A 2.5Gb/s Burst-Mode CDR based on a 1/8th rate Dual Pulse Ring Oscillator

Sander L.J. Gierkink

Conexant Systems, 100 Schulz Drive, Red Bank, NJ 07701

***Abstract*-A 2.5Gb/s burst-mode CDR uses a 1/8th rate ring oscillator with two pulses running simultaneously that are phase independent. One pulse sets the delay of the ring by phase locking it to a reference. The other pulse tracks the phase of the incoming data. Phase acquisition is instantaneous from a single data edge. Run length tolerance is greater than 72 bits. The 0.6mm^2 0.13μm CMOS chip includes a CML-to-CMOS input buffer, PLL with on-chip loop filter, PRBS checker, 1:8 demux and 8 output buffers. It has 2.7UI$_{pp}$ jitter tolerance at 100kHz and consumes 42mW from a single 1.2V supply.**

I. Introduction

A Passive Optical Network (PON) uses inexpensive optical splitters to connect multiple subscribers to the central office (CO). Data transfer to the CO occurs in bursts, frequency synchronous with the CO clock. Because subscribers have different fiber lengths and each uses its recovered clock to transmit, the bursts are phase asynchronous. As time between bursts is short, the Clock and Data Recovery (CDR) circuit in the CO must be able to acquire phase fast: within 20 bit intervals in some standards. Also, it must be able to recover large numbers of consecutive identical digits (CID) correctly.

A widely used burst-mode CDR topology is based on matched gated oscillators [1,2]. Using this approach, the maximum number of CID that is recovered correctly is limited by the mismatch between the oscillators. Figure 1 shows the relationship between frequency mismatch and maximum number of CID that can be recovered correctly, assuming no jitter. For example, the GPON standard allows 72 CID, requiring less than 0.7% frequency mismatch.

The CDR presented here uses a single 16-stage ring oscillator running at 1/8th of the data rate. It has two pulses running simultaneously around the ring: one "original" pulse to set the ring delay to 8 bit intervals using a PLL, and one "injected" pulse to track the phase of the incoming data by a process of pulse removal and reinsertion. The recovered clock phase is acquired instantly from one data edge. Because both pulses share the same delay stages, there is no frequency mismatch between the recovered clock and the incoming data stream, enabling correct operation in the presence of large run lengths. Eight taps of the injected pulse along the ring provide a 1:8 data demux. Power consumption is low since the oscillator runs at only a fraction of the data rate, allowing the use of standard CMOS logic.

In another application of this circuit, an injection-locked PLL is obtained by injecting the crystal reference in place of the data, reducing the phase noise by more than 15dB.

II. The Oscillator

Figure 2 details the principle of operation. For simplicity, only 8 oscillator stages are assumed. The plot shows the timing and location of the original- and the injected pulse along the ring. A phase update is implemented by removing and reinserting the injected pulse upon arrival of a data edge, without interfering with the original pulse. Interference is avoided by injecting into a stage that is roughly 180° out of phase with the original pulse. Since both pulses share the same ring, they have the same frequency of oscillation. The delay per stage is set to half a bit interval, by phase locking the original pulse to a reference. A data demux is obtained by tapping the injected pulse at every other stage along the ring and using it to clock the data in separate flip flops.

A suitable oscillator topology is chosen to enable dual pulse operation. Figure 3 compares the basic delay stage topologies and waveforms of the conventional- and sawtooth ring oscillator [3]. A conventional CMOS ring oscillator 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 have two phase-independent pulses running around the ring simultaneously. In the sawtooth oscillator, a rising edge at input P_{n-1} causes a rising edge at output P_n. This in turn causes transistor M_7 to discharge node P_{n-1}, giving a low duty cycle waveform. Note that the falling edge at P_{n-1} does not propagate to node P_n. As a result, only rising edges at P_n and falling edges at Q_n contribute to phase accumulation. These properties allow for two simultaneous pulses running around the ring that are phase independent. Transistors $M_{5,6}$ form a latch with dual function: it secures the node P_n when P_{n-1} is discharged and it keeps the capacitor at P_n shorted to ground in between pulses.

When the oscillator operates in dual pulse mode, all node voltages in a stage must be allowed to return fully to their "quiescent" value in between pulses. Therefore a certain minimum time interval between pulses must be kept. This ensures phase independency of the two pulses. Also, any common bias point between stages must be properly decoupled. This ensures that glitches caused by one pulse are not affecting the phase propagation of the other pulse.

III. Pulse Injection & Data Recovery

Figure 4 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 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. As shown in figure 4, the *en* signal must be activated before both the new and old injected pulses arrive at the stage of injection.

To allow for time to activate one of eight *en* signals, an incoming data edge first injects a pulse into a "merge lane". Figure 5 shows the 16-stage oscillator along with the merge lane, comprising of 8 stages, identical to those in the oscillator. The 6^{th} 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. New injections are accepted when the merge lane empties, 4 bit intervals after a previously accepted data edge.

Injection is enabled in a stage that lies roughly 180° out of phase with the original pulse, thus mitigating pulse interference. Figure 6 shows the schematic of one of eight enable inject blocks, driving the *en* signals in figure 5. The waveforms in figure 6 detail the timing of an injection into stage 1. Whenever the pulse $pulse_{in}$ at the input of the merge lane lies within a bit interval, defined by the original pulses $(P_1 \vee P_2) \wedge (\text{NOT } P_3)$, the latch SR_1 goes high. This latch is re-timed by original pulse P_3 and latch SR_2, setting signal en_1. Signal en_1 is reset by original pulse P_{12}. Thus the duration of the en_1 signal is set exclusively by original pulses $P_{3,12}$. This guarantees that the *en* signal cannot interfere with the original pulse P_1 of stage 1. 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 original pulse. 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* signal to the reset input en_{n-2} of the next enable block down the ring, only the first *en* signal wins.

Gating an oscillator node with a divide-by-two output or its complement selects either the original or injected pulse at that node. Figure 5 shows how the divide-by-two circuits gate the injected pulse towards the clock inputs of the data recovery flip flops. Also, the divider at the output of stage 1 gates the original pulse towards the PLL. Finally, the dividers provide the proper div_{set} and div_{reset} signals of figure 6 that select the original pulses in the enable circuits.

The recovered data D[0..7] is retimed on-chip by the original pulse. A bit rotator rotates the demuxed outputs by +/-1 bit whenever the injected pulse makes an early→late or late→early transition with respect to the injection boundaries (see figure 2). Finally, eight output buffers drive the output bonding pads.

IV. Experimental Results

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

Figure 8 shows the CML input and two of the 1:8 demuxed output waveforms. Input data is a 2.4Gb/s repeated pattern of 4 bytes of 01010101, alternately inverted, followed by 24 zeros, giving repetitions of either 1010000 or 010100 at the demuxed outputs. Phase is acquired with the first data edge.

The CDR can also function as an injection-locked PLL by using the crystal reference instead of the data to periodically realign the phase of the injected pulse. This reduces phase noise [4]. Interestingly, the ring now has two pulses running simultaneously with significantly different phase noise. Figure 9 shows the measured spectrum of both the original and injected pulse. The original pulse shows the regular PLL-shaped spectrum, with suppression of oscillator phase noise within the loop bandwidth. The injected 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 injected pulse is down by more than 15dB. The signal is taken after an on-chip logical OR of the 8 ring-taps of either the original or injected pulse.

The CDR is fabricated in a 0.13µm standard digital CMOS process. Figure 10 shows a die photograph. Active die size is 700µm X 850µm and total power consumption is 42mW from a single 1.2V supply. Area and power include the CML-to-CMOS input buffer, the PLL with on-chip loop filter, the 1:8 data demux, the eight output buffers and eight PRBS checkers, one per data demuxed output. It handles both 1.25 and 2.5Gb/s data rates and tolerates more than 72 CID.

V. Conclusions

A burst-mode CDR topology is presented, based on a $1/8^{th}$ rate dual pulse ring oscillator utilizing two phase independent pulses. Since the topology does not rely on oscillator matching, it is able to recover large numbers of CID correctly. The dual pulse technique is demonstrated to reduce phase noise by more than 15dB when applied in a realignment PLL.

References

[1] M. Banu, A. Dunlop, "A 660Mb/s CMOS Clock Recovery Circuit with Instantaneous Locking for NRZ Data and Burst-Mode Transmission", ISSCC Dig. Tech. Papers, pp 102-103, Feb. 1993.

[2] M. Nogawa, et. al. "A 10Gb/s Burst-Mode CDR IC in 0.13µm CMOS", ISSCC Dig. Tech. Papers, pp 228-229, Feb. 2005.

[3] S.L.J. Gierkink, 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.

[4] S. Ye, L. Jansson, 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.

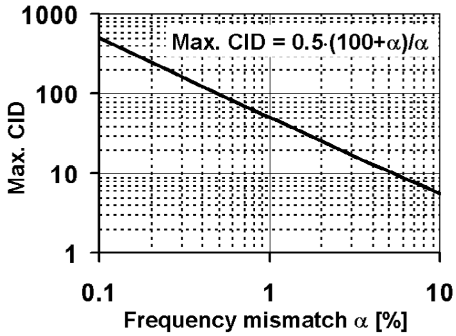

Fig. 1. Maximum tolerable CID versus frequency mismatch α assuming no jitter.

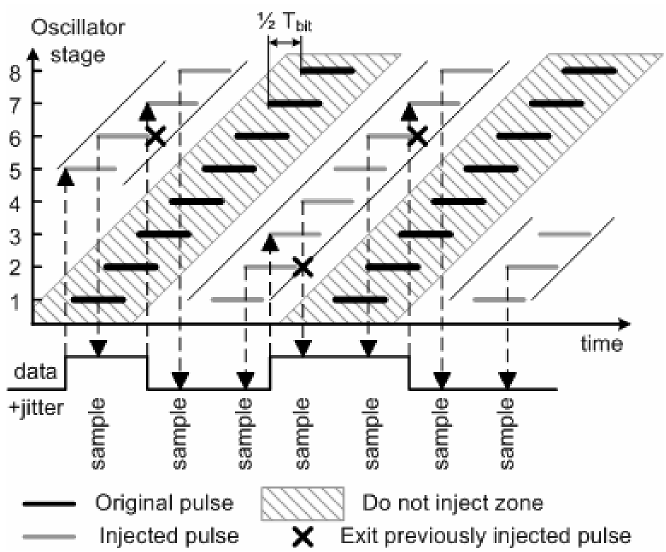

Fig. 2. Space-time plot demonstrating principle of operation.

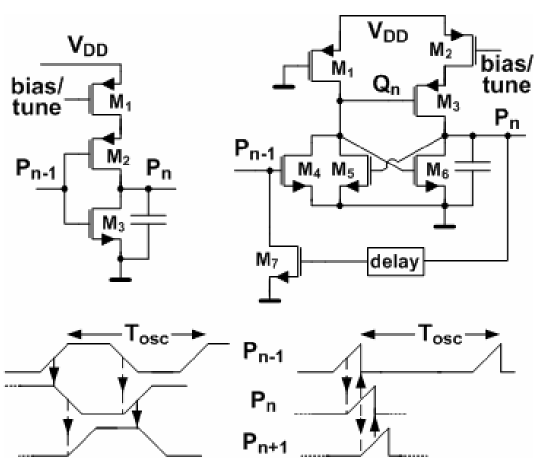

Fig. 3. Comparison of conventional- (left) and sawtooth ring oscillator delay stage (right).

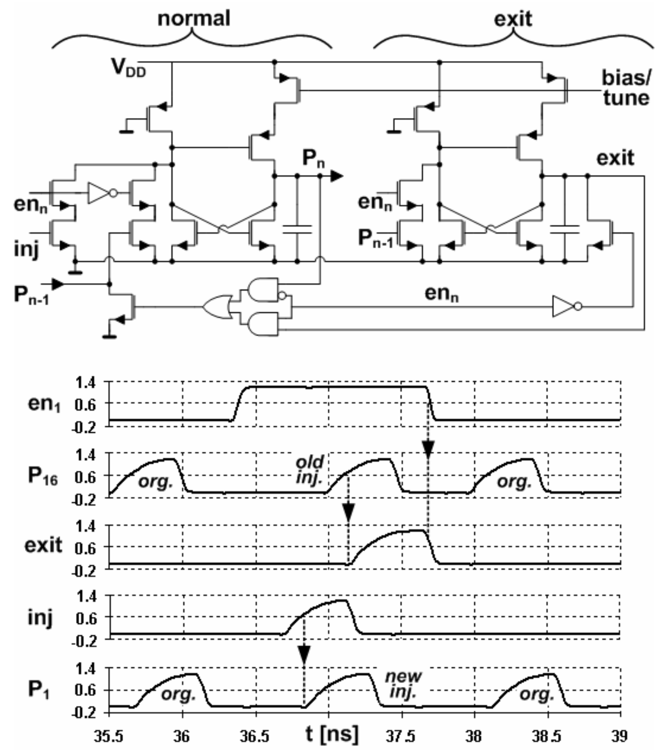

Fig. 4. Single stage of the 16 stage CDR oscillator and simulated waveforms.

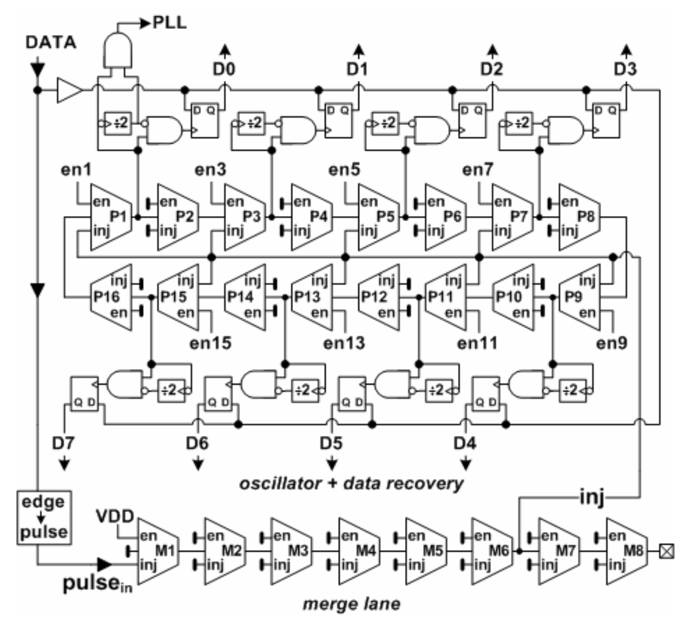

Fig. 5. Oscillator with merge lane and data recovery.

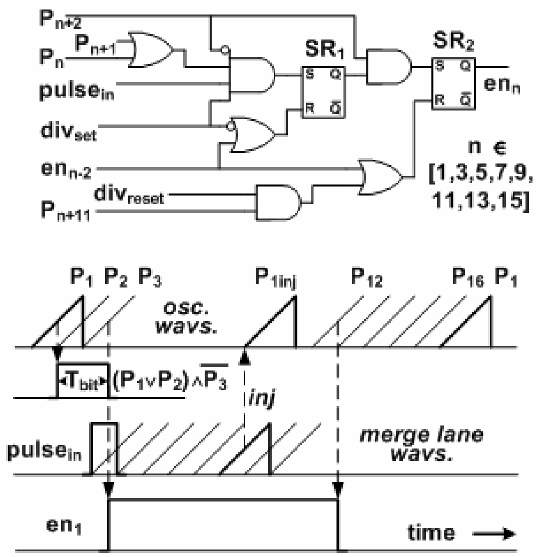

Fig. 6. Enable inject block and waveforms.

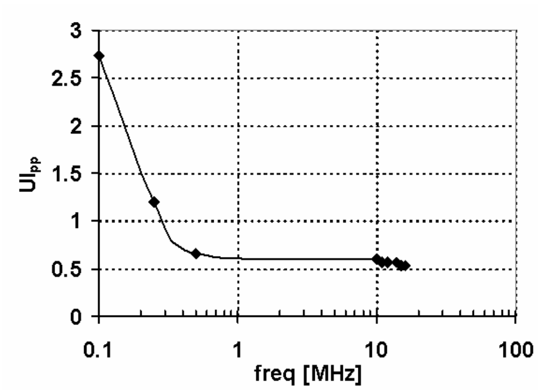

Fig. 7. Measured sinusoidal jitter tolerance (2.5Gb/s, 2^7-1 PRBS).

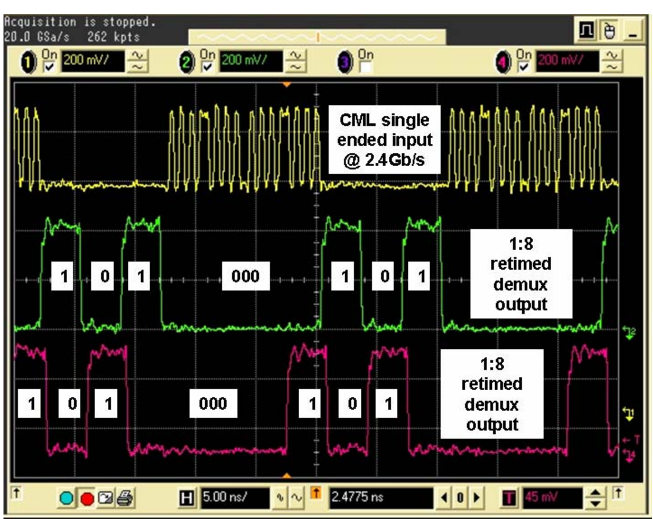

Fig. 8. Scope traces of bursts @ 2.4Gb/s and two 1:8 demuxed outputs.

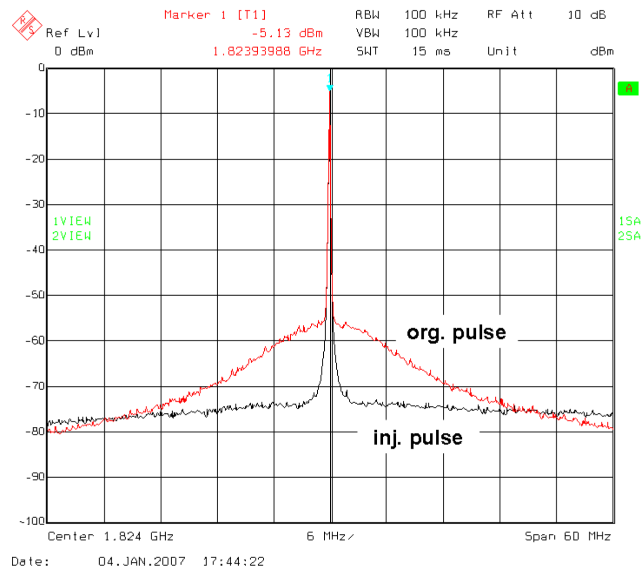

Fig. 9. Spectrum of original and periodically realigned injected pulse.

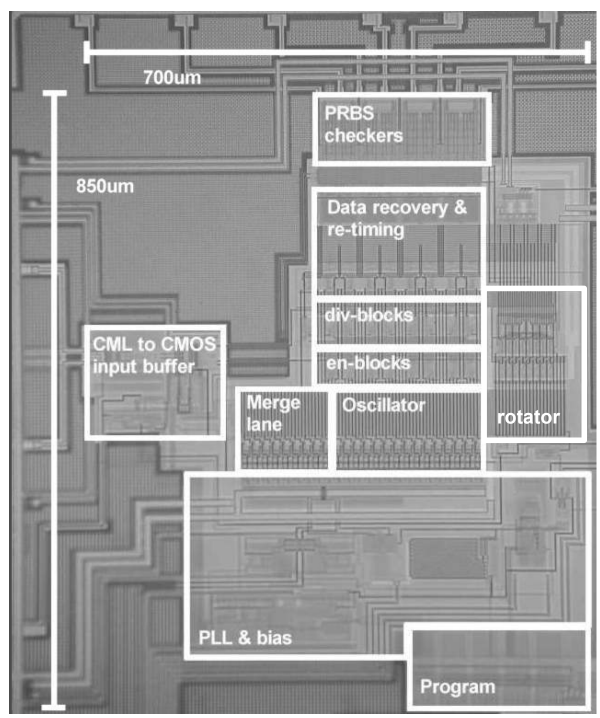

Fig. 10. Die photograph.