A 0.5-μm CMOS 4.0-Gbit/s Serial Link Transceiver with Data Recovery Using Oversampling

Chih-Kong Ken Yang, *Student Member, IEEE,* Ramin Farjad-Rad, *Student Member, IEEE,* and Mark A. Horowitz, *Senior Member, IEEE*

***Abstract*—A 4-Gbit/s serial link transceiver is fabricated in a MOSIS 0.5-μm HPCMOS process. To achieve the high data rate without speed critical logic on chip, the data are multiplexed when transmitted and immediately demultiplexed when received. This parallelism is achieved by using multiple phases tapped from a PLL using the phase spacing to determine the bit time. Using an 8 : 1 multiplexer yields 4 Gbits/s, with an on-chip VCO running at 500 MHz. The internal logic runs at 250 MHz. For robust data recovery, the input is sampled at 3$\times$ the bit rate and uses a digital phase-picking logic to recover the data. The digital phase picking can adjust the sample at the clock rate to allow high tracking bandwidth. With a 3.3-V supply, the chip has a measured bit error rate (BER) of $<10^{-14}$.**

I. Introduction

THE increasing demand for data bandwidth in networking has driven the development of high-speed and low-cost serial link technology. Applications such as computer-to-computer or computer-to-peripheral interconnection are requiring gigabit-per-second rates either over short distances in copper or longer distances in fiber. CMOS technology is used increasingly over GaAs or bipolar technologies because of the development toward faster and faster devices. In 0.18-μm CMOS technology, the n-channel f_T is expected to equal or exceed that of the standard 0.5-μm GaAs process. While other technologies are limited in the number of transistors due to yield or power, CMOS technology allows implementation of complex digital logic enabling more integration of the back-end processing, lowering the cost. Recent development has shown CMOS capability to achieve Gbit/s data rates [1], [5], [6], [8], [11]. This work pushes NRZ signaling rates to the bandwidth limitations of the process technology and explores the issues involved.

The primary components of a link are the transmitter, the receiver, and the timing recovery circuits. Section II describes the overall architecture of the link. Because many of the circuits in the transmitter and receiver blocks have been previously discussed [1], this paper focuses on the timing recovery technique. Section III evaluates the impact of timing recovery on performance and compares two different timing recovery techniques: phase-locked loops versus oversampled phase picking. This chip implements a phase-picking algorithm that is discussed in Section IV. The measured performance of the entire transceiver chip is presented in Section V. Finally, some conclusions are drawn from these results in Section VI.

Manuscript received September 1997; revised December 3, 1997.

The authors are with the Center for Integrated Systems, Stanford University, Stanford, CA 94305-4070 USA.

Publisher Item Identifier S 0018-9200(98)02225-2.

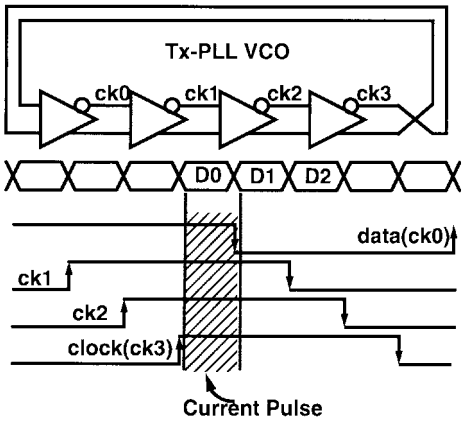

Fig. 1. Transmit architecture.

II. Architecture

A 0.5-μm CMOS technology is not fast enough to directly generate and receive a 4-Gbit/s stream (since the maximum ring oscillator frequency is <2 GHz). Instead, we use parallelism to reduce the performance requirements of each circuit. The transmitter generates the bit stream by an 8 : 1 multiplexer that multiplexes current pulses directly onto the output channel (Fig. 1). The receiver (Fig. 2) performs a 1 : 8 demultiplexing by sampling with a bank of input samplers. Similar to the transmitter, each sampler is triggered by individual clock phases. Furthermore, clock/data recovery is achieved by a 3$\times$ oversampling of each data bit. Thus, the receiver requires a total of 24 clock phases to support both the oversampling and the 1 : 8 demultiplexing. Various techniques exist for generating multiple clock phases [2], [3]. The receive side uses a six-stage ring oscillator (R_x-PLL) followed by phase interpolators to generate intermediate phases (ick[23 : 0]) between the ring oscillator edges (ck[11 : 0]) [1]. Similar to the R_x-PLL, eight different clock phases tapped from a four-stage ring oscillator (T_x-PLL) control the transmitter multiplexing.

A timing recovery circuit extracts the clock from the multiple samples per bit by finding the positions of the data transitions. Once the transitions are determined, a decision logic selects the samples furthest from data transitions (phase picking) as the received data byte. This approach is similar to

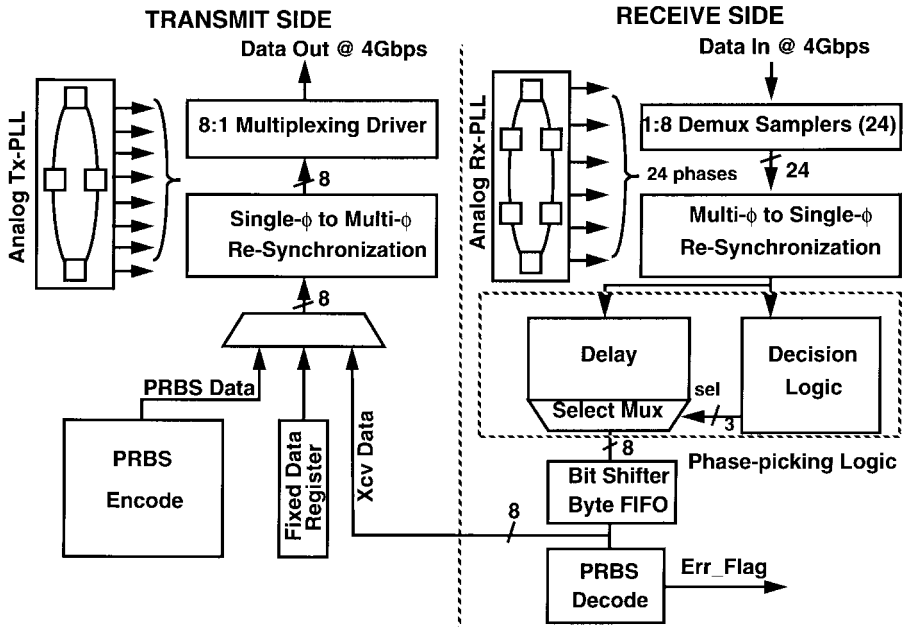

Fig. 3. Transceiver test-chip block diagram.

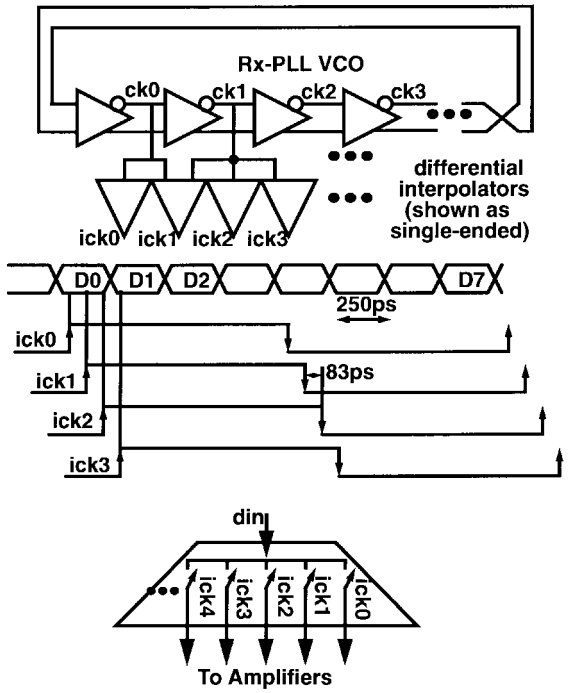

Fig. 2. Receive architecture.

what is done in UART's, and was first applied to a high-speed link by Lee *et al.* in [4].

Fig. 3 shows the full transceiver test-chip block diagram. Since the sampling clocks are different phases, the sampled results are resynchronized to a global clock. To facilitate the digital design, the on-chip data are further demultiplexed (2 : 1) to 250 MHz. Finally, in order to test the bit-error rate (BER), an on-chip parallel pseudorandom bit sequence (PRBS) encoder and decoder are used for a $2^7 - 1$ sequence. Serial data are commonly encoded with 8B10B coding which limits the run length to <5 consecutive zeros or ones. The PRBS sequence is a suitable substitute because it guarantees a maximum run length of 7. The transmitter can be optionally configured to transmit the PRBS sequence, a fixed sequence, or the received data for testing.

III. Timing Recovery

The goal of the timing recovery scheme is to maximize the timing margin—the amount that a sample position can err with the data still properly received. Errors that impact the timing margin can be classified into two sources: static phase error, and jitter (dynamic phase error). Fig. 4 illustrates the timing margin $t_{\text{margin}} = t_{bit} - t_{os} - t_{jc} - t_{jd}$ where t_{os} is the static sampling error, and t_{jd} and t_{jc} are the jitter on the data transition and the sampling clock. Since the sampling position is defined with respect to the data transition, jitter on both the clock and the data additively reduces timing margin. With ideal square pulses, as long as the sum of the magnitudes of the static and dynamic phase error is less than a bit time, the phase error does not impact signal amplitude. However, in a band-width limited system (for this work, due to the process technology), signal amplitude is lower with sampling phase error because the signals have finite slew rates. Correspondingly, this reduces the signal-to-noise ratio (SNR), hence impacting performance.

The amount of SNR degradation can be calculated based on the shape of the signal waveform. For static phase error, the SNR penalty is shown in Fig. 5 for a triangular signal waveform and a sinusoidal signal waveform. When the sample position phase offset is small, the sinusoidal waveform has a lower penalty than a triangular waveform due to the lower signal slew rate near the sample point.[1] For jitter, the SNR penalty is more complex to evaluate since it additionally depends on the statistics of the noise. For example, we can

[1] This penalty is only applicable to transitions.

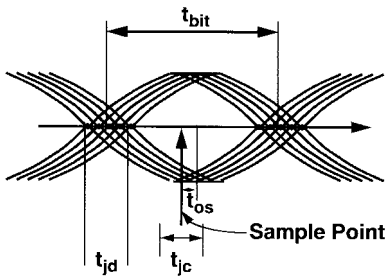

Fig. 4. Timing margin.

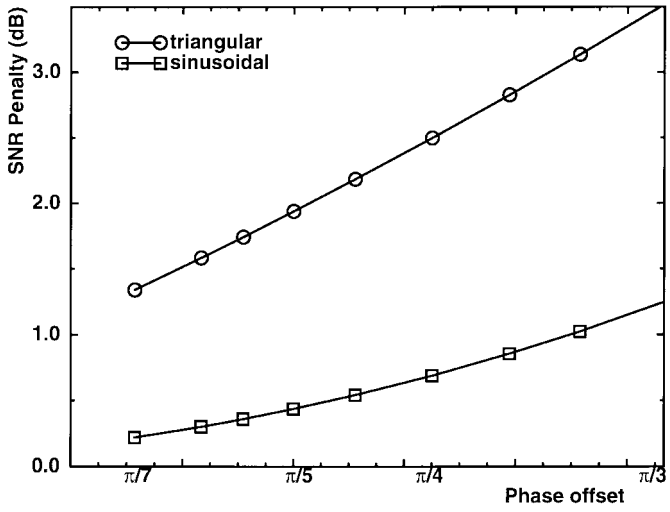

Fig. 5. SNR penalty for different phase offsets.

assume an idealized jitterless system with signal amplitude a and additive white Gaussian noise (AWGN) of standard deviation σ_A on the signal amplitude. In this system, we can determine the performance (BER) for various SNR [14]:

$$\text{ProbErr} = \int_a^\infty \frac{1}{\sqrt{2\pi\sigma_A^2}} \exp\left(-\frac{y^2}{2\sigma_A^2}\right) dy. \qquad (1)$$

This equation is plotted as the lowest dotted line in Fig. 6.

If we further assume jitter to be a AWGN as well, for a triangular waveform, the phase noise can be translated into amplitude noise using $\Delta A = a\Delta\phi/\pi$ (where the bit time spans 2π). Since the noise sources are additive, the probability of error can be simply expressed as

$$\text{ProbErr} = \int_a^\infty \frac{1}{\sqrt{2\pi\left[\sigma_A^2 + \left(\frac{a}{\pi}\sigma_\phi\right)^2\right]}} \cdot \exp\left\{-\frac{y^2}{2\left[\sigma_A^2 + \left(\frac{a}{\pi}\sigma_\phi\right)^2\right]}\right\} dy. \qquad (2)$$

Fig. 6 illustrate the BER versus amplitude SNR for various amounts of phase noise. The SNR penalty, as shown in the figure, increases at higher SNR because the phase noise eventually limits performance, a "BER floor." For a sinusoidal signal waveform (with a lower slew rate near the sample point), the behavior is similar, except with lower SNR penalty.

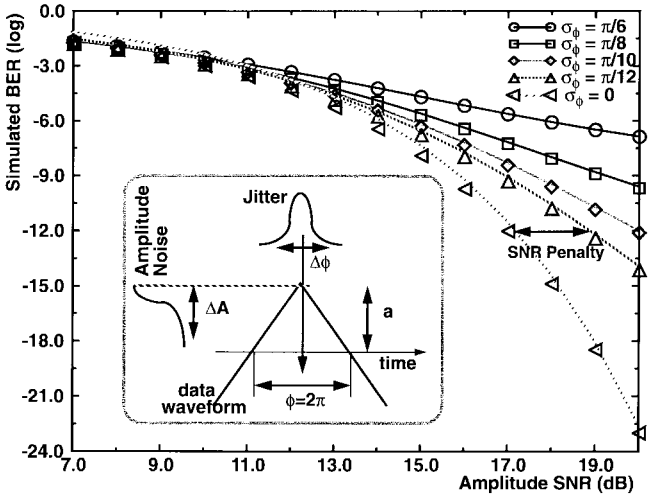

Fig. 6. BER versus SNR with various amounts of phase noise.

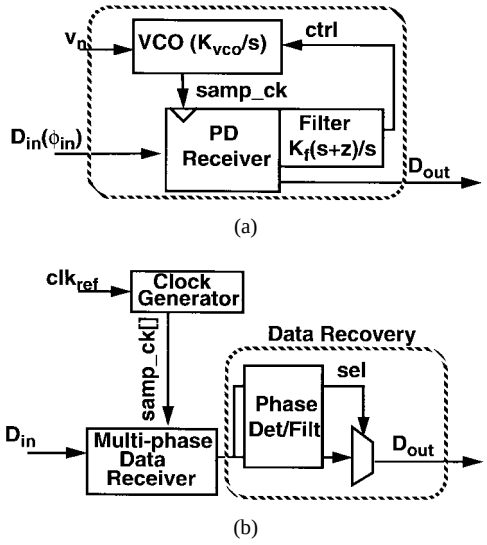

Fig. 7. Clock recovery architectures: (a) phase picking block diagram and (b) data/clock recovery architectures.

The amount of phase error and the jitter depends on the implementation of the clock recovery circuit. Two techniques are commonly used, a phase-locked loop (PLL) and a phase picker. A PLL employs a feedback loop that actively servos the sampling phase of an internal clock source based on the phase of the input [7]. Fig. 7(a) illustrates a common VLSI implementation using an on-chip voltage-controlled oscillator (VCO) as the clock source, and a charge pump following the phase detector to integrate the phase error. A phase picker, as shown in Fig. 7(b), oversamples each bit, and uses the oversampled information to determine the transition position (phase) of the data. Based on the transition information, the best sample is then selected as the data value (UART [10]). Each of the two architectures has a different tradeoff in terms of static phase error and jitter.

The static phase error of a PLL depends mainly on its phase detector design. Ideally, sampling at the middle of the bit window gives the maximum timing margin. However, if the sampler has a setup time, the middle of the effective bit

window is shifted by the setup time. Not compensating this shift causes significant static phase error. This error can be reduced by using the data samplers as the phase detector.[2] Additional phase error occurs due to inherent mismatches within the phase detectors and/or charge pump. Furthermore, any phase detector "dead band" (window in which the phase detector does not resolve phase information) limits the phase resolution, increasing the static phase error.

In a phase-picking architecture, the multiple samples per bit are used to find the transitions, effectively behaving as the phase detector. Sampler uncertainty limits the resolution of the transition detection. Sources of this uncertainty are sampler metastability window and data dependence of the sampler setup time. The uncertainty window for the sampler design used is <1/10 the bit time which does not impact performance significantly. More importantly, in this architecture, the phase information is quantized by the oversampling, causing a finite quantization error of 1/2 the phase spacing between samples. For a higher oversampling ratio, this static phase error is less, but it has a significant cost of increasing the number of input samplers, increasing the input capacitance, and hence limiting the input bandwidth. For a 3× oversampling system, the maximum static phase error is 1/6 the bit time.

In terms of jitter, a PLL tracks the phase of the input data with a tracking bandwidth limited by the stability of the feedback loop. The loop tracking is effectively a high-pass filter that rejects the phase noise of the input at lower frequencies. The noise not tracked appears as data jitter. Furthermore, because the PLL frequency source is an on-chip VCO, supply and substrate noise from on-chip digital switching can introduce additional jitter. The impact of these two sources is formulated for a second-order PLL in the following equation as the first and second terms:

$$\phi_{\text{err}} = \frac{\phi_{\text{in}} s^2}{s^2 + (K_f \cdot K_{\text{vco}})(s+z)} + \frac{v_n s K_n}{s^2 + (K_f \cdot K_{\text{vco}})(s+z)}. \tag{3}$$

Constants that determine the loop bandwidth in the equation are depicted in Fig. 7(a) with K_f (V/rad) the gain of the filter, z the stabilizing zero in the filter, and K_{vco} (rad-hertz/V) the gain of the VCO. v_n is the noise induced onto the VCO, and K_n is the sensitivity of the VCO to this noise. Thus, the total amount of "effective jitter" depends on the tracking bandwidth of the loop, the amount of supply and substrate noise, and the sensitivity of the loop elements to the noise. Because the feedback loop has a loop delay of at least one clock cycle, the bandwidth of the loop is often chosen to be <1/10 of the oscillation frequency for sufficient phase margin and stability. The delay makes tracking high-frequency phase noise ineffective because, if the phase error from on transition is independent of the phase of the next transition, correction based on the first transition's phase information could increase the phase error for receiving the next bit.

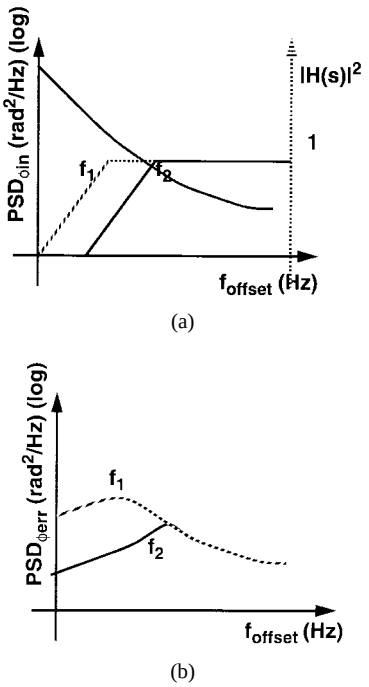

Fig. 8. Effect of tracking bandwidth on jitter.

The impact of different tracking bandwidth on jitter is illustrated in Fig. 8. The single sideband power spectral density (PSD) of an oscillator, such as the VCO of the transmitter, is shown to represent the phase noise in Fig. 8(a). Two hypothetical PLL's with different bandwidths (f_1, and f_2)[3] behave as high-pass filters that reject the lower frequency noise. Their transfer functions are overlaid in Fig. 8(a). The resulting phase error is shown in the PSD of Fig. 8(b). Note that this example excludes the additional noise from the phase-tracking circuit [second term of (3)]. The integral of the area beneath the curve is an indication of the amount of jitter [13] [σ_ϕ^2 for (2)]; thus, the phase noise of *Circuit I* is larger than that of *Circuit II.* Additionally, if a second-order PLL is not critically damped, the transfer function can exhibit peaking. This peaking accumulates phase noise at its loop bandwidth, increasing the noise.

For a phase picker, the sampling clocks experience similar jitter problems from supply and substrate noise since the phases for the oversampling are also generated from an on-chip VCO. The primary difference is the tracking bandwidth. A phase-picking system is a feedforward architecture (instead of feedback); thus, there are no intrinsic bandwidth limitations. The tracking rate depends on the rate at which new phase decisions are made, which in turn depends on the logic's cycle time. The importance of this fast tracking is that it can potentially track the accumulation of phase noise by the on-chip multiphase generator (PLL). We delay the data by the time to arrive at a decision so the corrections are applied to the appropriate bit (although with a latency overhead). However, the maximum phase change between two transitions must be

[2]This causes additional difficulties because such phase detectors can only determine if transitions are early or late. The control loop is "bang–bang" control instead of linear control, which is less stable, has inherent dithering, and requires additional frequency acquisition aid. Although a DLL (delay-line based PLL) [8] can be used to eliminate the stability and frequency acquisition problems, the phase spacing, when tapping phases from the buffer stages, is sensitive to the input clock's duty cycle and amplitude.

[3]The actual shape of the tracking transfer function $H(s)$ varies with implementation.

less than π, half the bit time, even if the peak-to-peak jitter can be much larger than a bit time. Changes ϕ_e greater than π are indistinguishable from a phase shift in the opposite direction, $\phi_e - 2\pi$.

Choosing between the two clock recovery systems depends on the system requirements and noise behavior. We chose a phase-picking architecture to explore the usefulness of the higher phase-tracking capability. In such VLSI implementations, supply noise can be significant enough for the peak-to-peak jitter to occupy a large fraction of the bit time, especially since a PLL accumulates jitter. For the 4-Gbit/s link, we chose a low oversampling ratio of 3× to maintain high input bandwidth and to keep the number of clock phases manageable (1 : 8 demultiplexing and 3× oversampling yields 24 phases). With a bit time of 250 ps, the phase-picking scheme[4] can track the noise of the on-chip multiphase generator (PLL) from both the transmit and receive sides to keep the total "effective jitter" below the 83-ps quantization spacing. One limitation of the phase-picker tracking is that the maximum rate of the tracking depends on the data transition density. Since the PRBS signal guarantees one transition per byte, the maximum tracking rate of one sample spacing every transition is fast (83 ps/2 ns).

Although the tracking rate is high, the maximum static phase error from the quantization is 41 ps (2% of the clock period, 8× bit time), causing an SNR penalty (Fig. 5). Whether or not a 3× oversampled phase-picking approach with higher tracking bandwidth than a PLL can achieve better performance with the larger static phase error depends on the amount of jitter induced by on-chip noise sources. If the lower SNR penalty from the lower jitter compensates the higher SNR penalty of larger static phase error, phase picking would be the better choice.

IV. Phase-Picking Algorithm and Implementation

The details of the phase-picking algorithm are illustrated in Fig. 9. Picking the center sample requires finding and tracking the bit boundaries. The decision logic first detects transitions by an XOR of adjacent samples, indicating the bit boundary to be in one of three possible positions. Fig. 10 shows an example of the boundary detection with a portion of a sampled stream. To find which of the three transition positions is the most likely bit boundary, transitions corresponding to the same bit boundary position are tallied. The position with the largest total determines the bit boundaries.

The decision logic makes a new decision per byte of data. In contrast to a higher order oversampling phase picker, the 3× oversampling limits the change of the selected sample position to one sample position per byte. To guarantee sufficient transitions for averaging any bit-to-bit variations of high-frequency noise (near the bit rate), the tally is across a sliding window of 3 bytes. The transitions are accumulated from the current byte, the previous byte, and the next byte (delaying the data allows the noncausal information) so that the decision is applied to the byte at the middle of the window. As a result

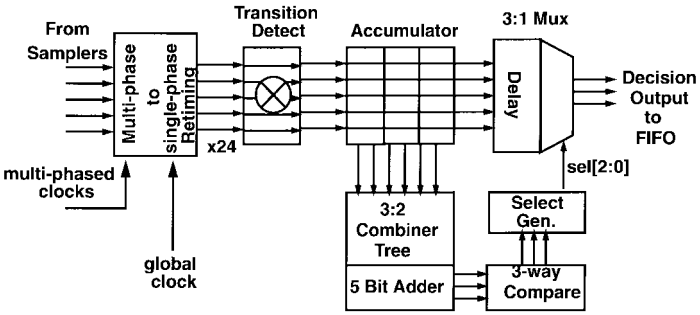

Fig. 9. Phase-picking algorithm block diagram.

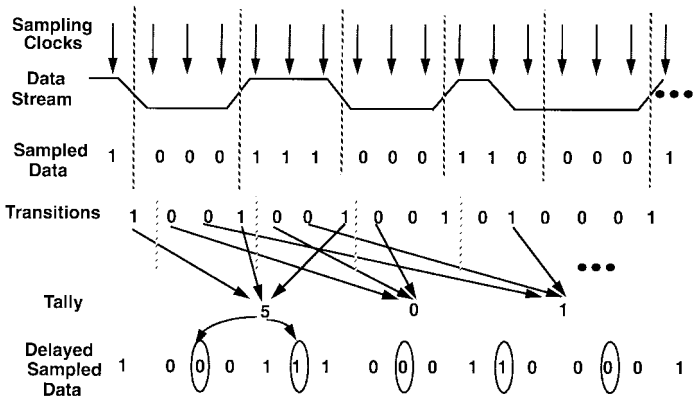

Fig. 10. Example of the phase-picking algorithm.

of the 3-byte sliding accumulation, the rate of phase change that the algorithm can track is slower than the maximum of 83 ps/2 ns. The algorithm picks the correct sample if the majority of the transition information within the 3-byte window (6 ns) indicates the correct phase. For example, if the input phase has a constant rate of change of <1 sample spacing per 3 ns (corresponding to a frequency difference of 4%), the transition information from >1.5 bytes of the 3-byte window would fall in the same phase quantization. Then the tally and compare would select the correct sample to track the phase change. This indicates a maximum phase-tracking rate of 83 ps/3 ns. The criterion of tracking both T_x and R_x-PLLs' accumulation is met because the VCO elements' supply noise sensitivity is $\sim$0.1%/% (percent of frequency change per percent of supply noise [1], [3]),[5] corresponding to 30 ps/3 ns for a 10% supply step, which is less than the tracking rate. If the phase change is slower than 83 ps/3 ns, the 3-byte accumulation offers some robustness by averaging any uncertainty in the transition detection due to high-frequency bit-to-bit noise. A smaller window of one byte can track phase faster, but has poorer performance without sufficient transitions within that byte to average the bit-to-bit variation. A larger window of 5 bytes (<83 ps/6 ns) would be too slow to track the T_x- and R_x-PLLs' phase accumulation under reasonable supply noise.

Once the transition position is determined, the middle sample within the bit boundaries is selected as the data.

[4] In our system, the oscillator is at 250 MHz so the PLL bandwidth is restricted to <25 MHz. This yields a 10× tracking rate difference between the two systems.

[5] Although the maximum phase error accumulation rate is based on the supply sensitivity of the VCO, the peak phase error depends on the loop bandwidth. The T_x-PLL and R_x-PLL generating the multiple clock phases have bandwidths of 15 and 5 MHz, respectively.

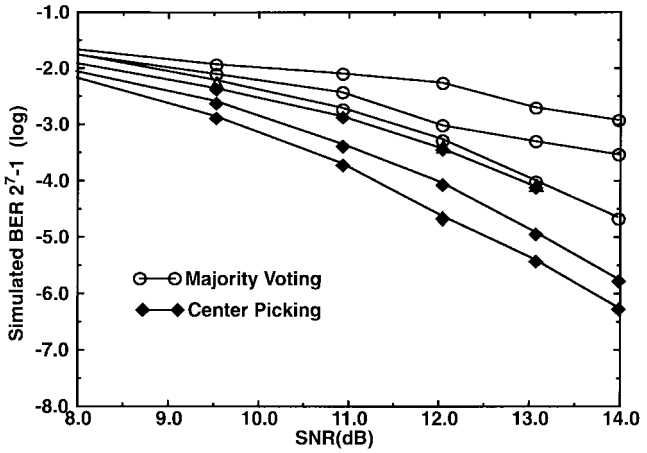

Fig. 11. Comparison between center picking versus majority voting.

The selection is implemented by multiplexers selecting the appropriate samples based on three select signals. In the case where no transitions are detected, the three select signals use previously stored values to maintain data through the multiplexers.

The actual algorithm for deciding the received data value from the oversampled information can be designed alternatively while still keeping the advantage of higher tracking bandwidth of a feedforward architecture. Instead of selecting the middle ("phase pick"), a simple alternative implementation is to take a majority vote based on the three sampled values. Fig. 11 shows the performance comparison. Majority voting works well with nonbandwidth-limited signals that have high-frequency noise because it averages the noise over many samples. In a bandwidth-limited system (low-pass filtered by the I/O RC time constant), it performs worse because at least one of the two nonmiddle samples is required to be valid, and the nonmiddle samples have a much higher probability of error.

Arbitration is required when two transition positions have equal counts. This occurs when two of the sample positions straddle the center of the bit and the third sampler samples at the transition. Picking either of the two straddling the center gives equivalent performance. More complex logic can be implemented by using the previous, current, and next cycles' comparison results to follow the direction of any phase transition. However, this only improves the performance by less than 1 dB.

If the peak-to-peak phase jitter is larger than one bit time, or if the transmitter and receiver operate at different frequencies, the tracking must allow bit(s) to overflow/underflow. For example, if the SEL[2:0] signal changes from 0–0–1 to 1–0–0, the selected sample of the first cycle corresponds to the same bit as the selected sample of the following cycle. This "underflow" condition must be appropriately handled by dropping one of the two samples. Typically, these samples are of the same bit, and thus have the same value. However, in the case where they are different, if phase movement changes directions (the SEL signal returns to 0–0–1) in the following cycle's decision, dropping the latter one gives a slight performance improvement. Similar to the "underflow"

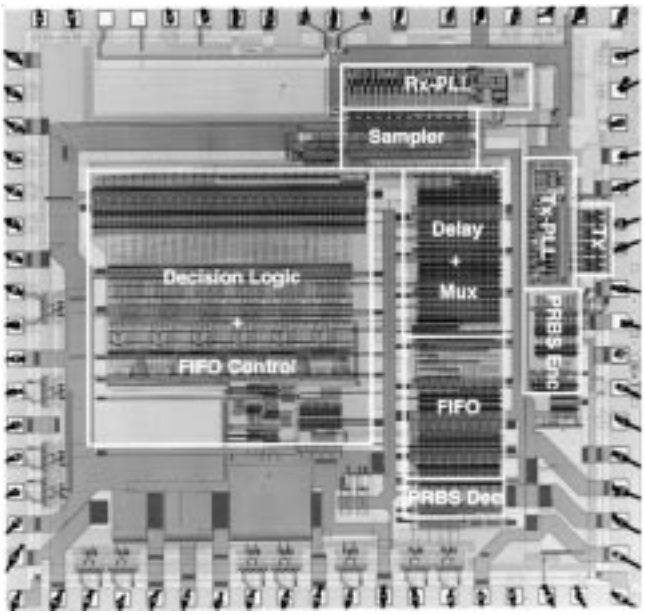

Fig. 12. Chip micrograph.

where only 7 bits are received, the opposite transition from 1–0–0 to 0–0–1 causes an "overflow," requiring an extra bit (9 bits total) to be stored. These conditions are handled by a bitwise FIFO built by shifting the input byte to accommodate the one extra/less bit. If the aggregate shift increases beyond 1 byte, a bytewise FIFO handles the overflow/underflow byte. The limited depth of the FIFO can only handle a finite number of byte overflow. If the application requires handling long streams of data with a slight frequency difference with the local reference clock, the local frequency can be corrected based on the phase information from the decision logic.[6]

V. Transceiver Experimental Results

The transceiver chip was implemented in a 0.5-μm CMOS process offered through MOSIS. The 3 mm $\times$ 3 mm die photo is shown in Fig. 12. The chip is packaged in a 52-pin CQFP package supplied by Vitesse Semiconductor which has internal power planes for controlled impedance. The size of the I/O bond pads are reduced to 70 μm $\times$ 70 μm to keep pad capacitance to a minimum because the capacitance would otherwise limit the I/O bandwidth. With an effective impedance at the I/O of 25 Ω (for a doubly terminated 50-Ω line), the total I/O capacitance can not exceed 4.5 pF for 4-Gbit/s operation without losing 10% of the bit height to the RC filtering. The 1:8 demultiplexing receiver and 8:1 multiplexing transmitter designs have capacitances of 2.2 and 1.2 pF, respectively, with 600 fF due to the pad and metal interconnects. An input time constant of $\sim$110 ps is estimated from measurements sweeping the reference voltage for a single-ended input pulse. The width of the pulse with a different reference voltage determines the time constant.

The performance of the link depends significantly on the I/O circuits. The minimum receivable amplitude of 50 mV was measured by using a fixed data pattern while changing

[6] This feature is not implemented as part of this test chip.

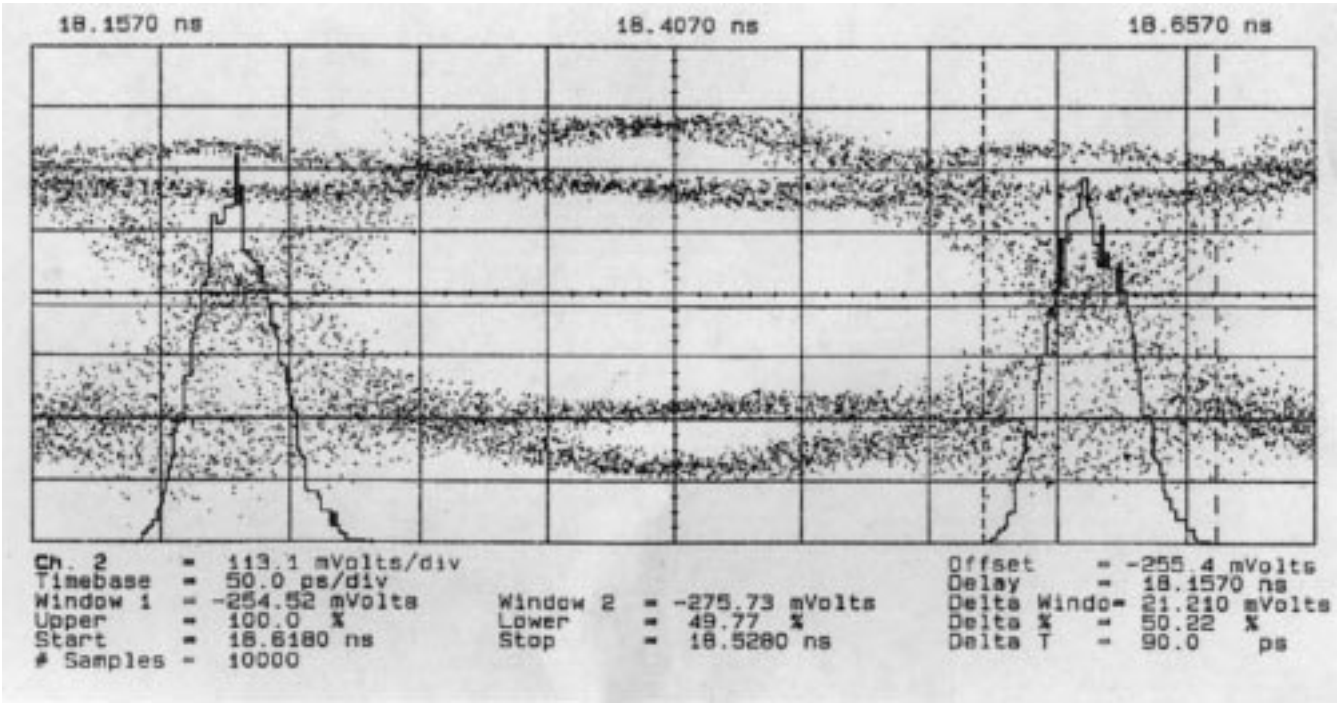

Fig. 13. Transmitter data eye.

the amplitude. This indicates the worst case input offset in the bank of samplers. The transmitter data eye at 3.0 Gbits/s is shown in Fig. 13 with the output driving a PRBS $2^7 - 1$ sequence. The measured data rate is limited by the triggering bandwidth of the oscilloscope. The maximum speed of the transmitter was 4.8 Gbits/s, and was limited by the maximum frequency of the ring oscillator used in the clock generation.

The multiple-phased clock generation (PLL) is crucial to the performance of the link because the phase spacing determines the bit time in the multiplexing/demultiplexing architecture, and the supply sensitivity and loop bandwidth determine the amount of jitter that needs to be tracked. Mismatches can cause one phase to be shifted with respect to the others. In the transmitter, the shift enlarges one bit, but reduces the next. By measuring the spacing between edges, we can evaluate the ability to match the phases tapped from the oscillators and interpolators [3]. The differential nonlinearity (DNL) of the phase spacing is plotted for the transmitter in Fig. 14 at various frequencies. The error is expressed as a percentage of the ideal bit time for all eight phase positions. While transmitting the PRBS pattern and using a trigger frequency of 1/8 the data rate (internal clock rate), these spacings are measured with a 20-GHz bandwidth digital oscilloscope by the width of each of the eight data-eye patterns.[7] If we use the data-rate frequency as a trigger instead of using a divided frequency, the data eye of Fig. 13 overlaps all eight of the bits. The overlaid histogram shows that the 333-ps bit time is degraded by 90 ps due to equal contributions from jitter and errors in the transmitter phase spacing.

The peak-to-peak variation, <±7% of the bit time, indicates very little degradation in bit width due to mismatches. The dominant cause of these bit-width variations is the V_T and K_P

[7]The measurement uncertainty is the DNL is ±2 ps.

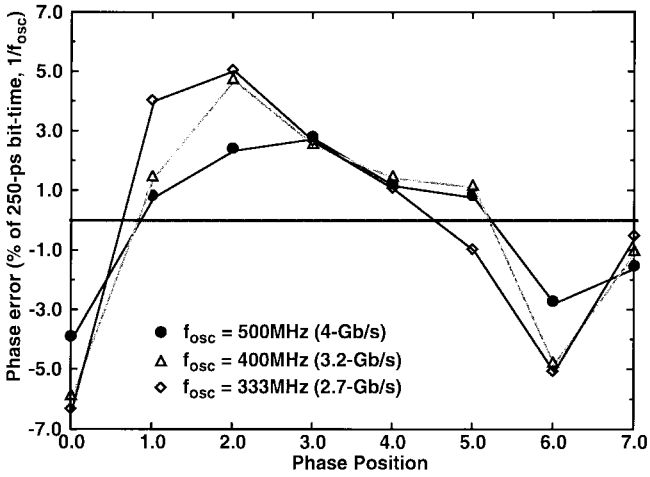

Fig. 14. Transmit-side DNL at various frequencies.

mismatches of the transistors in the clock generation circuits [12]. The increase in error with decreasing oscillation frequency, shown in Fig. 14, is an indication of these mismatches. The gate overdrive ($V_{\rm GS} - V_T$) is less at lower oscillation frequencies, making the phase spacing more sensitive to these mismatches. Fig. 15 shows the measurement of the DNL for four chips. The darker line indicates the average at each phase position. The variation of this average across phase positions potentially indicates some systematic error. However, because the average is over a sample size of only four chips, and the variation of the average is significantly smaller than the variation between chips, the random component is believed to be the dominant source of static phase spacing error.

Although a systematic component of the offset can also be expected from noise at any integer multiple of the oscillator

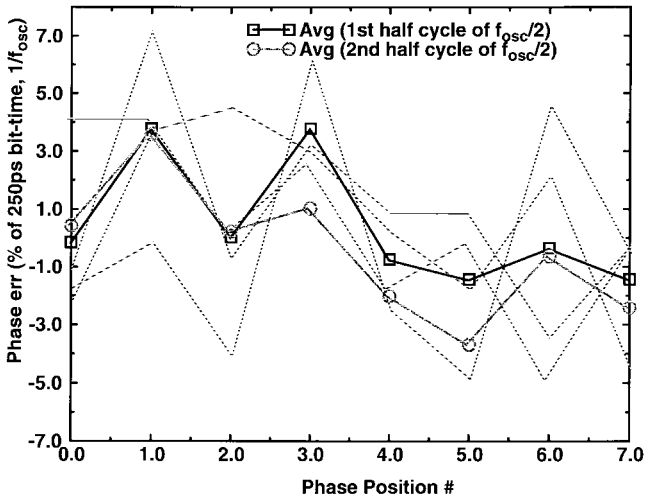

Fig. 15. Transmit-side DNL for four chips.

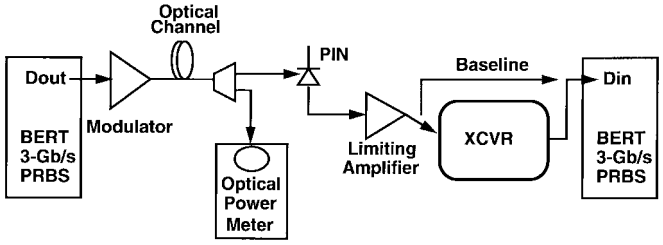

Fig. 16. BER testing configuration.

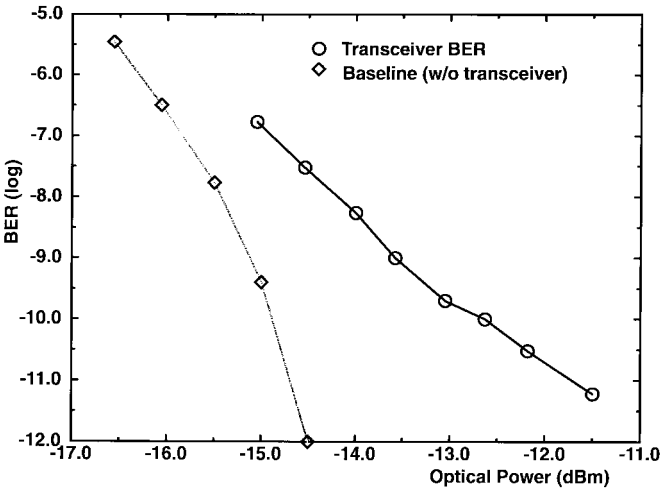

Fig. 17. Measured BER versus SNR.

frequency, it is not apparent in Fig. 15. Normally, noise such as substrate or supply noise at the same frequency as the oscillator would modulate the oscillator, causing a duty-cycle error which spreads the phases in the first half cycle and compresses the phases in the second half cycle. Since most of the digital logic clock on this chip switches at $f_{\rm osc}/2$ (250 MHz), this effect of the clock buffer switching on the 500 MHz oscillator would cause different phase spacings for two consecutive oscillator cycles. However, Fig. 15 shows that the average phase spacing errors from the second cycle is nearly the same as the first cycle, indicating that this coupling is negligible. Also, any systematic components from path mismatches (e.g., capacitive loading errors) are insignificant compared to the random source.

On the receive side, the DNL of the sample spacing is also measured, as was shown for a 0.8-μm process technology [1] to be <8% of the bit time. Receive clock phase spacing errors reduce the effectiveness of the oversampling by increasing the sample spacing, causing both increased static phase error and larger jitter.

Jitter in the transmitter can be measured by a outputting a fixed pattern and measuring the jitter on the data transition. We can also measure the sampling clock jitter by looking at the sampler output while sweeping a clean input transition. The window in which the sampler output is uncertain indicates the jitter with respect to the input. The supply sensitivitiy can also be measured by the increase in jitter due to induced supply noise with an internal switch that shorts between supply and ground. The sensitivities of the transmit and receive PLL's are 0.2 and 0.3 ps/mV, respectively, with a similar peak-to-peak quiescent jitter of 45 ps.

The BER testing is performed with two different configurations. The first measurement is by feeding the transmitted output directly back into the input. This yielded a BER of $<10^{-14}$. The second configuration is by placing the chip in a mock optical network (Fig. 16). A bit error rate tester (BERT) is used to generate the data pattern. The pattern is modulated onto a fiber-optic network. The optical power is measured by siphoning 1/10 of the total optical power. The optical signal is received and amplified by a avalanche photodiode (APD) followed by an amplifier. The output of the amplifier is either returned to the BERT for the baseline measurement, or sent into the chip configured in its transceiver mode. Because the BERT and optical amplifiers have a bandwidth limitation at 3 Gbits/s, the experimental results of this configuration are limited in data rate. As shown in Fig. 14, the phase spacing at lower frequencies is worse, so the performance is slightly worse than at 4 Gbits/s.

The BER versus SNR is plotted with SNR expressed in optical power showing both the baseline and the DUT with a 1.5-dB penalty at 10^{-9} BER (Fig. 17). The SNR penalty for not having the selected sample at the middle of the data eye is shown in Fig. 18. Because of the phase spacing errors on the receive side, the penalty shown here is worse than simulated. Since the quiescent jitter of the clock generation is smaller than the sample spacing (<83 ps), the phase tracking is not active. In order to test the effectiveness of the phase picking, voltage steps are induced on the supply, causing 250-ps jitter on both the T_x-PLL and R_x-PLL. While this causes the data eye to collapse, the receiver can still track this jitter and maintain BER $<10^{-9}$. Also, the transceiver is operated with the transmitter and receiver at different frequencies. The chip was able to track a frequency difference of 1 MHz with BER $<10^{-9}$.

Table I shows some additional performance measurements of the chip. The total power dissipated is 1.5 W, with 1/3 from the clock generation and 1/3 from the receive-side logic. The minimum amplitude that can still maintain $<10^{-9}$ BER is 90

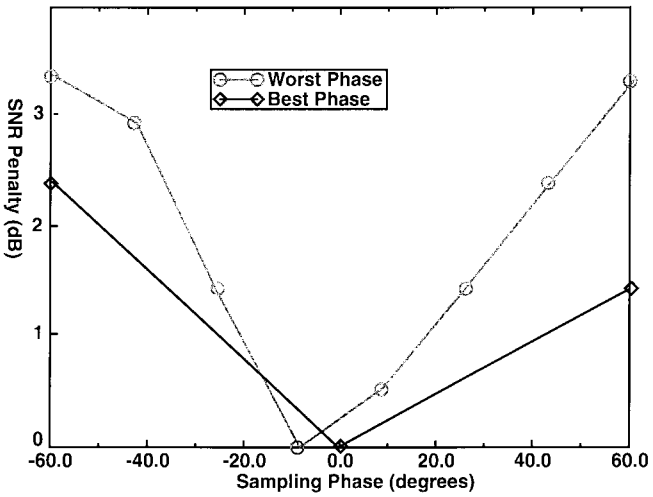

Fig. 18. Measured BER at various sampling phase.

TABLE I
TEST-CHIP PERFORMANCE

Supply Voltage	**2.7-4.0**
Max. Transmit Rate @3.3V	**4.8Gb/s**
Max. Receive Rate @3.3V	**4.3Gb/s**
Max. Frequency Difference	**1MHz (<10^{-9} BER)**
Power @ 4Gb/s- Total	**500mA**
Analog(2 PLL)	**150mA**
Input Samp,Rcv Logic	**25mA, 170mA**
Transmitter, Xmt Logic	**50mA, 10mA**
Parallel Data Driver	**90mA**
Min. Input Amplitude	**170mV <10^{-9} BER**

mV with an internal eye height of 65 mV. The 24 mV of amplitude noise is primarily due to ringing from the package inductance and on-chip output capacitance at the transmitter.

VI. CONCLUSION

Very high data rates are achievable in CMOS technologies by making extensive use of parallelism. Using an 8:1 demultiplexing at the input and a 8:1 multiplexing output transmitter, we achieved a 4-Gbit/s transceiver while keeping all internal signals <500 MHz in a 0.5-μm process technology. The fundamental limitations of this approach are the I/O capacitance (increased due to the parallelism), the sampler uncertainty, and the phase position accuracy of the multiple clock phases.

Provisions were made in this design to handle very large jitter accumulation of 83 ps/3 ns by a fast phase-picking algorithm. The effectiveness of this architecture critically depends on the jitter characteristics. Although a CMOS PLL can potentially exhibit this large jitter due to supply noise, the measured jitter while operating this transceiver is only 50 ps. This jitter is measured in a realistic noise environment because of the presence of significant digital switching noise from the large digital phase picker that can couple onto the VCO elements. Since the jitter is less than the quantization error, the advantage of the phase picking is only apparent when additional noise is induced. This low accumulated jitter implies that the lower tracking bandwidth of a PLL-based clock recovery circuit can potentially perform equally. The design of such a system is nontrivial, and still has challenges in maintaining small static phase offsets. However, since the phase picking has significant hardware overhead in the extra number of input samplers and large digital processing, a PLL would potentially offer similar performance with lower area and power.

ACKNOWLEDGMENT

The authors would like to thank S. Sidiropoulos, B. Amrutur, K. Falakshahi, Vitesse Semiconductor, Prof. T. Lee, Prof. L. Kazovsky, and their research groups for invaluable discussions and assistance.

REFERENCES

[1] C.-K. Yang and M. Horowitz, "A 0.8 μm CMOS 2.5 Gbps oversampling receiver and transmitter for serial links," *IEEE J. Solid-State Circuits*, vol. 31, Dec. 1996.
[2] C. Gray *et al.*, "A sampling technique and its CMOS implementation with 1-Gb/s bandwidth and 25 ps resolution," *IEEE J. Solid-State Circuits*, vol. 29, Mar. 1994.
[3] J. Maneatis and M. Horowitz, "Precise delay generation using coupled oscillators," *IEEE J. Solid-State Circuits*, vol. 28, pp. 1273–1282, Dec. 1993.
[4] K. Lee *et al.*, "A CMOS serial link for fully duplex data communications," *IEEE J. Solid-State Circuits*, vol. 30, pp. 353–364, Apr. 1995.
[5] A. Fiedler *et al.*, "A 1.0625Gb/s transceiver with 2×-oversampling and transmit signal pre-emphasis," in *ISSCC'97 Dig. Tech. Papers*, Feb. 1997, pp. 238–239.
[6] A. Widmer *et al.*, "Single-chip 4 × 500 Mbaud CMOS transceiver," *IEEE J. Solid-State Circuits*, vol. 31, pp. 2004–2014, Dec. 1996.
[7] F. M. Gardner, *Phaselock Techniques*, 2nd ed. New York: Wiley, 1979.
[8] W. Dally and J. Poulton, "A tracking clock recovery receiver for 4-Gb/s signaling," in *Hot Interconnect97 Proc.*, Aug. 1997, p. 157.
[9] S. Sidiropoulos and M. Horowitz, "A semi-digital DLL with unlimited phase shift capability and 0.08–400MHz operating range," in *ISSCC'95 Dig. Tech. Papers*, Feb. 1995, pp. 332–333.
[10] J. E. McNamara, *Technical Aspects of Data Communication*, 2nd ed. Bedford, MA: Digital, 1982.
[11] S. Kim *et al.*, "An 800Mbps multi-channel CMOS serial link with 3× oversampling," in *IEEE 1995 CICC Proc.*, Feb. 1995, p. 451.
[12] M. J. Pelgrom, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, p. 1433, Dec. 1989.
[13] J. A. Crawford, *Frequency Synthesizer Design Handbook.* Boston, MA: Artech House, 1994.
[14] J. Proakis, *Communication Systems Engineering.* Englewood Cliffs, NJ: Prentice-Hall, 1994.

Chih-Kong Ken Yang (S'93) received the B.S. and M.S degrees in electrical engineering from Stanford University, Stanford, CA, in 1992.

He is currently pursuing the Ph.D. degree at Stanford University in the area of circuit design for high-speed interfaces.

Mr. Yang is a member of Tau Beta Pi and Phi Beta Kappa.

Ramin Farjad-Rad (S'95) was born in Tehran, Iran, in 1971. He received the B.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, in 1993 and the M.Sc. degree in electrical engineering from Stanford University, Stanford, CA, in 1995, where he is currently a Ph.D. candidate in electrical engineering.

He worked at SUN Microsystems Laboratories, Mountain View, CA, on a 1.25-Gbit/s serial transceiver for the fiber channel standard during the summer of 1995. Over the summer of 1996, he worked at LSI Logic, Milpitas, CA, where he examined different multi-Gbit/s serial transceiver architectures.

Mr. Farjad-Rad holds one U.S. patent, and is also the Bronze Medal Winner of the 20th International Physics Olympiad, Warsaw, Poland.

Mark A. Horowitz (S'77–M'78–SM'95) received the B.S. and M.S. degrees in electrical engineering from MIT in 1978, and the Ph.D. degree from Stanford University, Stanford, CA, in 1984.

He is the Yahoo Founders Professor of Electrical Engineering and Computer Science at Stanford. His research area is in digital system design, and he has led a number of processor designs including MIPS-X, one of the first processors to include an on-chip instruction cache, TORCH, a statically scheduled, superscalar processor, and FLASH, a flexible DSM machine. He has also worked on a number of other chip design areas including high-speed memory design, high-bandwidth interfaces, and fast floating point. In 1990, he took a leave from Stanford to help start Rambus Inc., a company designing high-bandwidth memory interface technology. His current research includes multiprocessor design, low-power circuits, memory design, and high-speed links.

Dr. Horowitz is the recipient of a 1985 Presidential Young Investigator Award and an IBM Faculty Development Award, as well as the 1993 Best Paper Award from the International Solid-State Circuits Conference.