{err}. \end{aligned} \qquad (9)$$

$\Delta\Phi_{loop}$ and $\Delta\Phi_{fos}$ affect CDR locking in the following manner:

$$\begin{aligned} \Delta\Phi_{loop} &\ll \Delta\Phi_{fos} : \text{CDR does not lock} \\ \Delta\Phi_{loop} &< \Delta\Phi_{fos} : \text{CDR locks with cycle} - \text{slipping} \\ \Delta\Phi_{loop} &> \Delta\Phi_{fos} : \text{CDR locks without cycle} - \text{slipping}. \end{aligned} \qquad (10)$$

The change in the frequency error in Fig. 25 during an UP or a DN pulse can be written as

$$f_{err}(t+T) = f_{err}(0) + sgn\left[\Phi_{err}(0)\right] \frac{I_{CP} T}{C} k_{VCO}. \qquad (11)$$

In order to find the lock time, the reduction of the frequency error over each period of cycle-slipping should be found. This was found in [21] and it was shown that for a conventional bang-bang CDR

$$\begin{aligned} t_{(f_0 - f_{avg})} &= \frac{C}{2\Delta\phi_{loop} I_{CP} k_{VCO}} \Bigg[2\pi (f_{avg} - f_0) T \\ &+ \frac{\pi^2 - \Delta\phi_{loop}^2}{\pi} \times ln \left(\frac{1 - 2\pi T f_{avg}}{1 - 2\pi T f_0} \right) \Bigg] \end{aligned} \qquad (12)$$

where $t_{(f_0 - f_{avg})}$ is the time that it take for the frequency error to change from f_0 to f_{avg}.

For the proposed "phase reset" scheme, the situation is simpler, as theoretically, our proposed scheme eliminates cycle-slipping altogether. The CDR moves monotonically toward the locked position (this claim is verified in Fig. 17). Since $sgn[\Phi_{err}(t)]$ is now independent of time, (11) simplifies to

$$f_{err}(t+T) = f_{err}(0) \pm \frac{I_{CP} T}{C} k_{VCO}. \qquad (13)$$

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

$$t_{(f_0 - f_{avg})} = \frac{C}{I_{CP} k_{VCO}} |f_0 - f_{avg}|. \qquad (14)$$

Acknowledgment

The authors thank the anonymous reviewers for their comments and CMC Microsystems for providing measurement equipment and CAD tools and NSERC for funding.

References

[1] R. Inti, A. Elshazly, B. Young, W. Yin, M. Kossel, T. Toifl, and P. K. Hanumolu, "A highly digital 0.5-to-4 Gb/s 1.9 mW/Gb/s serial-link transceiver using current-recycling in 90 nm CMOS," in *Solid-State Circuits Conf. Digest of Technical Papers (ISSCC)*, Feb. 2011, pp. 152–154.

[2] J. G. Koomey, S. Berard, M. Sanchez, and H. Wong, "Implications of historical trends in the electrical efficiency of computing," *IEEE Ann. History Comput.*, vol. 33, pp. 46–54, Mar. 2011.

[3] M. Pozzoni *et al.*, "A multi-standard 1.5 to 10 Gb/s latch-based 3-tap DFE receiver with a SSC tolerant CDR for serial backplane communication," *IEEE J. Solid-State Circuits*, vol. 44, pp. 1306–1315, Apr. 2009.

[4] C. Kromer *et al.*, "A 25 Gb/s CDR in 90-nm CMOS for high-density interconnects," *IEEE J. Solid-State Circuits*, vol. 41, pp. 2921–2929, Dec. 2006.

[5] J. Lee and M. Liu, "A 20 Gb/s burst-mode clock and data recovery circuit using injection-locking technique," *IEEE J. Solid-State Circuits*, vol. 43, pp. 619–630, Mar. 2008.

[6] M. Nogawa *et al.*, "A 10 Gb/s burst-mode CDR IC in 0.13 μm CMOS," in *Solid-State Circuits Conf. Digest of Technical Papers (ISSCC)*, Feb. 2005, pp. 228–595.

[7] M. Banu and A. Dunlop, "A 660 Mb/s CMOS clock recovery circuit with instantaneous locking for NRZ data and burst-mode transmission," in *Solid-State Circuits Conf. Digest of Technical Papers (ISSCC)*, Feb. 1993, pp. 102–103.

[8] J. Terada *et al.*, "A 10.3125 Gb/s burst-mode CDR Circuit using a $\Delta\Sigma$ DAC," in *Solid-State Circuits Conf. Digest of Technical Papers (ISSCC)*, Feb. 2008, pp. 226–609.

[9] B. Abiri *et al.*, "A 1-to-6 Gb/s phase-interpolator-based burst-mode CDR in 65 nm CMOS," in *Solid-State Circuits Conf. Digest of Technical Papers (ISSCC)*, Feb. 2011, pp. 154–156.

[10] F. Gardner, *Phase-Locked Loops*, 2nd ed. New York, NY, USA: Wiley and Sons, 1979.

[11] B. Razavi, *Design of Integrated Circuits for Optical Communications*. New York, NY, USA: McGraw-Hill Science, 2002.

[12] R. C. H. van de Beek *et al.*, "A 2.5–10-GHz clock multiplier unit with 0.22-ps RMS jitter in standard 0.18- μm CMOS," *IEEE J. Solid-State Circuits*, vol. 39, pp. 1862–1872, Nov. 2004.

[13] A. Pottbacker, U. Langmann, and H. U. Schreiber, "A Si bipolar phase and frequency detector IC for clock extraction up to 8 Gb/s," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1747–1751, Dec. 1992.

[14] J. D. H. Alexander, "Clock recovery from random binary signals," *Electron. Lett.*, vol. 11, pp. 541–542, 1975.

[15] B. Lai and R. C. Walker, "A monolithic 622 Mb/s clock extraction data retiming circuit," in *Solid-State Circuits Conf. Digest of Technical Papers (ISSCC)*, Feb. 1991, pp. 144–306.

[16] S. K. Enam and A. Abidi, "NMOS IC's for clock and data regeneration in gigabit-per-second optical-fiber receivers," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1763–1774, Dec. 1992.

[17] I. Ahmed and D. A. Johns, "A high bandwidth power scalable sub-sampling 10-bit pipelined ADC with embedded sample and hold," *IEEE J. Solid-State Circuits*, vol. 43, pp. 1638–1647, Jul. 2008.

[18] 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, pp. 1356–1360, Aug. 2004.

[19] N. Kocaman *et al.*, "An 8.5–11.5 Gbps SONET transceiver with reference-less frequency acquisition," in *Proc. IEEE Custom Integrated Circuits Conf.*, Sep. 2012, pp. 1–4.

[20] M. Su *et al.*, "A 10 Gbps, 1.24 pJ/bit, burst-mode clock and data recovery with jitter suppression," in *Proc. IEEE Custom Integrated Circuits Conf.*, Sep. 2013, pp. 1–4.

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

[22] M. S. Jalali *et al.*, "An 8 mW frequency detector for 10 Gb/s half-rate CDR using clock phase selection," in *IEEE Custom Integrated Circuits Conf.*, Sep. 2013, pp. 1–4.

Ravi Shivnaraine received the Bachelors and Masters degrees in electrical engineering from the University of Toronto, Toronto, ON, Canada in 2010 and 2012 respectively.

In 2012 he joined Semtech-Snowbush IP, and has been engaged in the development of multistandard SerDes IP.

Mohammad Sadegh Jalali (S'13) received the Bachelors degree (with honors) in electrical engineering from the University of Tehran, Tehran, Iran, and the Masters degree from the University of British Columbia, Vancouver, BC, Canada, in 2008 and 2010, respectively. He is currently pursuing the Ph.D. degree in electrical engineering in the university of Toronto, Toronto, ON, Canada.

His research interests are integrated circuits for high speed chip-to-chip communication.

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

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

Dr. Sheikholeslami served on the Memory, Technology Directions, and Wireline Subcommittees of the ISSCC in 2001–2004, 2002–2005, and 2007–2013, respectively. He currently serves on the executive committee of the same conference as its Educational Events Chair, and on the editorial board of the SOLID-STATE CIRCUITS MAGAZINE as an Associate Editor. He was an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS for 2010–2012. He was the program chair for the 2004 IEEE ISMVL held in Toronto, Canada. He is a Registered Professional Engineer in the province of Ontario, Canada. He received the Best Professor of the Year Award four times (in 2000, 2002, 2005, and 2007) by the popular vote of the undergraduate students in the Department of Electrical and Computer Engineering, University of Toronto. He received the 2005–2006 Early Career Teaching Award and the 2010 Faculty Teaching Award, both from the Faculty of Applied Science and Engineering at the University of Toronto, in "Recognition of Superb Accomplishment in Teaching."

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.

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 multichannel high-speed I/O for server interconnects. Since then he has been working in the area of architecture- and transistor-level design for CMOS high-speed signaling circuits.