Design Metrics for Blind ADC-Based Wireline Receivers

(Invited Paper)

Ali Sheikholeslami[1] and Hirotaka Tamura[2]
[1]Department of Electrical and Computer Engineering, University of Toronto, Canada, [2]Fujitsu Laboratories Limited, Japan

***Abstract*—ADC-based receivers use an ADC in the front end to convert the incoming signal to digital where significant equalization can be done in digital domain. These receivers can be classified as phase-tracking and blind architectures. In the former, the VCO phase is controlled through a feedback loop so as to sample the received data in the middle of the data eye. In the latter, the received signal is sampled with a blind clock, i.e. not in a loop, and the data at the center is obtained by data processing techniques such as data interpolation and extrapolation. This paper compares the two architectures in terms of their design complexity and cost, and derives equations that relate the required ADC resolution to channel loss and to the characteristics of the FFE/DFE that follow the ADC.**

I. Introduction

As the data rates per single channel in chip-to-chip and backplane signaling march towards 100Gb/s, the main question still remains as whether the 100Gb/s receiver will be implemented as a binary receiver or as an ADC-based receiver. As shown in Fig. 1(a), a binary receiver consists of an analog equalizer at the front end followed by a Decision Feedback Equalizer (DFE). The equalized signal is then fed into a clock recovery (CR) unit that together with the slicer in the DFE form the clock and data recovery (CDR) block. This is in contrast with the ADC-based receivers, shown in Fig. 1(b)(c), where the received data is partially equalized in analog domain prior to being digitized by the ADC. The FFE, DFE, and the entire CDR are now implemented fully in digital. It is this capability of providing extensive equalization in digital domain that has given the ADC-based receivers an edge over their binary counterparts. However, this advantage comes at the price of power consumption, at least for now.

There are three main contributors to the power consumption in ADC-based receivers: the ADC, the digital processing, and the clock distribution network. Among the three, the digital processing power, and to some extent the clock distribution power, is expected to decrease over time as we move to smaller geometries. The ADC power, however, is expected to reduce at a lower rate, especially when a flash ADC is used to accommodate the high data rate. Since the ADC power grows exponentially with the ADC resolution, one remedy to the power issue is to devise architectures to minimize the required effective number of bits (ENOB) for the ADC. We will discuss this and other design considerations of the ADC-based CDRs in the second half of this paper. In the first half, we provide a primer on the basics of CDR for both binary and ADC-based receivers.

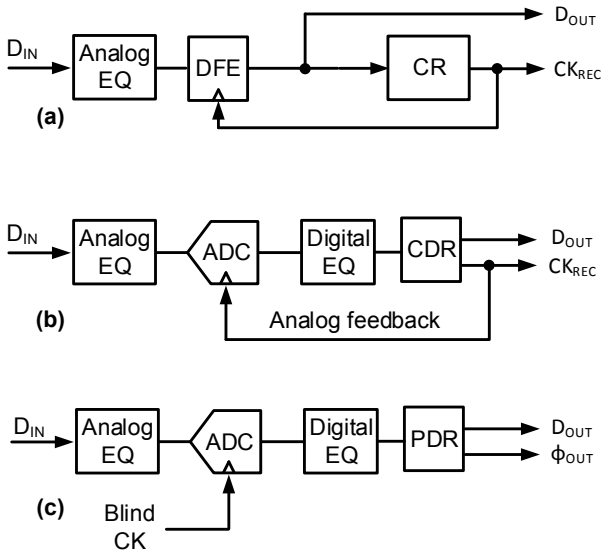

Fig. 1. Basic architectures for (a) binary receiver, (b) phase-tracking ADC-based receiver, and (c) blind ADC-based receiver

II. Clock and Data Recovery Primer

A. Binary CDR

A block diagram for a receiver architecture with the binary CDR is shown in Fig. 2(a). The receiver consists of a front-end analog equalizer, also known as continuous-time linear equalizer or CTLE, followed by a DFE and a CR unit. The analog equalizer partially compensates for the high-frequency attenuation introduced by the channel and hands over this partially-equalized signal to the DFE for further equalization. The equalized signal from the DFE is then sampled by a clean clock, which is provided by the CR unit.

The CR unit receives the equalized signal from the DFE and produces a clean clock. As shown in Fig. 2(b), The CR consists of a phase detector (PD), a charge pump (CP), a loop filter (LF) and a voltage-controlled oscillator (VCO). The PD compares the phase of its input with that of the recovered clock and accordingly produces early/late signals as shown in Fig. 2(c). The early/late signals control the flow of charge from the charge pump (CP) to the loop filter, and accordingly control the voltage of the VCO. Following a data transition (i.e. when $D_n \neq D_{n+1}$), an early signal is produced when the boundary data (B_n) matches D_n. Similarly, a late signal is produced when B_n matches D_{n+1}. When there is no transition, both

978-1-4673-6146-0/13/$31.00 ©2013 IEEE

early and late signals remain inactive. Since the loop gain in this architecture is very high at low frequencies and very low at high-frequencies, the closed-loop system passes the low-frequency jitter of its input to the recovered clock (that is the VCO output) but attenuates the high-frequency jitter. The recovered clock is considered clean in the sense that it contains little high-frequency jitter. Once the clock recovery is complete, assuming a full-rate CDR, one of the falling (or rising) edge of the VCO is expected to align with the data transition and its rising (falling) edge to align with the center of the data. As such, the rising edge is used to sample the equalized signal to form the recovered data.

Alternatively, a CDR may use a phase interpolator (PI), instead of a VCO, to recover the clock. In this case, the PI receives a phase code from the loop in order to interpolate between two phases of a clock signal with a fixed frequency.

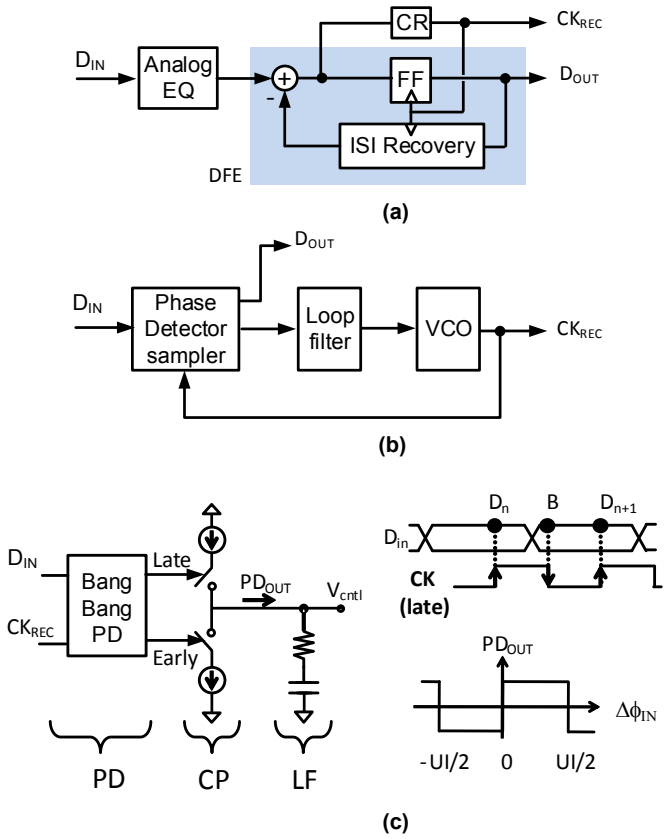

Fig. 2. (a) Binary receiver with CTLE and DFE, (b) bang-bang CDR, (c) bang-bang PD operation and its timing diagram

The binary CDR as described above is adequate for clock and data recovery if the linear equalizer and the DFE could open the eye to a sufficient level for the slicer. If this is not the case, i.e. when the channel is highly attenuative, more sophisticated equalization schemes become necessary, some of which are easy to implement in digital domain following an ADC.

B. Phase-Tracking ADC-Based CDR

As shown in Fig. 1(b), a phase-tracking ADC-based CDR consists of an analog equalizer in the front end followed by an ADC that samples the equalized signal at the center of the eye. These samples are converted to digital where significant signal processing can be applied in digital domain. This architecture is commonly used in applications with high channel loss and dispersion, [1][2].

Including an ADC at the front end and utilizing mostly digital circuits in the backend has several advantages. First, as we mentioned earlier, this architecture allows for more sophisticated signal processing, and hence it could cover a wider range of channel attenuation. Second, the digital circuits can be designed using Verilog and can be easily ported to other technologies. This reduces the design time significantly compared to that of its alternative mostly-analog design. Third, as we move to more advanced technologies, the cost of digital circuits (in terms of area and power) continues to decrease while the cost of analog circuits is expected to remain the same, or at least not to decrease at the same rate. This implies that the cost of ADC-based CDR will continue to become more competitive over time.

Another step towards ease of design and better portability is to eliminate the feedback from the digital domain to the ADC, i.e. to eliminate the feedback that provides the recovered clock to the ADC. We discuss this architecture next.

C. Blind ADC-based CDR

Fig. 3 shows the block diagrams of two architectures for a blind ADC-based CDR: the feed-forward architecture [3] and the DI-based architecture [4]. In both architectures, the sampling clock of the ADC is blind; i.e. it does not use any feedback from the digital backend. Nevertheless, the sampling clock is assumed to have a limited frequency offset, in the order of 100s or 1000s of ppm, with respect to the transmit clock. By using the blind clock, these architectures remove the feedback from the digital to the analog domain and hence greatly simplify the design as the ADC and the digital backend could be designed separately with minimum co-simulation.

The feed-forward architecture, Fig. 3(a), applies the blind clock (or its divided version in an actual implementation) to the digital backend. The digital backend consists of an FFE and DFE in addition to what we have called the Phase and Data Recovery (PDR) unit. We distinguish this block from a CDR as it does not recover *clock*; rather, it recovers the *phase* of the recovered clock in digits. The DI-based architecture (Fig. 3(b)) feeds back the recovered phase to a data interpolation (DI) block so as to reconstruct the received samples at the middle of the eye for ease of equalization by FFE and DFE. We will discuss this and the implications of not recovering the clock in the context of a 2x blind sampling in the next section.

III. Blind ADC-Based CDR Operation

A. 2x sampling

Fig. 4 shows the basic block diagram of a 2x blind ADC based receiver (the analog front end is not shown [5]). For

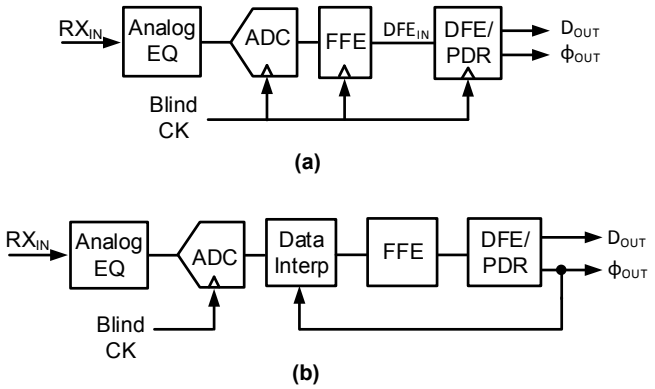

Fig. 3. Two blind ADC-based architectures: (a) feed-forward, (b) DI-based

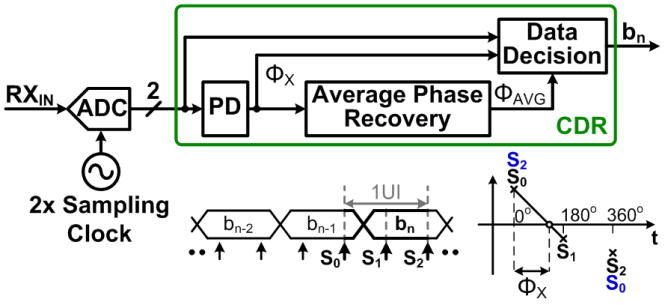

Fig. 4. Basic architecture of a 2x blind ADC-based CDR [3]

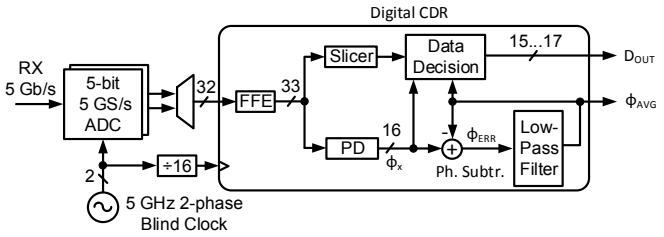

Fig. 5. Detailed implementation of a 2x blind ADC-based CDR [3]

simplicity, we assume the received signal is sampled by a *single* ADC at the rate of 2x the bit rate. That is, the ADC takes two samples for every UI of data. These samples are denoted by S_0, S_1, and S_2 in the example timing diagram. In this example, there is a data transition (zero crossing) between S_0 and S_1. The role of the phase detector is to determine the timing of this zero crossing, denoted by ϕ_x on the diagram. Once we collect a sequence of ϕ_x's corresponding to several transitions, we will average them in digital domain to determine the average location of the transitions, denoted by ϕ_{AVG}. This average phase represents, in digital, the falling edge location of the recovered clock, without a clock signal being present. Given the sequence of samples, ϕ_x, and ϕ_{AVG}, the Data Decision block then recovers the data, denoted by b_n.

In an actual implementation [3] in 65nm CMOS, shown in Fig. 5, two 5-bit 5GS/s ADCs are used to convert the received signal to two data streams of 5GS/s each. These two streams are then demuxed by a factor of 16 to provide 32 data streams of 312.5 MS/s where the digital logic can comfortably operate at 312.5MHz in 65nm CMOS. The details of FFE and its adaptation can be found in [6]. Here, we note that the PD takes 33 samples on each 312.5MHz clock period and produces up to 16 digital values for ϕ_x depending on the transition density of the received signal. The Data Decision block can output either 15, 16, or 17 bits depending on the relative position of ϕ_x and ϕ_{AVG}.

As noted in Fig. 5, the blind ADC-based CDRs do not recover clock; instead, they recover the digital phase corresponding to a recovered clock. The recovered phase is used inside the Data Decision block for data recovery, and to help handle the frequency offset between the transmitter and receiver clocks using a flexible FIFO. We explain this in greater detail in the following.

In a typical application, such as in USB3.0, the recovered clock is used to write data (say 16 bits at a time) into a flexible FIFO shown in Fig. 6(a). A different clock, with possible frequency offset, coming from the core logic, reads the content of the FIFO. The FIFO is designed with enough depth so as to handle transient frequency offsets (both positive and negative) between the two clocks. In our implementation of the blind CDR, we do not generate the recovered clock, so the question arises as how to handle the frequency offset using a FIFO. As shown in Fig. 6(b), we use the core clock for both writing and reading from the FIFO. However, we choose to write either 15, 16, or 17 bits at a time into the FIFO while we consistently read 16 bits from it at the output. This variable length in the input size allows for the effective bit rate at the input port to be different than that at the output, achieving the same effect as using of FIFO with two different clock frequencies but fixed data size.

One of the challenges of the blind ADC-based CDR as described above is the design of a corresponding DFE. Fig. 7 illustrates this challenge by comparing a one-tap DFE implantation in a phase-tracking versus blind architecture. In the former, the samples are taken in the middle of the eye where the ISI contribution from the previous bit is constant in amplitude (assuming only one post-cursor ISI for simplicity). In the latter, however, the ISI amplitude would depend on the phase of the sampling clock with respect to the center of the eye. And since this phase could change over time (because of any frequency offset), no fixed value could be used for the DFE coefficient. One design example [7] to address this challenge is shown in Fig. 8. Here, eight coefficient values are identified (α_1 to α_8) corresponding to eight relative locations of ϕ_{AVG} with respect to the sampling phase. The DFE Coefficient Selector then feeds two coefficients (c_1 and c_2) to the ISI Replica Generator. These coefficients correspond to two samples taken in one UI. The details of adaptive engine for this DFE are discussed in [8].

It is clear at this point that the feed-forward architecture of Fig. 3(a) requires a rather complex DFE architecture. This complexity can be avoided to a large extent by using the DI-based architecture of Fig. 3(b) where the use of data

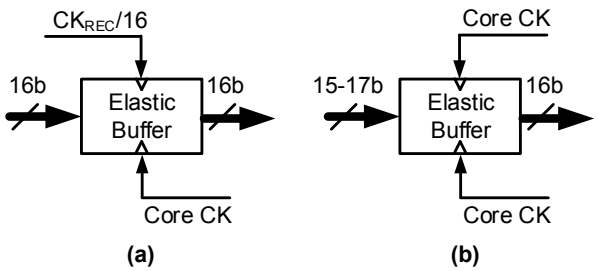

Fig. 6. Handling frequency offset in (a) USB3.0 and (b) 2x blind CDR

interpolation provides access to the data at the center and the edge. Therefore, a simpler, conventional DFE architecture can be used.

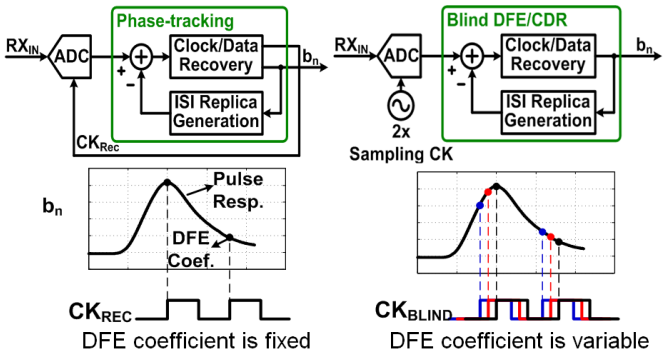

Fig. 7. DFE implementation in (a) phase tracking and (b) blind CDR [7]

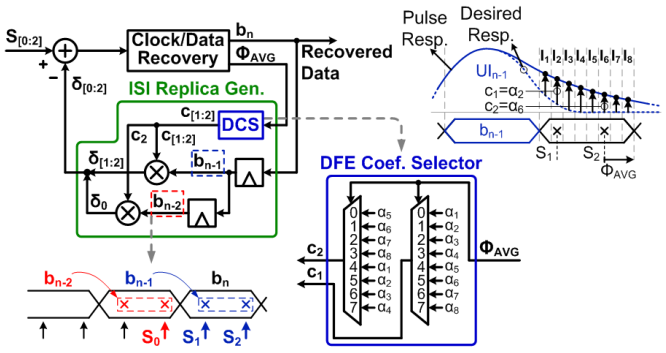

Fig. 8. Details of DFE implementation in 2x blind ADC-based CDR [7]

B. 1.45x sampling

The main purpose of sampling the received signal at twice the baud rate is to estimate with good accuracy the location of its zero crossings. However, if we allow a small degradation in this accuracy, or equivalently in jitter tolerance, it is possible to reduce the sampling rate to below 2x, hence reducing power consumption or increasing the bit rate. In an example implementation in [9], the ADC-based receiver samples the incoming signal at 1.45x the baud rate yet offers a reasonable estimate of the zero crossings as evidenced by its simulation and measurement results.

Fig. 9 shows how conceptually an eye diagram (and hence a zero crossing location) appears as we fold samples taken at 1.45x the baud rate into a one UI interval. This sampling rate corresponds to approximately 0.7UI distance between adjacent samples, or equivalently to a total of 16 samples per 11UI. The block diagram shows a 6.875Gb/s received signal is sampled by four phases of a 2.5GHz blind clock to produce an aggregate 10GS/s for the digital CDR. The design uses a PD and a Data Decision block, which are similar to those used in [3], and a Data Compactor, where 16 samples are translated to nominally 11 bits (corresponding to 11UIs). Finally, a FIFO, with its diagram in Fig. 10, takes 10-12 bit words at its input and spits out 16-bit words using the retiming clock. Note that this FIFO absorbs the frequency offset between the blind clock and the clock embedded in the data similar to what we described in connection with Fig. 6.

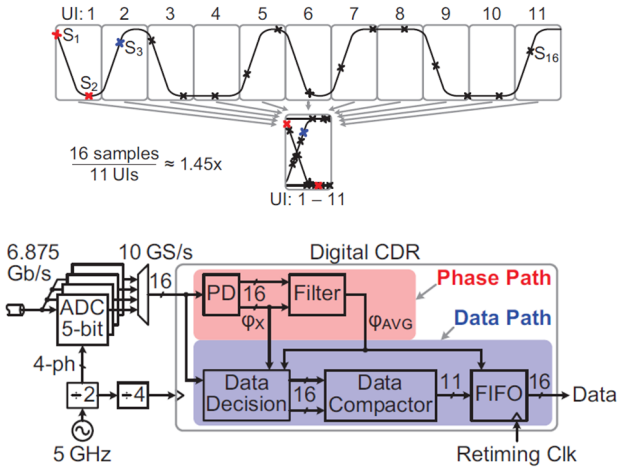

Fig. 9. 1.45x blind ADC-based CDR: (a) identifying the zero crossing from the samples, (b) implementation [9]

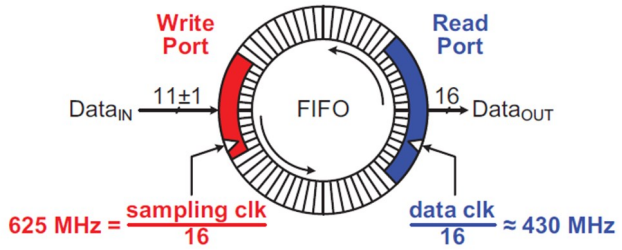

Fig. 10. FIFO implementation in 1.45x blind ADC-based CDR [9]

C. 1x sampling

Baud-rate sampling allows for data and timing recovery in an architecture known as Mueller-Muller [10] if feedback is provided directly to the ADC clock [11]. However, when the ADC clock is blind, it is possible that the received signal is sampled directly on its zero crossings, and the data is lost completely (since there is only one sample per UI). Yet, by

using an integrate and dump filter [5] in the analog domain prior to the ADC, Ting et al. [4] introduce intentional ISI in the data stream such that each bit is spilled over the adjacent bits. As a result, the data is contained not only in the center of the eye but also at zero crossings, and this makes 1x blind baud-rate CDR possible.

The block diagram of a 1x blind baud-rate CDR is shown in Fig. 11. Here the received signal is 10Gb/s and is sampled at 10GS/s using four time-interleaved 2.5GS/s ADCs. The ADCs are preceded by 1UI integrate and dump filters in analog and followed by a digital summer to form a 2UI I&D samples. These samples are then interpolated using the ϕ_{AVG} provided by the digital loop filter in the backend. A Mueller-Muller PD is used to estimate the phase, and a speculative 2-tap DFE is used to recover the bits. The measurement results provided in [4] confirm phase and data recovery operation with a high-frequency jitter tolerance of about 0.2UIpp even in the face of a 1000ppm of frequency offset, i.e. when the blind clock at the receiver samples the incoming signal at below the baud rate.

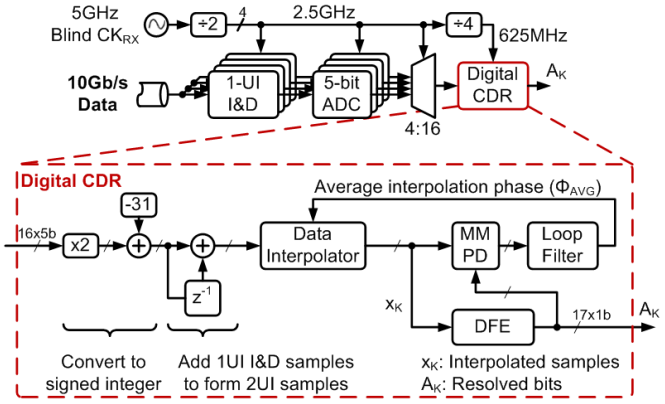

Fig. 11. Block diagram of a 1x blind ADC-based receiver [4]

IV. Binary versus ADC-based: Are they really different?

Fig. 12(a) shows a simplified block diagram of a binary CDR with 3-bit speculative DFE. The received signal is compared against 8 reference levels, corresponding to the ISI associated with three consecutive bits. The output of one of the comparators is selected as the current bit based on the three previously recovered bits. Note that the reference levels in this architecture are not equidistant since the 1st, 2nd, and 3rd post-cursor ISI's are subject to the channel impulse response. However, once the previous bits are recovered, they are used directly to control the select operation, i.e. to select the current bit among 8 candidates.

This architecture is similar to an ADC-based receiver followed by a non-speculative DFE, as shown in Fig. 12(b). Here, the input is compared against 7 reference levels, but the reference levels are distributed uniformly in the input range as it is typical in a flash ADC. However, instead of only passing one bit to the next stage, all the thermometer

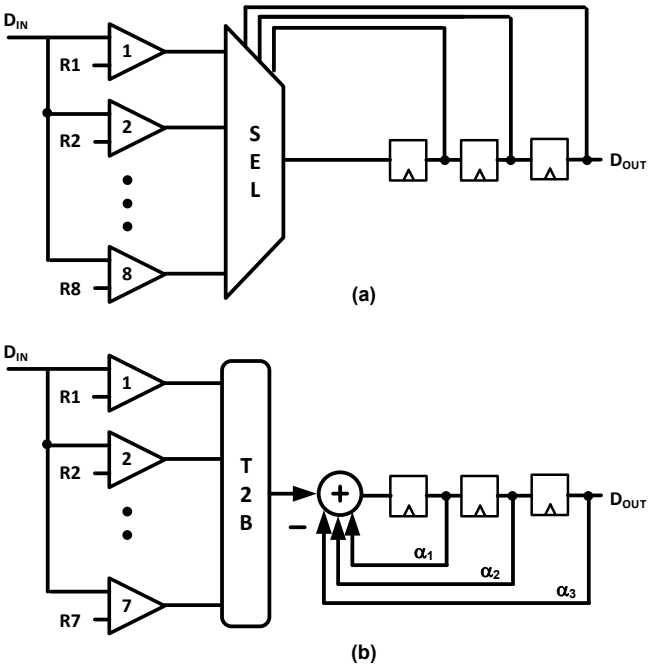

Fig. 12. (a) Speculative DFE in binary receivers, (b) ADC plus non-speculative DFE

bits (corresponding to the full digital output of a 3-bit ADC) are fed to a non-speculative DFE. Note that even though the number of bits produced by the comparators is about the same in both cases, they carry different information because the bits correspond to different reference levels. In the former, the reference levels incorporate the ISI information; and hence the previous bits are used directly to pick one bit among 8. In the latter, the bits carry no information of the past ISI (as the reference levels have nothing to do with the ISI values). To include the ISI information, the ADC-based architecture multiplies the previous bits by the ISI values $(\alpha_1, \alpha_2, \alpha_3)$ and subtracts these digital values from the digital output of the ADC.

Given the comparison above, it is natural to suggest that perhaps the ADC-based architecture would be identical to that of a speculative DFE if the ADC reference levels are chosen non-uniformly, to be consistent with the speculative DFE case. This is indeed true but with one important caveat. The ADC based CDR (with equidistant reference levels) allows for digital FFE (i.e. FFE in digital domain) prior to DFE. This is not true for the speculative DFE because the input waveform shape is lost once we pass the set of comparators at the front end. Nevertheless, it is possible to think of novel architectures where the benefits of the two architectures can be combined [12].

The similarity between the two architectures will vanish to a large extent once we assume the sampling clock is blind. In this case, the binary CDR with speculative DFE, in its current architecture, will not be functional, as the ISI information needs updating depending on the sampling phase. The ADC-based CDR, on the other hand, functions well as it preserves

the samples' magnitudes (except for the quantization error) of the incoming waveform for processing in the digital domain.

V. Design Considerations of ADC-Based CDRs

Fig. 13 shows the block diagram of a blind ADC-based CDR using data interpolation. An ADC digitizes the input data at a sufficient sampling rate. To compensate for the frequency-dependent signal loss in the channel, feed-forward equalizers (FFEs) are inserted in the signal path. The FFE can be analog, placed before the ADC, or digital, after the ADC, or both. Data at the decision timing (i.e., at the eye center) are then generated in the receiver by using interpolation in digital domain (Fig. 14). The interpolation ratio is obtained in the same manner as that in conventional PI-based receivers. A decision circuit with a decision feedback equalizer (DFE) performs a binary decision of the retimed data.

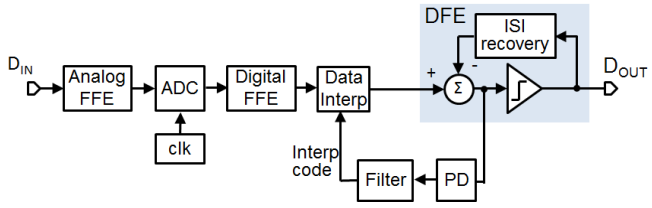

Fig. 13. Blind ADC-based CDR using data interpolation

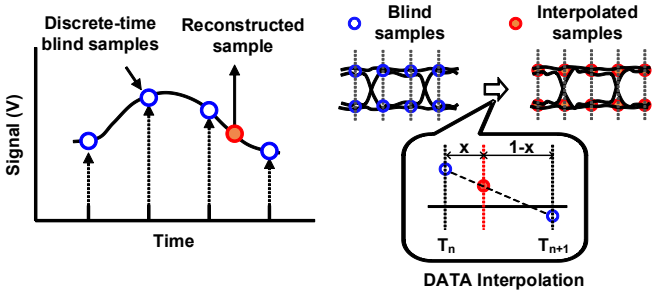

Fig. 14. Reconstructing data by interpolation

The finite resolution of the ADC affects the CDR performance through two mechanisms. One is that the ADC quantization error propagates to an amplitude error in the eye-center data generated by the data interpolation. This amplitude error reduces the vertical eye opening. The other is that the ADC quantization error generates a phase error in the phase detection performed in the digital signal processing. The error in the phase detection results in CDRs phase tracking error, further decreasing the eye opening.

Since the digital FFE combined with the data interpolation is a linear operation on the input data, transfer from the ADC peak-to-peak quantization error δV_{PP} to the retimed-data peak-to-peak amplitude error δA_{PP} can be expressed as

$$\delta A_{PP} = G_e \delta V_{PP} \tag{1}$$

where G_e is the quantization-error amplification factor, and $\delta V_{PP} = LSB$. When a digital FFE is used, it amplifies the ADC quantization noise, resulting in G_e greater than 1. The value of G_e is calculated in Appendix A. Assuming the channel attenuation in dB is proportional to frequency, we can write,

$$G_{ch}(f) = exp[-(ln10/10)L_0(f/f_b)] \tag{2}$$

Here L_0 is the channel loss in dB at half the baud-rate frequency, f_b. A digital FFE equalizes this channel transfer function to that of Gaussian-filter characteristics, $G_{tot}(f)$,

$$G_{tot}(f) = exp[-(ln2/2)(f/f_{3dB})^2] \tag{3}$$

Here f_{3dB} represents the 3-dB cut-off frequency of the equalized channel. Fig. 15 shows plots of the frequency responses for the channel ($G_{ch}(f)$), equalizer ($G_{eq}(f)$), and the equalized channel ($G_{tot}(f)$) for four values of L_0. The value of G_e increases by increasing the f_{3dB} of the equalized channel.

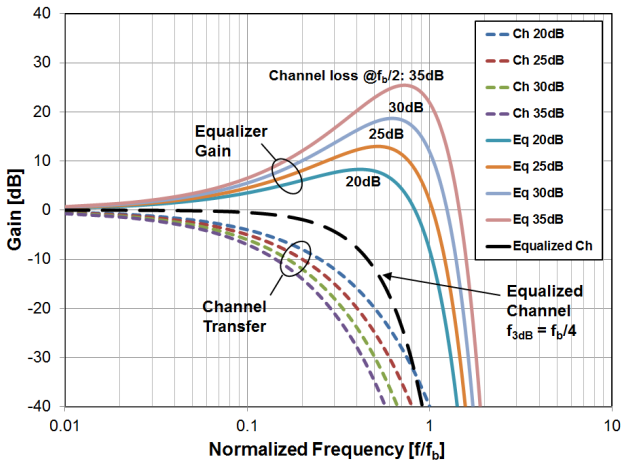

Fig. 15. Channel, equalizer, and the equalized channel frequency response

The phase error comes from the dead zone in the phase detection. For example, 2x bang-bang phase detection produces 0.5UIpp of a dead zone if a blind clock signal is used. Since both the non-linearity of quantization and the bandwidth limitation of the data come into play, it is hard to express the error in a compact analytical form for more general multi-bit cases. If, however, the waveform is adequately bandwidth limited and can be approximated as a straight line between two adjacent samples, the dead zone for the first-order interpolation is easily calculated. The dead zone is given by the overlapping of the two dead zones from the first and second samples (Fig. 16), and the timing error corresponds to the maximum of the two dead zones. As a result, the timing error becomes:

$$\delta\theta/\Delta T = Min[1, Max[1 - \frac{\lfloor\lfloor x \rfloor\rfloor}{\lfloor x \rfloor}, \frac{\lfloor\lfloor x \rfloor\rfloor + 1}{\lfloor x \rfloor} - 1]] \tag{4}$$

where ΔT is the sampling interval, dV/dt is the slope of the input waveform, x is the normalized slope ($= (dV/dt)/(LSB/\Delta T)$) and $\delta\theta$ is the dead zone error in terms of timing. The value of $\delta\theta/\Delta T$ oscillates between $1/x$ and

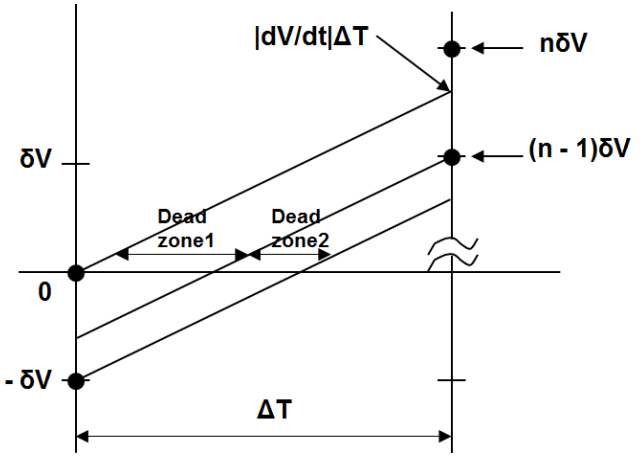

Fig. 16. Dead zones in phase detection

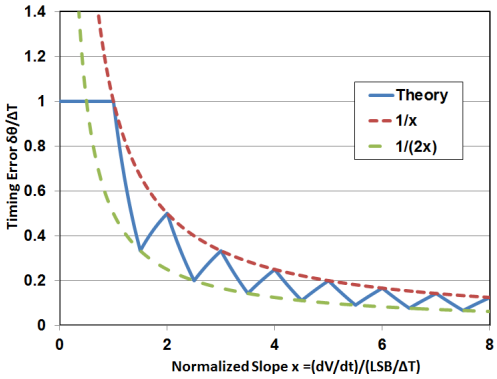

Fig. 17. Timing error as a function of normalized slope

$1/(2x)$ as shown in Fig. 17, and the worst-case timing error becomes

$$\delta\theta = Min[\Delta T, LSB/\mid dV/dt \mid] \tag{5}$$

Note that (5) is achieved by converting the ADC peak-to-peak quantization error, LSB, into the timing error using the slope $\mid dV/dt \mid$ as a timing-to-voltage conversion factor.

If the input waveform is well equalized by an analog FFE, the value of $\mid dV/dt \mid$ is on the order of V_{pp}/UI, where V_{pp} is the peak-to-peak value of the input signal and UI is the unit interval. We assume that the amplitude V_{pp} is adjusted by an automatic gain control such that $LSB \sim V_{PP}2^{-N}$, where N is the ADC number of bits. Then a rule of thumb expression is obtained from (5) as $\delta\theta \sim 2^{-N}$ [UI].

In theory it is possible to reduce the CDR tracking error to be less than the phase-detection dead zone given by (4) if means such as dithering is implemented. For example, adding a known amount of phase modulation would remove the dead zone error if the amount of the added phase exceeds the dead-zone width. The ADCs maximum quantization noise δV_{max} that a blind data-interpolated ADC-based CDR can tolerate is calculated from an eye diagram drawn under a given bit-error rate (e.g. 10^{-12}) criterion (Fig. 18). The condition for a correct signal reception in the CDR is given as

$$S_{eff} - \delta A_{ADC} - (\delta S/\delta t)\delta\theta > 0 \tag{6}$$

where S_{eff} is the effective signal strength that is given by the eye opening at the center of the eye, and $(\delta S/\delta t)$ is the slope of the eye opening at the eye center. These terms are calculated by the methods described in Appendix A and B, and the value of δV_{max} is calculated from (6).

Increasing the 3-dB bandwidth of the equalized signal by using a digital FFE improves the effective signal amplitude but it enhances the quantization noise at the equalizer output. Due to the noise enhancement, the range of the 3-dB bandwidth of the equalized channel where the correct signal reception is possible becomes narrower when the ADC quantization noise δA is increased. At a certain maximum value δA_{max}, the width of the correct operation range shrinks to zero (Fig. 19). We define the minimum required number of bits of the ADC as

$$\delta A_{max} = V_{pp}2^{-N_{min}} \tag{7}$$

The minimum required number of bit N_{min} increases with the channel loss L_0 (Fig. 20). The calculated curves matches the results obtained from behavioral simulations within ± 1 bit (Fig. 21).

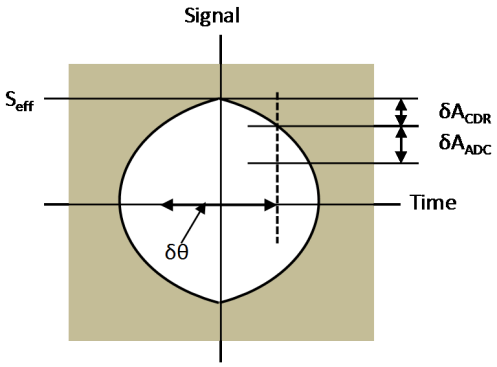

Fig. 18. Phase error and quantization error reduce the signal sense margin

ACKNOWLEDGMENT

The authors would like to acknowledge Masaya Kibune, Yanfei Chen, Mohammad Sadegh Jalali, and Clifford Ting for their contributions to this paper.

APPENDIX A

By using (2) and (3), the equalizer transfer function is given by

$$G_{eq} = \frac{G_{tot}}{G_{ch}} = exp[-\frac{ln2}{2}(\frac{f}{f_{3dB}})^2 + \frac{ln10}{10}L_0(\frac{f}{f_b})] \tag{8}$$

The ADC quantization noise is treated as a white noise with the rms value given by $< \delta V^2 >= LSB^2/12$ and the bandwidth up to $f_b/2$. The rms value of the output of the equalizer $< A_{ADC}2 >$ is calculated as

$$< (\delta A_{ADC})^2 >= \int_0^{f_b/2} \mid G_{eq}(f) \mid^2 < \delta V^2 > df \tag{9}$$

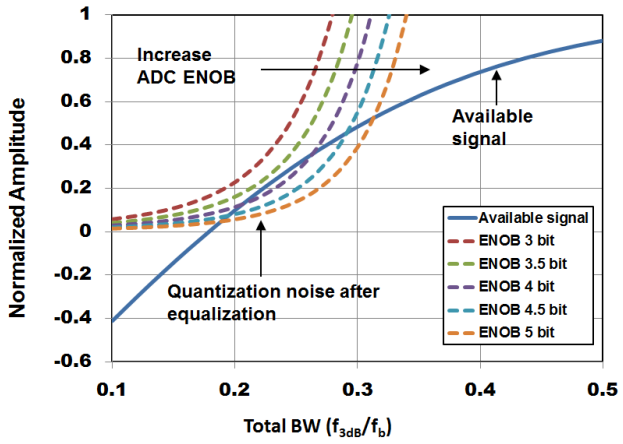

Fig. 19. Noise and signal versus equalized channel bandwidth

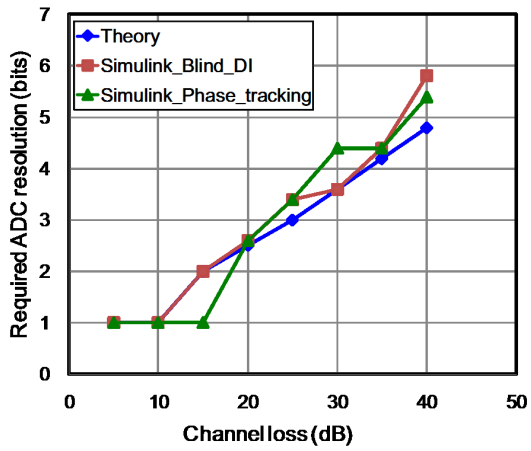

Fig. 21. Minimum ADC number of bits versus channel loss

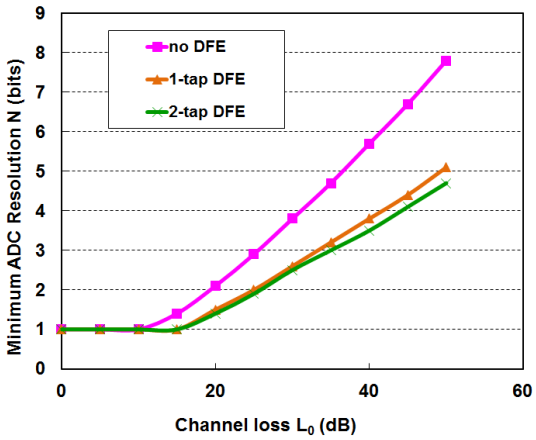

Fig. 20. Minimum ADC number of bits versus channel loss

Assuming the ADC quantization noise has a uniform distribution, this rms value can be multiplied by $2\sqrt{3}$ to provide the peak-to-peak value.

APPENDIX B

For a given channel characteristics, let the unit pulse response be written as $h_k = h(kT)$, $(k \in Z)$, T is the unit interval, and $h(0) = h_0$ is the main cursor where the unit pulse response has a peak. For Gaussian channel, the values of $h(kT)$ is non-negative and the maximum ISI is given as the sum of all post-cursor and pre-cursor taps. Thus we have

$$ISI_{max} = \sum_{i \neq 0} h_i = 1 - h_0 \tag{10}$$

$$S_{eff} = h_0 - ISI_{max} = 2h_0 - 1 \tag{11}$$

We assume that an ideal n-tap DFE makes the post cursor taps zero up to h_n. This decreases the ISImax by $\sum_{i=1}^{n} h_i$, producing

$$S_{eff} = 2h_0 + \sum_{i=1}^{n} h_i - 1 \tag{12}$$

The unit pulse response h(t) is calculated from the Gaussian filter step response s(t).

$$S(t) = 1 - 0.5erfc(\pi f_{3dB}(2/ln2)^{1/2}t) \tag{13}$$

REFERENCES

[1] D. Crivelli, et al., "A 40nm CMOS Single-Chip 50Gb/s DP-QPSK/BPSK Transceiver with Electronic Dispersion Compensation for Coherent Optical Channels," *ISSCC*, Dig. of Tech. Papers, pp. 338-329, Feb. 2012.

[2] J. Cao, et al., "A 40nm CMOS Single-Chip 50Gb/s DP-QPSK/BPSK Transceiver with Electronic Dispersion Compensation for Coherent Optical Channels," *IEEE J. of Solid-State Circuits*, Vol. 45, No. 6, pp. 1172-1185, June 2010.

[3] O. Tyshchenko, et al., "A 5Gb/s ADC-Based Feed-Forward CDR in 65nm CMOS," *IEEE J. of Solid-State Circuits*, Vol. 45, No. 6, pp. 1091-1098, June 2010.

[4] C. Ting, et al., "A Blind Baud-Rate ADC-Based CDR," *ISSCC*, Dig. of Tech. Papers, pp. 122-123, Feb. 2013.

[5] T. Tahmoureszadeh, et al., "A Combined Anti-Aliasing Filter and 2-tap FFE in 65-nm CMOS for 2x Blind 2-10 Gb/s ADC-Based Receivers," *IEEE Custom Integrated Circuits Conference*, pp. 1-4, Sep. 2010.

[6] H. Yamaguchi, et al., "A 5-Gb/s Transceiver with an ADC-Based Feed-Forward CDR and CMA Adaptive Equalizer in 65-nm CMOS," *ISSCC*, Dig. of Tech. Papers, pp. 168-169, Feb. 2010.

[7] S. Sarvari, et al., "A 5Gb/s Speculative DFE for 2x Blind ADC-based Receivers in 65-nm CMOS," *IEEE Symposium on VLSI Circuits*, Dig. of Tech. Papers, pp. 69-70, June 2010.

[8] B. Abiri, et al., "An Adaptation Engine for a 2x Blind ADC-Basead CDR in 65 nm CMOS ," *IEEE J. of Solid-State Circuits*, Vol. 46, No. 12, pp. 3140-3149, Dec. 2011.

[9] O. Tyshchenko, et al., "A Fractional-Sampling-Rate ADC-Based CDR with Feed-Forward Architecture in 65nm CMOS," *ISSCC*, Dig. of Tech. Papers, pp. 166-167, Feb. 2010.

[10] K. Mueller and M. Muller, "Timing Recovery in Digital Synchronous Data Receivers," *IEEE Trans. on Communications*, pp. 516-531, May 1976.

[11] M. Harwood, et al., et al., "A 12.5Gb/s SerDes in 65nm CMOS Using a Baud-Rate ADC with Digital Receiver Equalization and Clock Recovery," *ISSCC*, Dig. of Tech. Papers, pp. 436-437, Feb. 2007.

[12] J. Kim, et al., "Equalizer design and performance trade-offs in ADC-based serial links", *Custom Integrated Circuits Conference*, pp. 1-8, Sept. 2010.