A Digital Clock and Data Recovery Architecture for Multi-Gigabit/s Binary Links

Jeff L. Sonntag and John Stonick, *Member, IEEE*

***Abstract*—In this tutorial paper, we present a general architecture for digital clock and data recovery (CDR) for high-speed binary links. The architecture is based on replacing the analog loop filter and voltage-controlled oscillator (VCO) in a typical analog phase-locked loop (PLL)-based CDR with digital components. We provide a linearized analysis of the bang-bang phase detector and CDR loop including the effects of decimation and self-noise. Additionally, we provide measured results from an implementation of the digital CDR system which are directly comparable to the linearized analysis, plus measurements of the limit cycle behavior which arises in these loops when incoming jitter is small. Finally, the relative advantages of analog and digital implementations of the CDR for high-speed binary links is considered.**

***Index Terms*—Clock and data recovery (CDR), clock recovery, digital phase-locked loop (DPLL), jitter.**

I. Introduction

MULTI-GIGABIT per second (Gbps) serial binary links are fast replacing traditional parallel data links in many applications. Examples include Peripheral Component Interconnect (PCI) moving towards PCIexpress and Advanced Technology Attachment (ATA) moving towards Serial ATA (SATA). Additionally, there exist many other applications with multi-Gbps serial links such as IEEE 802.3ae XAUI, FibreChannel and RapidIO. Thus, the problem of architecting an effective clock and data recovery (CDR) for multi-Gbps rates is becoming increasingly common. At the same time, the trend is for the serial link to become a peripheral function at the edge of a large application specific integrated circuit (ASIC), rather than the core function of a mixed-signal application specific standard product (ASSP). For this reason, effective solutions must be extremely low in power, implementable in the cheapest of digital process technologies, insensitive to supply noise, and easily ported across multiple technologies and speed targets.

In this paper, we present and discuss a general architecture that meets these criteria. In Section II, we present a small-signal model and analysis for CDRs with bang-bang phase detectors. In Section III, we describe and analyze the digital CDR. In Section IV, we present measured results that corroborate the analysis of Section III. Finally in Section V, we summarize the results and describe the advantages of digital CDRs over analog implementations.

Manuscript received December 4, 2005; revised March 1, 2006.

J. L. Sonntag was with Synopsys, Inc., Hillsboro, OR 97124 USA. He is now with Silicon Laboratories, Beaverton, OR 97006 USA.

J. Stonick is with Synopsys, Inc., Hillsboro, OR 97124 USA (e-mail: jts@synopsys.com).

Digital Object Identifier 10.1109/JSSC.2006.875292

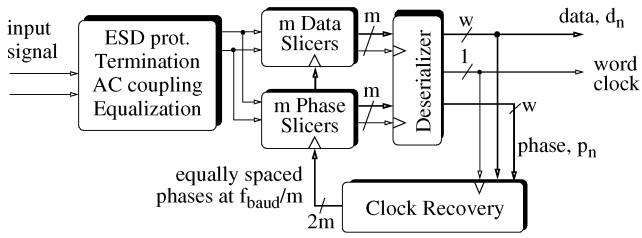

Fig. 1. Typical receiver and CDR.

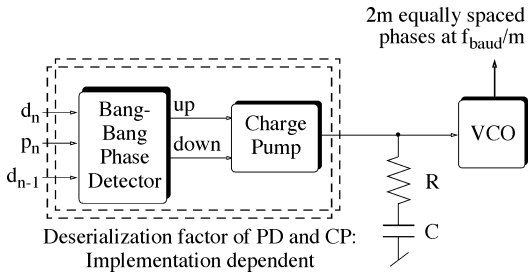

Fig. 2. Analog clock recovery unit.

II. General CDR Small-Signal Model

A. Typical Receiver and CDR

To identify (and limit) the scope of the problem, we refer to the block diagram of a typical high-speed receiver, illustrated in Fig. 1. We observe that receivers at these speeds typically comprise a bank of slicers to sample the incoming signal at a number of equally spaced phases, some type of deserialization and a clock recovery unit. The focus of this paper will be on the clock recovery unit. A common CDR uses an analog phase-locked loop (PLL), including a bang-bang phase detector, charge pump loop filter (CPLF) and a voltage-controlled oscillator (VCO), as shown in Fig. 2 [1]–[3].

Some analog CDR implementations run the phase detector and charge pump at the baud rate, while others deserialize to varying degrees before summing at the loop filter.

B. Bang-Bang Phase Detector

The bang-bang phase detector is common to many analog CDRs and the digital CDR proposed here. Lower speed transceivers (operating where the baud interval is much larger than multiple gate delays) often use phase detectors which produce more linear responses. In the multi-Gbps regime, the advantages (simplicity and accuracy) of the bang-bang phase detector overcome the drawbacks of nonlinearity and self-generated (also known as hunting) noise.

TABLE I
BANG-BANG PHASE DETECTOR LOOK UP

d_{n-1}	p_n	d_n	DECISION
-1	-1	1	EARLY (-1)
1	1	-1	
-1	1	1	LATE (+1)
1	-1	-1	
-1	X	-1	NO DECISION (0)
1	X	1	

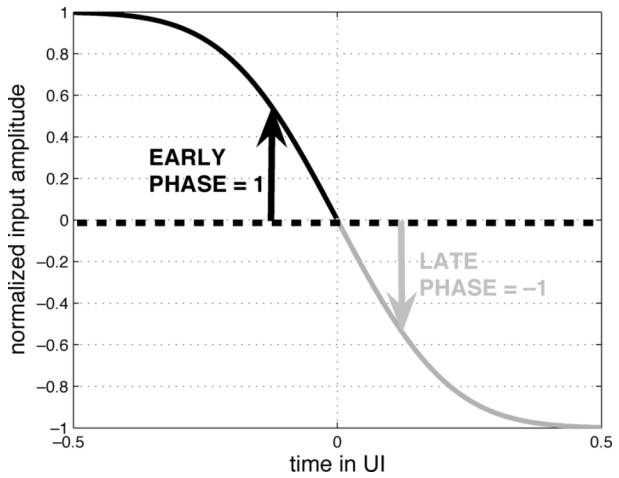

Fig. 3. Early–late phase sampling.

A bang-bang phase detector comprises a bank of slicers that sample the received signal at the nominal data and phase sampling points and some digital logic to produce early–late decisions based upon the relationship between the data and the phase samples. It produces a nonzero output of either $+1$ or -1 for data transitions and a zero output for nontransitions. The digital logic of a bang-bang phase detector operates as follows: For any data transition, if the phase bit agrees with previous the data bit the phase sample is early, if the phase bit agrees with next data bit the phase sample is late.

Table I provides the complete phase error decoding table, where the data before the phase sample is d_{n-1}, the phase sample is p_n and the data after the phase sample is d_n. This is graphically depicted in Fig. 3 for a phase sample that is being taken between a $+1$ bit and a -1 bit.

In Table I row two corresponds the case to the black phase sample in Fig. 3 and row four corresponds to the gray phase sample in Fig. 3.

C. Linearizing the Bang-Bang Phase Detector

Although it has been done in other papers [4], [5], we include an analysis of a bang-bang phase detector both for completeness and to perform the analysis in the terminology that we will be using throughout the paper.

First, consider an ideal comparator with an input signal that has a mean value of $V_{\rm DC}$ added to which is Gaussian noise with a standard deviation of σ_v. The ensemble average of the output is readily shown to be $1-2\cdot Q(V_{\rm DC}/\sigma_v)$, where $Q(x)$ is the integral of the tail of a unit variance Gaussian probability density function from x to ∞. This response is illustrated in Fig. 4.

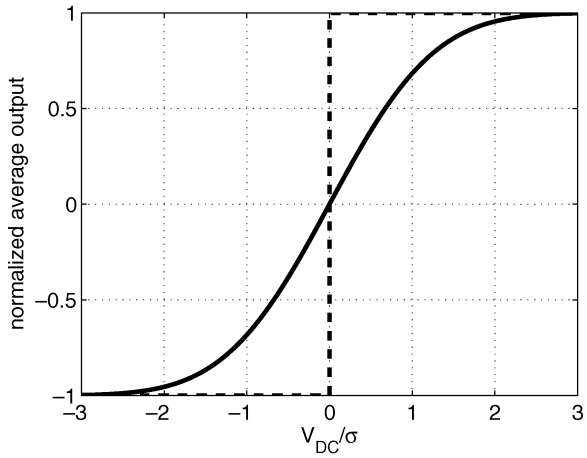

Fig. 4. Average output of ideal slicer, as function of mean input.

For small values of $(V_{\rm DC}/\sigma_v)$, this may be approximated as a straight line:

$$\text{mean(slicer output)} = \frac{V_{\rm DC}\sqrt{2}}{\sigma_v\sqrt{\pi}}. \tag{1}$$

Equation (1) is a voltage-to-voltage transfer function based upon an ensemble average. However, we are ultimately interested in what happens to the output of the comparator when the input is a random process, i.e., when it is used as a bang-bang phase detector. Consider the comparator in the presence of a small phase error, e. The mean sliced voltage (during a rising transition) is proportional to e, and to the slope of the signal at the center of the transition $(slope)$. Therefore, we can find the average output produced by a bang-bang phase detector in response to the phase error e by replacing V_{DC} in (1) with $e\cdot slope$:

$$\text{mean(slicer output)} = \frac{e\cdot slope\sqrt{2}}{\sigma_v\sqrt{\pi}}. \tag{2}$$

The linearized gain (time averaged mean) of the phase detector is derived by recognizing that rising and falling edges make equal contributions to the output and that (for random data patterns), the transition density is 1/2. The slope of the signal as it passes through the zero crossings depends upon the channel bandwidth and equalization. Assuming good equalization and a peak to peak signal amplitude of 2A, a good upper bound on the slope is $A/2$ (Volts/radian). This results in

$$K_{\rm PD} = \frac{A}{2\sigma_v\sqrt{2\pi}}\ (\text{units of radian}^{-1}). \tag{3}$$

At the zero crossing, additive voltage noise is indistinguishable from jitter. Using this equivalence $(\sigma_v = slope\cdot\sigma_j)$, the slope terms cancel and the small-signal gain of the phase detector can be written as

$$K_{\rm PD} = \frac{1}{\sigma_j\sqrt{2\pi}}\ (\text{units of radian}^{-1}). \tag{4}$$

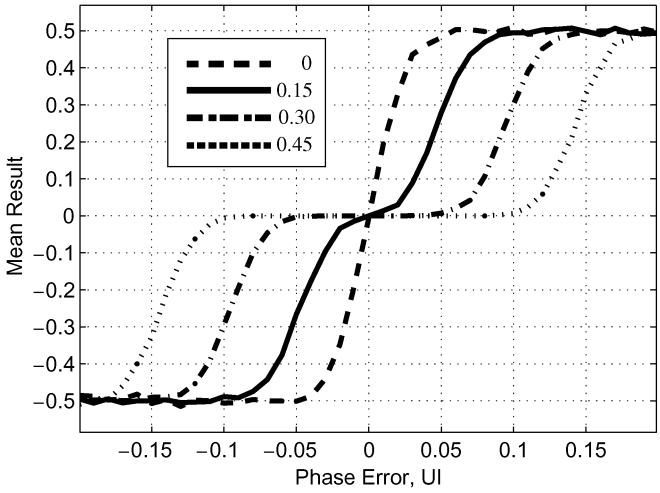

Fig. 5. Simulated phase detector transfer functions. Phase slicer offsets are: $\{0, 0.15, 0.30, 0.45\} * \mathrm{A(UI)}$; $\sigma_v = 0.06 * \mathrm{A(UI)}$.

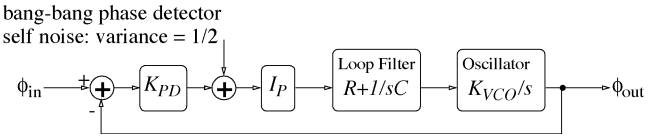

Fig. 6. Linearized model of analog clock recovery unit.

While a receiver may typically operate unimpaired when the offset of the data slicers is not small compared to the eye opening, offsets in the phase slicers produce substantially nonideal results. These offsets result in a difference in the desired sampling phase for rising and falling edges. Depending on the size of the offset relative to the eye opening and noise in the crossing times, this can result in a substantially reduced value of K_{PD}, or even a "dead zone" in the phase detector's transfer function. Such a dead zone leads to a reduction in jitter tolerance as the selected phase wanders within the dead zone. A family of simulated phase detector transfer functions with varied offset is shown in Fig. 5.

Note that in practice, much of the noise present at the signal zero crossings is not additive or Gaussian. Gaussian jitter sources in the transmitter, reference clocks, and receiver are present, and (due to the effects of jitter on a sloped signal) can reasonably be treated as described in this subsection. In many situations, substantial deterministic jitter (DJ) sources are present, generated both from nonideal transmitters and from uncanceled inter-symbol interference (ISI) arising in the channel. When such error sources are modelled, the standard deviation of their non-Gaussian distributions may still be used in (3).

D. Linearized Small-Signal Model

There have been many excellent papers on the design and analysis of this type of CDR system [1]–[5]. A linearized model is shown in Fig. 6.

The loop gain for the linearized system is given by

$$L(s) = I_P K_{\mathrm{PD}} \left(\frac{K_{\mathrm{VCO}}}{s}\right)\left(R + \frac{1}{sC}\right). \tag{5}$$

E. Self-Noise of the Bang-Bang Phase Detector

The self-noise of the bang-bang phase detector arises due to the fact that the output is full scale for every data transition. The result is that the standard deviation of the self-noise jitter is $\sqrt{2}$. By pushing the insertion point back to the input [and scaling by $1/K_{\mathrm{PD}}$ making use of (4)], we can consider the self-noise to be a broadband jitter source at the input the phase detector with a standard deviation of $\sigma_j \cdot \sqrt{\pi}$.

The effect of the self-jitter on the system can and must be controlled by limiting the bandwidth of the CDR, and retaining little of the self-jitter power in the passband.

Strangely enough, the input-referred jitter induced by self-noise is proportional to the jitter present at the phase detector input. In the limit as input jitter is reduced, K_{PD} rises and self-jitter falls until the CDR becomes small-signal unstable. This results in limit cycle behavior which prevents the jitter present at the phase detector from approaching zero.

III. Proposed System

In the previous section, we provided a general system overview that included a significant discussion of the phase detector. In this section, we build upon that previous discussion as we introduce the proposed digital CDR. The general architecture that is proposed is similar to those in [7]–[10] and is precisely that which we used in [11]. The purpose of this paper is to focus on the general architectural principles and issues that need to be understood in realizing a digital phase-locked loop (DPLL)-based CDR, rather than circuit-level details.

The goal of the proposed architecture is to overcome the limitations of the analog PLL of Fig. 2 by replacing each component with digital equivalents.

The decimation block is used to reduce the (effectively) baud rate phase error samples to a rate compatible with high-resolution digital signal processing. While this rate may not always match the byte rate, we will designate it as the *word* rate. Operating at this lower rate has a cost (latency), but makes the required computations both possible and power and area inexpensive. Decimation is described in Section III-B.

The digital-to-phase converter (DPC) is used as a generic term for any (typically mixed-signal) circuit which uses a multi-bit digital control bus to control the phase of a set of output clocks. For most applications, it is necessary that the DPC has infinite range, being capable of producing a continuous phase ramp (representing a frequency offset) in response to a repeatedly overflowing phase integrator. DPC circuits have been implemented using analog and digital delay-locked loops (DLLs), phase mixers/interpolators, and PLLs [7]–[13]. Implementation of the DPC is not covered in this paper.

It may seem pointless to be replacing an analog PLL used for CDR with a mostly digital implementation which may still include an analog PLL. However, note that the analog PLL used directly for CDR has severe disadvantages relative to the analog PLL which may be embedded in the DPC. The first PLL must use a jittered and ISI-impaired signal as a reference, and a noisy bang-bang phase detector, each of which forces the use of a low loop bandwidth and subsequent poor ability to reject thermal noise and power supply injected noise. The PLL embedded in the CDR, on the other hand, enjoys a precise reference clock as an input, operating at a convenient speed; a linear phase/frequency detector and a high loop bandwidth can be used, making possible the combination of low power, low jitter, and low area.

Regardless of the implementation of the DPC, production of an infinite range of output phase from a finite range of control inputs is achieved by "wrapping". After an input control change which corresponds to an integer multiple of 2π, the inputs and the output have returned to the same condition; it is only in the history of the signal produced that the change in output phase can be seen. While the linearity of the digital to phase transfer function could be poor in some DPC designs, or process, voltage, and temperature (PVT) sensitive, the average gain of the DPC cannot be other than 2π radians per the known control range which corresponds to a complete return (wrap-around) to the same input control state. Thus, the average gain of conceivable DPC implementations (the only mixed-signal component of the proposed system) is PVT insensitive.

A. Analogy to Analog Implementation

To illustrate the similarities between the analog and digital approaches, we map the VCO and CPLF using a backwards difference substitution $s = (1 - z^{-1})/T$. The result is the following:

$$\left(\frac{K_{\rm VCO}}{s}\right)\left(I_P R + \frac{I_P}{sC}\right) \rightarrow \left(\frac{TK_{\rm VCO}}{1-z^{-1}}\right)\left(I_P R + \frac{TI_P}{(1-z^{-1})C}\right). \tag{6}$$

Equation (6) offers an equivalent view of the basic architecture. In realizing this equation it is simplified to the following:

$$\left(\frac{K_{\rm VCO}}{s}\right)\left(I_P R + \frac{I_P}{sC}\right) \rightarrow \left(\frac{K_{\rm DPC}}{1-z^{-1}}\right)\left(phug + \frac{frug}{(1-z^{-1})}\right) z^{-N_{\rm EL}}. \tag{7}$$

By comparing (6) and (7) we can see that the **ph**ase **u**pdate **g**ain $(phug)$ models the proportional path gain in the CPLF, that the **fr**equency **u**pdate **g**ain $(frug)$ models the integral path gain in the CPLF, and that $K_{\rm DPC}$ models the gain of the VCO, $K_{\rm VCO}$. The extra term, $z^{-N_{\rm EL}}$, is included to model the pipe stages of latency required for implementation, delay through the control path of the DPC and delay through the deserialization process. If the latency, $T_{\rm word} \cdot N_{\rm EL}$, is not controlled and is allowed to approach $1/(4 \cdot f_{\rm unity\ gain})$ a severe loss in phase margin occurs. Design techniques which minimize $N_{\rm EL}$ must be used or the bandwidth of the loop must be reduced.

In realizing a CDR based upon the architecture of Fig. 7, there exist many important design tradeoffs in balancing power and performance. Much of the issue involves widening the bus to use slower clocks to save power at the cost of latency. In the following sections we discuss some of these tradeoffs while providing more detail on the blocks in Fig. 7.

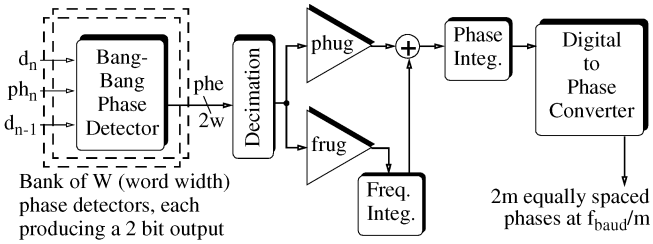

Fig. 7. Digital PLL architecture.

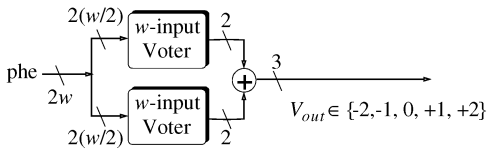

Fig. 8. Faster decimation with voting.

B. Decimation by Voting

In Fig. 7, each bang-bang phase detector produces a phase error at the data rate$/w$. These detectors operate in parallel on the deserialized data. The output of the bank of bang-bang detectors is thus a parallel group of w phase errors. Decimation comprises the function of producing a single, multi-bit description of the w deserialized phase errors. The most straightforward approach to the decimation operation is by the use of a finite impulse response (FIR) boxcar filter. All of the w deserialized 2-bit phase error samples are added together, producing a single multi-bit result per word clock cycle.

However, summing so many addends in a single clock cycle may be difficult, and there are substantial advantages in reducing latency in the DPLL. We have found that faster implementations are possible which start by voting across a modest number $(w/2)$ of phase error samples, as illustrated in Fig. 8.

Decimation via boxcar filter produces a DC gain, K_b, corresponding to the decimation factor, w. Decimation via voting has a reduced gain, $K_{\rm V}$, which can be determined through simulation. Clearly, a concern with using a nonlinear function such as voting is how much it will increase the input-reflected noise. However, simulations show that for voting across groups of modest size, the input-referred noise is increased by less than 1 dB.

Fig. 9 illustrates the result of a simulated comparison of a bank of four bang-bang phase detectors decimated both with a boxcar filter and via voting. In this example, decimation by voting across four inputs had a gain which was reduced to 54% relative to the decimation via boxcar filtering. Naturally, this gain reduction factor is dependent upon the population size across which voting is done. In conclusion, to perform analysis and fully understand the impact of voting, it is important to

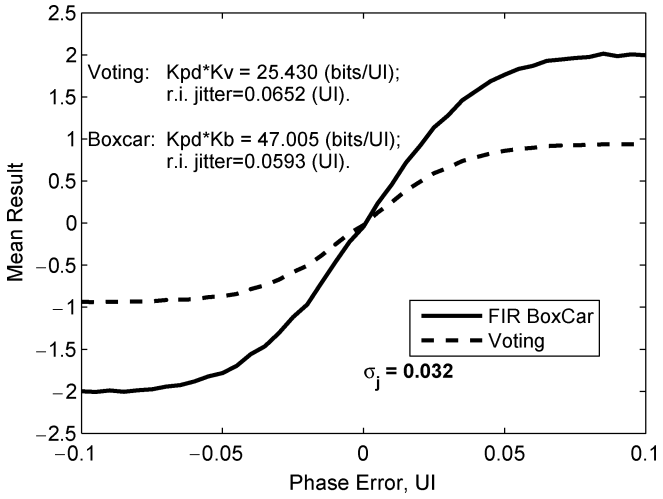

Fig. 9. Simulated decimation by voting and boxcar FIR.

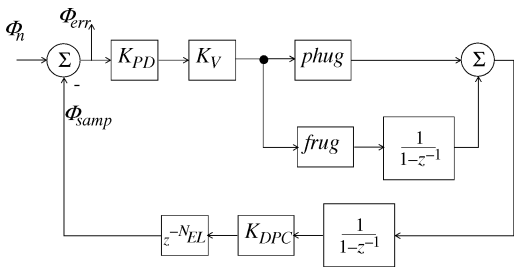

Fig. 10. Linearized model of proposed architecture.

TABLE II
TEST DEVICE DIGITAL CDR PARAMETERS

Parameter	Value
K_{DPC}	1 UI / 2^9 bits
K_V	8*.54=4.32
K_{PD}	10.6 per UI (for signal in figure 14)
phug	2^{-3}
frug	2^{-12}, 2^{-11} and 2^{-10}
N_{EL}	18

simulate the precise voting method to be used so as to be able determine the average gain $K_{\rm V}$.

C. Linearized Analysis of Sample System

In this section, we first present a linearized model of the proposed architecture in Fig. 10 and then proceed to analyze its transfer function and jitter tolerance. The linearized model that is equivalent to the architecture in Fig. 7 is shown in Fig. 10.

To analyze the performance of a system, we will use parameters which are consistent with the parameters of the CDR in the test device used in the measurements. These parameters are listed in Table II and described in the following text..

In the model, the element $K_{\rm PD}$ is the phase detector gain as given in (4). To get a meaningful value, we will use the jitter of 7.5 ps observed from the measured results provided later in the paper. To use this jitter value in (4), it must be converted into radians. For 5 Gbps operation, the period is 200 ps. Thus, σ_j is $7.5/200 * 2\pi$ radians. When this is substituted into (4), we get the value in the table.

The next element in the model is $K_{\rm V}$, the gain to handle any decimation that takes place. This includes the effects of decimation by voting. In the test device, the decimation factor was 8 and the factor for voting by 4 is arrived at in Section III-B.

The values of $phug$ and $frug$ correspond to the proportional and integral paths from the output of the voting to the DPC. In the measured results from the test device, three values of $frug$ were exercised.

The element $K_{\rm DPC}$ is the gain through the DPC. This corresponds to the resolution of the DPC in units of Unit Interval (UI) per bit. The resolution of the DPC is a tradeoff between the truncation noise induced by low resolution and the complexity and power required for high resolution.

Finally, recall that the term $z^{-N_{\rm EL}}$ incorporates all of the delay (analog and digital pipe stages) in going around the loop.

Two interesting functions to compute using the linearized model are the jitter tolerance function, $\Phi_{\rm in}/\Phi_{\rm err}$ and the transfer function, $\Phi_{\rm samp}/\Phi_{\rm in}$. To compute either of these, it is beneficial to first compute the loop gain, $L(z^{-1})$, from $\Phi_{\rm err}$ to $\Phi_{\rm samp}$:

$$L(z^{-1}) = \left(\frac{K_{\rm PD}K_{\rm V}K_{\rm DPC}}{1-z^{-1}}\right)\left(phug + \frac{frug}{(1-z^{-1})}\right)z^{-N_{\rm EL}}. \tag{8}$$

The jitter transfer function is proportional to the reciprocal of the phase error transfer function and is given in (9). Although any target may be used, here we chose to use a target bit error rate (BER) of 10^{-10} for which $12\sigma_j$ is appropriate.

$$\text{jitter tolerance fct} = \left(1 - \frac{12\sigma_j}{T_{UI}}\right) \quad \left(1 + L(e^{-j\omega})\right). \tag{9}$$

The first parenthetical term in (9) is the remaining horizontal eye opening remaining after considering the presence of Gaussian jitter with a standard deviation of σ_j. In the measured system the period of the unit interval ($T_{\rm UI}$) is 200 ps and the observed jitter was 7.5 ps and is assumed to be Gaussian. The jitter tolerance function is plotted in Fig. 11 for the three $frug$ values listed in Table II. It can be seen that all three settings readily beat the jitter tolerance limit. However, it is important to realize that when observing the jitter tolerance function of a linear model that it is an optimistic and inaccurate descriptor of the actual system for lower frequency values. In this range it is the large-signal slew-limiting caused by the saturation of the nonlinear phase detector transfer function that limits the performance.

The phase transfer function is given by the following well-known equation:

$$\Phi_{\rm samp}/\Phi_{\rm in} = \left(L(e^{-j\omega})\right) / \left(1 + L(e^{-j\omega})\right). \tag{10}$$

The transfer function is plotted in Fig. 12 for the three $frug$ values listed in Table II. It can be observed that for the design

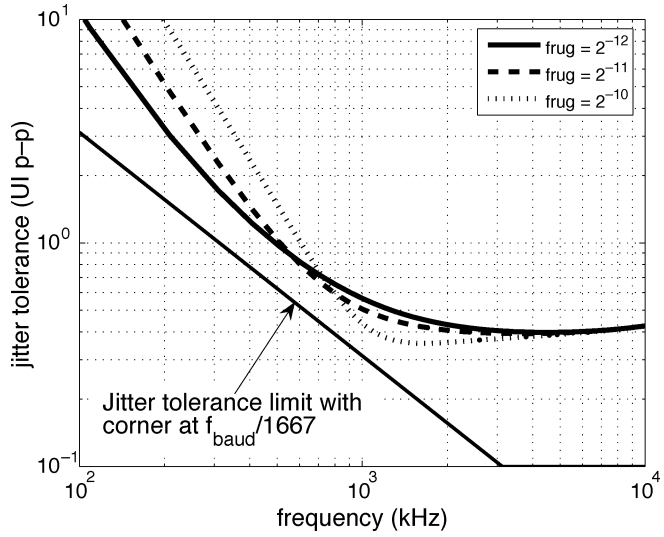

Fig. 11. Calculated jitter transfer function.

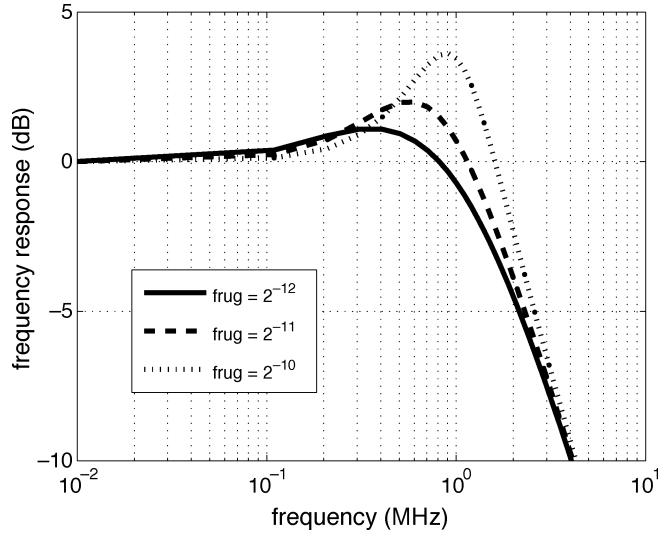

Fig. 12. Calculated phase transfer function.

the peaking takes on values of 1.1, 2, and 3.6 dB and the corresponding bandwidths are 1.6, 1.8, and 2.1 MHz.

Increasing the bandwidth of the system comes at the expense of jitter peaking. This is observed directly in Fig. 12 and its effect is seen by the crossing of the curves in Fig. 11. The "best" setting for a given application depends upon the spectrum of the incoming jitter.

D. Implementation Details

In this section, we will describe how the implementation in Fig. 13 matches the linearized model parameters listed in Table II. In this design, the phase integrator is unsigned and nonsaturating to allow the phase to move more than 1 UI. The frequency integrator is signed and saturating since it is used to track both $+/-$ parts per million (ppm) offsets. Saturation is required because we do not want the frequency register to "roll over" from large positive values to large negative values.

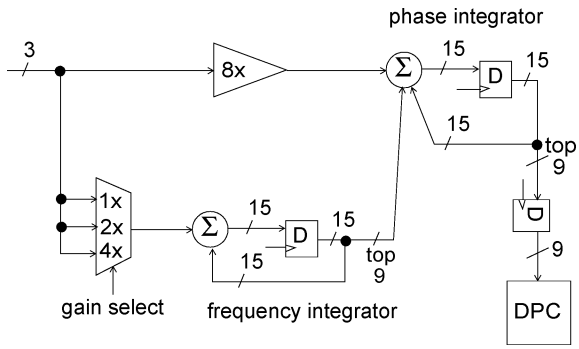

Fig. 13. Sample realization.

Finally, in the implementation, the phase and frequency integrators are fed from the sum of two 4-bit voting decimators as shown in Fig. 8, which provides an overall decimation factor of 8.

First, we describe how many bits are used for the phase integrator. One key aspect that we employ in the implementation to achieve fractional gains is sending only the top $N - D$ bits of an N-bit integrator to the next stage. In doing so, we achieve an effective gain of 2^{-D}, with the lower D bits being termed dither bits. We need to supply 9 bits to the DPC and we desire a $phug$ of 2^{-3}. Without considering the needs of the frequency register, the size of the phase integrator would simply be $9 + 3 = 12$ bits. However, in Fig. 13 it can be seen that the phase integrator is 15 bits wide, but that there is an 8X gain (3-bit shift) in the phase error path to the phase integrator. Thus, the $phug$ value is $8 * 2^{-6} = 2^{-3}$ as indicated in Table II. Next, we discuss why the need for the extra bits arises.

The purpose of the frequency integrator is to compensate for a ppm offset difference between the local reference clock and the incoming data. The frequency integrator must have enough top bits to hit the target maximum ppm and have enough resolution (dither bits) so as not to be a significant source of noise. The maximum ppm value that can be tracked is the fraction of a UI that the maximum frequency register value can move the output phase per UI times 1 million. To determine this value, we must include the fact that since the decimation factor is 8, the frequency integrator only gets to move the DPC once every 8 UI and that the top 9 bits $(8+\text{sign})$ get attenuated by 2^6 in passing to the DPC. Therefore, the frequency integrator can change the input to the DPC by 3.98 bits every 8 UI. Therefore, since the DPC has a 9-bit input in the implementation the maximum ppm offset that can be tracked is $(3.98/(8 * 512)) * 10^6 = 972$ ppm. The dither bits in the frequency integrator are included to provide the necessary attenuation and frequency resolution. The $frug$ value is calculated by the concatenating the effects of the dither bits in the frequency and phase registers which yields $2^{-6} * 2^{-6} = 2^{-12}$ as indicated in Table II. The frequency resolution of the top bits of the frequency integrator that are passed to the phase integrator is $972/2^8 = 3.8$ ppm/lsb.

In summary, we have truncated the phase to 1/512th of a UI and the frequency to 3.8 ppm/lsb. Simulations have shown that the quantization noise produced by these truncations is well into the noise floor.

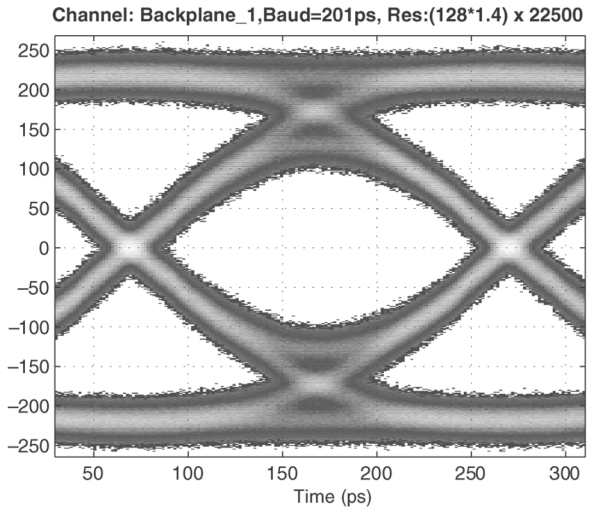

Fig. 14. Five Gbps receive signal used in obtaining measured results.

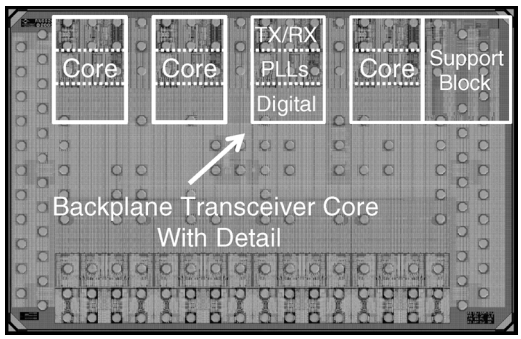

Fig. 15. Microphotograph of test chip.

IV. Measured Results

The measured results were obtained using a CDR integrated on a 0.13 μm CMOS test device (the details of which can be found in [11]) operating at 5 Gbps over a 34″ flame-resistant-4 (FR-4) backplane trace through two connectors. The data pattern was generated using a 31st order primitive polynomial, i.e., a pseudo random bit sequence-31 (PRBS-31) pattern. The device was programmed so as to produce an open eye using just transmit equalization. The signal present at the input to the receiver for all of the results described in this section is shown in Fig. 14. A microphotograph of the chip is shown in Fig. 15. As reported in [11], the entire transceiver consumes less than 150 mW at 5 Gbps and has an area of 0.56 mm^2.

A. $K_{\rm PD} * K_{\rm V}$ Measurement

With $frug$ set to zero and an offset programmed into the frequency register, the CDR must choose an offset phase such that the output of the decimator and the frequency register sum to zero. By reading the changes in the mean value of the phase register produced as different values of frequency offset are programmed, the transfer function of the combination of phase detector and decimator can be measured. In Fig. 16, this experiment is repeated for different programmed slicer offsets in order to see the effect of slicer offset on $K_{\rm PD}$. The results agree well with the simulated results of Fig. 5.

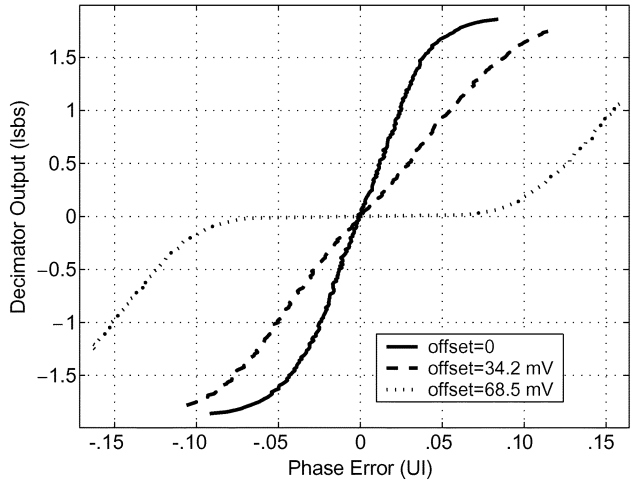

Fig. 16. Measured combined phase detector and decimator transfer function.

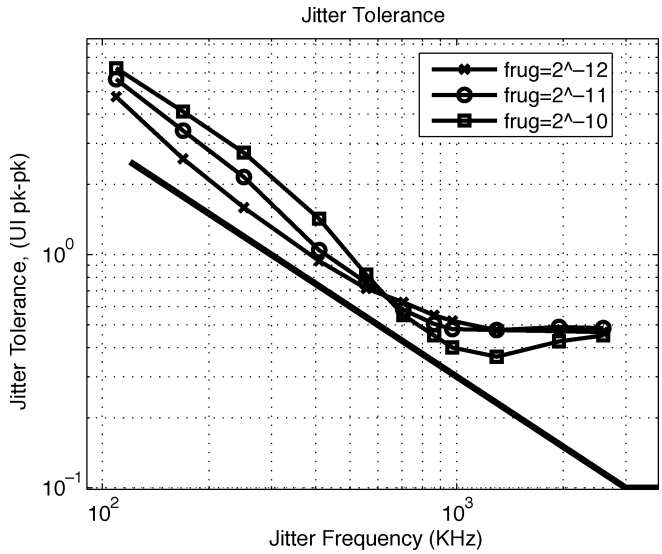

Fig. 17. Measured jitter tolerance.

B. Jitter Tolerance Measurement

Jitter tolerance was measured by determining the peak-to-peak (pk-pk) amplitude of jitter at each frequency which could be tolerated in order to produce a BER of approximately 1^{-10}. This was repeated for the three values of $frug$. The results are plotted in Fig. 17. These results agree well with the simulated results of the linear model shown in Fig. 11. The one point to note is the expected disagreement at lower frequencies as explained in Section III-C.

C. Limit Cycle Measurement

These measurements were made using a receiver containing another similar DPLL implementation which provides even higher $phug$ and $frug$ values and larger (more easily measured) limit cycle behavior than the DPLL described in Table II.

The measurement setup runs at 3.125 Gbps and includes a varied amount of FR-4, connectors, and programming of transmit equalization. These were varied to produce a wide range of loss-induced deterministic jitter in the received signal. Note that jitter of this type is wideband; little of the jitter is

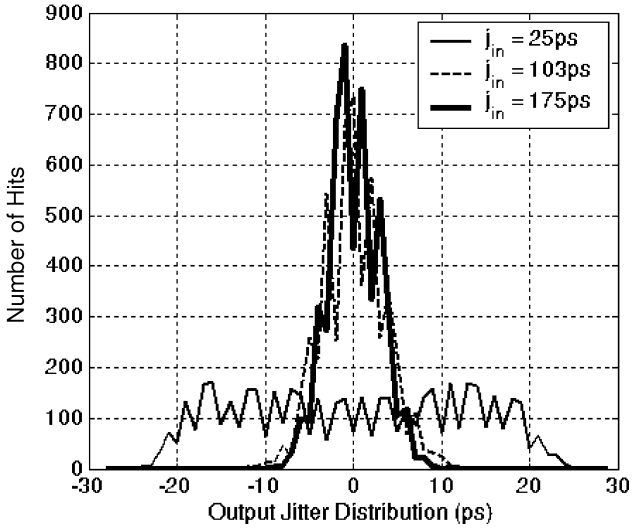

Fig. 18. Measured jitter amplitude distribution.

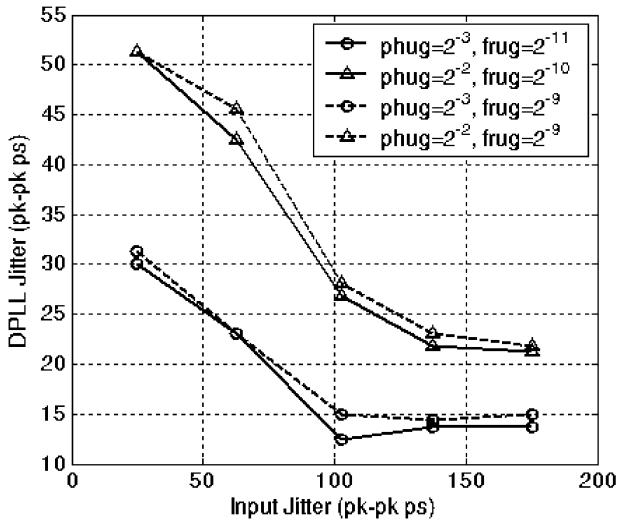

Fig. 19. Measured output jitter (pk-pk) as a function of input jitter.

within the bandwidth of the DPLL. Through repeated reads of the phase register, the DPLL output phase noise amplitude distribution can be measured.

Using $phug = 2^{-2}$, $frug = 2^{-10}$, for three different magnitudes of input DJ, the histograms of Fig. 18 were observed. Note that the wider, approximately flat distribution is indicative of limit cycle behavior, and is associated with the smallest input jitter magnitude (25 ps pk-pk).

Plotting only the pk-pk magnitude of the output jitter, and considering four different DPLL gain programming sets produces the result of Fig. 19. The results for the two smallest input jitter magnitudes are dominated by limit cycle behavior. For the three largest input jitter magnitudes, output jitter is nearly independent of input jitter magnitude. This occurs because larger input jitter produces lower DPLL gain at the high frequencies where the input DJ has the highest PSD.

Just how much limit cycle behavior can be tolerated? In Fig. 20, the same data is presented as the width of the remaining eye opening once the pk-pk values of the incoming

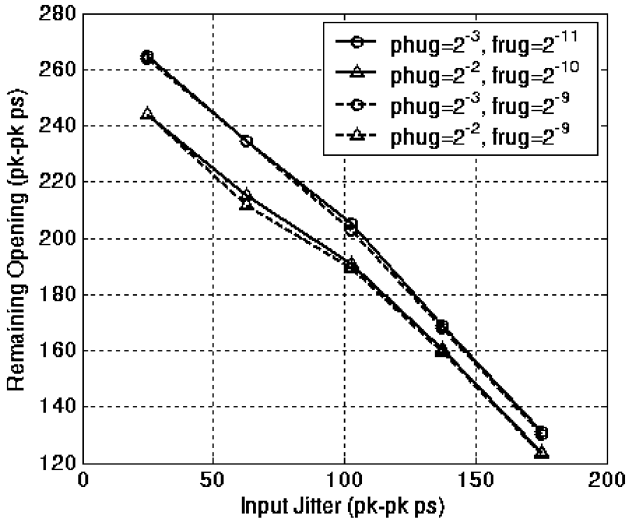

Fig. 20. Eye opening width remaining after subtracting pk-pk input and output jitter.

jitter and the (largely uncorrelated) output jitter are subtracted. As one would expect, overall system performance will be unaffected by the limit cycle behavior which occurs only when little jitter is present in the incoming signal.

V. Conclusion

A general DPLL-based architecture for CDR was presented. The key aspects to designing and understanding this architecture were set forth. Finally, results were presented validating the approach for use in multi-Gigabit binary data links.

Many advantages to the digital implementation of the CDR exist. These include at least: 1) substantial PSRR and thermal noise sources present in the analog implementation are absent in the digital implementation; the DPC is the only mixed-signal loop component remaining, which is locked to a quiet reference clock instead of jittered data, allowing the choice of higher loop bandwidth and subsequent low jitter; 2) insensitivity to long runs of transition-free data patterns because in the absence of transitions the phase error is exactly zero by definition, the only error source is any error trapped in the frequency register; 3) invariance of characteristics over PVT because the gain of the only analog component, the DPC, is naturally PVT invariant; 4) no possibility of false lock or the need for training mechanisms to avoid false locking; 5) analog process enhancements are not needed; 6) ease of porting a design across multiple technologies and foundries because the performance of the loop is derived through its digital functionality rather than by sensitive analog components; 7) production testing of logic gates is much more straightforward than analog circuits because standard digital scan techniques can be used for the digital portion; 8) ease of adding bench test hooks for characterization because phase programmability can be used for margining purposes in vector-only automated test environment (ATE); and 9) ease of allowing flexible control of design parameters.

In conclusion, DPLL-based CDRs are area and power efficient and provide flexible, effective functionality for Gbps data links.

ACKNOWLEDGMENT

The authors gratefully acknowledge B. Beale, A. Caffee, C. Jones, J. Giuliano, K. Krishna, B. Lefferts, J. Lessert, M. Loikkanen, J. Parker, S. Rockett, R. Segelken, A. Sengir, S. Titus, D. Weinlader, S. Wolfer, and D. Yokoyama-Martin for their useful discussions and help with the design, layout, and testing of the chip used to generate the results.

REFERENCES

[1] R. C. Walker, "Designing bang-bang PLLs for clock and data recovery in serial data transmission systems," in *Phase-Locking in High-Performance Systems*, B. Razavi, Ed.. New York: IEEE Press, 2003, pp. 34–45.

[2] M. Ramezani, C. Andre, and T. Salama, "A 10 Gb/s CDR with a half-rate bang-bang phase detector," in *Proc. Int. Symp. Circuits and Systems*, May 2003, vol. 2, pp. 181–184.

[3] R. Kreienkamp *et al.*, "A 10-Gb/s CMOS clock and data recovery circuit with an analog phase interpolator," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 735–743, Mar. 2005.

[4] 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.

[5] M. Ramezani, C. Andre, and T. Salama, "Jitter analysis of a PLL-based CDR with a bang-bang phase detector," in *Proc. 45th Midwest Symp. Circuits and Systems*, Aug. 2002, vol. 3, pp. 393–396.

[6] D. P. Atherton*, Nonlinear Control Engineering*. London & New York: Van Nostrand Reinhold Co., full edition 1975, student edition 1982.

[7] H. Takauchi *et al.*, "A CMOS multichannel 10-Gb/s transceiver," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2094–2100, Dec. 2003.

[8] H. Tamura *et al.*, "5-Gb/s bidirectional balanced-line link compliant with plesiochronous clocking," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2001, no. 375, pp. 50–51.

[9] R. Farjad-Rad *et al.*, "A 33-mW 8-Gb/s CMOS clock multiplier and CDR for highly integrated I/Os," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1553–1561, Sep. 2004.

[10] K. K. Chang *et al.*, "A 0.4–4-Gb/s CMOS quad transceiver cell using on-chip regulated dual-loop PLLs," *IEEE J. Solid-State Circuits*, vol. 38, no. 5, pp. 747–753, May 2003.

[11] K. Krishna *et al.*, "A 0.6 to 9.6 Gb/s binary backplane transceiver core in 0.13 μ CMOS," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2005, pp. 65–65.

[12] J. Parker *et al.*, "A 15 mW, 3.125 GHz PLL for serial backplane transceivers in 0.13 μm CMOS," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2005, pp. 412–413.

[13] 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.

[14] J. Yueming and A. Piovaccari, "A compact phase interpolator for 3.125 G Serdes application," in *Proc. Southwest Symp. Mixed-Signal Design*, Feb. 2003, pp. 249–252.

Jeff L. Sonntag received the B.S.E.E. degree from Carnegie-Mellon University, Pittsburgh, PA, in 1982, and the M.S.E.E. degree from Cornell University, Ithaca, NY, in 1983.

He joined Bell Labs in 1982, where he spent 18 years developing high-performance mixed-signal integrated circuits, focusing on applications in disk drive read channels while Bell Labs evolved into AT&T Microelectronics and Lucent Technologies. In recognition of his contributions in 1998, he was named a Bell Labs Fellow. In 2002, he joined Accelerant Networks in the development of high-speed serial transceivers, serving variously as mixed-signal circuit designer, system architect, and Chief Technical Officer. After the acquisition of Accelerant by Synopsys in 2004, he remained with Synopsys for two years, and was promoted to Synopsys Fellow. In 2006, he joined Silicon Laboratories, Beaverton, OR, where he serves as an Engineering Director in the Wireline division.

John Stonick (M'92) received the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, in 1992.

From 1993 to 1997, he held a postdoctoral research position in the Electrical and Computer Engineering Department at Carnegie Mellon University, Pittsburgh, PA. From 1997 to 2000, he was an Assistant Professor with the Electrical and Computer Engineering Department at Oregon State University, Corvallis, and a co-director for the NSF Center for the Design of Analog-Digital Integrated Circuits (CDADIC). Starting in 2000, he was a Principal Design Engineer with Accelerant Networks until they were acquired by Synopsys in 2004. Since 2004, he has remained with Synopsys, where he holds the title of Synopsys Scientist and was the 2004 Synopsys Distinguished Inventor. His interests include system architecture and simulation, clock and data recovery, using adaptive digital techniques to compensate for analog circuit imperfections in wireless and wireline transceivers.

Dr. Stonick has been a coauthor on two IEEE Best Paper Awards, 2006 IEEE CICC Best Invited Paper, and 1994 Matti S. Sukola Award for Best Paper Presented at the IEEE Broadcast Symposium, and a coauthor on a paper which won a DesignCon Paper Award in 2006. He is a member of the IEEE and a member of the ISSCC 2006–2007 technical program committee (Wireline).