Analytical modeling of jitter in bang-bang CDR circuits featuring phase interpolation

Analytical modeling of jitter in bang-bang CDR circuits featuring phase interpolation

Pierpaolo Palestri, *Senior member, IEEE,* Ahmed Elnaqib, Davide Menin, Klaid Shyti, Francesco Brandonisio, Andrea Bandiziol, Davide Rossi, Roberto Nonis

***Abstract*—This paper proposes compact expressions for the jitter in clock and data recovery (CDR) circuits based on bang-bang phase detector including the phase noise of the transmitter and receiver oscillators as well as the quantization noise associated to the finite number of phases of the phase interpolator that align the receiver clock to the incoming data. Different approaches to perform the Early/Late detection on deserialized data and edge samples are compared: the use of majority voting degrades the CDR bandwidth, increasing the impact of the clock jitter on the CDR jitter; on the other hand, counting the single Early/Late occurrences does not degrade the bandwidth but increases the noise related to the finite phases of the phase interpolator. The proposed analytical formulas are validated against event-driven behavioral simulations of the CDR system including free-running oscillators as well as PLL for clock generation.**

***Index Terms*—clock and data recovery, jitter, simulation, high-speed I/O**

I. Introduction

THE increasing demand for high-speed I/O in electronic systems has pushed the development of high-speed serial interfaces operating at rates up to 112 Gb/s [1]. Among the different clocking strategies [2], the use of embedded clock results in pin and power saving but requires clock and data recovery (CDR) systems to align the receiver clock to the received data [3]. The alignment is usually performed with a phase interpolator (PI) that derives a set of discrete phases from the receiver clock [4]. The use of bang-bang phase detectors based e.g. on the Alexander algorithm [5] makes the CDR system essentially digital [6][7].

Being the combination of different sub-blocks, the design of a CDR circuit should start from system-level considerations before actually placing and sizing the single transistors. In this respect, system-level modeling plays an important role in the initial design phase. Although faster than circuit level analysis, time-domain behavioral simulations [8] still require a level of detail that is not needed in the very initial phase of the system planning. For this reason, simple analytical formulas can serve as a starting point to set the main specs of the system.

Linearization of the bang-bang characteristic is the first step to derive analytical formulas for the CDR system [9]-[12]. Having as a target the estimate of the output jitter, a linear model can be used to compute the effect of the single noise sources on the overall jitter. A summary of the main approaches proposed in the literature is provided in [16]. Most of the works focus on the linearization of the phase detector and on the inclusion of loop delay (latency). However, in digital PLL/CDR other blocks feature quantization effects: [13] includes the quantization error of the DCO in a linearized PLL model.

In this paper, we extend the analyses available in the literature by considering a digital CDR with bang-bang phase detector and PI with discrete phases. We derive a set of compact formulas that include the effect of the phase noise of the oscillators (either free-running or PLL) and the finite number of phases of the PI on the output jitter. An event-driven, behavioral simulator was developed to test our model. These formulas are validated over a wide parameter space and provide a useful tool to determine an initial estimation of the required number of phases of the PI, of the main parameters of the digital control algorithm and to set limits to the phase-noise of the oscillators. Different strategies to elaborate the Early/Late (E/L) response of the Alexander algorithm on deserialized data are analyzed and compared. It is found that voting increases the output jitter due to the phase noise of the oscillators but keeps the jitter due to quantization of the PI under control, provided the loop delay is not so large.

The paper proceeds as follows. The derivation of the formulas and the comparison with an event-driven simulator (described in appendix B) is shown in Section II, starting with E/L detection on the serialized data considering free-running oscillators (section II.A) or PLLs (section II.B) as clock generators. The models for E/L detection on deserialized data are derived in Sections II.C and II.D considering an algorithm that accumulates the single E/L occurrence and one based on majority voting, respectively. Comparison with circuit-level simulations is provided in Section III. Conclusions are drawn in Section IV.

II. Derivation and Verification of the Analytical Model for Jitter

To derive the analytical formulas for the CDR jitter (difference between the timing of the reconstructed clock and the received data) we start from the linear model in Fig.1a, assuming a system with a single integrator (i.e. a first-order

Pierpaolo Palestri, Davide Menin and Klaid Shyti are with the DPIA, University of Udine, Italy (e-mail: pierpaolo.palestri@uniud.it).

Ahmed Elnaqib was with the university of Bologna (Italy).

Francesco Bandonisio, Andrea Bandiziol and Roberto Nonis are with Infineon Technologies Villach (Austria)

Davide Rossi is with the University of Bologna, Italy

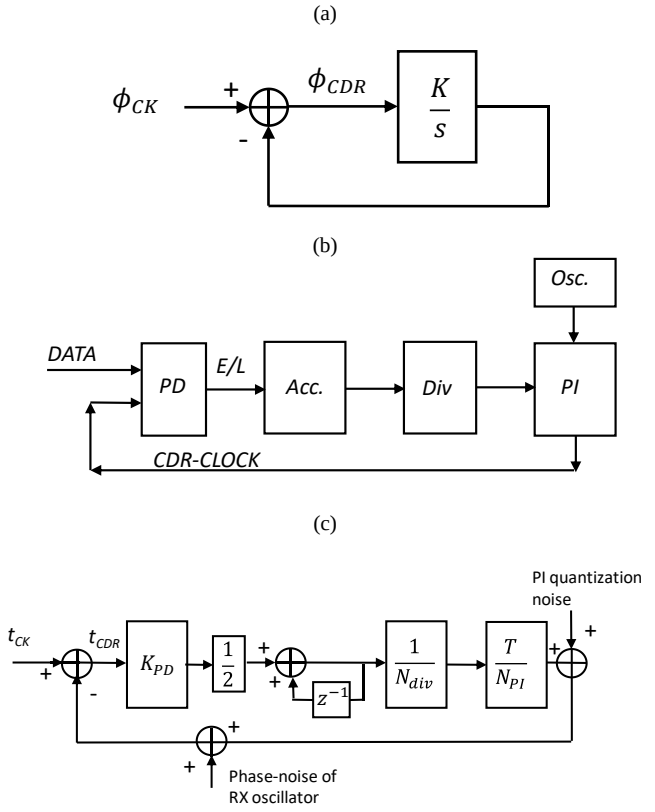


Fig. 1. (a) Linear model in the Laplace domain of a generic CDR with single integrator considering the phase of the transmitter clock as input. (b) CDR implementation considered in this paper: after the phase detector we have an accumulator, a divider (i.e. the output of the accumulator is divided by a given factor discarding some of the least significant bits) and then the PI that generates the CDR clock from a local oscillator. (c) Linear model in the z-domain of the scheme in (b). t_{CK} is the jitter of the transmitter clock, whose period jitter is indicated by σ in the text for a free-running oscillator, while t_{CDR} is the jitter of the reconstructed clock with respect to the timing of the received data. Plot c also indicates the other sources of jitter, i.e. the quantization of the PI phase (whose rms value is indicated as σ_{quant} in the text) and the noise of the RX oscillator (that in this model is indistinguishable from the one of the TX).

CDR). The applicability of our model to second order CDR will be discussed in Appendix C. As far as the phase noise of the oscillators is concerned, the linear model of Fig.1a gives:

$$\frac{\phi_{CDR}}{\phi_{CK}} = \frac{j\omega}{K+j\omega} = \frac{jf/BW}{1+jf/BW} \ , \qquad (1)$$

where the bandwidth is $BW=K/(2\pi)$. The estimate of the K parameter depends on the system architecture. Fig.1b considers the case where the E/L detection is performed directly on the serial data-stream, that will be analyzed in details in the next two sections. The case of E/L detection on deserialized data will be discussed later in Sections II.C and II.D.

A. Free-running oscillator and E/L detection on serial data

The simplest case concerns E/L detection on the serial stream, with transceiver and receiver clocks generated by free-running oscillators. For simplicity, we consider only the noise of the transmitter free-running oscillator. Its phase noise power spectral density (PSD) as a function of the offset frequency f is given by:

$$S_{\phi CK\phi CK} = \frac{A}{f^2} \, , \qquad (2)$$

Where we do not consider flicker noise to simplify the analysis and get close-form expressions. The inclusion of flicker noise would also require going to a second-order CDR.

To determine the constant A, we note that the period jitter corresponding to (2) is [15]:

$$\sigma^2_{osc,p} = \int_0^\infty |1-z^{-1}|^2 \left(\frac{T}{2\pi}\right)^2 S_{\phi\phi} df \cong \frac{AT^3}{2} \ . \qquad (3)$$

We can thus write:

$$S_{\phi CK\phi CK} = \frac{2\sigma^2_{osc,p}}{T^3 f^2} \, . \qquad (4)$$

Working with time is easier than using phases: we consider the block diagram of Fig.1c corresponding to the architecture in Fig.1b

If we assume for the transmitter oscillator the phase noise spectrum of Eq.4, the rms value of the absolute jitter of the recovered clock compared to the received signal (i.e. the rms value of the variable t_{CDR} in Fig.1c) due to the noise of the oscillator only is:

$$\sigma^2_{rj} = \frac{T^2}{4\pi^2} \int_0^\infty \frac{2\sigma^2_{osc,p}}{T^3 f^2} \frac{f^2/BW^2}{1+f^2/BW^2} df = \frac{\sigma^2_{osc,p}}{4\pi TBW} \, . \qquad (5)$$

Notice that Eq.5 links the absolute jitter of the CDR to the period jitter of the transmitter clock. In Fig.1a, consistently with the event-driven simulations, the jitter is applied to the transmitter clock. However, the jitter of the oscillator in the receiver has exactly the same effect (see where the noise of the RX oscillator is added in Fig.1c), so that $\sigma^2_{osc,p}$ in Eq.5 is indeed the sum of the square of the period jitter of both transmitter and receiver oscillators.

To compute the CDR bandwidth, although the model of the accumulator is in the z-domain, since the CDR bandwidth is much smaller than the data rate, we can approximate it as:

$$\frac{1}{1-z^{-1}} = \frac{1}{1-e^{-sT}} \sim \frac{1}{sT} \ . \qquad (6)$$

The block ½ comes from the consideration that, on average, we have 0→1 and 1→0 transitions only in half of the bit periods. Comparing Fig.1a and 1.c and using Eq.6 we get:

$$K = K_{PD} \frac{1}{2N_{div}N_{PI}} \ . \qquad (7)$$

That implies:

$$BW = \frac{K}{2\pi} = K_{PD} \frac{1}{4\pi N_{div}N_{PI}} \ . \qquad (8)$$

Linearization of the bang-bang PD characteristic is discussed in [9]- [11]. Since we indicate with σ_{rj} the rms value of the jitter out of the PD, we can write [16]:

$$K_{PD} = \frac{2}{\sqrt{2\pi}\sigma_{rj}} \ . \qquad (9)$$

Substituting Eqs.8 and 9 into Eq.5 we get σ^2_{rj} that depends linearly on σ_{rj} itself. We finally get:

$$\sigma_{rj} = \frac{\sigma^2_{osc,p}\sqrt{\pi/2}N_{div}N_{PI}}{T} \ . \qquad (10)$$

The results of Eq.10 are reported by red-dashed lines in Fig.2 and show that the jitter of the clock generators is the main contributor to the CDR jitter when the bandwidth is small (high N_{div} and N_{PI}). On the other hand, for small N_{div} and N_{PI}, Eq.10 significantly deviates from the results of the event-driven simulations (blue curve). This is due to the quantization noise associated to the finite number of PI phases [17].

To model this additional contribution, we consider that the desired phase will stay between two of the possible phases

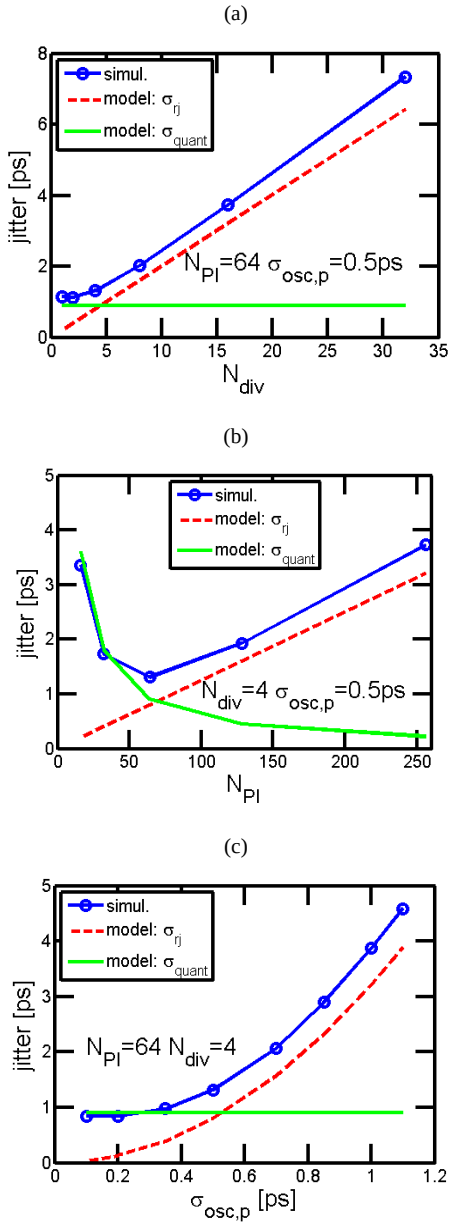


Fig.2. Blue lines with circles: simulated absolute jitter for a CDR operating at 10Gbps on the serial data considering a free-running oscillator as the transmitter clock source. The dashed red and solid green lines refer to Eqs.10 and 13 respectively. Plots a,b and c show the results as a function of respectively the divider ratio, the number of phases of the PI and the period jitter of the oscillator.

provided by the PI. Using time instead of phase, let us say that we want a delay "x" with respect to the delay of the n-th phase, whereas the (n+1)-th phase is $\Delta = T/N_{PI}$ from the n-th phase. The system will switch between the two phases since one time the CDR clock is "late" and the next time it will be "early". The rms error is:

$$\sigma^2(x) = \frac{1}{2}(\Delta - x)^2 + \frac{1}{2}x^2 = \frac{1}{2}\Delta^2 + x^2 - \Delta x \,. \qquad (11)$$

In absence of any random jitter, the jitter histogram of the CDR clock is composed by two Dirac's delta spaced by $\Delta = T/N_{PI}$, as verified by event-driven simulations (not shown). In the presence of random jitter from the oscillators, "x" continuously changes over time. The average error is:

$$\sigma^2_{quant} = \frac{1}{\Delta}\int_0^{\Delta} \sigma^2(x)dx = \frac{\Delta^2}{3} \,. \qquad (12)$$

In other words, the absolute jitter of the CDR due to quantization of the PI is:

$$\sigma_{quant} = \frac{T}{N_{PI}\sqrt{3}} \qquad (13)$$

This differs from the formula for jitter associated to PI quantization proposed in [17] that has a $\sqrt{12}$ instead of $\sqrt{3}$ of Eq.13. In fact, the quantization noise of the PI is not the same as in an ADC where we approximate a number with the closest level: here the PI oscillates between the two phases that are around the wanted phase.

Results of Eq.13 are reported by green lines in Fig.2 and reproduce the event-driven simulations (see Appendix B) in the regions where the jitter induced by the noisy oscillator (Eq.10) is low. Note that the quantization noise of the PI is a white noise source that must be added to the output of the PI (see Fig.1c) [17]. In principle Eq.13 gives the rms value of this contribution, whose PSD is $2T\sigma^2_{quant}$ (i.e. white over the Nyquist bandwidth). This noise sees the high-pass transfer function of the CDR loop as for Eq.1. However, since the CDR bandwidth is much lower than the Nyquist frequency *1/(2T)*, the integral of the PSD over the squared modulus of the transfer function essentially gives Eq.13.

From Fig.2a we see that the jitter increases with N_{div} (the CDR bandwidth is reduced) while the dependence on N_{PI} (Fig.2b) shows that small values increase the jitter due to phase quantization; on the other hand, when N_{PI} increases, the reduction of the jitter induced by PI quantization is then accompanied by the increase of the jitter due to the noisy oscillator (since the CDR bandwidth gets reduced and is not able to track the oscillator jitter anymore). We also see in Fig.2c that the CDR jitter goes quadratically with the period jitter of the oscillator (i.e. linearly with the phase-noise coefficient *A* of Eq.2, consistently with [13]). In Fig.2 we considered a data rate of 10Gbps, but everything scales with T so that the agreement between model equations 10, 13 and the simulation is good in any range of data rates (not shown).

Indications about how to combine the results of Eqs.10 and 13 and include the quantization noise of the bang-bang phase detector are provided in Appendix A. Visual observation of the figures in this one and in the following sections suggests that Eq.10 and Eq.13 can be summed together, an evidence that is consistent with [13].

As a final note, we should mention that simulations run with a clock-like ...01010... data sequence provide a jitter that can be reproduced by a combination of Eq.13 (the jitter due to PI quantization is the same) and Eq.10 divided by two (the bandwidth is twice as large since we have a 0→1 or 1→0 transition for each bit period). Simulations with different frequency of the transmitter and receiver clock without phase noise yield a jitter that can be reproduced by Eq.13 since the phase error between data and clock tends to increase linearly with time and the "wanted phase" out of the PI moves from 0 to 2π and experiences the quantization of the PI.

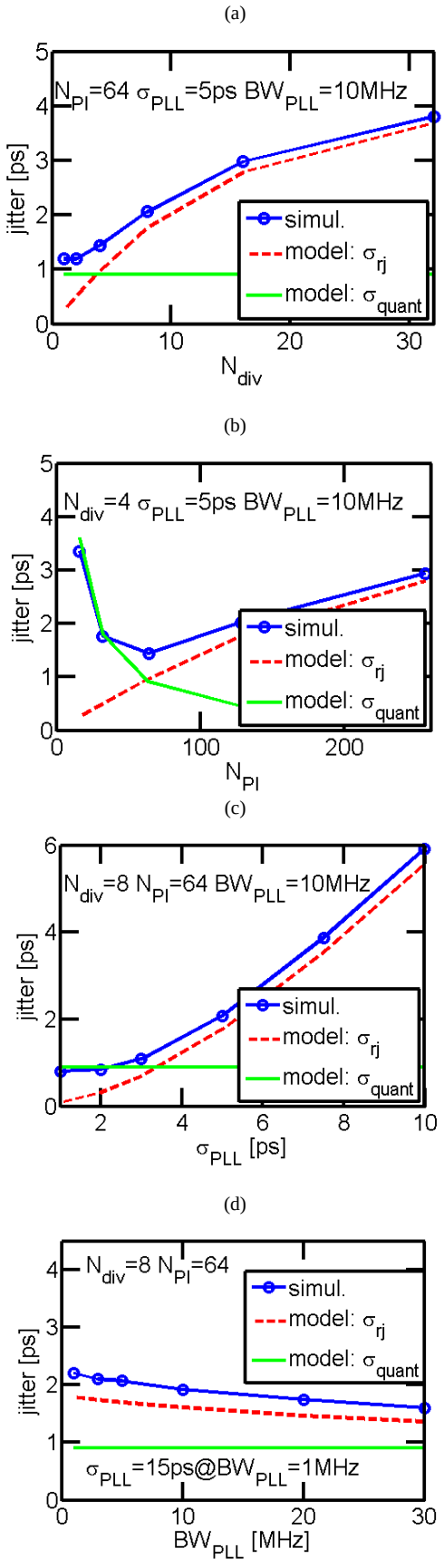


Fig.3. Blue lines with circles: simulated absolute jitter for a CDR operating at 10Gbps on the serial data considering a PLL in the transmitter. The dashed red and solid green lines refer to Eqs.17 and 13 respectively. Plots a,b,c and d show the results as a function of respectively the divider ratio, the number of phases of the PI, the absolute jitter of the PLL and the bandwidth of the PLL. In plot d, σ_{PLL} is scaled as $\frac{1}{\sqrt{BW_{PLL}}}$.

B. Clock from PLL and E/L detection on serial data

The derivation in the previous section can be adapted with little effort to a case where the transmitter is clocked by a PLL.

We approximate the phase noise PSD as:

$$S_{\phi CK\phi CK} = \frac{A}{BW_{PLL}^2+f^2} \quad , \tag{14}$$

i.e. a flat PSD up to the PLL bandwidth BW_{PLL} and then a $1/f^2$ slope. In this case the absolute jitter is given by:

$$\sigma_{PLL}^2 = \int_0^\infty \left(\frac{T}{2\pi}\right)^2 S_{\phi\phi}\, df \cong \frac{AT^2}{8\pi BW_{PLL}} \; . \tag{15}$$

The analysis starts with Eq.1. If we proceed as in Eq.5 but using the phase PSD of Eq.14 instead of the one from Eq.4, we get:

$$\sigma_{rj}^2 = \sigma_{PLL}^2 \frac{1}{1+\frac{1}{BW_{PLL}\sqrt{\pi/2}\sigma_{rj}^4 \pi N_{div} N_{PI}}} . \tag{16}$$

Where we have used Eqs. 8 and 9 for the CDR bandwidth. We can solve Eq.16 and get:

$$\sigma_{rj} = -\frac{1}{C} + \sqrt{\frac{1}{C^2} + \sigma_{PLL}^2} \quad , \tag{17}$$

$$C = BW_{PLL}\sqrt{\frac{\pi}{2}} 8\pi N_{div} N_{PI} \; .$$

Results of Eq.17 are reported by dashed red lines in Fig.3 and compared with the event-driven simulator. For small N_{div} and N_{PI} event-driven simulations tend to the jitter provided by Eq.13 (quantization of the PI). The trends of CDR jitter vs division ratio (Fig.3a), number of PI phases (Fig.3b) and clock jitter (Fig.3c) are the same as in Fig.2. Regarding the effect of the PLL bandwidth, in Fig.3d we change the BW_{PLL} and, at the same time, decrease σ_{PLL}^2 as $1/BW_{PLL}$, to describe a situation where the VCO of the PLL is the same, but the increase of the PLL bandwidth better removes the VCO phase noise at low frequencies. In other words, we consider a phase-noise PSD as in Eq.14 where A is constant. Since the transfer function between the clock jitter and the CDR jitter (Eq.1) is high-pass, increasing the PLL bandwidth has a minor effect on reducing the overall absolute jitter of the system since it affects the low frequency PSD of the clock phase-noise.

Since the results obtained with the PLL are in line with the ones using a free-running oscillator apart from more involved formulas, we will consider for simplicity only this latter case in the following (although PLL clock may be included also in the more complex situations described in the next sections).

C. Free-running oscillator and E/L detection after deserialization

We now consider CDR systems where the E/L detection is performed after deserialization. Two possible system architectures are shown in Fig.4. We start with the case in Fig.4a where the E/L detected from deserialized bits are added together. In this way, almost no information is lost compared to voting where a single E/L is derived from the parallel E/L vector instead (Fig.4b). In any case, even with the adder, some information is lost: we just cannot perform the E/L detection on

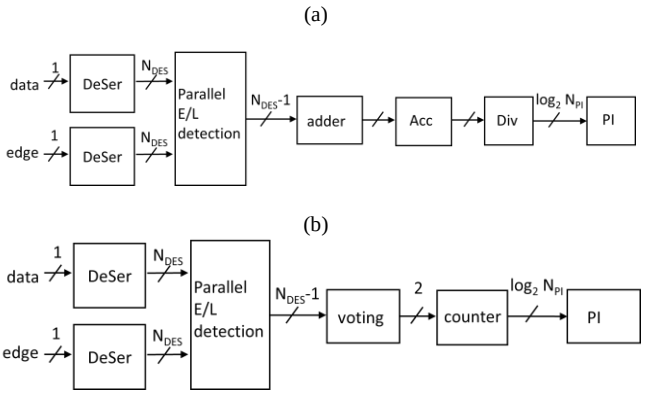


Fig. 4. Example of CDR algorithms performing the E/L detection on deserialized data and edge samples. In both cases the PI code is updated after N_{DES} data periods. In plot (a) the E/L resulting from the parallel data are summed together (with sign) and then accumulated and divided before driving the PI. The voting in plot b, instead, produces "up" and "down" signals for the counter that drives the PI.

the last bit in the parallel stream since this requires the first data sample in the next parallel word and this is why in Fig.4a the vector out of the E/L block is N_{DES}-*1* bit wide. When the CDR works on deserialized data, the PI code is updated only after N_{DES} bit periods

By performing event-driven simulations without transmitter jitter, we notice that the phase takes more than two values. In fact, the input of the PI is updated by a value that is the output of the adder after division. If the division ratio is small and a sufficient majority of Early or Late is detected at the same time from the deserialized data, the PI phase moves by more than one step. The number of lines in the jitter histogram is approximately N_{DES}/N_{div} (not shown). This is different from the cases analyzed in Sections II.A and II.B where only one E/L is added to the accumulator and thus to the PI input.

Sample simulation results including oscillator jitter are reported in Fig.5. We see that for small CDR bandwidth (high N_{div}) the jitter is almost the same, whereas working with deserialized data is detrimental in the cases where quantization of the PI dominates (small N_{div}).

The results in Fig.5 suggest that Eq.10 can still be used but one has to modify Eq.13. The fact that given the target phase, the PI switches between more than two phases, led us to propose

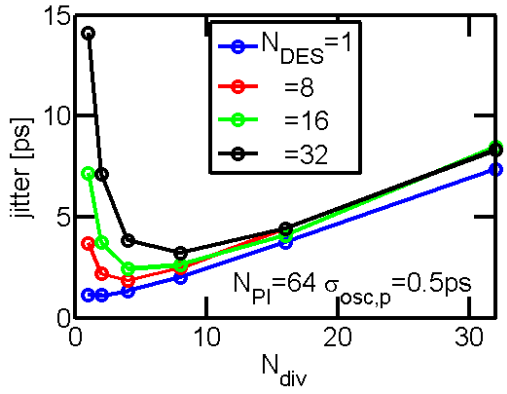


Fig. 5. Results of event-driven simulations for the scheme in Fig.4a considering a data-rate of 10Gbps considering a free-running oscillator in the transmitter.

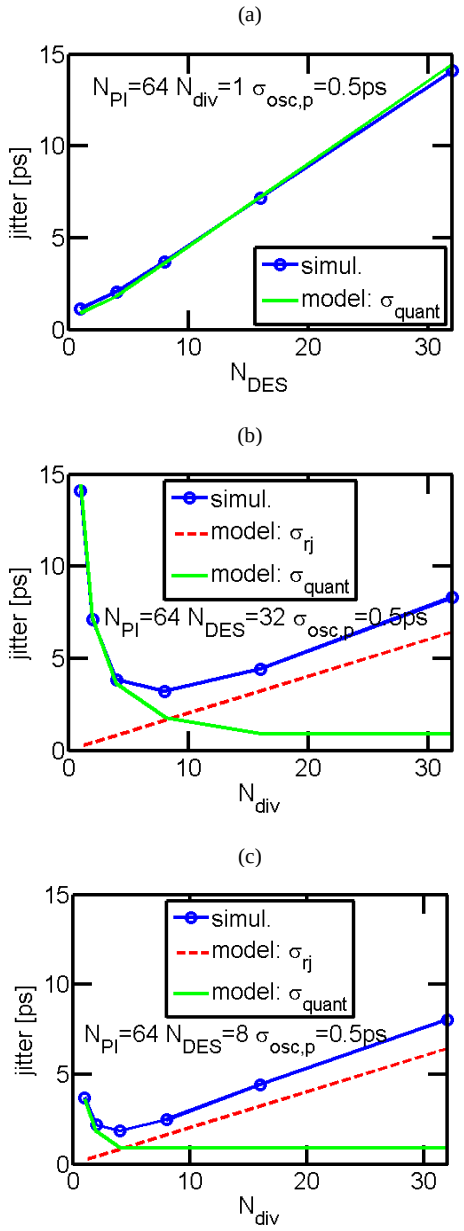


Fig. 6. Blue lines: simulated absolute jitter for a CDR operating at 10Gbps on the deserialized data (using the scheme in Fig.4a) considering a free-running oscillator in the transmitter. The red and green lines refer to Eqs.10 and 18 respectively. Plots a is for varying N_{DES} (parallel bits) while plots b and c are for varying divider ratios at fixed N_{DES}.

to modify Eq.13 as:

$$\sigma_{quant} = \frac{T}{N_{PI}\sqrt{3}} max\left(1, floor\frac{N_{DES}}{2N_{div}}\right) . \quad (18)$$

We recover Eq.13 when $N_{DES}<2N_{div}$. The results of Eq.18 are compared with event-driven simulations in Fig.6a for situations with small N_{div} so that the jitter contribution due to the oscillator noise is negligible. Fig.6b instead plots the results of Eq.10 and Eq.18 vs event-driven simulations for E/L detection after 32-bit deserialization, demonstrating that quantization noise due to discrete PI phases dominates at small N_{div} (high bandwidth), while noise of the oscillator dominates for high N_{div} (small bandwidth). This leads to a trade-off in the choice of the divider ratio, with the optimum division ratio that depends on N_{DES} and on the jitter of the oscillators (compare Fig.6b and Fig.6c).

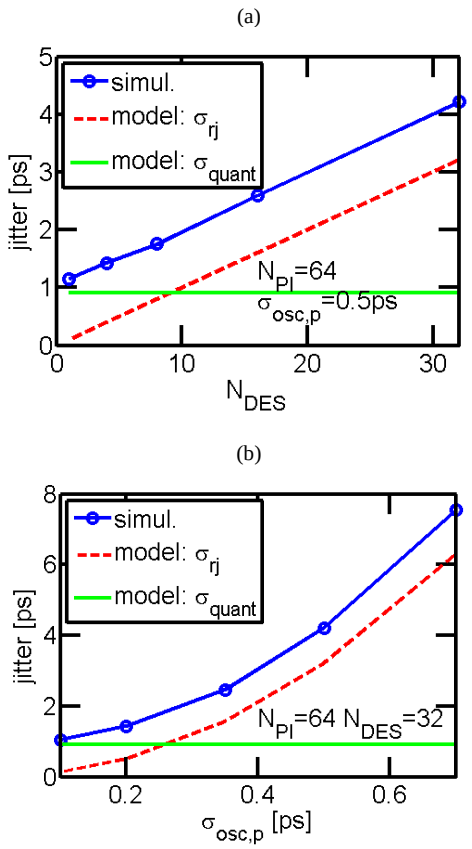


Fig. 7. Blue lines: simulated absolute jitter for a CDR operating at 10Gbps on the deserialized data (using the scheme in Fig.4b) considering a free-running oscillator in the transmitter. The red and green lines refer to Eqs.19 and 13 respectively. Plot a is for varying N_{DES} (parallel bits) while plot b is for varying period jitter of the transmitter oscillator.

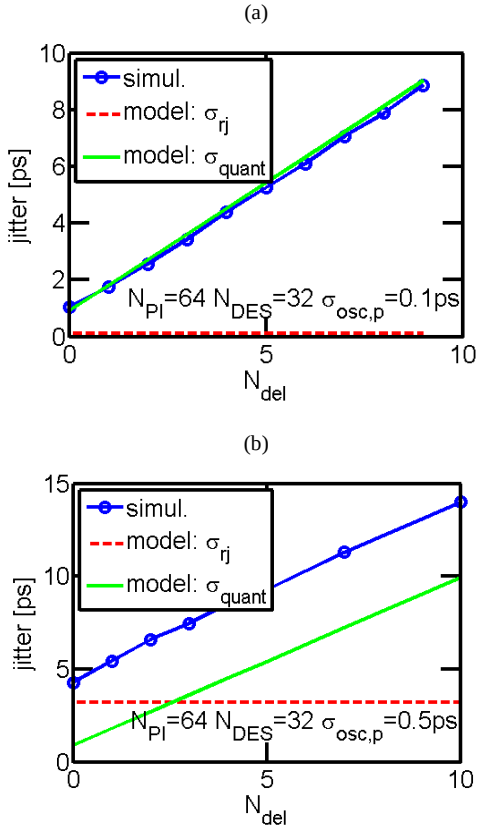


Fig. 8. Blue lines: simulated absolute jitter for a CDR operating at 10Gbps on the deserialized data (using the scheme in Fig.4b) considering a free-running oscillator in the transmitter. The red and green lines refer to Eqs.19 and 20 respectively. Plots a and b consider two different values of the period jitter of the clock and plot the CDR jitter for varying N_{DEL} (delay of the digital core).

D. Free-running oscillator and E/L detection after deserialization with voting

An alternative scheme for E/L detection on deserialized data and edges is depicted in Fig.4b. In this case, majority voting is performed on the parallel E/L vector: the output of the voting block controls the up and down inputs of a counter that drives the PI. This guarantees that the phase is always updated by +/- 1. As a result, the noise related to quantization of the PI is the same as when E/L detection is performed on serial data, i.e. Eq.13 is valid.

On the other hand, the voting reduces the bandwidth of the CDR, meaning that Eq.10 (jitter due to the noise of the transmitter oscillator) has to be modified as follows

$$\sigma_{rj} = \frac{\sigma_{osc,p}^2 \sqrt{\pi/2} N_{DES} N_{PI}}{2T} \ . \qquad (19)$$

In other words, the bandwidth is reduced by a factor $N_{DES}/2$ (we have on average $N_{DES}/2$ useful transitions but the voting counts only one) compared to the CDR working on serial data (where ½ of the bit periods show useful transitions).

Fig.7 compares the model consisting of Eqs.19 and 13 with event-driven simulations. In this system architecture, deserialization leads to an increase of the impact of the phase noise of the oscillators on the overall CDR jitter, while the effect of PI quantization is limited.

Another important aspect related to the digital implementation of the CDR is the latency of the loop [6][17][18]. In fact, E/L detections as well as voting require some clock cycles (at the rate of the deserialized data that we denote as N_{del}). The associated delay is $N_{DES}N_{del}T$. This delay has an effect on the loop bandwidth (but small, as we will see). The relevant effect is on the noise associated to PI quantization. In fact, the outcome of the E/L detection arrives delayed to the PI so that the output phase features limit cycles including more than two phases, due to outdated control words computed previously. Simulations using the event-driven simulator of Appendix B when the update of the PI code is delayed are reported by blue lines in Fig.8. Even when the jitter of the clock is small, the jitter of the CDR increases linearly with N_{del}. In particular, the jitter contribution due to quantization can be modeled as:

$$\sigma_{quant} = \frac{T}{N_{PI}\sqrt{3}} (1 + N_{del}) \ , \qquad (20)$$

In fact, the state of the CDR evolves in limit cycles whose amplitude is proportional to N_{del} [19]. As a consequence, the output of the PI switches among a larger set of phase values centred on the phase to be tracked, thus increasing the quantization noise proportionately to N_{del}. Therefore, Eq. 20 should be interpreted as the quantization noise due to the intrinsic one-cycle delay of the CDR plus another quantization

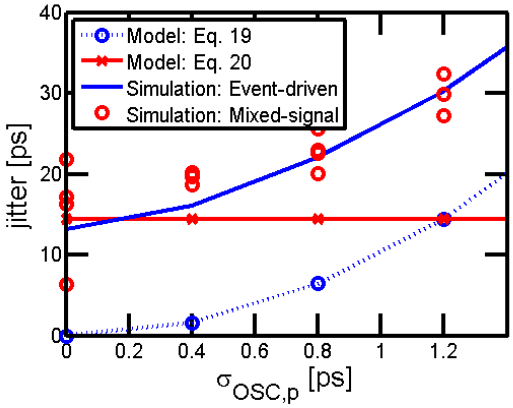


Fig. 9. Simulated jitter of the CDR described in [20] operating at 6.25Gbps. Mixed-signal simulations are compared with the event-driven simulator of Appendix B and with the model in Section II.D.

noise contribution proportional to N_{del}.

We see that the sum of Eqs.19 and 20 reproduces quite well the jitter of the event-driven simulator, suggesting that Eq.19 is still valid, i.e. that the additional $z^{-N_{DES}N_{del}}$ term in the loop transfer functions has a small impact on the transformation of the clock jitter into CDR jitter. Notice that a latency of approximately $0.5N_{DES}T$ is in any case present when working with deserialized data even if the digital core latency is not included.

The effect of the digital core latency on the CDR jitter is severe and requires the inclusion of a divider after the counter in Fig.4b. This reduces the noise associated with quantization by roughly N_{div}, but unfortunately reduces the bandwidth by the same amount, increasing the CDR jitter due to clock jitter.

III. Comparison with Circuit-Level Simulations

To show the applicability of the models proposed in the previous section to realistic situations and validate it against schematic-level simulations, we have simulated the digital CDR circuit of [20] using the mixed-signal simulator XA-VCS. E/L detection is performed after 1:40 deserialization with majority voting. Then the output of the voter is deserialized 1:2 and a further majority voting is implemented. The resulting signal changes the phase of the PI by -1,0 or +1.

Results are reported in Fig.9. The mixed-signal environment couples the gate-level analog circuit to the HDL description of the digital circuit. The oscillators and their noise are not included at schematic level: rather we apply to the CDR noiseless clock with jittered edges to implement period jitter with rms value $\sigma_{OSC,p}$. Due to the non-linearity of the PI, we consider different delays between TX and RX clocks to span different portions of the PI characteristic and average out the effect of PI non-linearity. This is why we have different red circles in Fig.9 for the same $\sigma_{OSC,p}$ value. The cloud formed by the red circles matches quite well the results of the event-driven simulator described in Appendix B (that for this analysis has been slightly modified to handle the double voting algorithm of the CDR considered here).

Fig.9 also reports the results of the compact formulas proposed in Section II.D. Since we have a 1:2 deserialization after the 1st voting on the 1:40 deserialized data and edge, in Eq.19 we set $N_{DES}=80$. Concerning the value of N_{del} to insert into Eq.20, the delay of the digital part of the CDR is about 8 cycles of the clock that times the 1:40 deserialized data. Since the final voting is essentially on 80 bits, Ndel=4 has been inserted into Eq.20. We see that Eq.20 gives a quite fine estimate of the CDR jitter due to the limit cycle of the PI phases (i.e. $\sigma_{OSC,p} = 0$ in Fig.9), while Eq.19 captures the increment of the CDR jitter when $\sigma_{OSC,p}$ increases.

TABLE I
Summary of Main Formulas for Free-Running Oscillator

Architecture	Contribution of oscillator jitter σ_{rj}	Quantization noise σ_{quant}
E/L on serial data	$\frac{\sigma_{osc,p}^2\sqrt{\pi/2}N_{div}N_{PI}}{T}$	$\frac{T}{N_{PI}\sqrt{3}}$
E/L on deserialized data with adder	$\frac{\sigma_{osc,p}^2\sqrt{\pi/2}N_{div}N_{PI}}{T}$	$\frac{T}{N_{PI}\sqrt{3}}max\left(1, floor\frac{N_{DES}}{2N_{div}}\right)$
E/L on deserialized data with voting	$\frac{\sigma_{osc,p}^2\sqrt{\pi/2}N_{DES}N_{PI}}{2T}$	$\frac{T}{N_{PI}\sqrt{3}}$
E/L on deserialized data with voting including loop latency	$\frac{\sigma_{osc,p}^2\sqrt{\pi/2}N_{DES}N_{PI}}{2T}$	$\frac{T}{N_{PI}\sqrt{3}}(1+N_{del})$

IV. Conclusion

We have derived a set of simple formulas (summarized in Tab.I for the case of free-running oscillator) for a first-order estimate of the CDR jitter due to noisy oscillators as well as quantization of the PI phases including also effects such as voting and loop latency. These formulas match quite well the event-driven behavioral simulations. The latter requires in any case less than a minute to perform a simulation, so that the main advantage of the compact formulas is to show the main trade-off and guide in the initial phases of the design.

The analysis identifies a trade-off in the choice of the number of PI phases: few phases results in higher quantization jitter, whereas a large number of phases reduces the CDR bandwidth. When E/L detection is performed on deserialized data and edge samples, the use of voting results in a significant bandwidth penalization. On the other hand, accumulating the single E/L occurrence amplifies the quantization noise of the PI. This latter choice is then preferable when the noise of the oscillators is large and the bandwidth penalty associated with voting cannot be afforded. Care has to be taken since the loop latency increases the jitter due to PI quantization.

The analysis presented here assumed an ideal channel without inter-symbol interference and noise. Inclusion of a realistic channel response into the model is ongoing. Preliminary results point out that the main trends described in this paper (effect of PI steps, impact of deserialization) are still valid, although the absolute value of jitter shows an additional dependence on the channel attenuation. An example of such

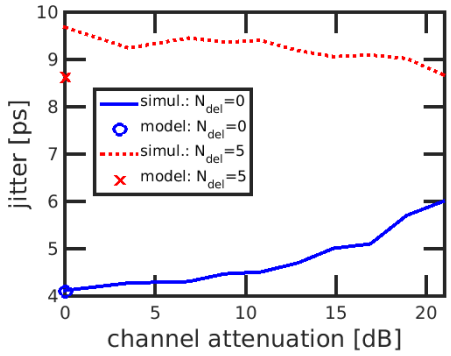


Fig. 10. Jitter for the same situations as in Fig.8b obtained with a simulator (under development) that includes effect of channel ISI on the received data. The symbols are the results of Eq.19 and 20 summed together.

results is reported in Fig.10: when the jitter associated to PI quantization dominates (case with N_{del}=5 in the figure), the impact of the data-dependent jitter introduced by the channel is limited (we even see a slight decrease of the total jitter); on the other hand, in cases dominated by the oscillator phase-noise, the presence of the data-dependent jitter further reduces K_{PD} and the overall jitter increases. The small difference between model and simulation without channel for N_{del}=5 is in line with the fact that Eq.(20) is indeed approximated.

As a final remark, in this paper we limited the analysis to 1st-order CDR architectures, where closed-form expressions can be derived. The applicability to 2nd-order cases is discussed in Appendix C, showing that the expressions found here are adequate in most cases also to describe 2nd-order CDR architectures.

APPENDIX A: COMBINING THE DIFFERENT JITTER SOURCES

We have provided formulas for the CDR jitter due to noisy oscillators (σ_{rj}) and finite PI phases (σ_{quant}). The two contributions have been derived independently, but since K_{PD} depends on the jitter itself, they are interdependent. Furthermore, we have neglected the quantization noise due to the binary output of the bang-bang phase detector [16]. For simplicity, we consider a CDR operating on serial data with a free-running oscillator (same as Section II.A). The total CDR jitter can be expressed as:

$$\sigma^2_{CDR,tot} = +\sigma^2_{quant} + \sigma^2_{qPD} \ . \quad (21)$$

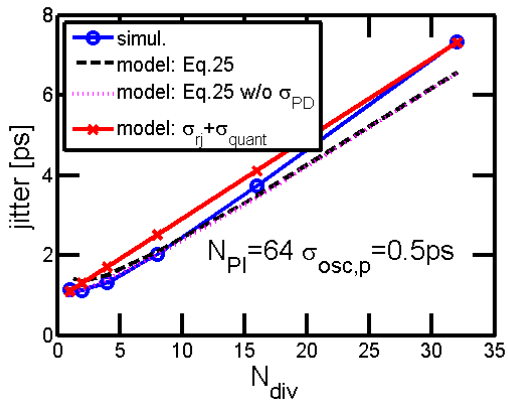
Where the first term is the contribution to the CDR jitter of the oscillator period jitter. It can be computed as in Section II.A but considering that K_{PD} would depend on $\sigma_{CDR,tot}$ instead of σ_{rj}. So, adapting Eq.5 one finds:

$$\sigma^2_{CDR|osc} = \frac{\sigma^2_{osc,p}}{4\pi TBW} = \frac{\sigma^2_{osc,p}\sqrt{\pi/2}N_{div}N_{PI}}{T}\sigma_{CDR,tot} = \sigma_{rj}\sigma_{CDR,tot} \ . \quad (22)$$

where σ_{rj} is given by Eq.10.

The term σ^2_{quant} in Eq.21 is the square of what is expressed in Eq.13. The term σ^2_{qPD} is the contribution of the quantization of the PD output to the CDR jitter. Following [16], we model it as a white noise at the input of the accumulator with rms value $0.5\cdot(1-1/\pi)$. This corresponds to a white PSD on the Nyquist band from 0 to $1/(2T)$ that is filtered by the CDR loop. We can

(a)



(b)

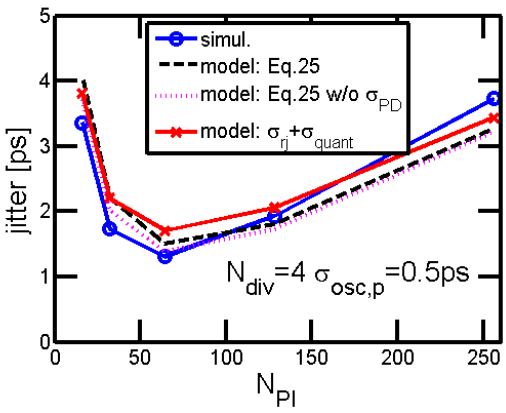


Fig. 11. Blue lines: simulated absolute jitter for a CDR operating at 10Gbps on the serial data (same as in Fig.2a and b). The results of Eq.25 with and without the term σ_{PD} are compared with the sum of σ_{rj} and σ_{quant} from Eqs. 10 and 13.

compute:

$$\sigma^2_{qPD} = \frac{1}{2}\left(1-\frac{1}{\pi}\right)2T\frac{4}{K^2_{PD}}\int_0^\infty \frac{BW^2}{BW^2+f^2}df = \sigma_{PD}\sigma_{CDRtot} \ , \quad (23)$$

where we introduce the term

$$\sigma_{PD} = \left(1-\frac{1}{\pi}\right)\frac{T}{2N_{div}N_{PI}}\sqrt{\frac{\pi}{2}} \ . \quad (24)$$

By inserting Eqs.22 and 23 in Eq.21, one gets a second order equation giving:

$$\sigma_{CDR,tot} = \frac{1}{2}\left(\sigma_{rj}+\sigma_{PD}+\sqrt{\left(\sigma_{rj}+\sigma_{PD}\right)^2+4\sigma^2_{quant}}\right) \quad (25)$$

The results of Eq.25 are compared with the event-driven simulations in Fig.11. Comparison between black and magenta lines points out that σ_{PD} has a minor effect on the CDR jitter, consistently with [17]. Overall Eq.25 matches the event/driven simulations, with some deviation at high division ratios or high number of PI steps. On the other hand, the event-driven simulations are matched quite well also by the simple sum $\sigma_{rj}+\sigma_{quant}$, that is consistent with [13] and can be verified by visual inspection of the other figures in the previous sections.

APPENDIX B: DESCRIPTION OF THE EVENT-DRIVEN SIMULATOR

We describe here the simple event-driven simulator for CDR systems used to verify the proposed models. Each simulation time step corresponds to the occurrence of an *event*, i.e. the data and clock transitions, that are computed similarly to what is done in [14] for the simulation of PLLs.

A random sequence of bits is initially generated with an associated vector containing the time border between two adjacent bit periods. In absence of jitter, such timing samples are separated by T seconds, where T is the inverse of the data rate. When considering a free-running oscillator, we add to each period T a random time t^*_{PER} normally distributed with variance σ. When considering a PLL, Inverse Fourier transform of $\sqrt{S_{\phi CK\phi CK}}$ from Eq.14 with an additional random phase (uniformly distributed between 0 and 2π) in each frequency bin is then used to generate the displacements t^*_{ABS} so that the n-th bit transition occurs at $nT + t^*_{ABS}$.

Two additional vectors are used at the receiver for the clock edges where data and edges are sampled, *T/2* seconds apart (we assume that transmitter and receiver have the same frequency and only their phases need to be aligned). As the simulation evolves during time, the n-th elements of data and edge clocks are determined based on the data and edges sampled in the previous instants.

First, the sampled values of data and edges are found by looking at the alignment of data and edge clocks with the transmitted data. This implicitly assumes that the channel does not introduce any distortion or inter-symbol interference.

The sampled data and edges are then used to derive the E/L information based on the Alexander algorithm [5].

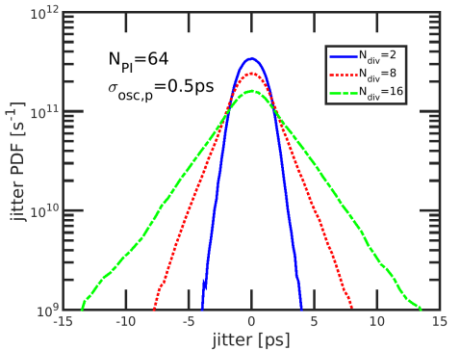
The E/L information is used to update the PI code. When E/L detection is performed on the serial data, they are simply accumulated, scaled by N_{div} (that is usually a power of two, hence truncating some of the accumulator's least significant bits) and used to drive the PI that has N_{PI} phases. Once the phase of the PI is known, we can derive the next data and edge sampling times*:* in a nutshell, the phase of the PI determines a delay between *–T/2* and *T/2* in addition to the clock period *T*. Then the algorithm proceeds for a number of time steps sufficient to gather enough statistics.

Implementation of CDR on deserialized data is straightforward: the PI code is updated every N_{DES} bit periods summing up the E/L values of the last N_{DES}*-1* bit periods.

The jitter histogram is derived by comparing the transmitted data transitions with the edge clock. Sample distributions are reported in Fig.12: the presence of the oscillator noise dominates and makes them fairly Gaussian. Small deviations from Gaussian distributions are observed for small N_{div}, where quantization noise dominates.

It would be straightforward to apply jitter also to the receiver clock: the distance between two consecutive data (or edge) clock samples would not be simply *T* plus the delay associated to the PI, but would also include additional jitter terms as above. The effect on the overall CDR jitter is exactly the same as above, at least in the cases considered here where there is no channel and thus no amplification of the transmitter jitter.

(a)



(b)

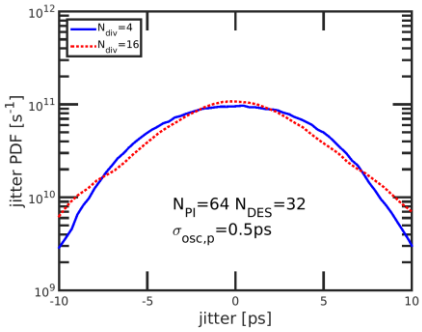


Fig. 12. Sample jitter distributions obtained with the event-driven simulator. Plot (a) refers to a CDR on serial data with parameters as in Fig.2, whereas plot (b) considers CDR on deserialized data as in Fig.5.

APPENDIX C: VALIDITY OF THE MODEL IN 2ND-ORDER CDR

Although the analysis carried out in this paper has been developed for a 1st-order CDR, we have verified (using the event-driven simulator described in Appendix B, where adding the second integrator is quite trivial) that the jitter is in most cases quite similar in 1st- and 2nd-order CDRs. This is particularly true when the integral path in the 2nd-order CDR is made strong enough to compensate for difference in frequency between the TX and RX oscillators as large as 100ppm, but not as strong as to affect the loop transfer function experienced by phase noise. An integral path stronger than needed may affect the CDR transfer function, thus modifying the output jitter due to the noisy oscillators. On the other hand, when the jitter due to PI quantization is dominant, the effect of the CDR transfer function is negligible (the quantization noise is white up to the Nyquits frequency that is orders of magnitude higher than the CDR bandwidth) and 1st- and 2nd-order CDRs display essentially the same jitter, so the expressions for σ_{quant} reported in this paper apply to both 1st- and 2nd-order CDRs.

A compact formula for σ_{rj} can be obtained only in the simplified case of CDR performing E/L detection on serial data and featuring a free-running oscillator (i.e. the case analyzed in Section II.A). To derive this expression, we consider for the 2nd-order CDR the same scheme as in Fig.1, but with an additional block before the accumulator. This block features a direct path with unitary gain (i.e. the gain of the proportional k_P path will be the factor K from Eq.7) in parallel with an

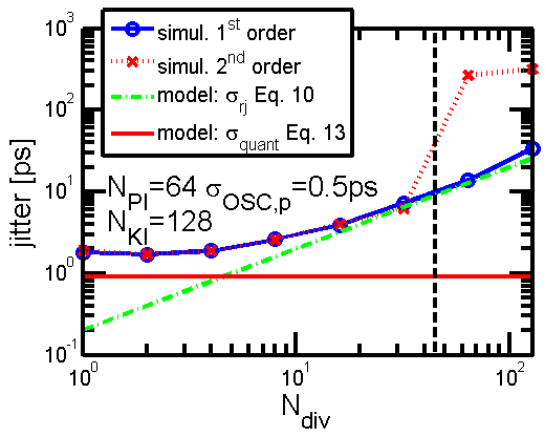


Fig. 13. Absolute jitter of 1st- and 2nd-order CDRs considering free-running oscillators and different division ratios. Symbols correspond to the event-driven model of appendix B, while symbols are the equation of Section II.A for 1st-order CDR. The vertical line displays the value of N_{div} corresponding to the condition $K=4/(TN_{KI})$

accumulator followed by division factor N_{KI} (i.e. the gain of the integral path is $k_I=K/(TN_{KI})$). In this structure, Eq.1 becomes

$$\frac{\phi_{CDR}}{\phi_{CK}} = \frac{(j\omega)^2}{(j\omega)^2+j\omega k_P+k_I} \quad , \tag{26}$$

So that the transfer function for jitter to be inserted into Eq.5 is given by:

$$|H|^2 = \frac{f^4}{f^4+f^2\left(\frac{K^2}{4\pi^2}-\frac{2K}{4\pi^2TN_{KI}}\right)+\left(\frac{K}{4\pi^2TN_{KI}}\right)^2} = \frac{f^4}{(f^2+f_1^2)(f^2+f_2^2)} \tag{27}$$

where the expressions for f_1^2 and f_2^2 are omitted.

By inserting (27) into (5) in place of $(f^2/BW^2)/(1+f^2/BW^2)$ one obtains an integral giving σ_{rj} that can be worked out only assuming that both f_1^2 and f_2^2 are real (i.e. $K>4/(TN_{KI})$). Surprisingly, the integral gives exactly the same result as in (5), that then leads to (10), with no dependence on the I-path gain. In other words, if $K>4/(TN_{KI})$ the 2nd-order CDR has exactly the same jitter as the 1st-order one when working on data clocked by a free-running oscillator. This is verified by the results in Fig.13: the event-driven simulator described in Appendix B gives exactly the same results in terms of jitter as far as $K>4/(TN_{KI})$ (vertical dashed line). In that range, the jitter is also well descried by (10). Making the CDR work above this limit is not a good choice since it corresponds to poles with imaginary part resulting in overshooting.

ACKNOWLEDGMENT

We would like to thank Werner Grollitsch and Luigi Grimaldi of Infineon Technologies Villach as well as Luca Benini of the University of Bologna for support and for many helpful discussions.

REFERENCES

[1] C. Menolfi et al., "A 112Gb/s 2.6pJ/b 8-Tap FFE PAM-4 SST TX in 14nm CMOS", ISSCC, 2018

[2] B. Casper, "Clocking Wireline Systems: An Overview of Wireline Design Techniques, " IEEE Solid-State Circuits Magazine, vol. 7, no.4 pp. 32-41, Nov.2015, 10.1109/MSSC.2015.2476015

[3] A. Amirkhany, "Basics of Clock and Data Recovery Circuits: Exploring High-Speed Serial Links," IEEE Solid-State Circuits Magazine, vol. 12, n.1, pp.25-38, Jan. 2020, 10.1109/MSSC.2019.2939342

[4] R. Kreienkamp, U. Langmann, C. Zimmermann, T. Aoyama, H. Siedhoff, "A 10-Gb/s CMOS Clock and Data Recovery Circuit With an Analog Phase Interpolator," IEEE Journal of Solid-State Circuits, vol. 40, no. 3, pp. 736- 743 Mar. 2005, 10.1109/JSSC.2005.843624

[5] J. D. H. Alexander, "Clock Recovery from Random Binary Signals," Electronics Letters, vol. 11 no. 22, pp.541-542, Oct. 1975, 10.1049/el:19750415

[6] J. L. Sonntag and J. Stonick, "A digital clock and data recovery architecture for multi-gigabit/s binary links," IEEE J. Solid- State Circuits, vol. 41, no. 8, pp. 1867– 1875, 2006, 10.1109/JSSC.2006.875292

[7] J. Liang, A. Sheikholeslami, H. Tamura, Y. Ogata and H. Yamaguchi, "6.7 A 28Gb/s digital CDR with adaptive loop gain for optimum jitter tolerance," 2017 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, 2017, pp. 122-123, 10.1109/ISSCC.2017.7870291

[8] M. H. Perrott, "Fast and accurate behavioral simulation of fractional- N frequency synthesizers and other PLL/DLL circuits," in Proc. 39th Design Autom. Conf., Jun. 2002, pp. 498–503, 10.1109/DAC.2002.1012676

[9] N. Da Dalt, "Markov chains-based derivation of the phase detector gain in bang-bang PLLs," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 11, pp. 1195–1199, Nov. 2006, 10.1109/TCSII.2006.883197

[10] N. Da Dalt, "Linearized Analysis of a Digital Bang-Bang PLL and Its Validity Limits Applied to Jitter Transfer and Jitter Generation", IEEE Trans. Circuits Syst. I: Regular Papers, vol. 55, no. 11, pp. 3663- 3674, Dec. 2008, 10.1109/TCSI.2008.925948

[11] Y. Choi, D. K. Jeong, and W. Kim, "Jitter transfer analysis of tracked oversampling techniques for multigigabit clock-and-data recovery," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 50, no. 11, pp. 775–783, Nov. 2003, 10.1109/TCSII.2003.819070

[12] J. Lee, K. S. Kundert, and B. Razavi, "Analysis and modeling of bang–bang clock and data recovery circuits," IEEE J. Solid-State Circuits, vol. 39, no. 9, pp. 1571–1580, 2004, 10.1109/JSSC.2004.831600

[13] G. Marucci, S. Levantino, P. Maffezzoni, C. Samori, "Analysis and Design of Low-Jitter Digital Bang-Bang Phase-Locked Loops," IEEE Trans. Circuits Syst. I: Regular Papers, vol. 61, no. 1, pp.26-36, Jan. 2014, 10.1109/TCSI.2013.2268514

[14] K. Kundert. Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers. Accessed: Apr. 2020. [Online]. Available: http://www.designers-guide.org

[15] C. Azeredo-Leme, "Clock Jitter Effects on Sampling: A Tutorial", IEEE Circuits and System Magazine, vo.11, no. 3, pp.26-37, Aug. 2011, 10.1109/MCAS.2011.942067

[16] M.-J. Park, J. Kim, "Pseudo-Linear Analysis of Bang-Bang Controlled Timing Circuits," IEEE Trans. Circuits Syst. I: Regular Papers, vol. 60, no. 6, pp. 1381- 1394, Jun. 2013, 10.1109/TCSI.2012.2220502

[17] J. Liang, A. Sheikholeslami, H. Tamura, Y. Ogata and H. Yamaguchi, "Loop Gain Adaptation for Optimum Jitter Tolerance in Digital CDRs," in IEEE Journal of Solid-State Circuits, vol. 53, no. 9, pp. 2696-2708, Sept. 2018, doi: 10.1109/JSSC.2018.2839038.

[18] M. Talegaonkar, R. Inti and P. K. Hanumolu, "Digital clock and data recovery circuit design: Challenges and tradeoffs," 2011 IEEE Custom Integrated Circuits Conference (CICC), San Jose, CA, 2011, pp. 1-8, doi: 10.1109/CICC.2011.6055346.

[19] N. Da Dalt, "A design-oriented study of the nonlinear dynamics of digital bang-bang PLLs", IEEE Trans. Circuits Syst. I: Regular Papers, vol. 52, no. 1, pp. 21–31, Jan. 2005, doi: 10.1109/TCSI.2004.840089

[20] A. Bandiziol, W. Grollitsch, F. Brandonisio, M. Bassi, R. Nonis, and P. Palestri, 'Design of a half-rate receiver for a 10Gbps automotive serial interface with 1-tap-unrolled 4-taps DFE and custom CDR algorithm', in 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, May 2018, pp. 1–5, doi: 10.1109/ISCAS.2018.8351310