8.6 A Fractional-Sampling-Rate ADC-Based CDR with Feedforward Architecture in 65nm CMOS

Oleksiy Tyshchenko[1], Ali Sheikholeslami[1], Hirotaka Tamura[2], Yasumoto Tomita[2], Hisakatsu Yamaguchi[2], Masaya Kibune[2], Takuji Yamamoto[2]

[1]University of Toronto, Toronto, Canada
[2]Fujitsu Laboratories, Kawasaki, Japan

ADC-based CDRs take digital samples of the received signal to recover the clock and data. Digital representation of the signal allows for extensive channel equalization in the digital domain. Recently-reported ADC-based CDRs sample the signal at 1× or 2× the baud rate. The 1× CDR aligns the sampling clock with the signal using a phase-tracking feedback loop [1-2], which requires a voltage-controlled oscillator or phase interpolator, both analog circuits, to adjust the phase of the sampling clock. To eliminate these analog circuits (and their phase control) in favor of an all-digital implementation, a blind-sampling ADC-based CDR (top of Fig. 8.6.1) samples the received signal at 2× without phase locking to the signal. The CDR then interpolates between the blind samples to obtain a new set of samples in order to recover the phase and data [3-4]. The doubling of the sampling rate, however, increases the ADC power consumption or, equivalently, reduces the maximum baud rate due to the conversion-rate limitations of ADCs.

This paper presents a new fractional-sampling-rate (FSR) CDR architecture, shown in Fig. 8.6.1, that samples the received signal blindly at a fractional rate of 1.45×, hence reducing the ADC power per Gb/s of data rate by 27.3% compared to the 2× architecture. This architecture uses a digital phase detector (PD) that estimates the data phase directly from the blind digital samples, thus eliminating the need for interpolation. This PD enables data recovery in a feed-forward path, further simplifying the CDR architecture. Measurements of a test-chip fabricated in 65nm CMOS confirm that the FSR CDR successfully recovers data with BER<10^{-13} at 6.875Gb/s from samples taken at 10GS/s.

A block-diagram of the CDR architecture is shown in Fig. 8.6.2. We blindly sample a 6.875Gb/s signal with four time-interleaved 2.5GS/s 5-bit flash ADCs for a total sampling rate of 10GS/s, corresponding to 1.45 samples per unit interval (UI). This sampling rate makes the sampling interval (SI) equal to 11/16 UIs, which causes the sampling instances to span the full duration of a UI. A 4:16 DeMUX then feeds 16 samples at a time, corresponding to 11 UIs, to the digital CDR. The PD estimates the instantaneous zero-crossing phase, $\varphi_X[1:16]$, for every UI, using a scheme we describe later. We use φ_X to recover the average zero-crossing phase, φ_{AVG}, in two steps. First, the phase subtractor generates the phase error, φ_{ERR}, with a modulo-subtraction of φ_{AVG} from φ_X, bounding φ_{ERR} within [-0.5; 0.5) UI. Then, φ_{ERR} is fed into a third-order low-pass filter to recover φ_{AVG}. The filter consists of three discrete-time integrators with programmable gains, K_1, K_2, and K_3, that control the CDR's jitter-tracking bandwidth. The data decision block picks one sliced sample per UI as the recovered data by comparing $\varphi_X[n]$ and φ_{AVG}, and marks duplicate samples, present in some UIs due to the FSR, as invalid samples. We remove these invalid samples from the data-decision vector, $\hat{S}[1:16]$, with a vector compactor (described later), which outputs 11 data bits, D[1:11]. For measurement purposes, we retime the recovered data from the blind-sampling clock domain to the baud-rate clock domain, $f_B/16$, using a FIFO.

The PD, shown in Fig. 8.6.3, consists of an average-transition-slope calculator and a data-phase calculator. From the 16 samples at its input, the PD linearly estimates φ_X for every pair of adjacent samples with opposite polarities. This linear phase estimation proves sufficient when there is enough ISI in the channel. Otherwise, an anti-aliasing filter needs to precede the sampling ADC. The PD uses the transition slope between the samples to estimate the phase. As shown in Fig. 8.6.3, due to the FSR some slopes lead to small errors in φ_X, while others lead to larger errors. To calculate a running average of slopes, we select only those slopes that lead to low φ_X error. When two transitions occur around one sample (top waveform in Fig. 8.6.3) only the transition with the higher slope contributes to the average (S[n] to S[n+1]). When such comparison is impossible (bottom waveform in Fig. 8.6.3), a slope contributes to the average only if both its samples exceed a threshold level, V_{TH}, that is extracted from sample magnitudes. Since the time between adjacent samples is constant, finding the slope simplifies to a sum of sample magnitudes; the slope calculator thus outputs $(|S[n]|+|S[n+1]|)_{AVG}$. The phase calculator, shown in Fig. 8.6.3, estimates the zero-crossing time, $\varphi_{ZC}[n]$ (in units of SI), as the ratio $|S[n]|/(|S[n]|+|S[n+1]|)$. To maintain low circuit complexity, the accuracy of $\varphi_{ZC}[n]$ is limited to 2 bits. For transitions with low-error slopes, we use an instantaneous sum in the 2-bit $\varphi_{ZC}[n]$ calculation, while for transitions with high-error slopes we use the average sum. We then convert $\varphi_{ZC}[n]$ from SI to UI using $\varphi_X[n]=TS[n]+SI\cdot\varphi_{ZC}[n]$, where TS[n] is the time stamp – the sample's position in UI. Our choice of sampling rate causes TS[n] to repeat every 16 samples. Since $\varphi_{ZC}[n]$ is only 2-bit accurate, we convert $\varphi_{ZC}[n]$ to $\varphi_X[n]$ using a selector with constant inputs, as shown in Fig. 8.6.3.

The data decision block, shown in Fig. 8.6.2, picks one sliced sample per UI by comparing $\varphi_X[n]$, φ_{AVG}, and TS[n]. This block also marks the duplicate samples by setting their valid flags (VF[n]) to '0'. To remove these duplicates, whose positions are unknown *a priori*, we use the vector compactor presented in Fig. 8.6.4. It accepts 16 sliced samples, $\hat{S}[n]$, with their VF[n] and produces 11 data bits, D[k], such that every UI corresponds to a single data bit. The compactor consists of an array of conditional data selectors, which pass data bits either from the left or from top to bottom according to the state of the enable signals. The rows with VF[n]='0' (shaded) pass the data from top to bottom. As a result, the output data vector is free of duplicate samples. To reduce area and power, we eliminate the cells that only pass data from top to bottom. With this, the compactor reduces to only 33 cells instead of a full 176-cell array, resulting in a single-cycle compaction.

To experimentally verify the ADC functionality at the FSR, we sampled a 6.875Gb/s 2^7–1 PRBS signal at 10GS/s and captured the DeMUXed ADC samples. We then assigned constant TS[n] to every channel of the DeMUX according to the channel number, n. Finally, we arranged the samples along the time axis in the ascending order of their TS[n]. Figure 8.6.5 presents the resulting measured eye diagram reconstructed from 516800 ADC samples. For every vertical slice of the eye diagram, we annotate the DeMUX channel number and the corresponding time stamp. An open eye at the ADC output confirms that the ADC is functional and that error-free data recovery is possible in the FSR CDR.

Figure 8.6.6 presents the simulated and measured jitter tolerances of the FSR CDR. Both simulations and measurements were performed with a 6.875Gb/s 2^7–1 PRBS input and a sampling rate of 10GS/s. We used an event-driven model [5] in Simulink to simulate the CDR. Our simulations show that the CDR tolerates up to $0.5UI_{PP}$ of sinusoidal jitter at high frequencies (simulated for 2×10^5 UIs, no random jitter at TX and RX). To validate our simulated results, we fabricated and characterized the FSR CDR in 65nm CMOS. The inset in Fig. 8.6.6 presents a measured eye diagram at the receiver input. In addition to the 16ps ($0.11UI_{PP}$) of jitter already present because of the setup, we applied sinusoidal jitter from 50kHz to 8MHz (the range was limited by the available equipment) to measure the jitter tolerance of the CDR. We generated the receiver input with a Centellax PRBS board and recorded the jitter tolerance at BER=10^{-12}. Our measurements closely match the simulation results and confirm that the FSR CDR tolerates $0.3UI_{PP}$ of high-frequency sinusoidal jitter. The CDR tolerates up to 49MHz (0.98%) of frequency offset (with BER≤10^{-12}) between the transmitter and receiver beyond the nominal offset due to the FSR.

Figure 8.6.7 shows a die photo of the test-chip. The ADC, DeMUX and clock divider are analog custom-designed blocks while the CDR and test-structures were synthesized. The CDR consists of 75644 gates and consumes 58.8mW while the ADC consumes 116.4mW. The FSR CDR reduces the ADC power by 27.3%, in comparison with a 2× feed-forward architecture, at the cost of doubling the gate count; however, the power per Gb/s of data rate and the total receiver area are reduced by 12.5%. The receiver occupies 0.3683mm^2.

Acknowledgment:
We thank Chihiro Sannomiya for assistance with test-chip design and verification.

References:

[1] O. Agazzi *et al.*, "A 90nm CMOS DSP MLSD Transceiver with Integrated AFE for Electronic Dispersion Compensation of Multi-mode Optical Fibers at 10Gb/s," *ISSCC Dig. of Tech. Papers*, pp. 232-233, Feb. 2008.

[2] M. Harwood *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.

[3] F.M. Gardner, "Interpolation in digital modems – Part I: Fundamentals," *IEEE Tran. on Communications*, Vol. 41, Issue 3, pp. 501-507, Mar. 1993.

[4] M. Spurbeck *et al.*, "Interpolated timing recovery for hard disk drive read channels," *IEEE Intern. Conf. on Communications*, Vol. 3, pp. 1618-1624, Jun. 1997.

[5] M. van Ierssel *et al.*, "Event-Driven Modeling of CDR Jitter Induced by Power-Supply Noise, Finite Decision-Circuit Bandwidth, and Channel ISI," *IEEE Tran. on Circuits and Systems I*, Vol. 55, Issue 5, pp. 1306-1315, Jun. 2008.

 978-1-4244-6034-2/10/$26.00 ©2010 IEEE

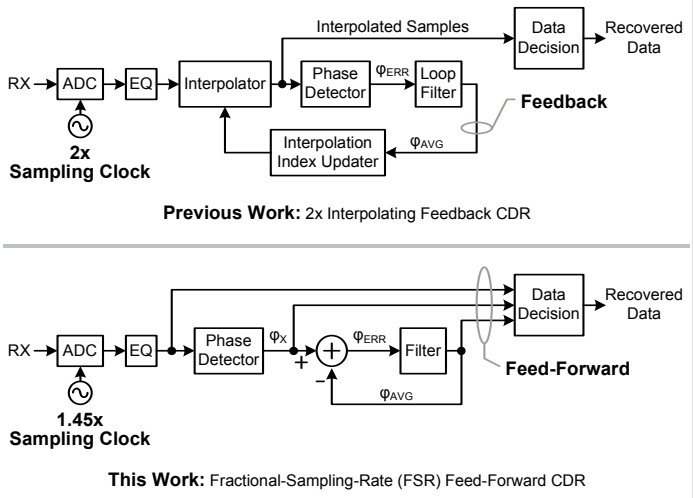


Figure 8.6.1: Blind-Sampling ADC-Based CDR Architectures.

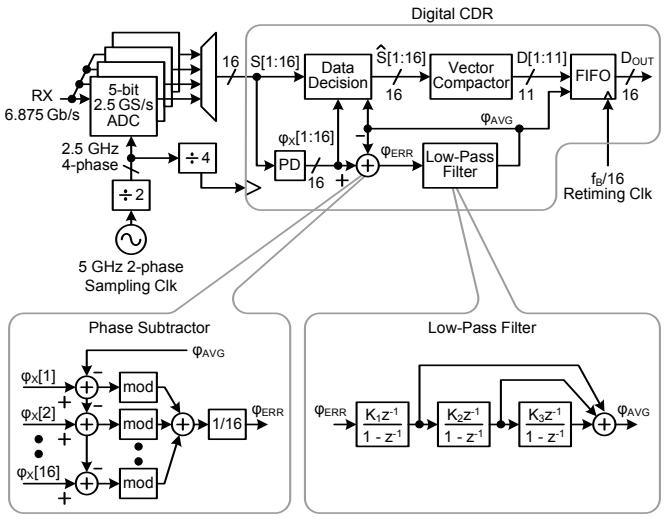


Figure 8.6.2: Fractional-Sampling-Rate (FSR) CDR Block Diagram.

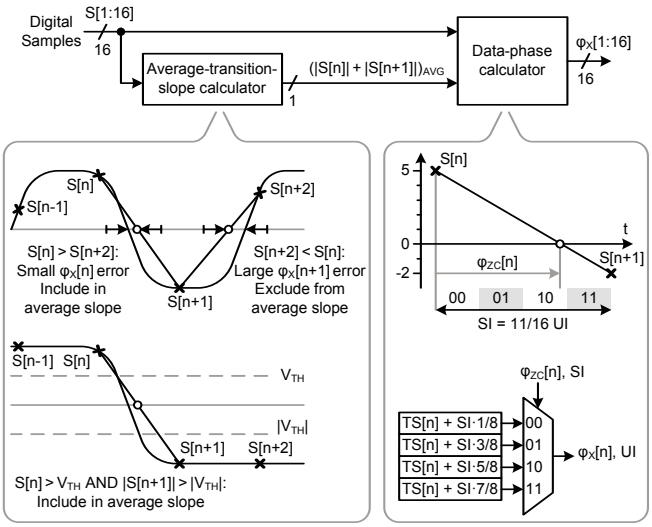


Figure 8.6.3: Phase-Detector Implementation.

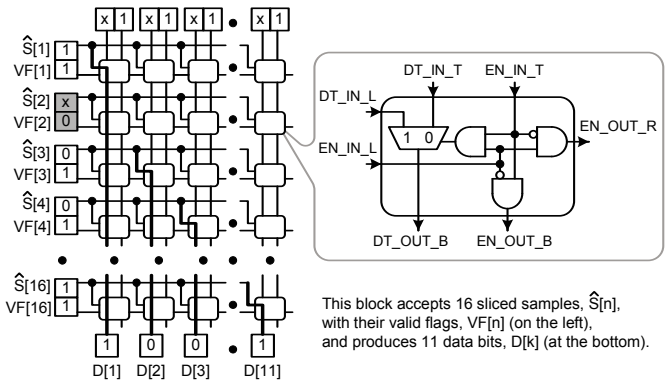


Figure 8.6.4: Vector-Compactor Implementation.

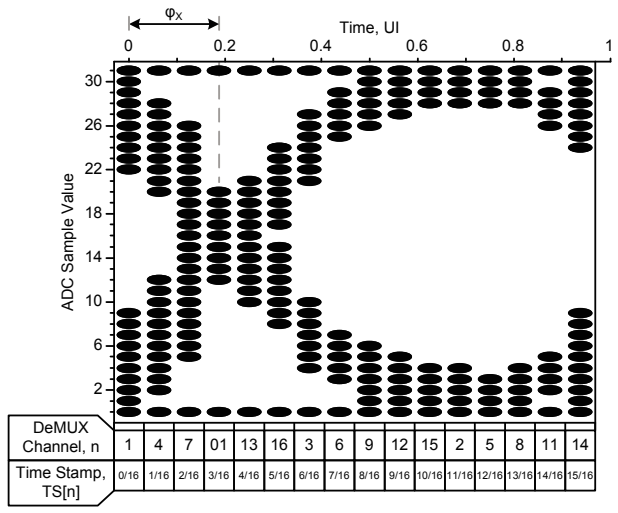


Figure 8.6.5: Measured Eye Diagram at ADC Output (reconstructed from 5x10^5 ADC samples).

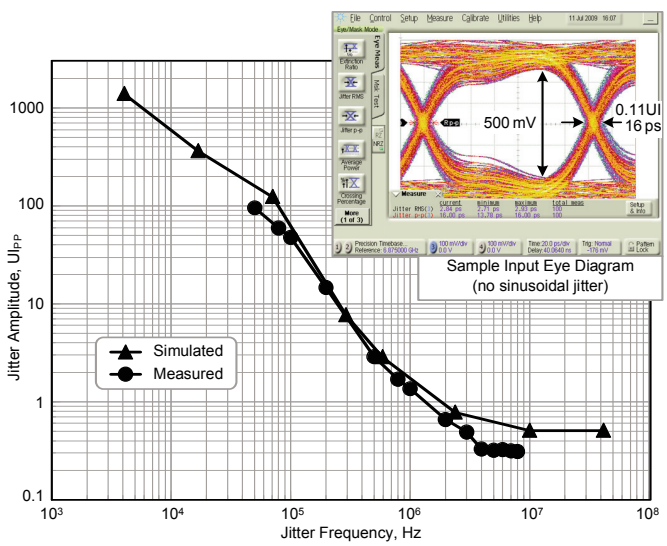
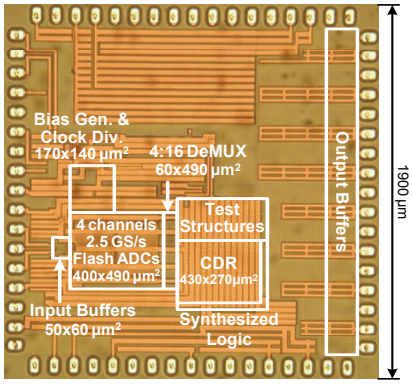


Figure 8.6.6: Measured and Simulated Jitter Tolerance.

8



Process	65 nm CMOS
Data Rate	6.875 Gb/s
Supply	1.2 V
ADC Power	116.4 mW
Digital Power	58.8 mW
Receiver Area	0.3683 mm²

Figure 8.6.7: Die Photograph.

978-1-4244-6034-2/10/$26.00 ©2010 IEEE