

An 8×5 Gb/s Parallel Receiver With Collaborative Timing Recovery

Ankur Agrawal, *Student Member, IEEE*, Andrew Liu, Pavan Kumar Hanumolu, *Member, IEEE*, and Gu-Yeon Wei, *Member, IEEE*

Abstract—This paper presents the design of an 8 channel, 5 Gb/s per channel parallel receiver with collaborative timing recovery and no forwarded clock. The receiver architecture exploits synchrony in the transmitted data streams in a parallel interface and combines error information from multiple phase detectors in the receiver to produce one global synthesized clock. This collaborative timing recovery scheme enables wideband jitter tracking without increasing the dithering jitter in the synthesized clock. Circuit design techniques employed to implement this receiver architecture are discussed. Experimental results from a 130 nm CMOS test chip demonstrate the enhanced tracking bandwidth and lower dithering jitter of the recovered clock.

Index Terms—High-speed serial link, source-synchronous link, parallel receiver, clock and data recovery, jitter tolerance, jitter tracking bandwidth.

I. INTRODUCTION

THE exponential growth in IC technology has pushed on-chip clock rates well into the multi-GHz regime. In multi-chip digital systems, in order for the entire system to benefit from the increased on-chip computation speeds, the off-chip input/output (I/O) bandwidth should also scale. This has led to the widespread use of parallel links, where interfaces employ tens to hundreds of I/O links in parallel to achieve their bandwidth targets. Conventional parallel links are implemented either as an ensemble of serial links (e.g., FB-DIMM) [1] or as a source-synchronous link (e.g., Quickpath) [2].

An ensemble of serial links consists of identical transceivers, and each transmitter sends data to its receiver over a channel—typically a printed circuit board trace for microprocessor interfaces. Since the clock is embedded within the data, the receiver must recover both the clock and data from the incoming symbol stream. In contrast, in source-synchronous links, a clock is sent to the receiver along with multiple data signals. The receivers can use this forwarded clock for efficient timing recovery (i.e., phase alignment) and avoid complex circuitry required for frequency acquisition. Since inter-channel skew can be a significant fraction of the data unit interval (UI)

for high data rates, these receivers often employ per-pin skew compensation to maximize timing margins [3].

As process technology scales, parallel interfaces need to deliver the high aggregate I/O bandwidth requirements of future microprocessor systems while meeting latency, power consumption, and form-factor constraints. They must also deal with increasing amounts of on-chip power-supply noise and channel losses that can add inter-symbol interference (ISI) and cause jitter amplification. This paper describes the design and experimental results from a 8×5 -Gb/s parallel receiver that seeks to merge the desirable characteristics of both embedded- and forwarded-clock links. The proposed receiver employs *collaborative timing recovery* instead of relying on a forwarded clock or duplicating clock-recovery circuitry across all data channels. Given synchrony between parallel data channels, per-channel clock recovery is replaced by a single global timing recovery (TR). This collaborative architecture incurs approximately 10–15% area and power overhead when compared to source-synchronous links and is more power efficient than an ensemble of serial links. Collaborative timing recovery also greatly enhances the effective edge transition density, which enables higher bandwidth in the TR loop—without increasing recovered clock jitter or susceptibility to long sequences of 1's and 0's—to track midfrequency supply noise that plague large digital systems.

The remainder of this paper is organized as follows. Section II motivates this work with a brief overview of how power-supply noise affects link performance. Details of the proposed receiver architecture is then presented in Section III with implementation details of the main blocks and peripheral correction circuitry following in Section IV. The performance advantages of the proposed collaborative architecture are discussed in Section V based on experimentally measured results from the prototype chip. Lastly, Section VI summarizes the paper.

II. BACKGROUND

Before presenting the details of the receiver architecture and its circuit design, it is instructive to understand the disadvantages of existing architectures to motivate a new architecture. In an ensemble of serial links (Fig. 1), each transceiver has its own dynamic clock phase- and frequency-tracking loop, but this replication consumes excess area and power. Also, low edge-transition density in the data streams limits the tracking bandwidth of these links. While encoding schemes like 8b/10b can

Manuscript received March 03, 2009; revised July 04, 2009. Current version published October 23, 2009. This paper was approved by Associate Editor Jafar Savoj.

A. Agrawal and G.-Y. Wei are with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA (e-mail: ankur@eecs.harvard.edu).

A. Liu is with The Mathworks, Natick, MA 01760 USA.

P. K. Hanumolu is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA.

Digital Object Identifier 10.1109/JSSC.2009.2033399

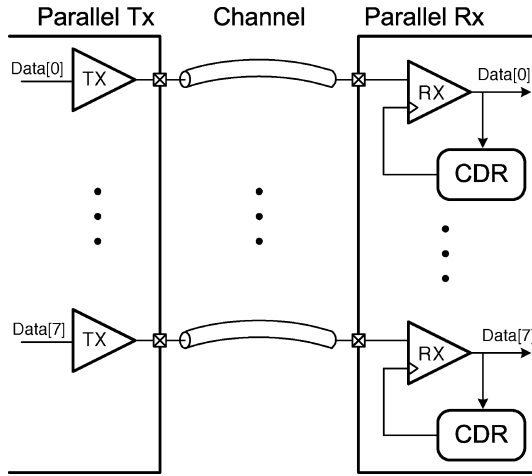


Fig. 1. Ensemble of serial links.

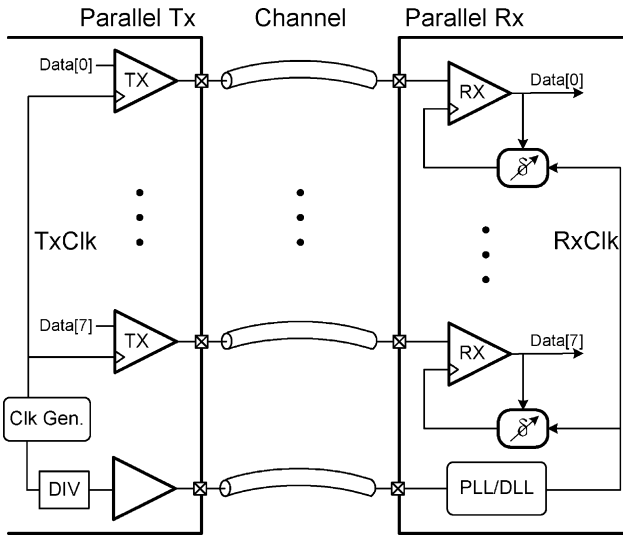


Fig. 2. Block diagram of source-synchronous links.

ameliorate this issue, it comes with throughput and power overheads. Source-synchronous links (Fig. 2) are more power efficient as they do not require clock-recovery hardware. The dynamic phase-tracking bandwidth of these links depends on the correlation between the clock and data jitter, and is limited by the mismatch in path length between the sampling clock and data. However, to save clock power, the frequency of the forwarded clock is often stepped down and a multiplying PLL or DLL is used in the receiver to step the frequency back up. This reduces the correlation in the jitter between the clock and data, and degrades phase-tracking bandwidth. Moreover, limited channel bandwidth causes jitter amplification at high data rates [4], further reducing the correlation between the clock and data jitter.

A simple noise analysis can highlight the differences in phase-tracking performance of serial links, source-synchronous links, and the proposed collaborative timing recovery. Time-domain simulations, based on a quad-tree model, shown in Fig. 3, elucidate link performance trade-offs across different noise scenarios given supply-induced jitter on the transmitted clock and data streams. In the model, total jitter is divided into

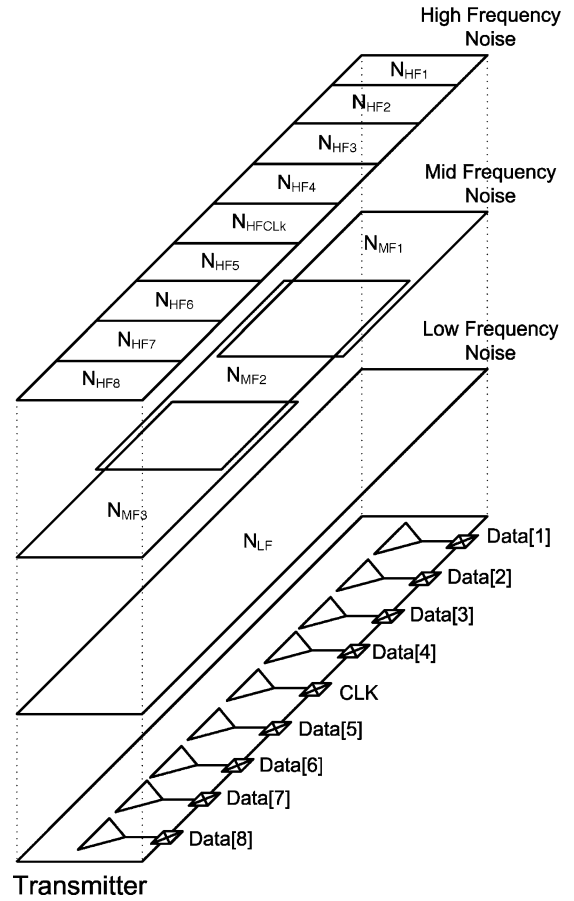


Fig. 3. Quad-tree model for jitter injection in a parallel transmitter.

three components: low-frequency jitter (N_{LF}) centered around 1 MHz; midfrequency jitter (N_{MF}) centered around 30 MHz; and high-frequency jitter (N_{HF}) centered around 1 GHz. The model assumes low-frequency jitter is correlated across all channels, because it stems from a shared transmit-side PLL. Midfrequency jitter incorporates the effects of spatially-correlated noise across adjacent groups of channels due to midfrequency resonance in the impedance of the power-delivery network, derived from analytical models of processor on-chip power-delivery networks that show midfrequency noise exhibits spatial correlations [5]. High-frequency jitter is not correlated across channels. Fig. 4 plots the simulated rms values of phase-tracking error for four different link configurations and three noise scenarios, assuming a data rate of 5 Gb/s and 0.2 UI p-p jitter added to the data streams.

For the low-frequency jitter dominated case, we find that all links have acceptable phase-tracking performance. The higher BW serial link, whose bandwidth is the same as that of the collaborative link, has higher dithering jitter. The links also have comparable performance in the high-frequency jitter dominated case, as the receivers filter out most of the jitter on the incoming data. The midfrequency noise dominated scenario is interesting. The source-synchronous link exhibits high phase-tracking error because midfrequency jitter on the clock and data channels is not strongly correlated for data channels not located close to the clock channel. The low bandwidth serial link filters out the

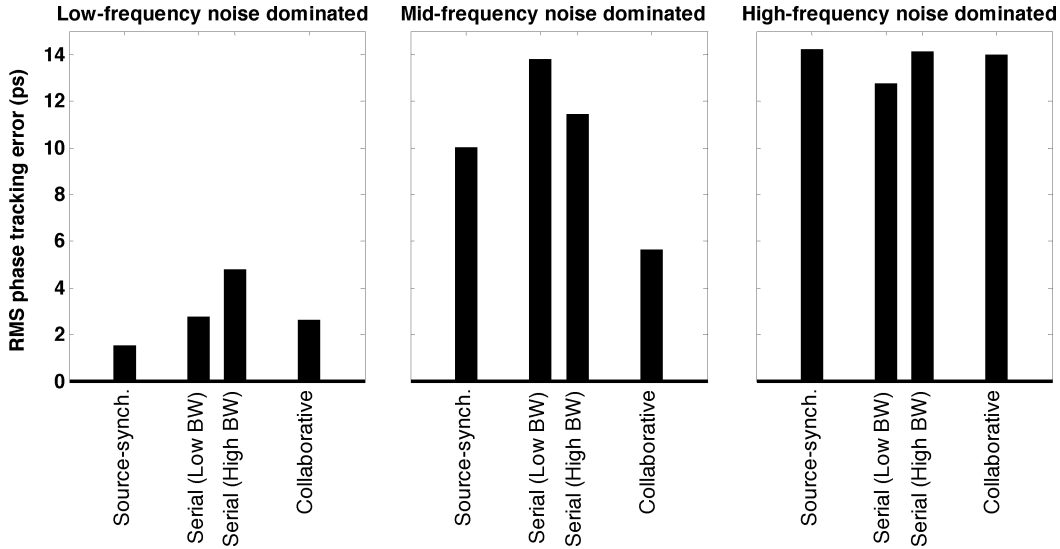


Fig. 4. RMS phase-tracking error for different parallel link configurations across three noise scenarios.

midfrequency noise and the high bandwidth serial link has prohibitively high dithering jitter, leading to conflicting trade-offs. In comparison, the collaborative link tracks the average midfrequency jitter across all of the channels with low dithering jitter and exhibits the lowest amount of phase-tracking error. While the collaborative architecture requires 10–15% area and power overhead when compared to source-synchronous receivers, it is more power efficient than serial links with high phase tracking bandwidth. Trends for on-chip power supply noise over the past few years show that the midfrequency component of the noise is growing [6], which in turn leads to higher amounts of midfrequency jitter in the transmitted data signals. Thus, it is important for high-speed receivers to have the ability to track midfrequency jitter and motivates this collaborative architecture.

Although the analysis above only uses discrete values of jitter frequencies, it provides insight into the behavior of the collaborative architecture. The phase tracking error is low for jitter frequencies within the tracking bandwidth of the loop and when noise across the channels are correlated. However, when the jitter is uncorrelated, or beyond the tracking bandwidth of the loop, performance degrades. This suggests designing the tracking bandwidth of the collaborative timing recovery loop to be greater than the midfrequency noise frequency, which has been in the 100–300 MHz range for multiple generations of modern microprocessors. While the test-chip prototype, implemented in a 130 nm technology, exhibits much lower bandwidth (1–10 MHz), it ought to scale well with technology. Higher clock rates, lower-latency clock buffers, and lower degrees of interleaving can enable the collaborative architecture, implemented in aggressively-scaled technologies, to track midfrequency jitter with low tracking error.

III. PROPOSED ARCHITECTURE

The block diagram of the proposed receiver architecture is shown in Fig. 5 [7]. It consists of multiple local receiver slices, one for each data channel of the parallel receiver, and one global timing recovery (TR) block. Each local receiver slice

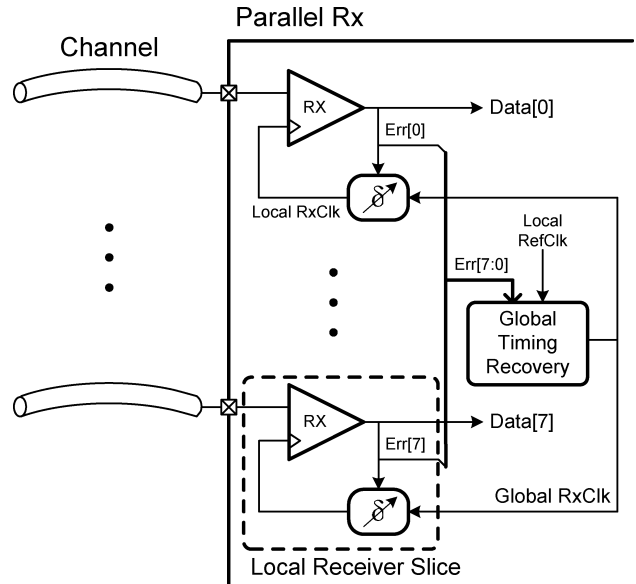


Fig. 5. Receiver architecture.

uses Alexander-type bang-bang phase detectors, and early/late timing error information from each of these detectors is sent to the global TR block. The global TR aggregates the error information and produces the recovered clock (Global RxClk) that tracks the frequency and the correlated jitter of the data signals. No synchrony is assumed between the reference clock and the clock on the transmitter, and the burden of frequency synthesis is placed on the global TR block. This recovered clock is distributed to the local receivers. The local receivers manipulate only the phase of the incoming global RxClk to compensate for static inter-channel skew. They also generate local multiphase clocks (Local RxClk) that drive local time-interleaved receiver samplers.

Fig. 6 illustrates the details of the global TR block with respect to the design parameters used in the prototype chip. The popular dual-loop architecture [8] is employed for timing

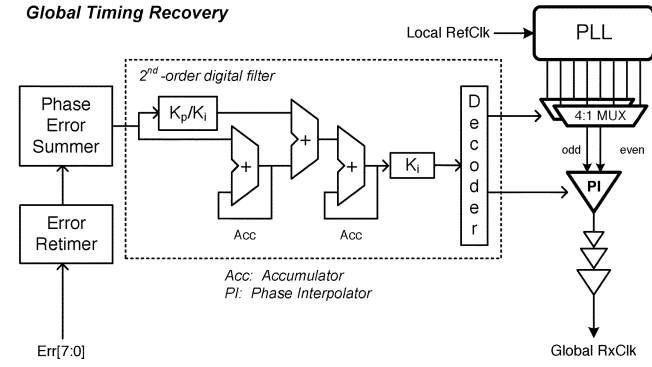


Fig. 6. Global timing recovery block diagram.

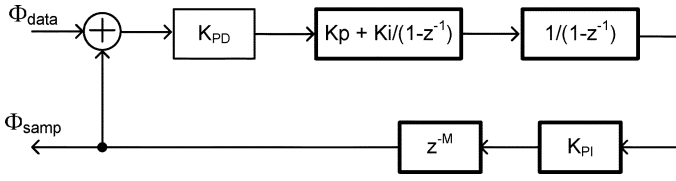


Fig. 7. Linearized CDR model.

recovery. A phase-locked loop generates eight clocks evenly spaced by 45° , which are then used by the clock recovery loop to generate the synthesized clock. In the clock recovery loop, the early/late timing error signals from 8 data channels are first aligned to a common clock, and then summed up in the phase error summer. The error information arrives once every 2 clock cycles to alleviate speed requirements of downstream digital circuitry at the expense of tracking bandwidth. The error sum is passed to a second-order digital filter having both proportional (K_p) and integral (K_i) paths. Using second-order control enables the loop to track frequency offsets between reference clocks in the transmitter and receiver, and extend the frequency-tracking range of the timing recovery loop [9]. Decoded bits out of the filter drive a clock synthesizer consisting of two 4:1 MUXes and a 5b phase interpolator (PI) to generate the Global RxClk.

A linear analysis of the global timing recovery loop is used to set the K_p and K_i values, and ensure stability of the system. A number of recent works have developed small-signal models for digital clock and data recovery (CDR) loops, as shown in Fig. 7 [9], [10]. In this model, K_{PD} is the linearized gain of the bang-bang phase detector (PD), K_p is the proportional gain of the filter, K_i is the integral gain of the filter, K_{PI} is the gain of the phase interpolator, and M is the latency in the loop. The overall open-loop transfer function is

$$L(z^{-1}) = \left(\frac{K_{PD} \cdot K_{PI}}{1 - z^{-1}} \right) \left(K_p + \frac{K_i}{1 - z^{-1}} \right) z^{-M}. \quad (1)$$

The phase transfer function is given by

$$\frac{\Phi_{\text{samp}}}{\Phi_{\text{data}}} = \frac{L(z^{-1})}{(1 + L(z^{-1}))} \quad (2)$$

where Φ_{samp} and Φ_{data} are the phase of the sampling clock and the data, respectively.

The collaborative timing recovery loop can also be modeled using (2), with the only difference being the representation of the PD. The models described in [9], [10] linearize the grossly non-linear transfer function of the PD by calculating its gain in the presence of sampling clock jitter. It has been shown that if the rms value of the clock jitter is σ_j , the linearized gain K_{PD} is equal to $1/(\sqrt{2\pi}\sigma_j)$ [11]. The K_{PD} for the collaborative loop is the effective gain for the ensemble of PDs. The original K_{PD} is scaled by a factor of 8 since error from 8 receivers are summed together. Also, the interleaved front-end samplers have small voltage offsets caused by inherent device mismatch in deep sub-micron process technologies. The spread in sampler offsets further linearizes the transfer function of the phase detector. This model is used to set the phase tracking bandwidth of the collaborative timing recovery loop.

Fig. 8 presents details of each receiver slice, which consists of two cascaded DLLs, interleaved data and edge samplers, skew-compensation logic, and phase spacing error correction circuits. The global RxClk first goes through a duty cycle corrector (DCC) to compensate for distortion from the clock distribution buffers. The first DLL, called the phase de-skew DLL, adds a variable amount of delay to the clock path such that its output clock (Local RxClk) is phase aligned to the incoming data. The second DLL produces the 8 evenly-spaced clock phases that drive 8 interleaved edge and data samplers. The phase detector logic generates early/late information sent to the global TR block and to the digital filter in the local phase de-skew DLL. The filter controls a simple 4b thermometer encoded DAC to add offsets between the up and down currents of the charge pump, which translates to skewing the DLL output clock [12]. The digital filter uses $\Delta\Sigma$ modulation to reduce quantization noise in this loop with phase filtering provided by the low-pass transfer function through the DLL (I_{CP} to Φ_{out}). The tuning range of the de-skew circuit is greater than ± 0.5 UI. Also shown in the figure is the phase spacing error correction loop that corrects imbalances in the delays through each stage of the delay line due to any nonidealities in the reference clock entering the second DLL.

The cascaded-DLL architecture was chosen for a variety of reasons, although other architectures can also provide similar functionality. It only requires one phase of the global RxClk to generate the de-skewed local RxClk, which avoids having to distribute multiple clock phases with phase mixers at each receiver slice. DLLs exhibit substantially all-pass phase transfer characteristics and, hence, does not limit the global TR bandwidth. No extra preshaping delay buffers are required in the clock path before the second DLL used for multiphase clock generation. Lastly, this topology offers a convenient location for placing the duty cycle correction circuit. The duty cycle of the clock entering the second DLL is sensed and tuned to 50% via correction circuitry placed before the first DLL to not disturb the shape of the clock signal into the second DLL. Given the large number of loops in this design, care must be taken to ensure that loops do not adversely interact with one another. Since DLLs in the local receiver slices do not introduce phase filtering in the clock path of the local receiver, they do not impose additional constraints on the phase-tracking bandwidth of the global

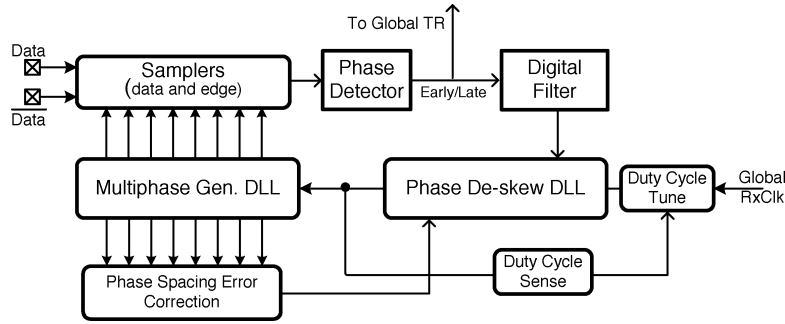


Fig. 8. Local receiver slice block diagram.

TR loop. With a digital filter, local de-skew loop bandwidth can be programmable, and is set 2 orders of magnitude lower than the bandwidth of the global TR loop. Also, the local de-skew loop only operates periodically to compensate for static skew and temperature drift. This periodic operation minimizes additional dithering jitter on the sampling clock, which would otherwise occur as a result of the global TR loop trying to track the combined dithering of the parallel de-skew loops.

IV. CIRCUIT DESIGN

This section describes the implementation details of the building blocks for both the global timing recovery loop and the local receiver slices.

A. Global Timing Recovery Loop Components

We begin with the main components of the global TR loop. The error summer, digital filter, and decoder all rely on digital circuitry, implemented with a standard digital CAD flow. The PLL, used for multiphase clock generation, uses a conventional design and its details are omitted for brevity. The phase interpolator and error retimer are described in detail below.

The 5b phase interpolator, combined with the 8 VCO clock phases, provides a total of 256 phase steps per clock period. The unmodified digital interpolator shown in Fig. 9 uses weighted inverters to mix two phases and generate a full-swing clock. Thermometer-weighted inverters are used for the MSBs to minimize the impact of process mismatch in order to achieve a monotonic phase-versus-code characteristic. One drawback of this digital interpolator is that, large RC time constants at nodes A, B, and C are desirable such that the clock transition times are roughly 2–3x the VCO phase spacing to provide good linearity. However, this reduces the signal swing at node C. Also, due to mismatched drive strengths between the pMOS and nMOS devices in the inverters and process variations, the DC voltage at node C can shift away from the switching threshold of inverter Z. The shifted DC voltage and reduced signal swings can cause severe duty-cycle distortion out of inverter Z or failure to switch at all. A remedy is to add a series capacitor before inverter Z and a feedback resistor to self-bias its input voltage to precisely the switching threshold. With the series capacitor, even if node C sees a large DC shift with small signal swing, node D will be biased to the optimal voltage for inverter Z to amplify the signal with minimal duty-cycle distortion. The series capacitance is large (100 fF) with respect to the gate capacitance of

inverter Z (2 fF) to minimize signal attenuation, and the feedback resistance is large (90 k Ω) to minimize static current draw. Interleaved metal-fingers comprise the series capacitor, which introduce little parasitic capacitance and negligible amounts of additional power.

Fig. 10 illustrates the error-retimer circuit in the global TR block to align error information from multiple receiver slices to its local clock domain. This circuit is required since the arrival times of the E/L signal to the global TR block depends on the timing in each local Rx slice and the propagation delay from the local slice to the global TR block. Since frequency is matched, this circuit samples the incoming E/L[n] signal with a pair of rising-edge clock signals separated by more than the setup-and-hold window of the flip-flops. A mismatch between the two samples tells the XOR based logic to select the output data sampled on the falling clock edge.

B. Local Receiver Slice Components

The details of the local receiver building blocks are discussed next. The digital logic for the phase detectors and loop filter in the local receivers were created using a standard digital CAD flow. The design details of the DLL is described below.

Nearly identical DLLs are used for phase de-skew and multiphase clock generation. Fig. 11(a) shows the details of the DLL. The incoming clock goes through a delay line consisting of four differential buffers. The phase detector in the loop locks the delay of the delay line to 180°. The PD is designed to have only two states, to avoid the false-lock problem associated with using phase-frequency detectors in DLLs [13]. When the loop is in lock, the PD generates narrow UP and DN pulses of equal widths to prevent a dead zone in its transfer function. The use of an active loop filter offers two advantages. It provides a second-order low pass transfer function from I_{CP} , the charge pump current, to Φ_{out} that is able to filter out the quantization noise in the control bits to the charge-pump DAC in the phase de-skew DLL. The feedback amplifier also biases the output node of the charge pump to V_{REF} , irrespective of the delay of the delay line. By adjusting V_{REF} , small mismatches in the charge pump up and down currents of the multi-phase generating DLL can be compensated.

The delay cell uses pseudo-differential current starved inverters with rail-to-rail swing, and is shown in Fig. 11(b). The current source is split into a 4X device and a 1X device that are controlled by the coarse and fine voltage control signals, respectively. In this test chip, the coarse control voltage is set

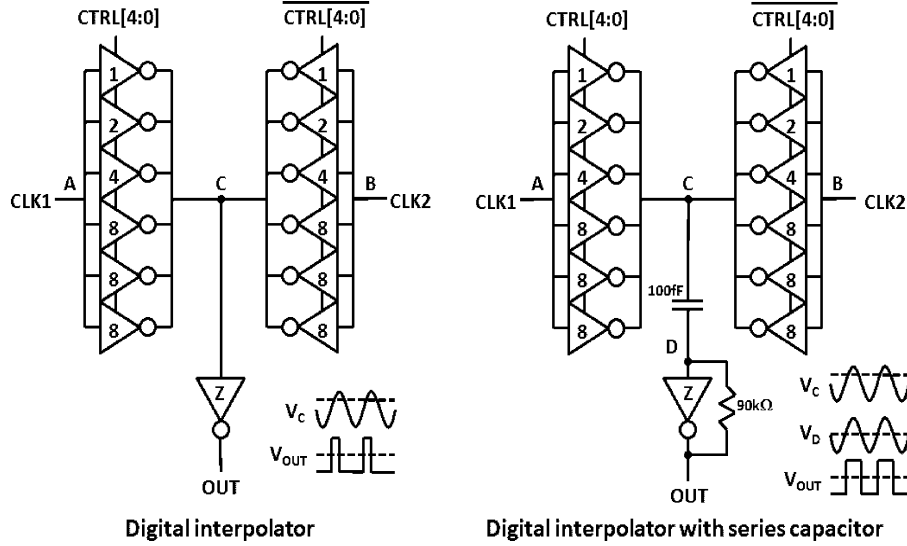


Fig. 9. Phase interpolator.

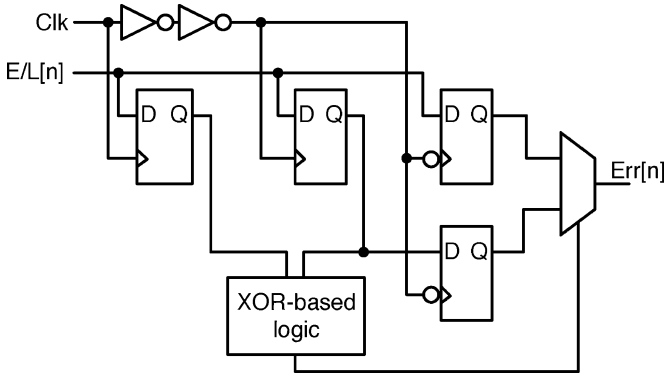


Fig. 10. Error retimer.

manually and the feedback loops determines the fine control voltage. In a real system, a peripheral loop can be designed to set the coarse control voltage during the initialization of the receiver. The simulated coarse and fine gain of the delay line are 800 ps/V and 200 ps/V, respectively. While these delay cells produce nearly full-swing clock signals, they have asymmetric rise and fall times due to the difference in strengths of the pull-up and pull-down paths. To correct for this, duty-cycle restoring buffers similar to those described in [9] are used at the output of the delay buffers to drive the samplers. Series-connected sense amplifier based samplers are employed in this design [14].

C. Phase Spacing Error Correction Circuit

The local receivers use two DLLs for phase de-skew and multi-phase clock generation, respectively. Both of these functions can be achieved by using a single phase-locked loop [15], but DLLs are preferred as they do not perform any phase filtering on the incoming synthesized clock. However, there are some caveats to using DLLs: 1) A small difference between the charge pump up and down currents can result in static phase spacing error at the end of the delay line in a DLL; 2) DLLs that lock the delay of the delay line to half a reference clock cycle are

very sensitive to the duty cycle of the reference clock signal; and 3) the shape of the reference clock entering the delay line can exacerbate phase spacing mismatches in DLLs. We have discussed a technique to correct for the charge pump current mismatch while describing the DLL. The duty-cycle corrector used is similar to that used by Tan *et al.* [16]. This section describes a technique to adjust the shape of the reference clock entering the delay line to reduce phase spacing errors.

In a voltage-controlled delay line, the voltage swing and slope of the clock entering the first delay cell can differ from those entering subsequent stages, resulting in adjacent delay stages having slightly different delays. All stages would have equal delays if the input and output signals were identical, as is the case for a voltage-controlled oscillator. Hence, designers frequently employ preshaping buffers to preshape the clock entering the delay line. However, preshaping buffers introduce additional latency in the timing recovery loop. The proposed local receiver has a full delay line from the phase de-skew DLL to shape the clock. Unfortunately, its control voltage is set by the phase de-skew loop, which can differ from the second DLL's control voltage. Fig. 12 shows the simulated nonlinearity in the phase spacings for a DLL consisting of a four-stage differential delay line after preshaping using an identical delay line with a different control voltage. Despite the nearly-ideal simulation environment, deviations in stage-to-stage delay are noticeable and can be attributed to our delay cell, which is sensitive to the slope of the rising edge of the incoming signal. Other delay cell topologies may be less sensitive to the shape of the clock entering the delay line.

Simple correction circuitry can further shape the clock signal entering the delay line to equalize stage delays in second DLL, used for multiphase clock generation. The rise and fall times of the input clock to the multi-phase generating DLL are adjusted by sinking or sourcing small amounts of current from the penultimate delay stage of the phase de-skew DLL. A feedback loop, shown in Fig. 13, consists of XOR-based delay detectors, charge pump, loop filter, and a V/I converter [17]. As can be

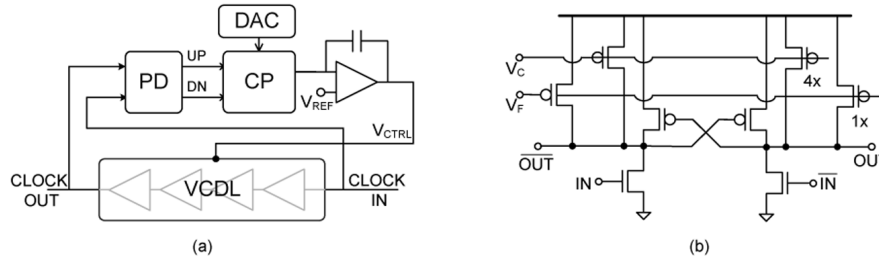


Fig. 11. (a) Delay-locked loop block diagram and (b) delay cell schematic.

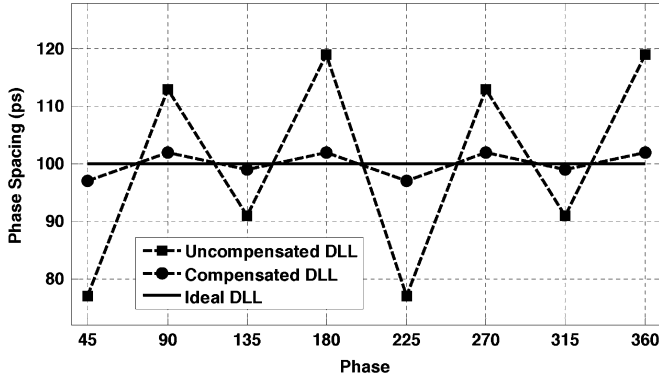


Fig. 12. Phase spacing errors caused by the shape of the reference clock (simulated).

seen from Fig. 12, imperfect clock slope of the incoming clock causes the delays of the delay cells to be either distributed as fast-slow-fast-slow or as slow-fast-slow-fast. Thus, the error is such that all odd delay stages are either faster or slower than all even delay stages. The charge pump dumps or removes charge proportional to the difference between the odd and even delays. Multiple error detectors are used to average out the effects of random mismatches between the adjacent stages. The charge pump employs feedback biasing to ensure that static and dynamic mismatches in the up and down currents do not introduce offsets that can result in residual delay mismatches.

Simulation results (Fig. 12) indicate that the RMS DNL reduces from 16.9 ps to 2.1 ps. However, the efficacy of this method is limited by mismatches in the loading of the delay cells and random mismatches in the delay cells and XOR gates. To prevent interaction between the loops, the bandwidth of this loop is set to be greater than the bandwidth of the phase de-skew loop. The shape of the reference clock entering the second DLL changes when there is a change in the control voltage of the first DLL, which in turn is controlled by the phase de-skew feedback loop. Thus, as long as the error-correction loop bandwidth is greater than this, it is able to compensate for the change in the control voltage of the first DLL.

V. MEASUREMENTS AND RESULTS

A test chip was fabricated in the UMC 130 nm CMOS logic process and the die photo is shown in Fig. 14. The floorplan of each local receiver slice is shown in Fig. 15 for additional detail. The chip occupies a total area of 6.25 mm², and an active area of 3.2 mm². The die is bonded directly to a four-layer test

board using gold bond wires to minimize parasitics. This is critical as no equalization is included in this receiver test chip and maximum signal integrity of the data signals is desirable at the sampler inputs.

To test this chip, up to eight synchronous PRBS data signals are required with the ability to add controlled amounts of correlated jitter to the data signals. While the simplest way to achieve this is to use a high-speed parallel bit error rate tester (parBERT), we rely on two lower-cost options available to us. For initial tests, a Tektronix 5334 parallel data generator was used. Its maximum data rate is 3.35 Gb/s and we performed a comprehensive set of tests at 3.2 Gb/s to verify functionality and performance of the test-chip at lower data rates.

To test the chip at higher data rates, we rely on an FPGA-based parallel transmitter with the ability to add correlated jitter on multiple data streams as shown in Fig. 16, consisting of a Xilinx Virtex4 FPGA with *RocketIO* transceivers that can generate multiple parallel data streams. Correlated jitter is added by combining voltage noise to one of the differential clock input signals using a power combiner/splitter. Inside the FPGA, the output of the differential to single-ended converting buffer is a clock with jitter. This clock is then frequency multiplied by a PLL and distributed to the transceiver slices. The -3 dB bandwidth of this PLL is in the 20–30 MHz range, facilitating the injection of relatively wideband jitter on the transmitted data.

The local reference clock for the test chip is generated from a separate clock generator. This allows the addition of a frequency offset between the clock and data. The recovered clock is driven to a sampling scope to measure the jitter histogram. An on-chip PRBS verifier is used to perform BER measurements.

A. System Results

Several parameters are swept to experimentally demonstrate the merits of the collaborative timing recovery architecture. The receiver can be configured to combine error information from different numbers of channels and have different settings for K_p and K_i in the digital filter of the global TR. Different frequency offsets can be added to the local reference clock of the receiver with respect to the transmitter clock and PRBS data of different run-lengths can be transmitted to the receiver. Fig. 17 plots the jitter histogram of the global recovered clock (triggered by the transmitter clock) when the global TR is combining error information from 7 channels, $K_p = 2^{-5}$, $K_i = 2^{-11}$, 7 bit PRBS data, and no frequency offset. Access to the 8th channel is unfortunately blocked by insufficient connector spacing on the board. The rms value of the recovered clock jitter is 5.24 ps.

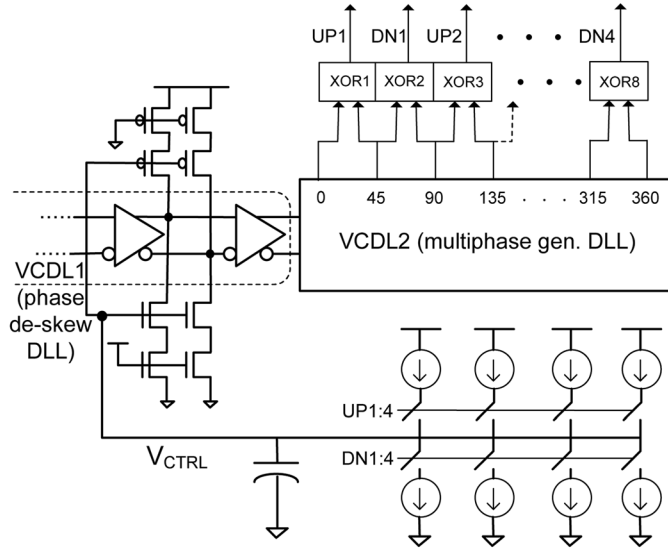


Fig. 13. Phase spacing error corrector schematic.

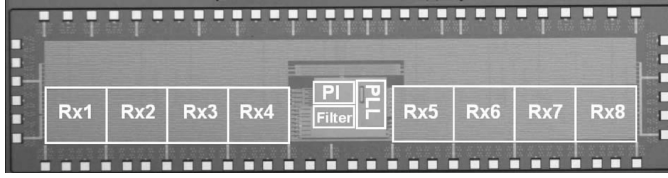


Fig. 14. Die micrograph with floorplan overlay.

Fig. 18 plots the sideband in the spectrum of the recovered clock, relative to a clean clock, when correlated wide band jitter is added to the transmitted data. The number of channels sharing timing information is swept while the K_p and K_i settings of the global TR block are fixed to 2^{-5} and 2^{-11} , respectively. The plot shows that the jitter tracking bandwidth of the receiver improves as the number of channels that contribute timing error information increases from 1 to 6. Results for 7 or 8 channels could not be obtained due to limitations of the FPGA-based test setup. The plot also reveals jitter peaking, which can be attributed to additional latency inadvertently added to the global TR loop in the latter stages of design.

The jitter tracking bandwidth of the global TR loop can also be increased by increasing the gain coefficients in its digital filter, while fixing the number of channels that share timing information. Fig. 19(a) plots the dithering jitter on the recovered clock for the two scenarios. Shown in the dark line is the case where the number of channels is fixed to 1 and the proportional and integral gains are increased to increase the loop bandwidth. The K_p/K_i ratio is fixed to 2^{-6} . As expected, we see that as the coefficients are increased, the dithering jitter also increases from 2.7 ps to 3.8 ps. The other scenario, plotted using the lighter line, shows the dithering jitter for the collaborative TR. We sweep the number of channels that share timing information while keeping the gain coefficients fixed. In this case, we see that as the bandwidth of the loop increases, the dithering jitter reduces from 2.7 ps to 2.2 ps. The numbers reported in this plot indicate only the dithering jitter that is obtained by subtracting the baseline PLL jitter of 4.7 ps from the total clock jitter. This plot confirms

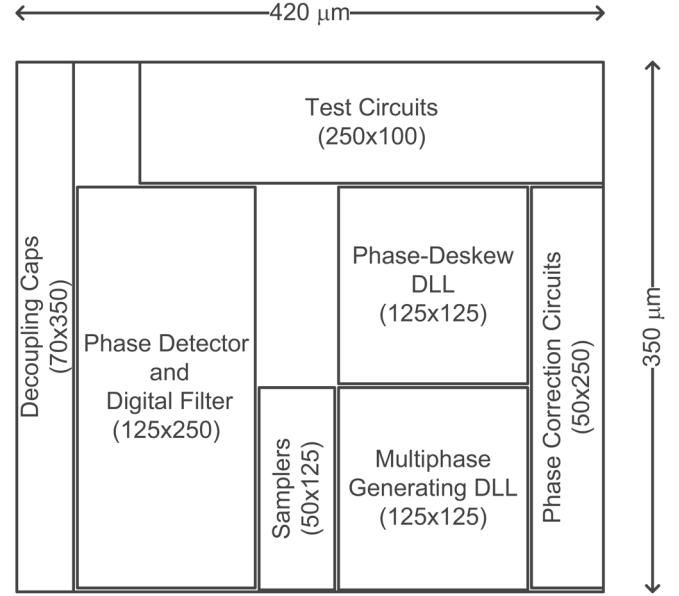


Fig. 15. Local Rx slice floorplan.

that increasing the effective edge transition density via collaboration enables the timing recovery loop to operate with higher tracking bandwidth settings while reducing dithering jitter. To further demonstrate this, Fig. 19(b) shows dithering jitter measured on the recovered clock reduces as the number of channels increases while tracking bandwidth is held roughly constant by modifying gain coefficients.

The increased edge transition density also helps to improve the robustness of the timing recovery in the face of long strings of 1's and 0's. Fig. 20 plots the dithering jitter in the recovered clock for 7-bit PRBS case and 15-bit PRBS case, when 200-ppm frequency offset is present between the local reference clock and the data. For the 1 channel case, the dithering jitter is higher when the longer run-length PRBS sequence is used. As the number of channels sharing timing information increases, the difference in dithering jitter for the two PRBS lengths reduces, demonstrating the resilience of the collaborative architecture to long strings of 1's and 0's. A 200-ppm frequency offset is used in this experiment, as the CDR becomes more sensitive to the run-length when it is functioning as a frequency synthesizer. This also explains the higher dithering jitter than those reported in Fig. 19. Finally, the measured linearity of the phase interpolator is shown in Fig. 21. The worst case INL and DNL of the interpolator are 5.6 LSBs and 0.8 LSBs, respectively.

B. Correction Circuit Results

Fig. 22 plots the measured phase spacings with the PSEC loop turned on and off for one of the DLLs in the parallel receiver. The reference clock frequency is 1.25 GHz and the nominal delay of each stage is 100 ps. The rms differential nonlinearity (DNL) reduces from 21 ps to 10 ps. The residual error can be attributed to random delay cell mismatches and offsets in the correction circuit. The rms DNL for seven DLLs from seven receiver slices are plotted in Fig. 23. On average, the rms DNL reduces by 42% when the PSEC loop is turned on. All phase

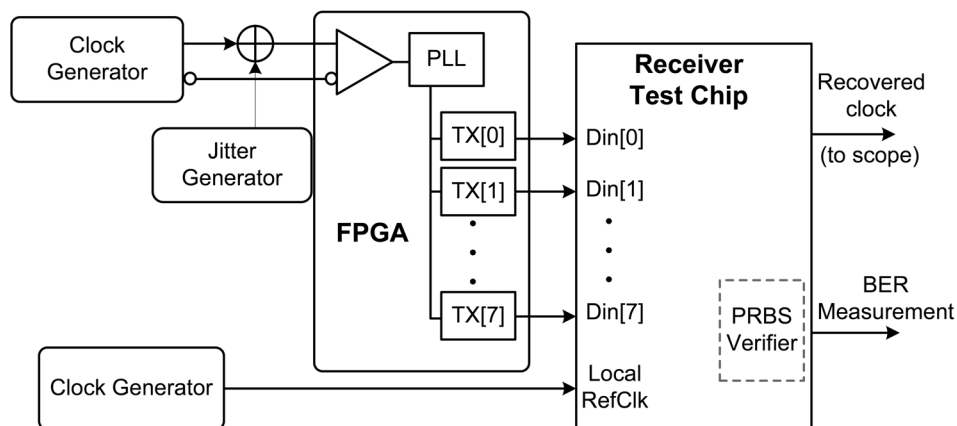


Fig. 16. Experimental setup.

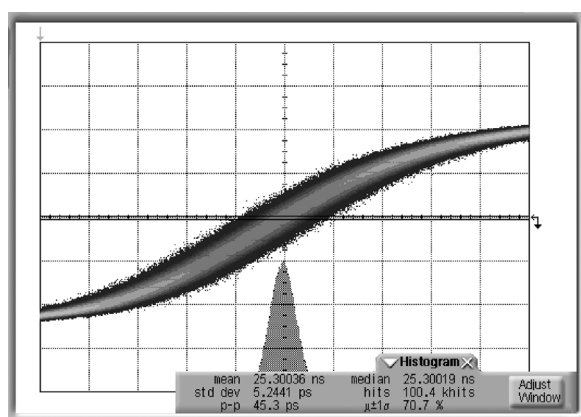


Fig. 17. Recovered clock jitter measurement.

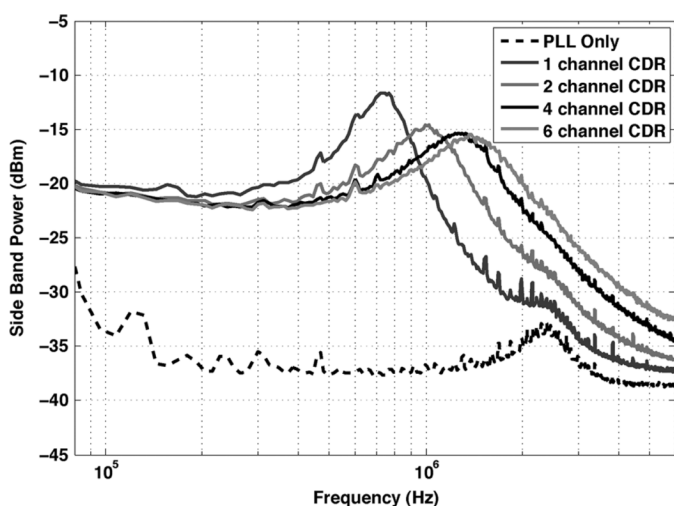


Fig. 18. Sideband of the spectrum of recovered clock when wideband jitter is added to the data-streams for different number of channels sharing timing information.

spacing measurements were made by sweeping a 1010... data pattern at 5 Gb/s across half a reference clock cycle and observing the sampler outputs.

Overall, the 8×5 Gb/s parallel receiver consumes 310 mA of current from a 1.45 V supply (including all output drivers), which translates to a power efficiency better than

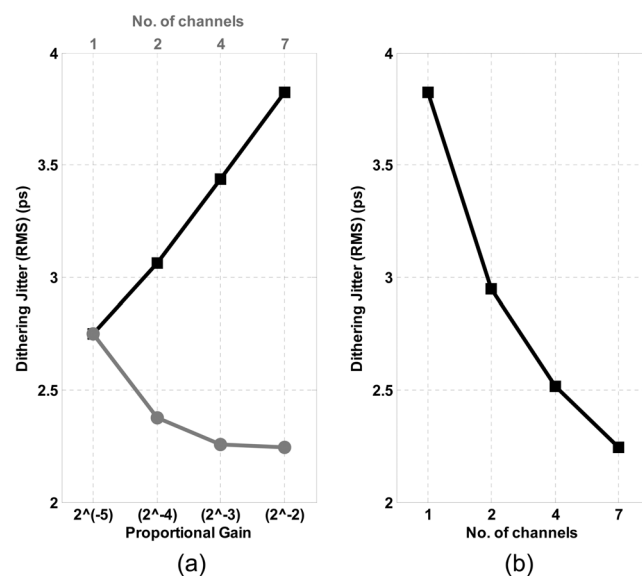
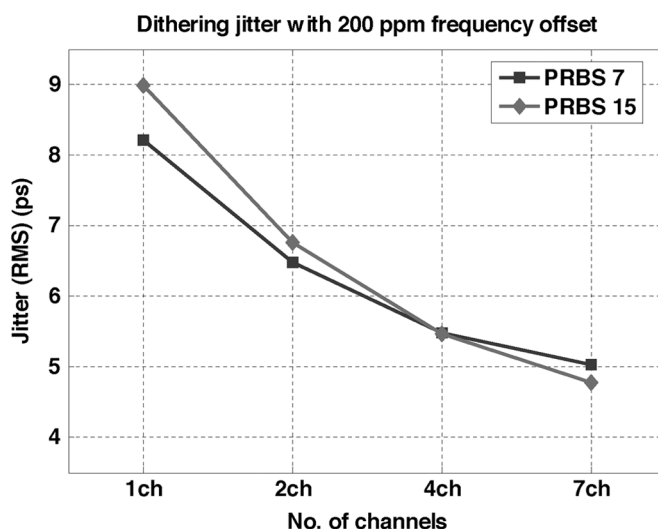
Fig. 19. (a) Measured rms jitter versus K_p , K_i settings (squares), and versus number of channels sharing timing information (circles). (b) Measured rms jitter for different degrees of collaboration while keeping loop tracking bandwidth fixed.

Fig. 20. Dithering jitter on the recovered clock versus number of channels sharing timing information versus PRBS run-length.

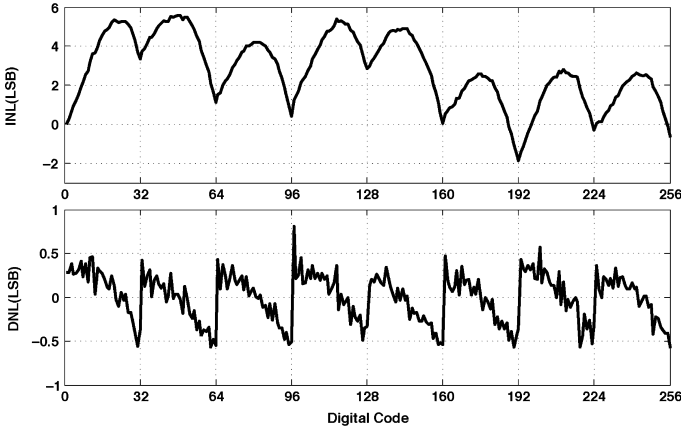


Fig. 21. Phase interpolator INL and DNL.

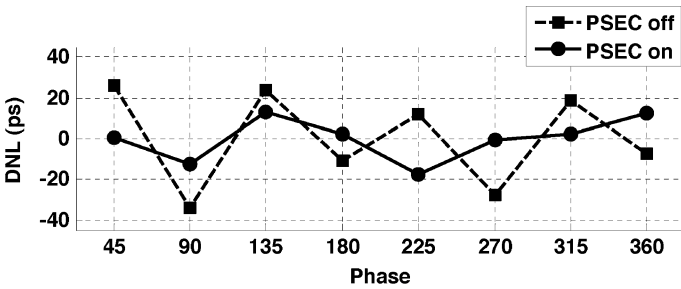


Fig. 22. DNL in phase spacings with the PSEC loop disabled and enabled for a representative DLL.

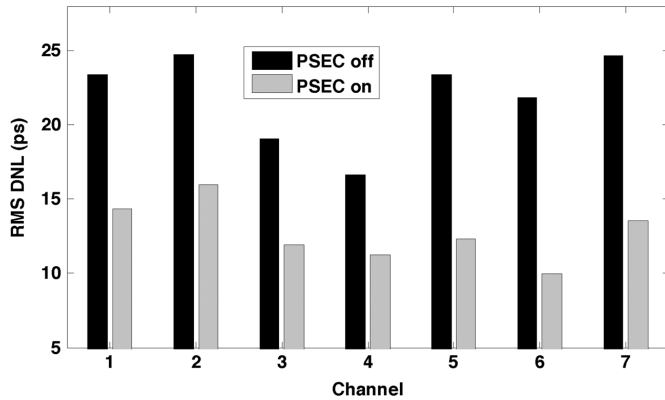


Fig. 23. Comparison of rms DNLs for 7 DLLs with the PSEC loop disabled and enabled.

11.2 mW/Gb/s. The performance of the test chip is summarized in Table I.

C. Discussion

While the collaborative timing recovery architecture enables wideband jitter tracking, some attention must be paid to the area and power overheads of this design. Table II provides a power breakdown of this loop, calculated from post-layout schematic simulations. The power overhead of implementing this architecture over traditional source-synchronous receivers is around 8.5%. The area overhead of the phase interpolator, global digital loop filter, error summer and routing is around 15% per receiver. The power and area costs of the PLL has not been taken into account while calculating the overheads.

TABLE I
PERFORMANCE SUMMARY

Technology	0.13 μ m CMOS
Supply voltage	1.45V
Active Area	3.2 mm ²
Throughput	8 \times 5 Gb/s
Power consumption	450 mW
Tracking range	± 5000 ppm
BER	$< 10^{-12}$

TABLE II
DETAILED POWER BREAKDOWN

Local Rx Slices	81.9%
PLL	3.6%
Clock Distribution	6.0%
Phase Error Signal Routing	2.7%
Phase Interpolator and Global Control Logic	5.8%

The additional power and area cost is used to achieve the increased jitter tracking bandwidth while limiting the dithering jitter. As was shown in Section II, source-synchronous receivers fail to track the spatially varying components of the jitter on incoming data. If one were to design a serial receiver with large tracking bandwidth, the update rate of the control logic in the receiver would have to be very fast, making each receiver consume too much power. Moreover, some techniques to control the higher dithering jitter would have to be implemented. Recently, O'Mahony *et al.* presented an injection-locked parallel receiver architecture that achieves wideband jitter tracking while having very good power efficiency [18]. However, the use of *LC* oscillators in each receiver path leads to high area overheads.

VI. SUMMARY

A parallel receiver with a collaborative timing recovery architecture is presented. Sharing timing information from several synchronous data streams enables wideband jitter tracking by reducing dithering jitter on the received clock. The proposed design incorporates several techniques: a global digital clock synthesizer that sums up timing error information from multiple receiver slices and filters it using a second-order digital filter; a dual DLL based receiver architecture that performs skew compensation and samples 4-way interleaved data without introducing phase filtering in the path of the sampling clock; and a phase-spacing error correction circuit that compensates for phase-spacing mismatches introduced by the shape of the reference clock in DLLs. The design techniques are validated by a test-chip fabricated in a 130 nm CMOS process. An FPGA-based parallel transmitter is developed to test the receiver test-chip at high data rates. Experimental results confirm that the collaborative architecture improves the overall jitter performance of the timing recovery loop.

ACKNOWLEDGMENT

The authors thank UMC for chip fabrication; IBM and Intel for research gifts that partially supported this work; and Hayun Chung and Mark Hempstead for helpful discussions.

REFERENCES

- [1] *FBDIMM Specification: High Speed Differential PTP Link at 1.5 V*, JEDEC Standard-JESDS-18, 2006.
- [2] "Intel quickpath architecture," Intel Corp., Whitepaper 2008 [Online]. Available: www.intel.com
- [3] E. Yeung and M. A. Horowitz, "A 2.4 Gb/s/pin simultaneous bidirectional parallel link with per-pin skew compensation," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1619–1628, Nov. 2000.
- [4] G. Balamurugan and N. Shanbhag, "Modeling and mitigation of jitter in multiGbps source-synchronous I/O links," in *Proc. 21st Int. Conf. Computer Design*, 2003, pp. 254–260.
- [5] M. S. Gupta, J. L. Oatley, R. Joseph, G.-Y. Wei, and D. M. Brooks, "Understanding voltage variations in chip multiprocessors using a distributed power-delivery network," in *Proc. Conf. Design, Automation and Test in Europe*, 2007, pp. 624–629.
- [6] K. Aygun *et al.*, "Power delivery for high performance microprocessors," *Intel Technology J.*, Nov. 2005.
- [7] A. Agrawal, P. K. Hanumolu, and G.-Y. Wei, "A 8×3.2 Gb/s parallel receiver with collaborative timing recovery," in *IEEE ISSCC Dig. Tech. Papers*, 2008, pp. 468–469.
- [8] S. Sidiropoulos and M. A. Horowitz, "A semidigital dual delay-locked loop," *IEEE J. Solid-State Circuits*, vol. 32, no. 11, pp. 1683–1692, Nov. 1997.
- [9] P. K. Hanumolu, G. Y. Wei, and U. Moon, "A wide-tracking range clock and data recovery circuit," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 425–439, Feb. 2008.
- [10] J. Sonntag and J. Stonick, "A digital clock and data recovery architecture for multi-gigabit/s binary links," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 2005, pp. 532–539.
- [11] 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, Sep. 2004.
- [12] K. L. J. Wong, H. Hatamkhani, M. Mansuri, and C. K. K. Yang, "A 27-mW 3.6-Gb/s I/O transceiver," *IEEE J. Solid-State Circuits*, vol. 39, no. 4, pp. 602–612, Apr. 2004.
- [13] G.-Y. Wei, "Energy efficient I/O interface design with adaptive power-supply regulation," Ph.D. dissertation, Stanford Univ., Stanford, CA, 2001.
- [14] B. J. Lee, M. S. Hwang, S. H. Lee, and D. K. Jeong, "A 2.5–10-Gb/s CMOS transceiver with alternating edge-sampling phase detection for loop characteristic stabilization," *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 1821–1829, Nov. 2003.
- [15] P. Larsson, "A 2–1600-MHz CMOS clock recovery PLL with low-V_{dd} capability," *IEEE J. Solid-State Circuits*, vol. 34, no. 12, pp. 1951–1960, Dec. 1999.
- [16] A. H. Tan and G.-Y. Wei, "Phase mismatch detection and compensation for PLL/DLL based multi-phase clock generator," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 2006, pp. 417–420.
- [17] A. Agrawal, P. K. Hanumolu, and G. Y. Wei, "A 8×5 Gb/s source-synchronous receiver with clock generator phase error correction," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 2008, pp. 459–462.
- [18] F. O'Mahony, S. Shekhar, M. Mansuri, G. Balamurugan, J. E. Jaussi, J. Kennedy, B. Casper, D. J. Allstot, R. Mooney, and H. Intel, "A 27 Gb/s forwarded-clock I/O receiver using an injection-locked LC-DCO in 45 nm CMOS," in *IEEE ISSCC Dig. Tech. Papers*, 2008, pp. 452–453.



Ankur Agrawal (S'06) received the B.Tech. degree in electrical engineering from Indian Institute of Technology (IIT), Madras, India, in 2004, and the M.S. degree from Harvard University in 2006. He is currently working towards the Ph.D. degree in the School of Engineering and Applied Sciences at Harvard University.

In fall 2005, he was a co-op intern at Intel Massachusetts in Hudson, MA. In the summers of 2008 and 2009, he interned at IBM T.J. Watson Research Center in Yorktown Heights, NY, where he worked on the design of equalizers for high-speed (up to 20 Gb/s) electrical links. His research interests include transceivers for high speed wireline communications, PLL/DLL design and design of analog circuits in deep-submicron technologies.



Andrew Liu received the B.S. and M.S. degrees from Stanford University in 2003 and 2004, respectively, and the M.S. degree from Harvard University in 2008, all in electrical engineering.

He is currently with The MathWorks, Natick, MA.



Pavan Kumar Hanumolu (S'99–M'07) received the B.E. (Hons.) degree from the Birla Institute of Technology and Science, Pilani, India, in 1998, the M.S. degree from the Worcester Polytechnic Institute, Worcester, MA, in 2001 and the Ph.D. degree from Oregon State University, Corvallis, in 2006.

He is currently an Assistant Professor in the school of Electrical Engineering and Computer Science at Oregon State University. His research interests include high-speed, low-power I/O interfaces, digital techniques to compensate for analog circuit imperfections, time-based data converter techniques, and power-management circuits.

Dr. Hanumolu currently serves as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING and on the Technical Program Committee of the IEEE Custom Integrated Circuits Conference and the Analog Signal Processing Technical Committee of the IEEE Circuits and Systems Society.



Gu-Yeon Wei (S'94–M'00) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1994, 1997, and 2001, respectively.

He is an Associate Professor of electrical engineering in the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA. After a brief stint as a Senior Design Engineer at Accelerant Networks, Inc., Beaverton, OR, he joined the faculty at Harvard as an Assistant Professor in January 2002. His research interests span several

areas, including mixed-signal circuits for wireline communications, co-design of circuits and architecture to address power and PVT variability in processors, and ultra-low-power computing and power electronics for flapping-wing microrobots.