A 2.5 Gb/s Run-Length-Tolerant Burst-Mode CDR Based on a 1/8th-Rate Dual Pulse Ring Oscillator

Sander L. J. Gierkink, *Member, IEEE*

***Abstract*—A 2.5 Gb/s burst-mode clock and data recovery (CDR) circuit is presented that uses a 1/8th-rate ring oscillator with two pulses running simultaneously that are phase independent. One "tune" pulse sets the delay of the ring by phase locking it to a reference. The other "clock" pulse tracks the phase of the incoming data by a process of pulse removal and reinsertion. Because both pulses share the same ring, there is no frequency mismatch between the incoming data stream and the recovered clock in frequency synchronous systems, allowing for large data run lengths. A 1:8 data-demux clock is naturally generated by tapping the clock pulse along the ring. Phase acquisition is instantaneous from a single data edge. Run length tolerance is larger than 72 bits. The 0.6 mm^2 0.13 μm CMOS chip includes a CML-to-CMOS input buffer, PLL with on-chip loop filter, PRBS checker, 1:8 data demux, and eight output buffers. It has 2.7 UI_{pp} measured jitter tolerance at 100 kHz and consumes 42 mW from a single 1.2 V supply.**

***Index Terms*—Clock and data recovery (CDR), phase-locked loop (PLL), voltage-controlled oscillator (VCO).**

I. Introduction

A PASSIVE optical network (PON) uses inexpensive optical splitters to connect multiple subscribers to the central office (CO) over a single, optical trunk fiber in a point-to-multipoint configuration. PONs are called "passive" because, other than at the CO and subscriber endpoints, there are no active electronics within the network. PONs apply time division multiplex access (TDMA) with data transfer in full duplex, using different wavelengths for transmit and receive. Data transmitted from the CO to any of the subscribers is frequency and phase synchronous with the CO clock at all times, allowing for the use of a relatively slow CDR with small bandwidth at the subscriber. Each individual subscriber, connected to the common fiber over possibly different fiber lengths, recovers the clock and uses it to transmit bursts of data back to the CO. As a result, data bursts arriving at the CO from different subscribers have different phase offsets relative to the CO master clock. In summary, data burst arriving at the CO are frequency synchronous but phase asynchronous with the CO clock. As time between bursts is short, the CDR in the CO must be capable of acquiring phase fast: within 20 bit intervals in some standards. Also, it must be able to recover large numbers of consecutive identical digits (CID) correctly.

This paper describes the design of the burst-mode CDR that resides at the CO. A widely used burst-mode CDR topology is based on matched gated oscillators [1], [2]. Fig. 1(a) shows an implementation example. It employs three matched gated oscillators. The incoming data starts oscillators A and B alternately, thus implementing instantaneous phase synchronization. Oscillator C is permanently locked to a reference by a phase-locked loop (PLL). It sets the tune voltage for all three oscillators. In PON the incoming data rate is an exact multiple of the reference clock due to the frequency synchronous nature of the network. However, during long periods without data transition, the frequency mismatch between the oscillators accumulates to a phase error that ultimately leads to a misalignment between recovered clock and data. Hence, mismatch between the three oscillators limits the maximum number of CID that is recovered correctly. The accumulated phase error is reset with each new data transition. In general, bit errors occur during recovery when the phase error due to jitter and/or frequency mismatch between two updates accumulates to more than half a bit interval. Fig. 2 shows the relationship between the percentage frequency mismatch α and maximum number of CID that can be recovered correctly, assuming no jitter and no data pulse-width distortion. For example, the GPON standard allows 72 CID, requiring less than 0.7% frequency mismatch. The allowable frequency mismatch decreases further in the presence of data jitter.

The burst-mode CDR presented here uses a single 16-stage ring oscillator running at 1/8th of the data rate [3]; see Fig. 1(b). It has two phase independent pulses running simultaneously. One "tune" pulse is used to permanently lock the oscillator to a reference, while the other "clock" pulse is used to track the phase of the incoming data by a process of pulse removal and reinsertion. This allows instantaneous phase synchronization, similar to the gated oscillator approach. Because both pulses share the same delay stages, there is no frequency mismatch between the clock pulse and the incoming data stream, enabling correct operation in the presence of large run lengths. Eight taps of the clock pulse along the ring are used to clock the incoming data into eight separate D-flops, thus implementing a 1:8 data demux. Power consumption is low since the oscillator runs at only 1/8th of the data rate, allowing wide use of standard CMOS logic. Even though the oscillator has 16 stages, its power consumption compares favorably to the gated oscillator approach that uses three oscillators, each having at least 3 stages. Of the three oscillators, two are running at the full data rate at any given time. Moreover, a 1:8 data demux is naturally provided for; in the gated oscillator approach the data demux consumes additional power.

Finally, in another application of this circuit, a low-noise clock multiplier is obtained by injecting the crystal reference instead of data, reducing phase noise of the clock pulse by more than 15 dB compared to the tune pulse.

Manuscript received January, 2008; revised April, 2008. Published July 23, 2008 (projected).

S. L. J. Gierkink is with Conexant Systems Inc., Red Bank NJ 07701 USA (e-mail: sander.gierkink@conexant.com).

Digital Object Identifier 10.1109/JSSC.2008.926736

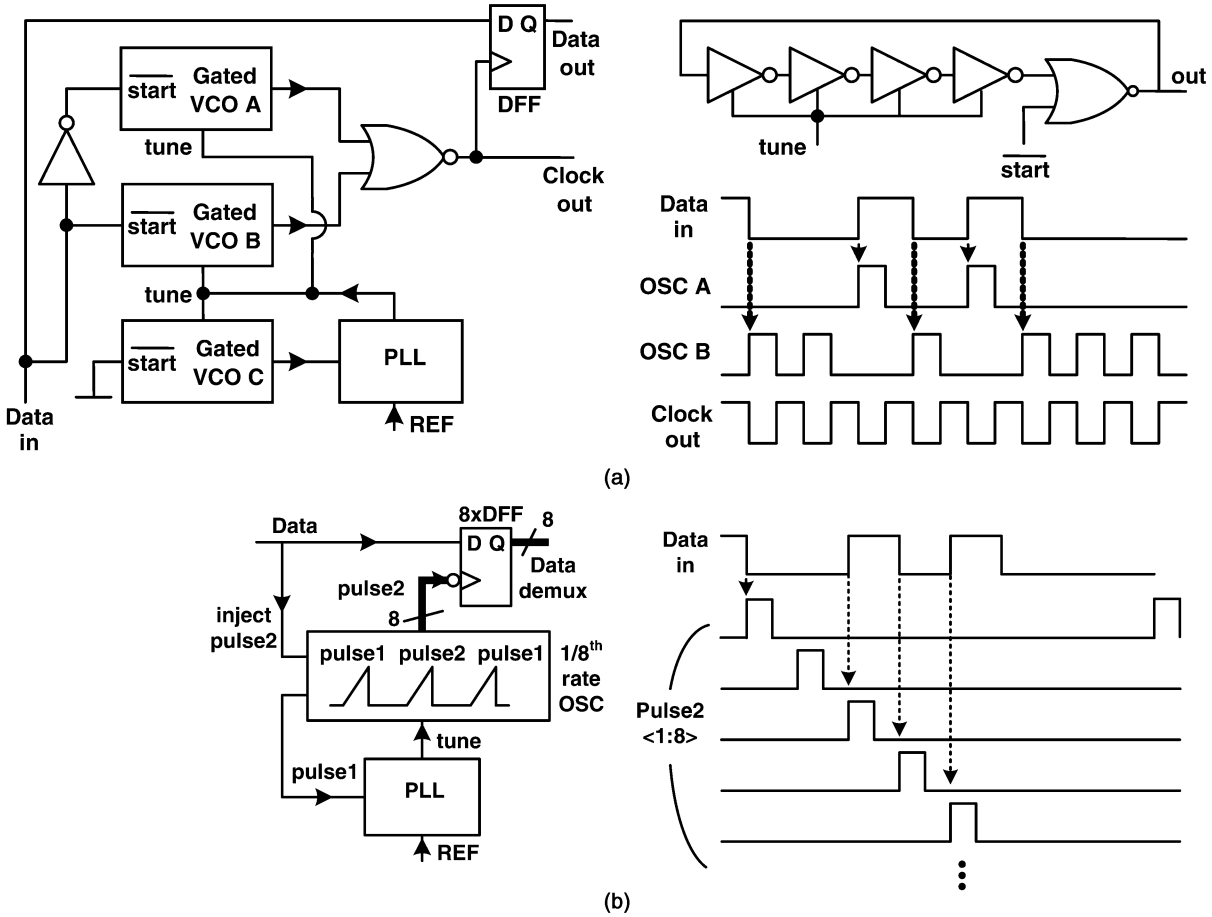

Fig. 1. (a) Burst-mode CDR based on matched gated oscillators and (b) based on 1/8th rate dual pulse oscillator.

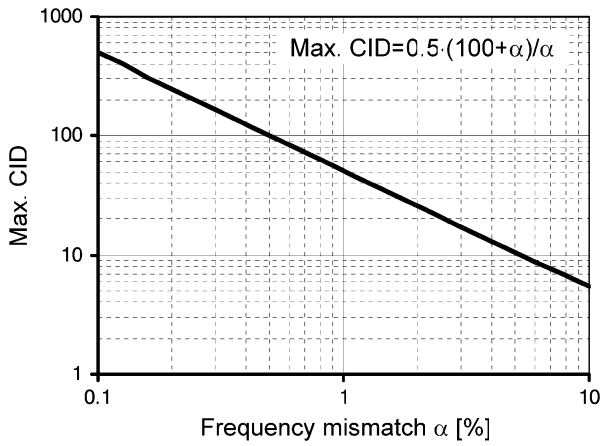

Fig. 2. Maximum tolerable CID versus percentage frequency mismatch α assuming no jitter.

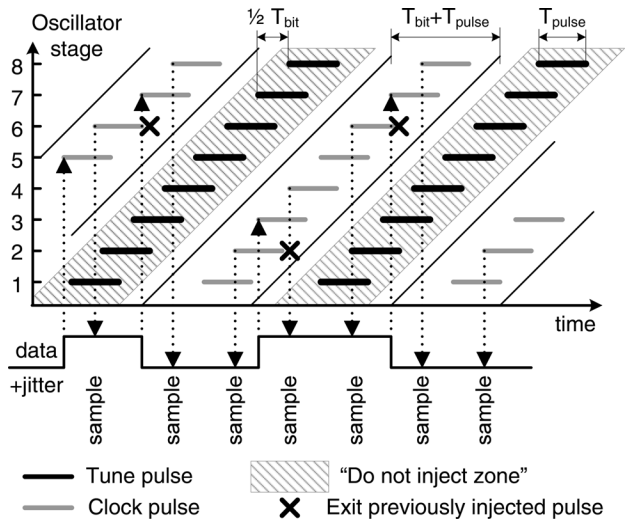

Fig. 3. Space-time plot demonstrating the principle of operation of the proposed CDR.

II. Principle of Operation

Fig. 3 visualizes the principle of operation. For simplicity, the oscillator is assumed to be comprised of only eight oscillator stages. The plot shows timing and location of the "tune" and "clock" pulses along the ring. The pulses are represented by short horizontal lines, identifying their pulse width. The tune pulse sets the delay per oscillator stage to half a bit time by permanently phase locking it to the reference signal. The rising edge of the clock pulse tracks the phase of the incoming data. A phase update of the clock pulse is implemented by completely removing and subsequently reinserting it in-phase with each incoming data edge. A data demux is obtained by tapping the clock pulse at every other stage along the ring and using it to clock the data in separate flip-flops.

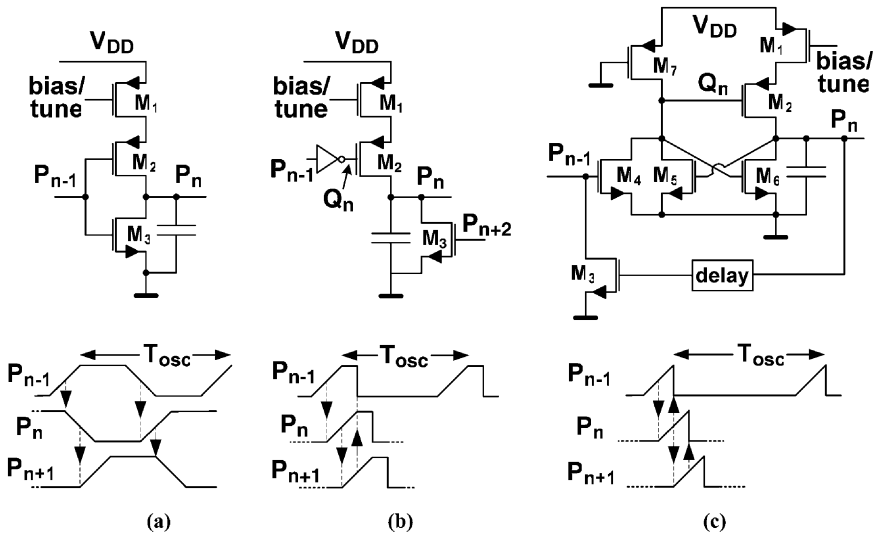

Fig. 4. Comparison of (a) standard CMOS and (b), (c) sawtooth ring oscillator delay stages.

The specific ring oscillator topology must allow the coexistence of two pulses that are frequency synchronous and phase asynchronous with one another. Frequency synchronicity is ensured by simply having the pulses run through the same delay stages as they pass along the ring. Phase a-synchronicity is harder to achieve and requires careful selection of the oscillator topology, as will be explained in Section III.

Any phase update of the clock pulse is not allowed to interfere with the phase accumulation of the tune pulse. Interference is avoided by restricting the injection of the clock pulse to a zone that is always roughly 180° out of phase with the tune pulse. The position of the tune pulse at the time of a data edge determines which stage to inject into, as will be explained in more detail in Section IV. The injection zone must be wide enough to allow seamless injection of a data edge into an injection stage at any given time. In the example, injection is accepted in the odd stages only. Since the delay per stage is half a bit interval, seamless phase tracking requires the injection-stages to accept a data edge over one bit interval. Finally, the recovered clock is taken from the even numbered stages, thus giving clock edges that occur half a bit interval later than the data edge. This positions the recovered clock in the center of the bit when no jitter is present.

III. The Dual Pulse Oscillator

A suitable oscillator topology is chosen that allows dual pulse operation. Fig. 4 compares the basic delay stage and waveforms of the conventional- and sawtooth ring oscillator [4]. Fig. 4(a) shows the basic stage of a conventional CMOS ring oscillator employing an odd number of stages. It has an approximate 50% duty cycle waveform with rising and falling edges alternately contributing to phase accumulation at node P_n. This makes it impossible to run two phase-independent edges or pulses around the ring simultaneously: any additional edge or pulse inserted will affect the timing of *all* subsequent edges along the ring. What is needed is an oscillator that utilizes only *one* type of edge (either rising or falling) to keep track of phase accumulation. Such oscillators are not new: examples can be found in [4], [5]. New here is that the oscillator operates with two phase independent pulses simultaneously.

Fig. 4(b) shows an oscillator stage of the sawtooth oscillator, which oscillates with both an even or odd number of stages. This type of oscillator requires startup: all nodes P_n except one must be pulled low briefly to start oscillation. Only rising edges at nodes P_n and falling edges at nodes Q_n contribute to phase accumulation. To verify whether or not an edge accumulates phase one can imagine adding a small time error to a particular edge and check whether all subsequent edges in the ring shift by the same amount. For example, if for some reason a falling edge at node P_n is delayed somewhat, it will not affect the timing of the next rising edge at node P_n. This is because a falling edge at P_n does not initiate any other edge further down the ring. It merely functions as a reset of the P_n node, making it available to accept the next rising edge. Since only one edge accumulates phase at any given node and waveform duty cycle is low, a second pulse can be inserted in between two occurrences of the first pulse, without affecting the latter's phase accumulation.

Compared to the conventional stage the sawtooth stage has an additional internal node Q_n that limits the maximum operating frequency. However, this is not problematic for the application since the oscillator runs at only a fraction of the data rate. To allow for further reduction of duty cycle latch $M_{5,6}$ is added in Fig. 4(c). The latch has dual function: it secures node P_n when P_{n-1} is discharged and it keeps P_n shorted to ground in between pulses. The latter feature eliminates any parasitic charge build-up during intervals between pulses that might affect the time to charge this node. A build-up of parasitic charge causes undesired phase coupling between pulses.

In the CDR the delay per stage is fixed at half a bit interval by the PLL. As a result the data-demux ratio equals half the number of stages in the ring. One can reduce the duty cycle of the waveform further by adding more stages to the ring. This increases the time separation between the pulses, making the

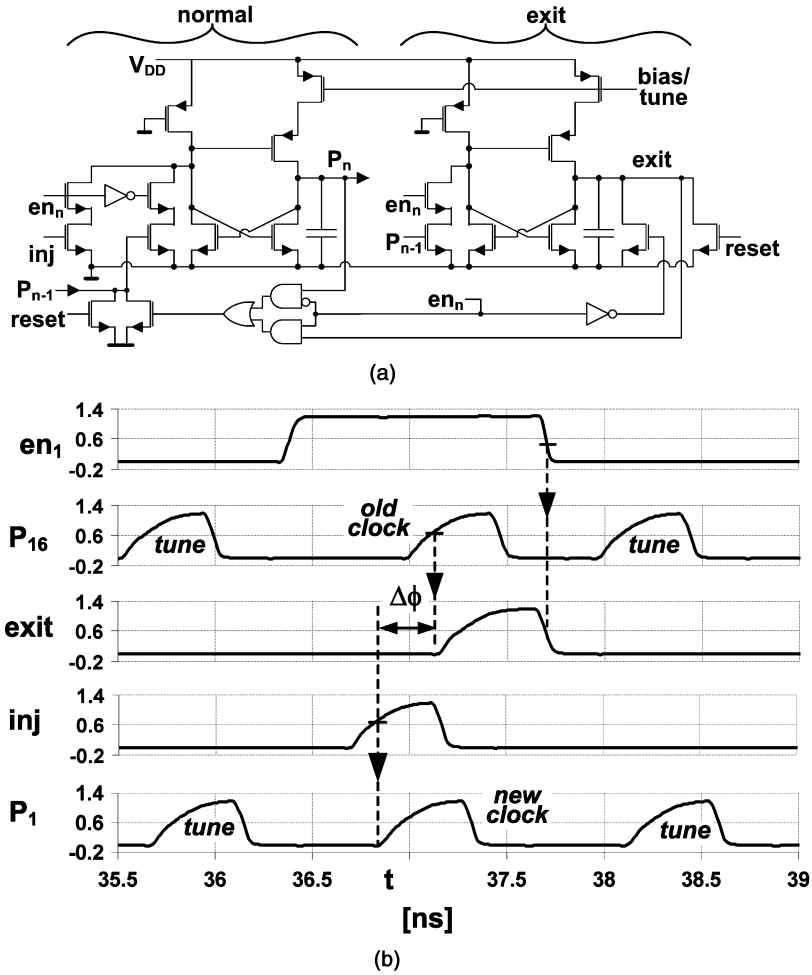

Fig. 5. (a) Single stage of the 16-stage CDR and (b) simulated waveforms during pulse removal/reinsertion.

pulse removal/reinsertion process less time critical with respect to pulse interference.

The number of possible phase-independent pulses in the sawtooth oscillator is not restricted to two. In fact, it can accommodate as many phase-independent pulses as the ring length and waveform duty cycle allow. To avoid a situation with more than two pulses one must ensure that the old clock pulse is always fully removed from the ring during a phase update.

Fig. 5(a) shows the complete schematic of the *n*th-oscillator stage used in the CDR. It consists of two parts: the branch that is normally active and the "exit" branch that is active only during pulse injection/removal. During normal operation signal en_n is low and a pulse at input P_{n-1} causes a pulse at output P_n. When a data edge arrives, one of the oscillator stages is selected for injection. The en_n signal enables the *inj* input and redirects the previously injected pulse arriving at P_{n-1} to the exit section, where it terminates. The exit branch acts as a pulse "trap" that waits for a pulse arriving at P_{n-1} while en_n is high. Since the "exit" branch is identical in structure to the "normal" branch the loading of node P_{n-1} is not affected by the state of the en_n signal.

Fig. 5(b) shows the simulated waveforms of a successful injection into stage 1. As shown the en_n signal must be activated before both the new- and old clock pulse arrives at the stage of injection. It must be wide enough to span the durations of both the new- and old clock pulse and any possible phase difference between the two, limited to a maximum of one bit interval. This potential phase difference is caused by data jitter that is to be tracked. In-between pulses all node voltages in a stage must be allowed to return fully to their "quiescent" value to avoid phase coupling between the two pulses. Hence, a certain minimum time interval between pulses must be kept. Finally, any common bias point between stages must be properly decoupled. This ensures that possible glitches caused by one pulse are not affecting the phase accumulation of the other pulse.

The oscillator stage of Fig. 5(a) is single-ended, making it sensitive to supply disturbances. Although not included in the present design, a robust design would preferably use a local on-chip supply regulator for the oscillator. Potential random noise from the supply (regulator) is less of a concern in a burst-mode CDR compared to a conventional CDR because jitter accumulation is reset with each incoming data edge.

IV. Pulse Removal and Reinsertion

The requirements of the *en* signal pose a causality problem: it must be made high *prior* to the arrival of a data edge. To allow for time to activate one of eight en signals, an incoming data edge first injects a clock pulse into a "merge" delay line. Fig. 6

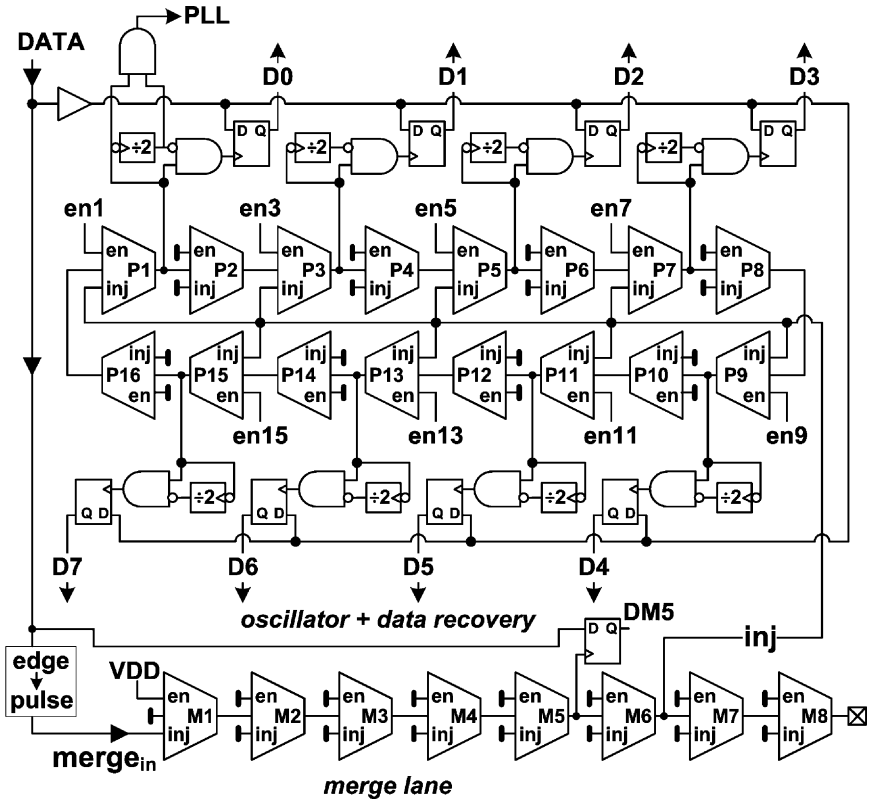

Fig. 6. Oscillator with merge lane and data recovery.

shows the 16-stage oscillator along with the merge lane, comprised of 8 stages, identical to those in the oscillator. Sixteen stages were chosen in the oscillator because it reduces the duty cycle enough to allow for secure dual pulse operation with sufficient time margin between pulses. Also it results in a practical 1:8 data demux ratio and a frequency of oscillation of 1/8th of the data rate. More stages can be chosen, but this was deemed to complicate the routing in the layout too much. Note that the single-ended sawtooth stage, similar to the conventional CMOS stage in Fig. 4, consumes power only during transitions. Since the delay per stage is kept constant at half a bit interval by the PLL, adding more stages will not increase power consumption of the ring; it only increases layout complexity.

The merge lane shares its tune voltage with the oscillator. The 6th stage of the merge lane connects to the *inj* input of the odd oscillator stages, implementing a 3-bit delay between data edge and actual oscillator injection. To match the loading of the 6th stage of the merge lane all remaining CDR stages are loaded with eight additional dummy transistors, equal in size to the injection transistor in Fig. 5(a). As mentioned, the merge lane allows for time to select the proper injection stage. Mismatch between the delay of the merge lane and oscillator will align the clock pulse edge away from the data bit center. This reduces jitter tolerance; however, once the clock pulse is inserted into the ring, the potential time error does not grow over time during long run lengths, as is the case for the gated oscillator approach of Fig. 1. Thus, matching is still required; however the matching requirements are relaxed compared to the gated oscillator approach.

While the clock pulse, initiated by a data edge, travels down the merge lane one of eight en signals is activated. New injections are accepted once the merge lane has injected a pulse into the ring oscillator, 3 bit intervals after a previously accepted data edge. This restricts the maximum phase update rate of the recovered clock compared to the gated oscillator approach of Fig. 1(a), where in principle every data edge realigns the phase of the recovered clock. As a result, the tolerance to jitter frequencies approaching 1/3UI of this approach is somewhat less.

Injection is enabled in a stage that is roughly 180° out of phase with the original pulse, thus mitigating pulse interference. Fig. 7(a) shows the schematic of one of eight enable-inject circuits, driving the en_n signals in Fig. 6. Indici of waveforms are to be taken modulo 16. The waveforms in Fig. 7(b) detail the timing of an injection into stage 1. Whenever the pulse $merge_{in}$ at the input of the merge lane lies within a bit interval, defined by the tune pulses $(P_1 \vee P_2) \wedge$ (NOT P_3), the latch SR_1 goes high. This latch is re-timed by tune pulse P_3 and latch SR_2, setting signal en_1. The retiming eliminates the uncertainty in the rising edge of the en_n signal. Signal en_1 is reset by tune pulse P_{12}. Thus, the duration of the en_1 signal is set exclusively by tune pulses $P_{3,12}$. This guarantees that the en_1 signal cannot accidently send the tune pulse P_1 of stage 1 to the exit branch. The role of the div signals in Fig. 7(a) is to distinguish the tune pulse from the clock pulse as will be explained in Section V. While the en_1 signal is high, the incoming pulse travels down the merge lane and injects into stage 1, approximately 180° out of phase with the tune pulse. Simultaneously the previously injected clock pulse is sent to the exit branch where it terminates.

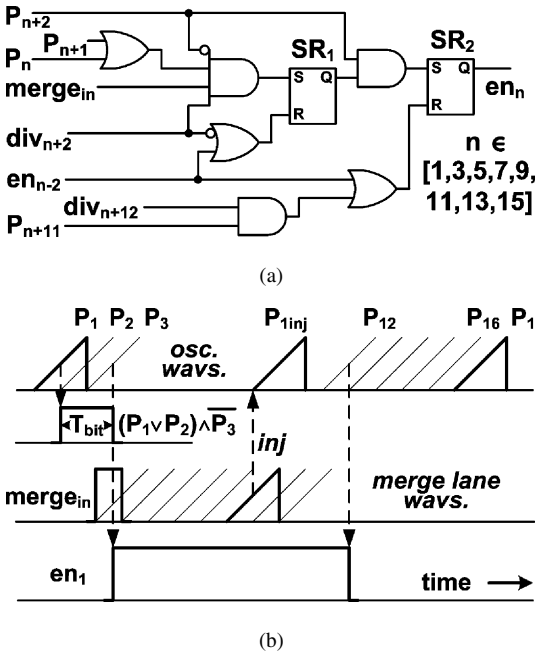

Fig. 7. (a) One of eight enable inject circuits (b) with waveforms.

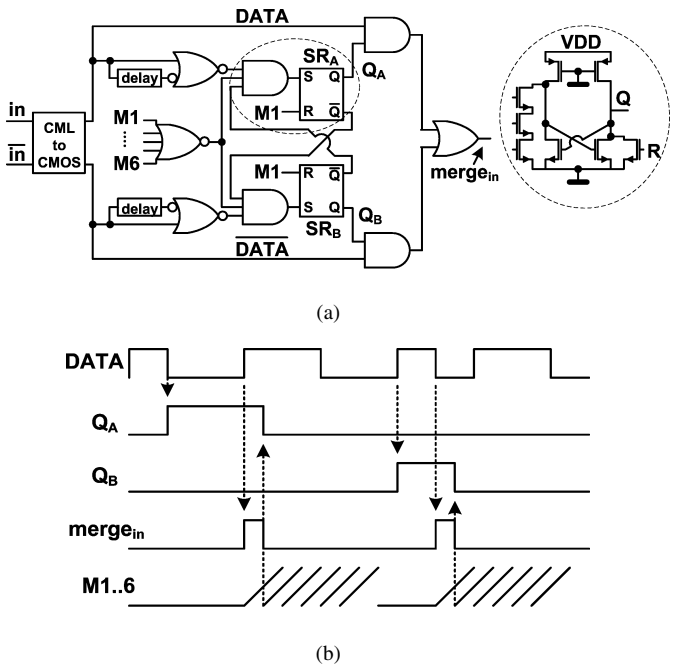

Fig. 8. (a) Data edge-to-pulse converter and (b) waveforms.

Due to the non-zero width of the pulse_{in} pulse, the en signals in two successive enable inject blocks may go high. By feeding each en_n signal to the reset input en_{n-2} of the next enable block down the ring, only the first en signal wins.

Fig. 8(a) shows the circuit that converts a data edge into a pulse $merge_{in}$ that drives the merge lane. The CML input is first converted to digital CMOS levels. Latch SR_A is set by a falling data edge and enables rising data edges to generate a $merge_{in}$ pulse; latch SR_B does the same for the edges of the inverted data stream. Only one latch can be set at a time, due to the cross coupling between latches. Furthermore, a latch can be set only when the merge lane is empty, i.e., when $M_{1...6}$ are zero, as shown in Fig. 8(b). The latches are reset by M_1, ensuring the proper duration of the $merge_{in}$ pulse. Custom latches are used instead of standard CMOS digital library components because of the speed requirements of this particular circuit.

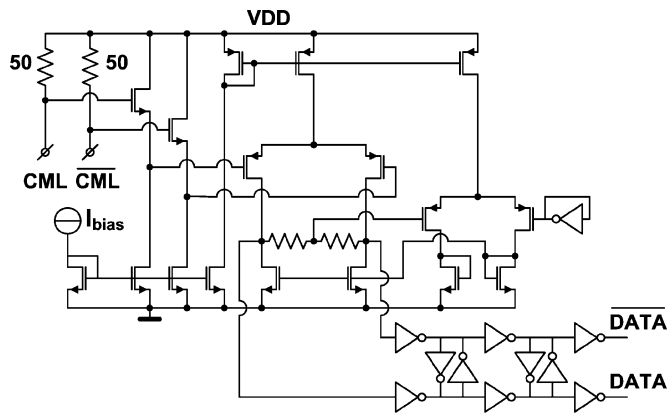

Fig. 9. CML-to-CMOS converter.

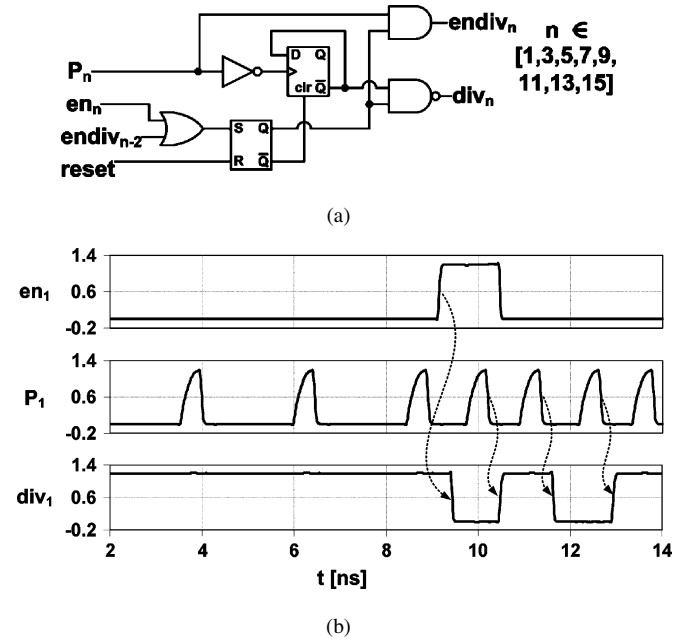

Fig. 10. (a) One of eight divide-by-two circuits (b) with simulated waveforms.

Fig. 9 shows the CML-to-CMOS converter [6]. The requirements for this circuit are that it must have balanced CMOS outputs and that it must not introduce data dependent jitter. The latter is generally introduced by slow settling internal nodes that cause bits to widen or shrink depending on the transition density of the input data pattern. Since we are building a burst-mode CDR, the CML-to-CMOS converter must be able to convert the very first bit of a pattern correctly, even if it is preceded by a long string of zeros or ones. The circuit is fully balanced. The input data is first level shifted and then converted to CMOS levels using a single differential gain stage whose common mode level is set equal to the transition point of a CMOS inverter. Additional output inverters with cross coupled weak inverters further sharpen the final output edges.

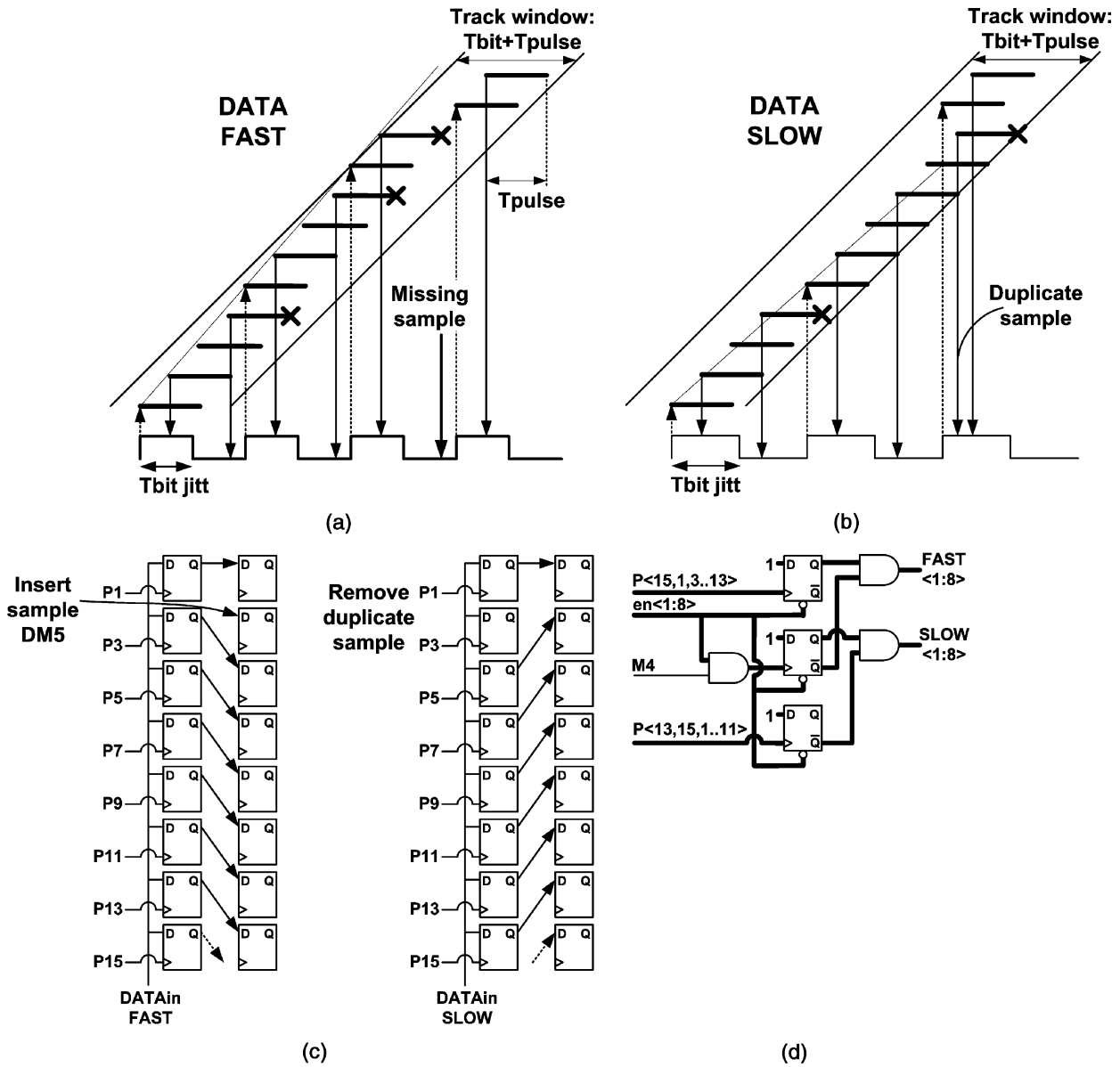

Fig. 11. (a) Missing sample when data is fast; (b) duplicate sample when data is slow; (c) bit rotations; (d) circuit that detects fast or slow data.

V. Separating the Pulses

So far we have described the oscillator topology that allows dual pulse operation and the mechanism by which to remove and reinsert the clock pulse without interfering with the tune pulse. One important aspect is ignored: how to distinguish the clock pulse from the tune pulse? Looking at any given node of the ring oscillator the two pulses will pass by in alternating order. Thus, a divide-by-two circuit can be used to keep track of which pulse is which. However, this requires the divider to start from a known state by a known pulse so that from thereon the dividers output uniquely relates to one of the two pulses. Fig. 10(a) shows one of eight divide-by-two circuits. At startup the SR latch is reset, forcing the signals div_n high and $endiv_n$ low. Initially the oscillator has only one running pulse: the tune pulse. Just prior to an injection of the clock pulse one of the en_n signals goes high. The very first en signal that goes high starts one of eight divide-by-two circuits. It in turn starts the next divide-by-two circuit along the ring with its $endiv$ signal. Finally, all divide-by-two circuits are activated in the proper sequence. Fig. 10(b) shows the simulated waveforms in case the very first clock injection after startup occurs in stage 1. The rising edge of the en signal toggles the div signal. From thereon the div signal toggles on the falling edges of the pulses P_n. At the same time all other divide-by-two circuits are activated sequentially by passing their $endiv$ signals to their next-neighbor divide circuit. Gating an oscillator node with a divide-by-two output or its complement selects either the tune or clock pulse at that node, as shown also in Fig. 6.

VI. Bit Rotator

An essential part of the dual pulse CDR is the bit rotator. Fig. 11(a) and (b) illustrates its necessity. To avoid interference between the tune and clock pulse the latter's position is restricted to a "tracking window". Its width allows the clock pulse in an injection stage to track the data edges over a full bit interval. When the data is temporarily faster or slower that the

tune pulse, the clock pulse will eventually be positioned near the boundaries of the tracking window. In these situations the CDR essentially performs a modulo-2 π operation on the recovered clock phase: it moves the injected pulse back to the other injection boundary. This operation is implemented by the enable inject circuitry: it always selects a stage for injection that bounds the clock pulse to the tracking window. However, in the process of shifting the clock pulse to the other boundary the data flip-flops (DFF) of Fig. 6 produce either a missing- or a duplicate recovered bit, as illustrated by Fig. 11(a) and (b), respectively. The bit rotator corrects for this situation by retiming the DFFs into a second set of DFFs, using the tune pulse. Depending on whether the incoming data is fast or slow the rotator sequentially rotates the recovered data bits of the first set of DFFs one bit backward or forward as shown in Fig. 11(c). When the data is fast, the missing recovered bit is provided by a DFF, clocked by pulse M5 of the merge lane (see Fig. 6). Fig. 11(d) shows the circuit that detects when a bit rotation is necessary. The rotator itself essentially consists of MUXes that connect the first row DFFs to the second row of DFFs in the right order and eight counters that keep track of the amount of rotation. The rotator does not have to be fast: its speed of operation is 1/8th of the data rate. It is completely built in standard CMOS logic.

VII. Experimental Results

The burst-mode CDR is fabricated in a 0.13 μm standard digital CMOS process. The 16-stage oscillator layout consists of a single row of stages placed in the order 1-16-2-15-3-14-4-13-5-12-6-11-7-10-8-9. This ensures equal parasitic delays between stages. The merge lane is added to the left of the ring oscillator, on the same row. The enable inject circuits are placed on a row just above the ring oscillator, followed by another row of divide-by-two circuits. Symmetry in routing is essential and maintained from the oscillator all the way to the final data flip-flops.

Fig. 12 shows a die photograph of the complete burst mode CDR. Active die size is (700 μm $\times$ 850 μm) and total power consumption is 42 mW from a single 1.2 V supply. Area and power include the CML-to-CMOS input buffer, the PLL with on-chip loop filter, the 1:8 data demux including the bit rotator, eight output buffers and eight PRBS checkers, one per data demuxed output. It handles both 1.25 and 2.5 Gb/s data rates and tolerates more than 72 CID.

Fig. 13 shows the measured sinusoidal jitter tolerance at 2.5 Gb/s with a 2^7-1 PRBS pattern, using the on-chip PRBS checker. Above 1 MHz, the jitter tolerance approaches 0.5 $\mathrm{UI_{pp}}$; this can be shown to be the theoretical limit for burst-mode CDRs that sample data at a fixed time offset of $0.5 \cdot \mathrm{T_{bit}}$ from the phase acquisition data edge.

Fig. 14 shows the differential CML input and two of the 1:8 demuxed output waveforms. Input data are two short 2.4 Gb/s bursts of each two bytes of "01010101", alternately inverted, separated by a string of zeros, giving either a "01" or "10" pattern per burst at the 1:8 demuxed outputs. The measurement shows that phase is acquired with the first data edge, even when the data is preceded by a long string of identical bits.

Fig. 15 again shows the differential CML input and two of the 1:8 demuxed output waveforms, now for a 2^7-1 PRBS pattern.

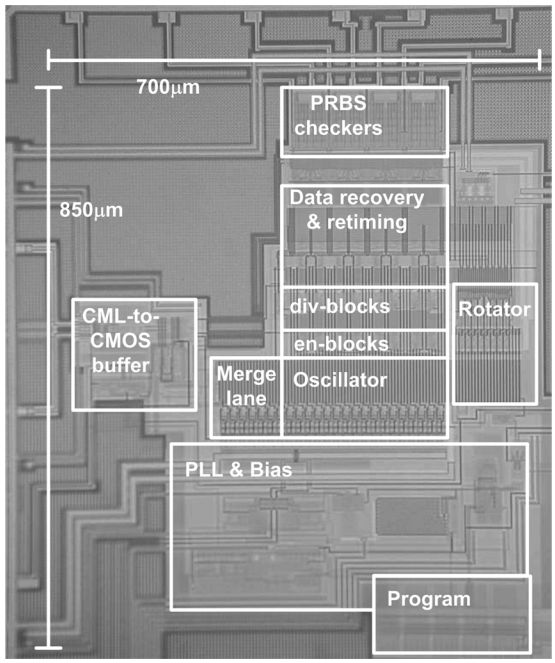

Fig. 12. Die photograph.

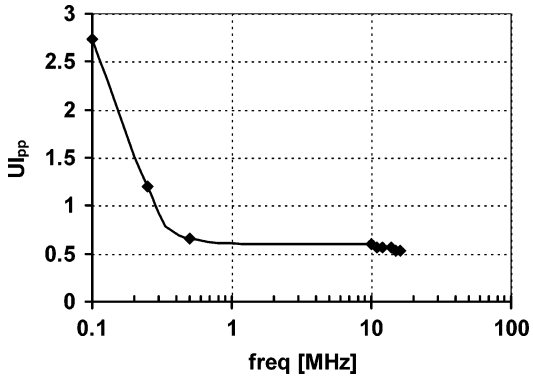

Fig. 13. Measured sinusoidal jitter tolerance (2.5 Gb/s, 2^7-1 PRBS).

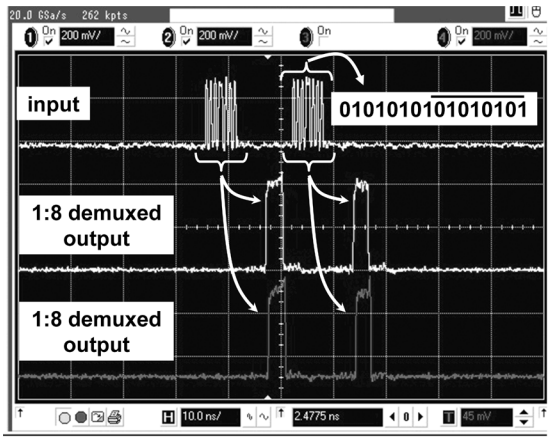

Fig. 14. Scope traces of bursts at 2.4 Gb/s and two 1:8 demuxed outputs.

The 1:8 demuxed outputs are themselves 2^7-1 PRBS patterns, at one eighth of the input data rate.

The CDR can also function as a low phase noise clock multiplier. It functions as a combination of PLL and recirculating

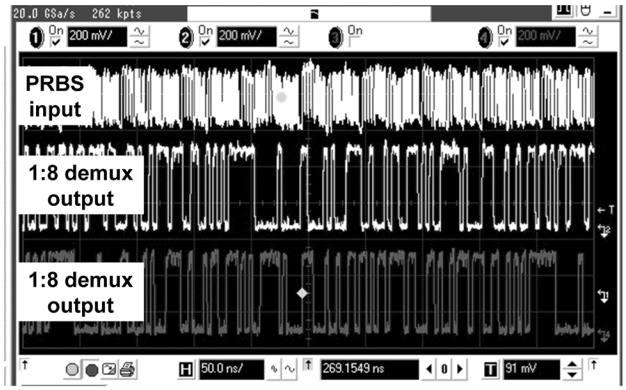

Fig. 15. Scope traces of 2^7-1 PRBS input and two 1:8 demuxed outputs.

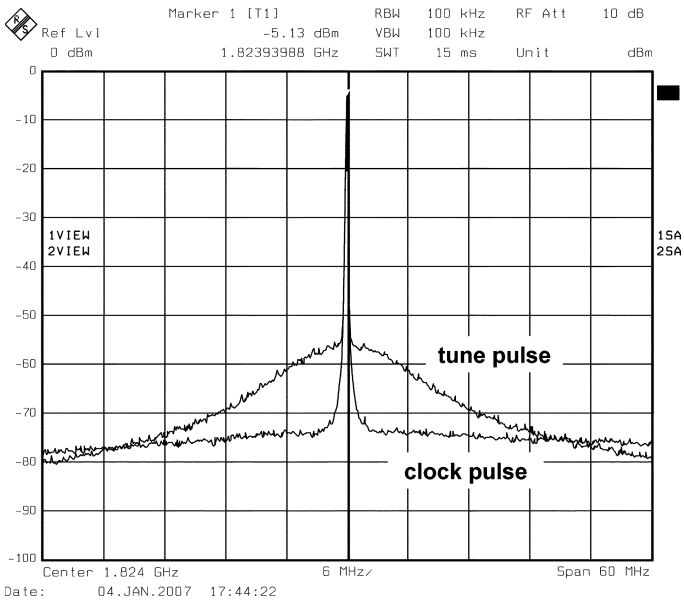

Fig. 16. Measured spectrum of tune pulse and periodically updated clock pulse, both after eight-times pulse combiner.

delay line by using the crystal reference instead of the data to periodically realign the phase of the clock pulse. This reduces the phase noise of the clock pulse [7], [8]. Compared to a classical recirculating or multiplying Delay Locked Loop (DLL) it offers the additional benefit that it avoids phase discontinuity between the pulses that are substituted for one another, because the realigned pulse is not used for tuning the delay line [9]. Interestingly, the ring oscillator now has two pulses running simultaneously with significantly different phase noise. Fig. 16 shows the measured spectrum of both the tune and clock pulse. The tune pulse shows the regular PLL-shaped spectrum, with suppression of oscillator phase noise within the loop bandwidth. The clock pulse shows phase noise suppression over a much wider frequency range, theoretically up to half the reference frequency. At low offsets, the sideband of the clock pulse is down by more than 15 dB. The signals are taken after two on-chip pulse combiners consisting of a logical OR of the 8 ring-taps of either the tune or clock pulse. In this experiment the oscillator is tuned to a frequency of 228 MHz, giving a 1.824 GHz signal after the pulse combiner. The phase noise measurement confirms that the two pulses are indeed phase independent; otherwise they would not be able to show different phase noise spectra while running around the same ring oscillator.

VIII. Conclusion

A burst-mode CDR circuit is presented that does not rely on oscillator matching. As a result it is able to recover large numbers of CID correctly in frequency synchronous systems. At the heart of the CDR is a sawtooth ring oscillator that allows the coexistence of two phase independent pulses. One "tune" pulse tunes the oscillator, the other "clock" pulse tracks the phase of the incoming data through a process of pulse removal/reinsertion. The ring oscillator runs at 1/8th of the data rate, enabling the use of standard CMOS logic throughout the circuit. A 1:8 data demux is naturally provided for by tapping the clock pulse along the ring. The dual pulse technique is demonstrated to reduce phase noise by more than 15 dB compared to a conventional PLL when applied in a clock multiplier.

Acknowledgment

The author would like to thank his colleagues K. Lakshmikumar and E. Säckinger for technical discussions, and the latter for providing the visualization idea of Fig. 3.

References

[1] M. Banu and A. Dunlop, "A 660 Mb/s CMOS clock recovery circuit with instantaneous locking for NRZ data and burst-mode transmission," in *IEEE ISSCC Dig. Tech. Papers*, 1993, pp. 102–103.

[2] M. Nogawa *et al.*, "A 10 Gb/s burst-mode CDR IC in 0.13 μm CMOS," in *IEEE ISSCC Dig. Tech. Papers*, 2005, pp. 228–229.

[3] S. L. J. Gierkink, "A 2.5Gb/s burst-mode CDR based on a 1/8th rate dual pulse ring oscillator," in *Proc. IEEE CICC*, Sep. 2007, pp. 357–360.

[4] S. L. J. Gierkink and A. J. M. van Tuijl, "A coupled sawtooth oscillator combining low jitter with high control linearity," *IEEE J. Solid-State Circuits*, vol. 37, no. 6, pp. 702–710, Jun. 2002.

[5] M. P. Flynn and S. U. Lidholm, "A 1.2-μm CMOS current-controlled oscillator," *IEEE J. Solid-State Circuits*, vol. 27, no. 7, pp. 982–987, Jul. 1992.

[6] S. Pellerano, "Fully-integrated frequency synthesizers for multi-standard WLAN applications," Ph.D. dissertation, Politecnico di Milano, Milan, Italy, 2004.

[7] S. Ye, L. Jansson, and I. Galton, "A multiple-crystal interface PLL with VCO realignment to reduce phase noise," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1795–1803, Dec. 2002.

[8] R. Farjad-Rad *et al.*, "A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1804–1812, Dec. 2002.

[9] S. L. J. Gierkink, "An 800 MHz -122 dBc/Hz @ 200 kHz clock multiplier based on a combination of PLL and recirculating DLL," in *IEEE ISSCC Dig. Tech. Papers*, 2008, pp. 454–455.

Sander L. J. Gierkink (M'08) received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Twente, The Netherlands, in 1994 and 1999, respectively.

In 1999, he joined Bell Laboratories, Lucent Technologies, Murray Hill, NJ, where he mostly worked on RF circuit design for WLAN. He continued to work at Agere Systems, Allentown, PA, after it spun off from Lucent. In 2004, he joined Conexant Systems, Red Bank, NJ, where his work currently focuses on CDR, PLL/DLL and line driver design for wire-line applications.

Dr. Gierkink received the 1998 Young Scientist Award and the 2003 Best Paper Award, both of the European Solid-State Circuits Conference, and the 2007 Best Regular Paper Award of the Custom Integrated Circuits Conference.